

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Белгородский государственный технологический
университет им. В.Г. Шухова «БГТУ им. В. Г. Шухова»



На правах рукописи

Каратач Сергей Александрович

**Разработка высокопроизводительных методов
интеллектуального анализа данных на основе нечетких
систем при несинглтонной фаззификации**

Специальность 02.03.01 —
«Системный анализ, управление и обработка информации, статистика»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
кандидат технических наук, профессор
Синюк Василий Григорьевич

Белгород — 2025

Оглавление

	Стр.
Введение	5
 Глава 1. Системы нечеткого вывода как метод анализа	
зашумленных или неопределенных входных данных . .	9
1.1 Нейро-нечеткие системы	10
1.2 Методы нечеткого вывода	11
1.3 Несингтонная фаззификация	11
1.4 Нечеткое значение истинности	16
1.4.1 Вычисление нечеткого значения истинности, когда функции принадлежности формализуются гауссовыми функциями	22
1.5 Выводы	22
 Глава 2. Метод нечеткого вывода на основе нечеткого	
значения истинности	23
2.1 Постановка задачи нечеткого вывода	23
2.2 Вывод на основе нечеткого значения истинности	24
2.3 Вывод для систем логического типа	27
2.4 Анализ эффективности методов нечеткого вывода	28
2.5 Сравнительный анализ логического подхода к нечетком выводу с подходом Мамдани	28
2.6 Сравнение нечеткого вывода с использованием различных методов дефаззификации	28
2.6.1 Дефаззификация по методу среднего центра	28
2.6.2 Дефаззификация по методу центра тяжести	30
2.6.3 Дефаззификация по методу центра области	34
2.6.4 Дефаззификация по методу среднего максимума	34
2.7 Прогнозирование временных рядов на основе нечетких систем логического типа с использованием нечеткого значения истинности	35
2.8 Выводы	37

Глава 3. Программная реализация разработанного метода нечеткого вывода с применением технологии CUDA . . .	38
3.1 Вычисление нечеткого значения истинности посредством дискретизации	40
3.2 Алгоритм свертки НЗИ при $T_1 = \min$ и T_3 - неубывающая по всем аргументам	45
3.3 Реализация дефаззификации	47
3.4 Использование библиотеки ArborX	48
3.5 Алгоритм построения базы правил	50
3.6 Анализ производительности	50
3.7 Выводы	50
Глава 4. Применение разработанной нечеткой модели для прогнозирования временных рядов в задачах экономики и финансов	51
4.1 Задача прогнозирования стоимости ценных бумаг	52
4.2 Задача прогнозирования динамики использования банковского продукта / CLTV по продукту	52
4.3 Описание набора данных	52
4.4 Решение задачи с использованием разработанной нечеткой модели	52
4.5 Выводы	52
Заключение	53
Словарь терминов	54
Список литературы	55
Список рисунков	59
Список таблиц	60
Приложение А. Примеры вставки листингов программного кода	61

Приложение Б. Очень длинное название второго приложения, в котором продемонстрирована работа с длинными таблицами	67
Б.1 Подраздел приложения	67
Б.2 Ещё один подраздел приложения	69
Б.3 Использование длинных таблиц с окружением <i>longtblr</i> из пакета <code>tabularray</code>	73
Б.4 Форматирование внутри таблиц	76
Б.5 Стандартные префиксы ссылок	78
Б.6 Очередной подраздел приложения	79
Б.7 И ещё один подраздел приложения	79
Приложение В. Чертёж детали	80

Введение

Высокопроизводительный интеллектуальный анализ данных дает возможность принимать обоснованные решения на основе знаний, получаемых посредством обработки данных со скоростью близкой к реальному времени. Семейство методов мягких вычислений с применением техник высокопроизводительного анализа данных открывает возможность находить закономерности и взаимосвязи в данных, содержащих неоднозначность, с высокой скоростью.

Существующие реализации кейсов мягких вычислений опираются на использование методов нечеткого вывода выработанных Мамдани, Ларсеном, Такаги и Сугено. [Написать про логический подход](#). В этих случаях системы вывода работают с четкими входами, что позволяет упростить реализацию процедуры вывода. Однако такое упрощение порождает отличие с классическим подходом Заде.

В теории нечеткой логики нечеткий логический вывод реализуется за с помощью обобщенных нечетких правил *modus ponens* и *modus tollens* на основе *композиционного правила вывода*. При нескольких входах вычисление по данным правилом приводит к экспоненциальной зависимости вычислительной сложности от количества входов. Данное ограничение является основным препятствием для применения нечеткого логического вывода с несколькими посылками, тогда как необходимость анализа многомерных данных **является** актуальной задачей, **например,**

[Возможно стоит описать изученность проблемы в России?](#)

Целью данной работы является повышение производительности анализа неопределенных данных путем разработки математического и программного обеспечения на основе нечетких систем при несинглтонной фаззификации.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Провести обзор проблем и предлагаемых подходов построения и реализации нечетких систем анализа данных с качественным описанием.
2. Разработать метод вывода на основе нечеткого значения истинности для системы MISO-структуры логического типа, обеспечивающий полиномиальную вычислительную сложность.

3. Выполнить программную реализацию выработанного метода нечеткого вывода с использованием технологии параллельных вычислений CUDA, обеспечив эффективность реализации за счет внедрения оптимизаций алгоритма вывода.
4. Применить разработанный модуль нечеткого логического вывода для высокопроизводительного анализа зашумленных данных в выбранной предметной области.

Научная новизна:

1. Впервые применено нечеткое значение истинности и принцип обобщения для получения выходного значения при нескольких нечетких входах в соответствии с обобщенным нечетким правилом вывода *modus ponens* для нечетких систем логического типа, в результате чего была получена новая структура базы правил: «Если *истинно*, то B_k ».
2. Разработан метод нечеткого вывода логического типа с использованием нечеткого значения истинности, имеющий полиномиальную вычислительную сложность при многих нечетких входах.
3. Разработан метод регрессии временных рядов с нечеткими оценками измеренных значений на основе предложенного метода нечеткого вывода логического типа и алгоритм построения базы правил
4. Разработан параллельный алгоритм, реализующий нечеткий вывод на основе нечеткого значения истинности с применением отбора ...

Теоретическая значимость заключается в расширении класса задач анализа данных, эффективно решаемых при помощи нечеткого моделирования, соответствующего теории нечеткого вывода Л. Заде.

Практическая значимость ...

Методология и методы исследования. В работе использованы методы теории нечетких множеств, нечетких отношений, нечеткого логического вывода, принятия решений и мягких вычислений.

Основные положения, выносимые на защиту:

1. Метод вывода для нечетких систем логического типа на основе нечеткого значения истинности, имеющий полиномиальную вычислительную сложность при многих нечетких входах.
2. Метод регрессии для временных рядов с нечеткими оценками измеренных значений на основе метода нечеткого вывода логического типа.
3. Разработанный вид нечетких правил «Если *истинно*, то B_k ».

4. Разработанный параллельный алгоритм для предложенного метода вывода и его эффективная реализация на графическом процессоре с поддержкой технологии CUDA.

5. ...

6. ...

Соответствие диссертации научной специальности. Диссертационная работа соответствует паспорту специальности 2.3.1. «Системный анализ, управление и обработка информации, статистика» по следующим областям исследования:

— п. 10 «Методы и алгоритмы интеллектуальной поддержки при принятии управленческих решений в технических системах».

Внедрение результатов диссертационного исследования. Результаты диссертационной работы внедрены Предложенные алгоритмы также использованы при выполнении научного проекта при поддержке РФФИ №20-07-00030 «Разработка высокопроизводительных методов интеллектуального анализа данных на основе нечеткого моделирования и создание компьютерной системы поддержки принятия решений для классификации и прогнозирования».

Достоверность полученных результатов обеспечивается корректным применением математического аппарата, экспериментальными исследованиями, апробацией на научно-практических конференциях, доказанностью выводов.

Апробация работы. Основные результаты работы докладывались на:

1. Международная конференция «Перспективные компьютерные и цифровые технологии» (ACDT 2021)», г. Белгород, 2021.
2. XV Международная научная конференция «Параллельные вычислительные технологии (ПаВТ) 2021», г. Волгоград, 2021.
3. XI Международной научно-практической конференции «Интегрированные модели и мягкие вычисления в искусственном интеллекте (ИММВ-2022)», г. Коломна, 2022 г.
4. XX Национальная конференция по искусственному интеллекту с международным участием (КИИ-2022), г. Москва, 2022.
5. XVII Международная научная конференция «Параллельные вычислительные технологии (ПаВТ) 2021», г. Санкт-Петербург, 2023.

6. XXI Национальная конференция по искусственному интеллекту с международным участием (КИИ-2023), г. Смоленск, 2023.

Личный вклад. Все изложенные в диссертации результаты исследования получены либо соискателем лично, либо при его непосредственном участии.

Публикации. Основные результаты по теме диссертации изложены в 9 печатных изданиях, 2 из которых изданы в журналах, рекомендованных ВАК, 2 — в периодических научных журналах, индексируемых Web of Science и Scopus, 4 — в тезисах докладов. Зарегистрированы 1 патент и 1 программа для ЭВМ.

Объем и структура работы. Диссертация состоит из введения, 4 глав, заключения и 3 приложений. Полный объём диссертации составляет 80 страниц, включая 21 рисунок и 4 таблицы. Список литературы содержит 0 наименований.

Глава 1. Системы нечеткого вывода как метод анализа зашумленных или неопределенных входных данных

Высокопроизводительный анализ данных (HPDM) позволяет обрабатывать и анализировать огромные массивы данных с использованием современных вычислительных технологий – как специализированных аппаратных средств (высокопроизводительных вычислительных кластеров, GPU ускорителей), так и оптимизированного программного обеспечения. Этот подход объединяет имеющиеся модели анализа данных с возможностями масштабируемых технологий Big Data и высокопроизводительных вычислений (HPC), что обеспечивает получение знаний на основе данных практически в реальном времени и значительно сокращает время от сбора данных до принятия обоснованных решений. Широкое промышленное применение получили программные пакеты (например, Apache Spark, NVIDIA RAPIDS), включающие традиционные модели анализа данных, а также инструменты JIT-компиляции готового программного кода (например, Numba).

Для определенных задач традиционные «жесткие» вычисления (основанные на точных математических моделях) оказываются неэффективными или неприменимыми. Модели мягких вычислений принимают неопределенность, шум в данных и частичную истинность, что позволяет находить решения в условиях реального мира, где информация часто неполна или противоречива. Вместо строгих математических моделей используются эвристические и адаптивные методы, имитирующие человеческое мышление (например, лингвистические переменные вида «высокая температура»). Мягкие вычисления часто комбинируются с нейросетями (для обучения) и эволюционными алгоритмами (для оптимизации), усиливая их способность к обработке сложных систем.

Во многих случаях решение таких задач является вычислительно сложным, из-за чего технологии HPC оказываются востребованными в методах мягких вычислений. Наиболее ярко эта потребность выражается в задачах: планирование «умных» городов с использованием эволюционных алгоритмов для многокритериальной оптимизации [1, 2], использование систем на основе нечеткой логики для управления и мониторинга производственных процессов в режиме реального времени [3, 4], анализ сложных структур с большим количеством связей в задачах фармацевтики и генетики с применением генетических и

роевых алгоритмов [5, 6]. Из этих примеров видно, что широкий спектр методов мягких вычислений хорошо поддаются распараллеливанию их алгоритмов.

Некоторые высокопроизводительные реализации нейро-нечетких систем были оформлены в самостоятельные программные модули. Частой практикой достижения высокой производительности нечетких систем является эффективная утилизация аппаратных ресурсов [7], в том числе, встраиваемых систем и плат ПЛИС [8]. В [9] и [10] представлена реализация схемы *MapReduce* для классификации несбалансированных больших данных с использованием подхода [11]. В ситуации, когда большой набор данных целиком или фрагментарно может быть проанализирован с использованием памяти только одного вычислительного узла, но требуется провести много итераций анализа, для быстрого получения промежуточного результата нечеткого анализа целесообразно использовать (графический) ускоритель [**<empty citation>**].

1.1 Нейро-нечеткие системы

Текущее развитие нейро-нечетких систем направлено на повышение применимости таких моделей за счет повышения точности моделирования и увеличения скорости получения результата моделирования. Прогресс по этим двум направлениям подкреплён экспериментами по использованию различных способов фаззификации [12, 13, 14], импликации [15, 16, 17], дефаззификации [18, 19], выборе t -норм, в том числе, t -норм в композиционном правиле вывода [20].

Например, в

Существует также комбинированный подход с использованием так называемых гибких нейро-нечетких систем [21, 22], сочетающих в себе два метода нечеткого вывода. Использование параметрических t -норм дает возможность осуществлять плавный переход от одного метода вывода к другому. Это позволяет совершать подбор оптимального метода вывода для конкретной задачи посредством оптимизации данного дополнительного гиперпараметра.

Чаще всего нейро-нечеткие системы используются для анализа четких входных данных взятых из четких наборов данных. Однако существуют при-

меры использования нечетких систем для моделирования нечетких данных. Например, прогнозирования временных рядов [**<empty citation>**].

Исследуются также и подходы для обучения нейро-нечетких систем. В работах широко применяется метод [23] из-за своей простоты. Также появляются методы с использованием эволюционного подхода [**<empty citation>**], градиентного спуска [**<empty citation>**] и непрерывного обучения на потоковых данных [24, 25].

1.2 Методы нечеткого вывода

Эффективная организация нечеткого вывода является одной из главных точек повышения производительности при использовании нечетких систем. Проблема разрабатывается. Предложенная в 1965 году Заде теория нечетких множеств [**<empty citation>**] позже была использована для построения схемы нечеткого логического вывода и нечетких логических систем. Использование этой схемы нечеткого вывода для составных посылок оказывается затруднено из-за экспоненциальной временной сложности операций над нечеткими отношениями. Во входных

В ответ на проблему сложности использования классического нечеткого вывода были предложены методы Мамдани, Такаги-Сугено, Ларсена и Цукamoto. Эти подходы напрямую используют четкие входные значения при формировании нечетких посылок. Недостатком такого упрощения является искажение ??????. Кроме того эти методы используют t -норму вместо импликации.

1.3 Несинглтонная фаззификация

Большинство имеющихся реализаций нечеткого вывода предоставляют возможность анализа лишь четких — числовых данных.

Нечеткая система представляется в виде композиции фаззификатора, базы правил, модуля вывода и дефаззификатора, как показано на рисунке ??. Фаззификация — отображение входных данных из исходного простран-

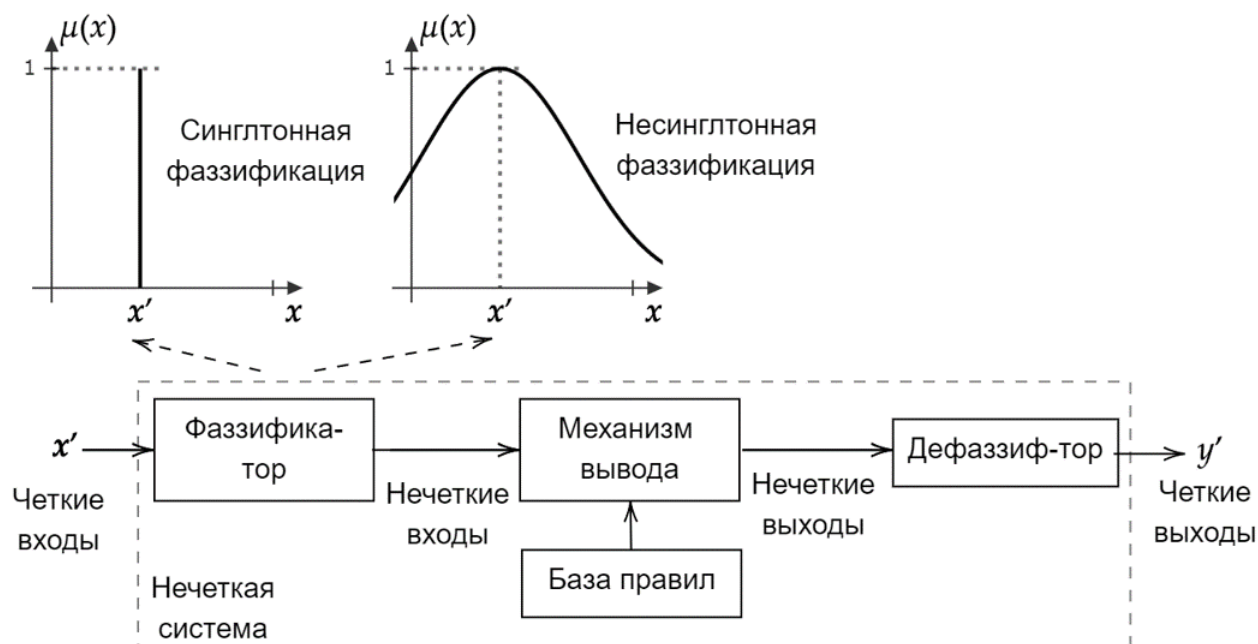


Рисунок 1.1 — Схема системы нечеткого вывода с использованием синглтонной и несинглтонной фаззификации

ства в пространство нечетких множеств. Наиболее распространенным является метод синглтонной фаззификации. Основная причина его широкого использования является существенное упрощение реализации систем нечеткого вывода. При использовании синглтонной фаззификации поданное на вход значение x' интерпретируется как истинное значение измеренной величины, что эквивалентно использованию функции принадлежности входного нечеткого множества $\mu_{A'}(x) = [x = x']$. Информация о неопределенности входных данных при этом игнорируется.

Альтернативный подход с использованием несинглтонной фаззификации предусматривает формализацию входного значения нечетким множеством, содержащим информацию о неопределенности значения точки входных данных. Эти неопределенности могут возникать как результат несовершенства процедуры измерений (например, шумом измерительного оборудования, дефектами или деградацией качества датчиков), или когда входные данные описываются качественными понятиями естественного языка.

В случае с анализом числовых данных, неопределенность измерений формализуется функцией принадлежности, которая $\mu_{A'}(x') = 1$ и $\mu_{A'}(x)$ уменьшается по мере удаления от x' . При таком способе моделирования измеренное

значение x' рассматривается как истинное, а значения в его окрестности — как возможные.

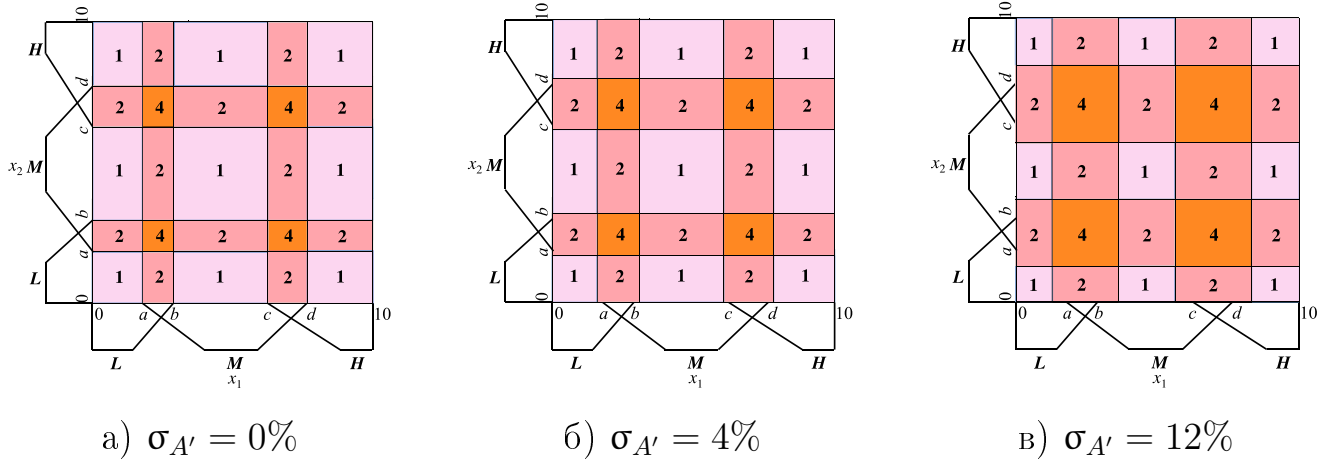


Рисунок 1.2 — Сравнение количества активированных правил при переходе от сингלטонной фаззификации к несингלטонной и при увеличении ширины окрестности погрешности $\sigma_{A'}$.

Влияние перехода от сингלטонной фаззификации к несингלטонной и величины окрестности погрешности на внутреннее поведение и итоговое качество нечеткой системы продемонстрировал Мендель в [1].

Для анализа от использования того или иного способа фаззификации на корректность получаемого результата можно для системы логического типа. Мендель в своей книге проводил такое сравнение для систем типа Мамдани. Поскольку в системах типа Мамдани в качестве функции импликации выступает t -норма, то разница от применения двух способов фаззификации проявляется в различных максимальных уровнях линии пересечения ф. п. входной посылки и антецедента правила. В зависимости от минимум или произведение.

Применение композиционного правила вывода \sup здесь дает эффект *пре-фильтрации* (или *корректировки*) входного значения. То есть ядра функций принадлежности антецедентов правил выполняют функцию эталонов, а уровень пересечения функции принадлежности для входного значения и ф. п. антецедента правила позволяет интерпретировать входное значение как суперпозицию эталонных значений антецедентов с долями равными этому уровню.

В случае с нечеткой системой логического типа, разница от использования различных способов фаззификации будет проявляться в различных формах выходного нечеткого множества.

В статье [1] Мендель также дополнительно иллюстрирует описанное выше с помощью карты разбиений первого и второго порядка на декартовом произведении базовых множеств входных лингвистических переменных. Разбиение первого порядка

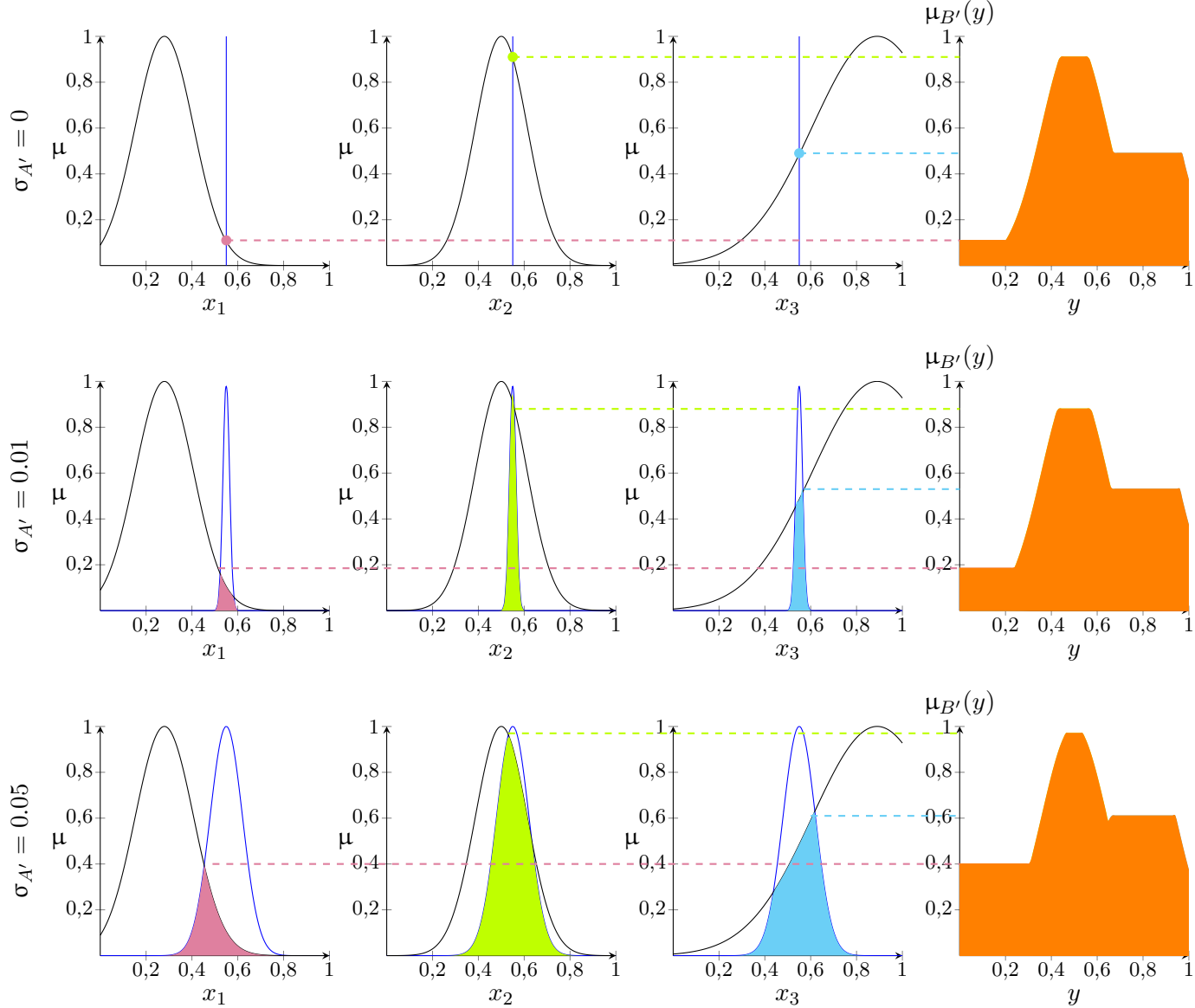


Рисунок 1.3 — Сравнение формы функций принадлежности выходных нечетких множеств для подхода Мамдани

Показанная на этих схемах динамика более ясно раскрыта на рисунке 1.3 для различных значениях среднеквадратичного отклонения в гауссовой ф. п. входного значения на примере агрегации выходной ф. п. нечеткой системы с тремя правилами в базе правил. Видно, что при переходе от синглтонной фаззификации к несинглтонной и при дальнейшем увеличении ширины среднеквадратичного отклонения ф. п. входного нечеткого множества, повышается

уровень срабатывания первого правила, и, как следствие использования импликации Мамдани, уровень задействования ф. .п выходного нечеткого множества этого правила в результирующей агрегации. Кроме того, можно пронаблюдать, упомянутый ранее, эффект префильтрации входного значения антецедентом третьего правила.

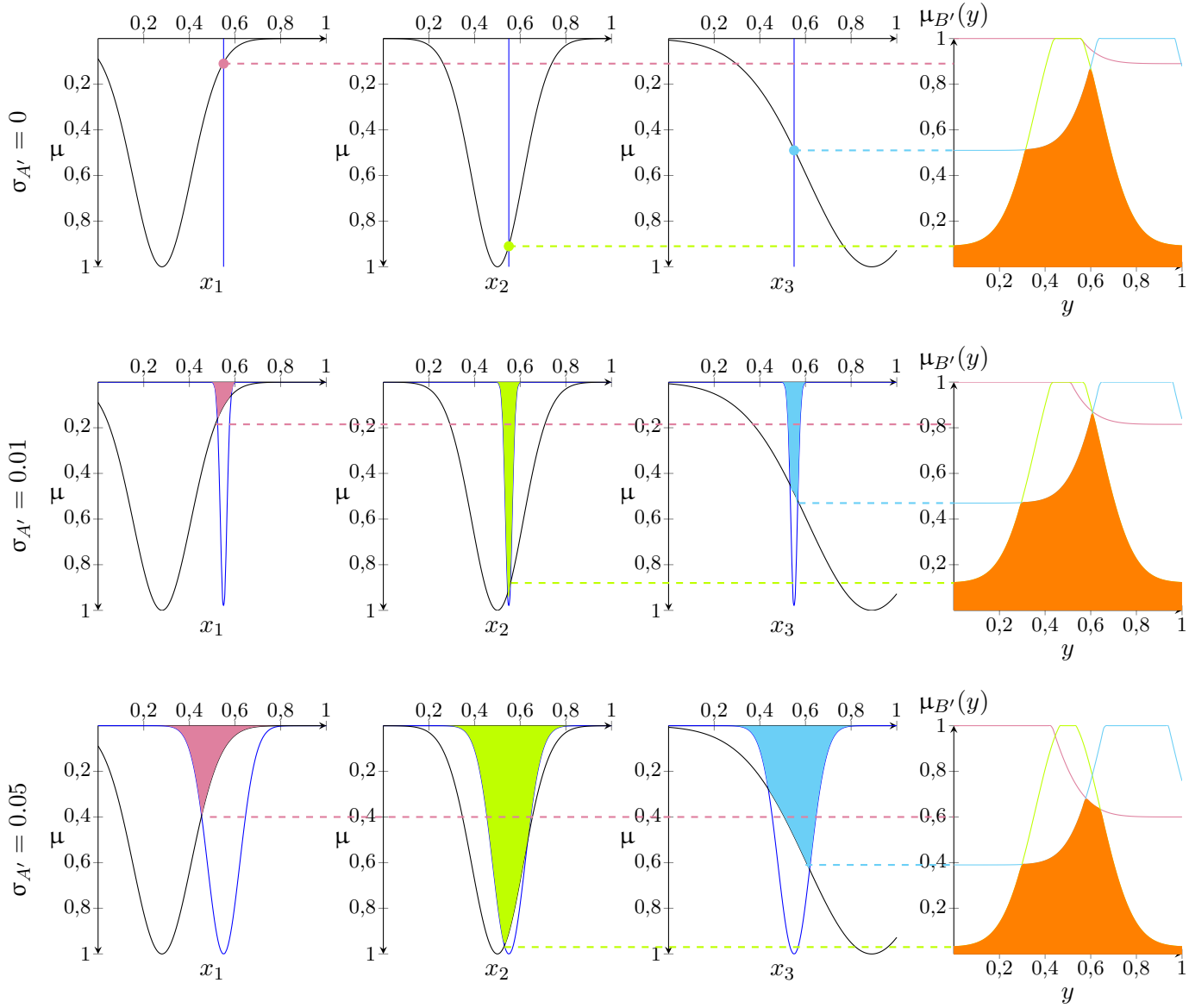


Рисунок 1.4 — Сравнение формы функций принадлежности выходных нечетких множеств для логического подхода

Теперь можно проследить за влиянием увеличения ширины окна для измеренного значения на область выходного нечеткого множества нечеткой системы при использовании логического подхода к нечеткому выводу. При логическом подходе функция принадлежности выходного нечеткого множества формируется как результат пересечения (в данном случае операцией *min*), что

можно представить как постепенное вырезание области функции принадлежности выходного нечеткого множества. Из рисунка 1.4 видно, что при увеличении ширины в пересечении проекций импликации на пространство выходной переменной оказывается более «точно выкроенная» область.

1.4 Нечеткое значение истинности

Определение. Нечеткой истинностью множества A относительно нечеткого множества A' называется нечеткое множество $CP(A, A')$ такое, что:

$$\mu_{CP(A, A')}(v) = \sup_{\substack{\mu_A(x)=v \\ x \in X}} \{\mu_{A'}(x)\} \quad (1.1)$$

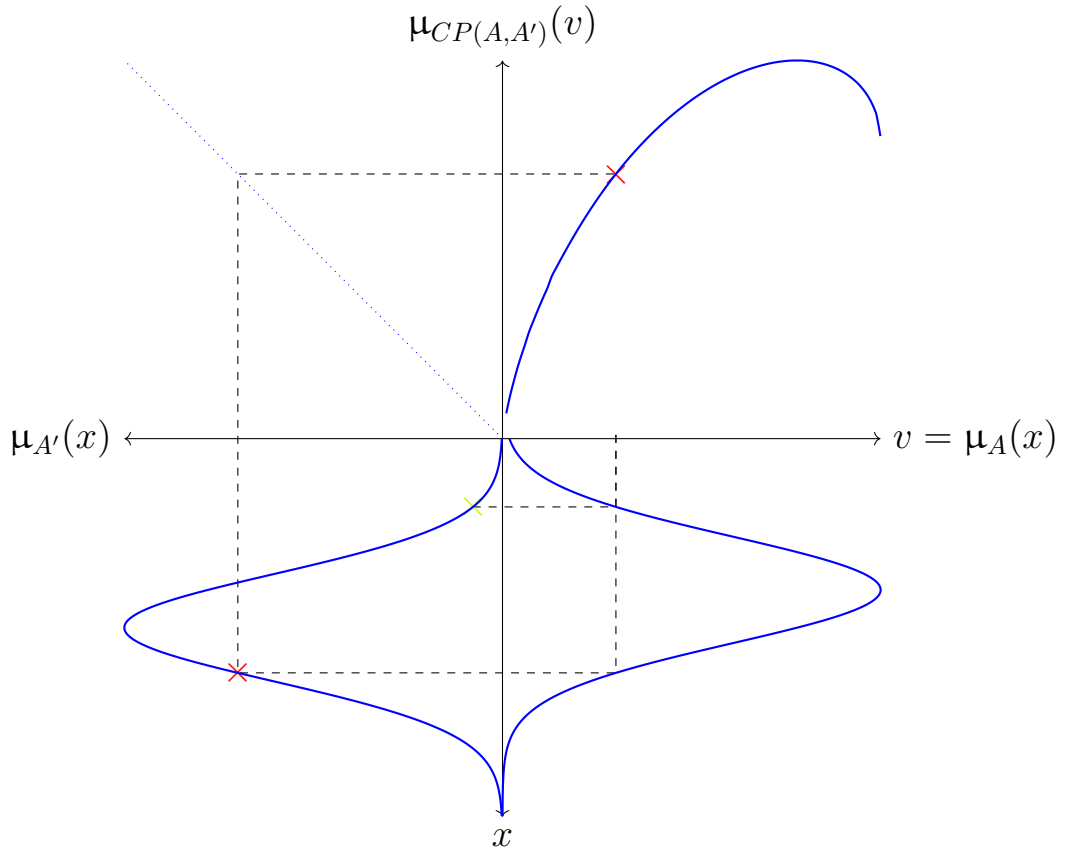


Рисунок 1.5 — Пример вычисления нечеткого значения истинности

Приведенные на рисунке 1.6 функции принадлежности термов лингвистической переменной истинности можно задать следующими выражениями:

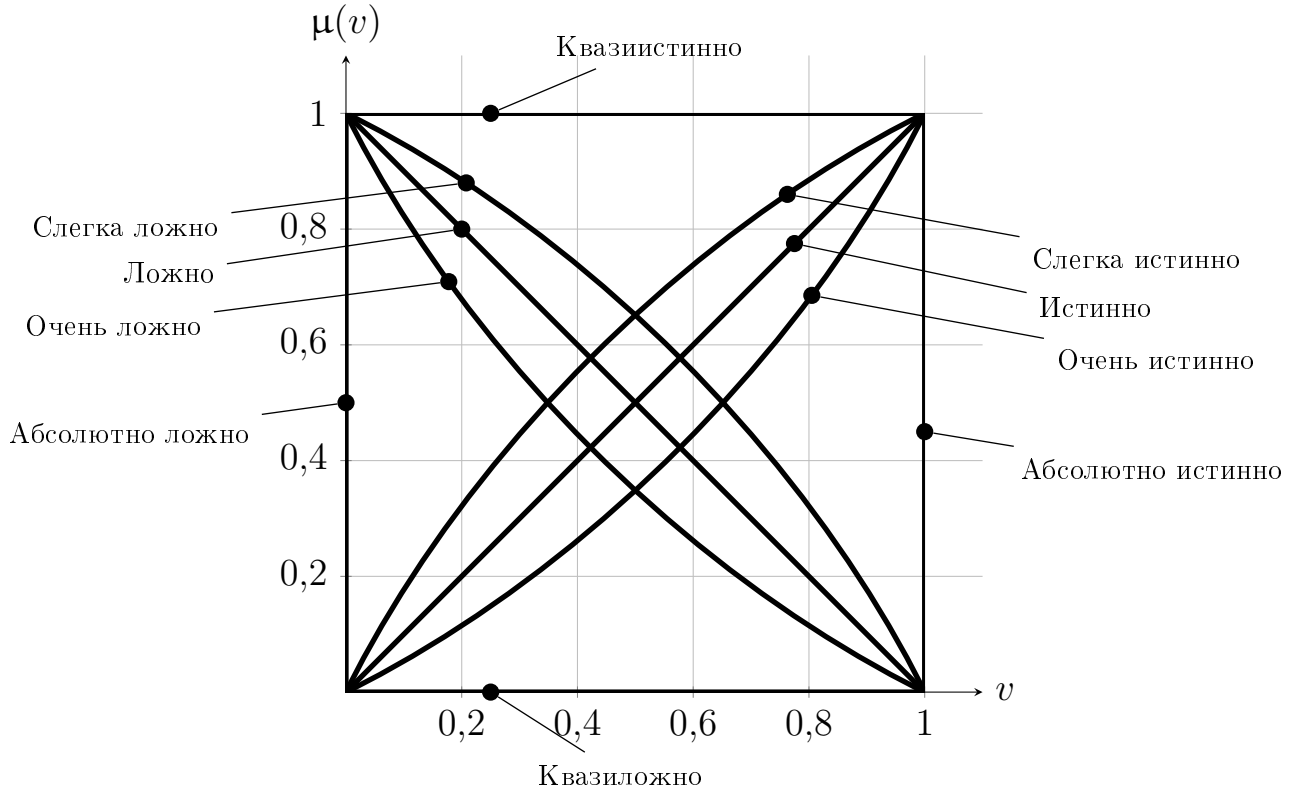


Рисунок 1.6 — Значения лингвистической переменной «истинность»

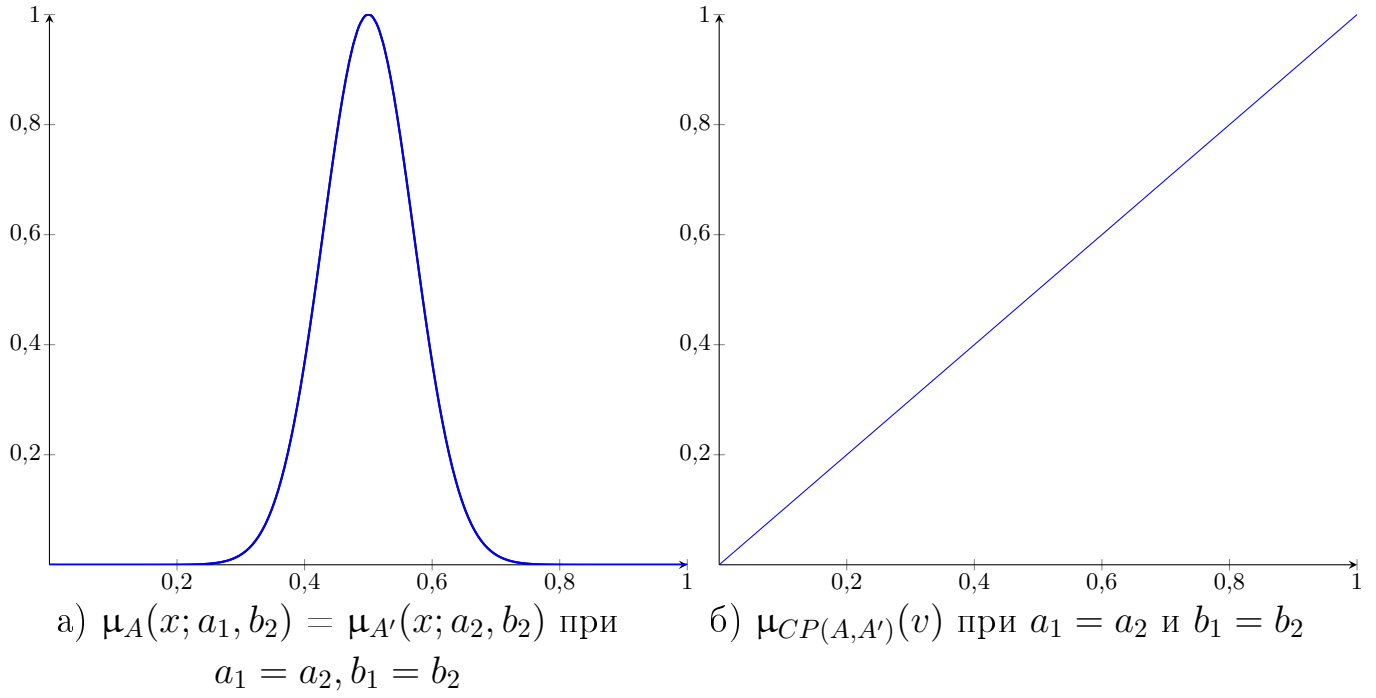
$$\begin{aligned}
 M[\text{«истинно»}] &= \int_0^1 v/v; & M[\text{«ложно»}] &= \int_0^1 1 - v/v; \\
 M[\text{«слегка истинно»}] &= \int_0^1 \sqrt{v}/v; & M[\text{«слегка ложно»}] &= \int_0^1 \sqrt{1 - v}/v; \\
 M[\text{«очень истинно»}] &= \int_0^1 v^2/v; & M[\text{«очень ложно»}] &= \int_0^1 \frac{(1 - v)^2}{v}; \\
 M[\text{«абсолютно истинно»}] &= \frac{1}{1} + \int_0^1 \frac{0}{v}; & M[\text{«абсолютно ложно»}] &= \frac{1}{0} + \int_0^1 \frac{0}{v}; \\
 M[\text{«квазиистинно»}] &= \int_0^1 1/v; & M[\text{«квазиложно»}] &= \int_0^1 0/v.
 \end{aligned}$$

Существуют следующие *аксиомы значения истинности*:

- *Аксиома истинности*. Нечеткое значение истинности ИСТИННО задается нечетким множеством:

$$CP(A, A') = \{ \langle \mu_{CP(A, A')}(v), v \rangle \} = \{ v/v \}, v \in [0; 1],$$

что выполняется тогда и только тогда, когда A относительно соответствует A' , т. е. функции принадлежности нечетких множеств A' и A совпадают.



На рис. ?? представлены графики совпадающих функций принадлежности высказываний и построенной функции принадлежности нечеткого значения истинности.

- *Аксиома ложности.* Нечеткое значение истинности ЛОЖНО задается нечетким множеством:

$$CP(A, A') = \{ \langle \mu_{CP(A,A')}(v), v \rangle \} = \{ (1-v)/v \} = \{ v/(1-v) \}, v \in [0; 1],$$

что выполняется тогда и только тогда, когда утверждаемое высказывание A противоположно утверждаемому в A' , т. е. функции принадлежности высказываний A' и A удовлетворяют одному из условий:

$$\mu_{A'}(x) = 1 - \mu_A(x)$$

или

$$\mu_{A'}(x) = \begin{cases} 1 - \mu_A(x), & x \leq x_{max} \\ 0, & x > x_{max} \end{cases}$$

или

$$\mu_{A'}(x) = \begin{cases} 0, & x < x_{max} \\ 1 - \mu_A(x), & x \geq x_{max}, \end{cases}$$

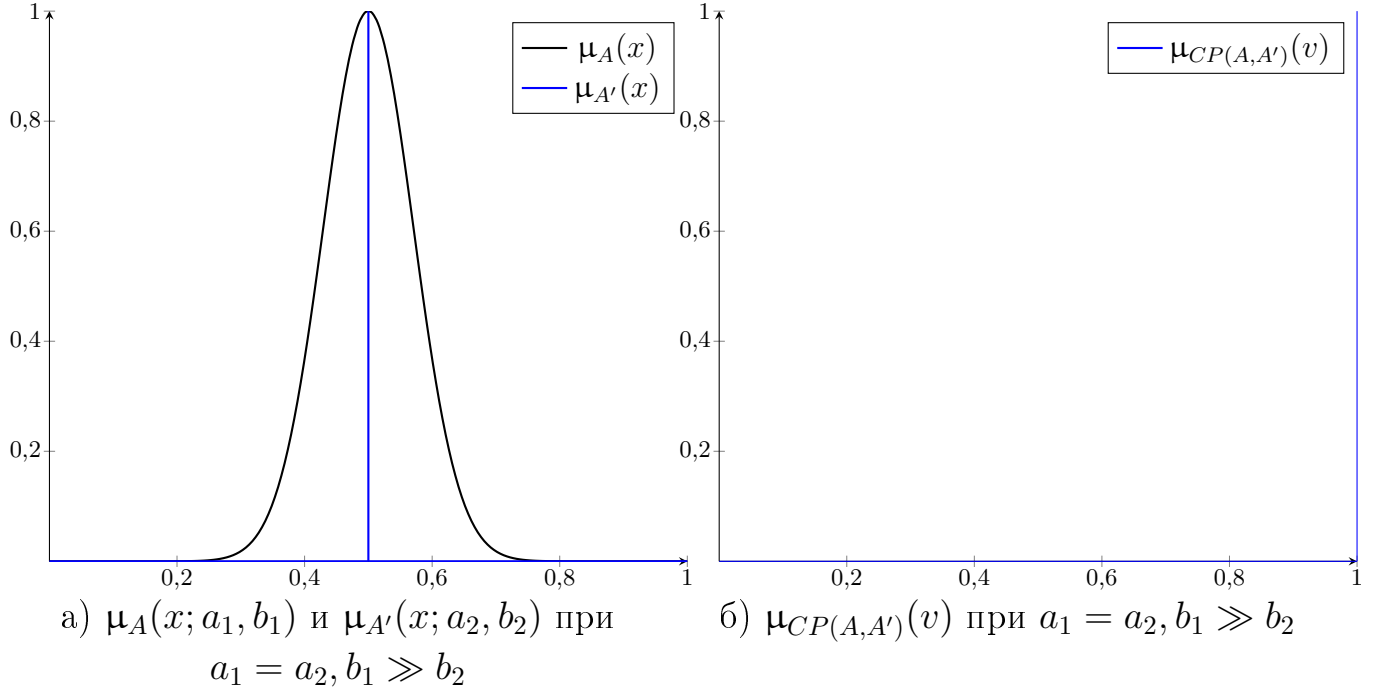
где $x_{max} = \arg \max_x \mu_A(x)$.

На рис. ?? представлены графики противоположных по значению функций принадлежности высказываний A' и A и построенной функции принадлежности нечеткого значения истинности.

- *Аксиома абсолютной истинности.* Значение истинности АБСОЛЮТНО ИСТИННО задается нечетким множеством:

$$CP(A, A') = \{\langle \mu_{CP(A, A')}(v), v \rangle\} = \{v/1\} = \{1/1\}, v \in [0, 1],$$

что выполняется тогда и только тогда, когда A' абсолютно соответствует A , то есть в случае когда оценка данная в высказываниях A' и A является четкой или нечеткой, когда носитель высказывания A' включен в носитель высказывания A .



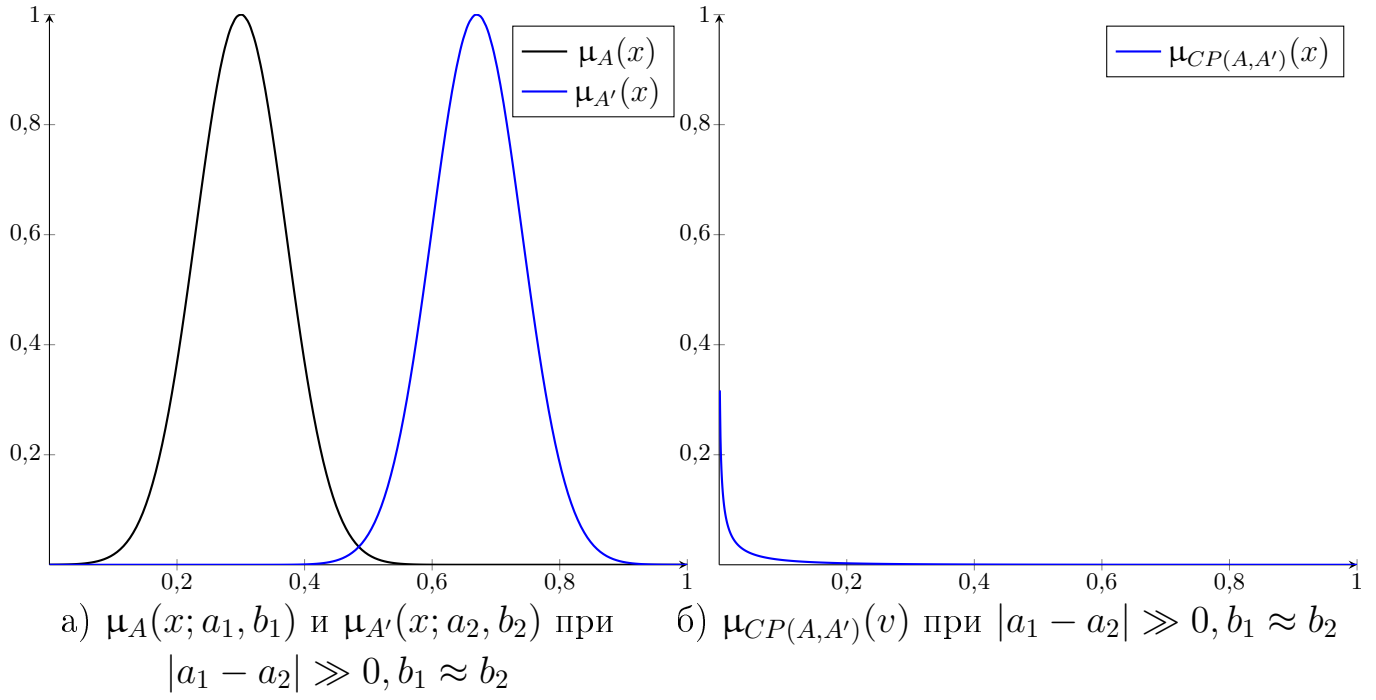
На рис. ?? представлены графики функций принадлежности высказывания A' , включенного в A и функции принадлежности нечеткого значения истинности, соответствующие данной ситуации. Для моделирования четкого значения функции принадлежности (синглтона) взята гауссова функция кривая с дисперсией, стремящейся к нулю.

- *Аксиома абсолютной ложности.* Значение истинности АБСОЛЮТНО ЛОЖНО задается нечетким множеством:

$$CP(A, A') = \{\langle \mu_{CP(A, A')}(v), v \rangle\} = \{v/0\} = \{1/0\}, v \in [0, 1],$$

что выполняется тогда и только тогда, когда A' абсолютно не соответствует A , то есть в случае когда оценки данные в высказываниях A' и A имеют несовпадающие носители.

На рис. ?? представлены графики непересекающихся гауссовых функций принадлежности высказываний A' и A с удаленными центрами и



построенная для этого случая функция принадлежности нечеткого значения истинности.

- *Аксиома квазиистинности.* Нечеткое значение истинности КВАЗИИСТИННО задается нечетким множеством:

$$CP(A, A') = \{ \langle \mu_{CP(A, A')}(v), v \rangle \} = \{ 1/v \}, v \in [0; 1],$$

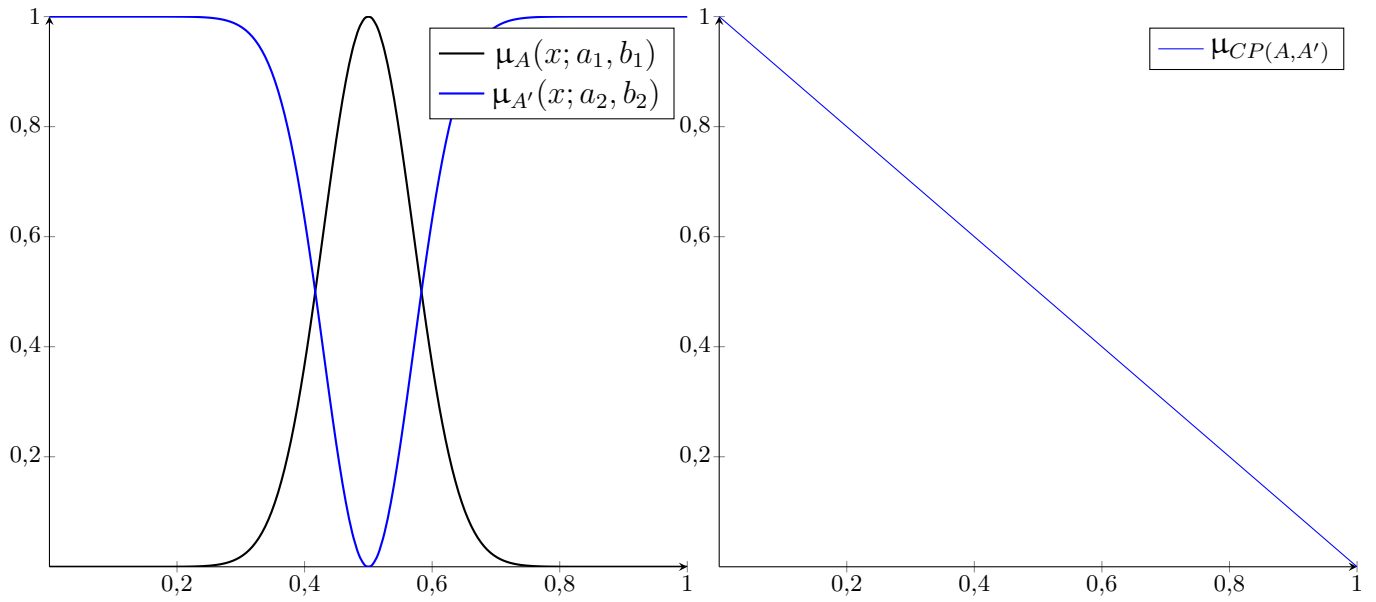
что выполняется тогда и только тогда, когда утверждение в высказывании A' является частным по отношению к A , то есть оценка данная в высказывании A интервальная и совпадает с носителем высказывания A .

На рис. представлены графики совпадающих функций принадлежности высказываний и рассчитанной функции принадлежности нечеткого значения истинности.

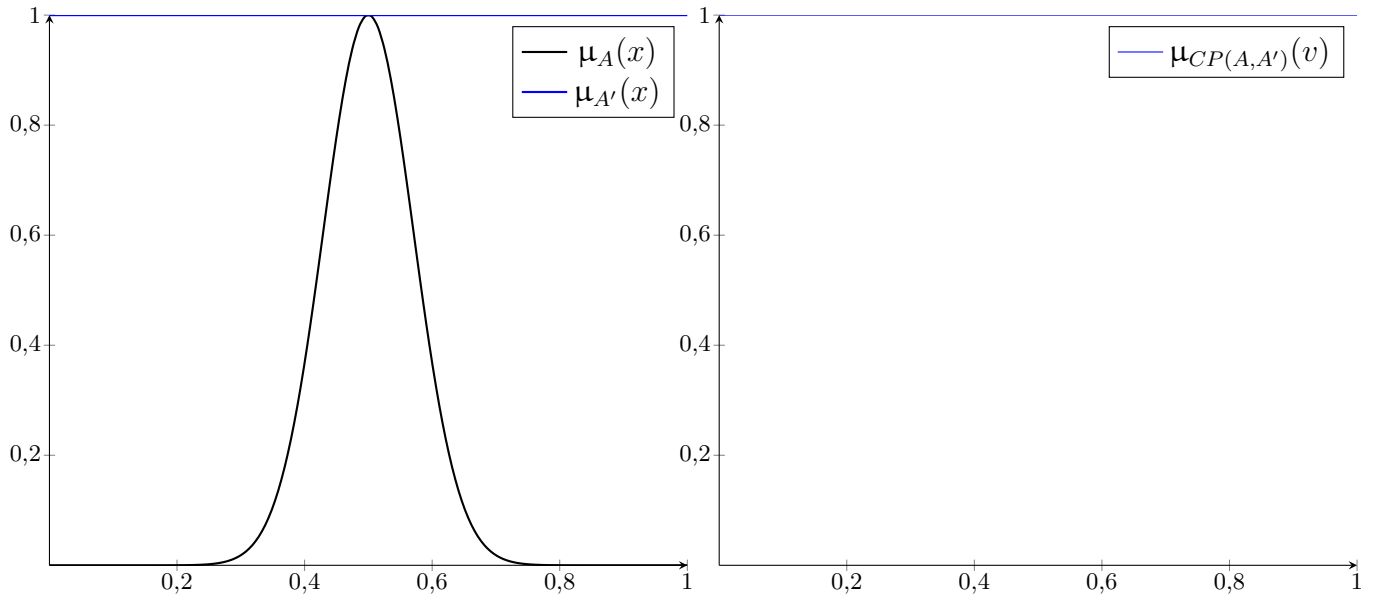
- *Аксиома квазиложности.* Нечеткое значение истинности КВАЗИЛОЖНО задается нечетким множеством:

$$CP(A, A') = \{ \langle \mu_{CP(A, A')}(v), v \rangle \} = \{ 0/v \}, v \in [0; 1],$$

что справедливо тогда и только тогда, когда утверждаемое в A' не имеет реального подтверждения в действительности. Иными словами, отсутствует возможность установления истинности высказывания A' , так как не определено, существует ли в действительности то, что утверждается в A' .



а) $\mu_{A'}(x; a_2, b_2) = 1 - \mu_A(x; a_1, b_1)$ при $a_1 = a_2, b_1 = b_2$ б) $\mu_{CP(A,A')}(v)$ при $\mu_{A'}(x) = 1 - \mu_A(x)$



а) $\mu_A(x; a_1, b_1)$ и $\mu_{A'}(x; a_2, b_2)$ при $a_1 = a_2, b_1 \ll b_2$ б) $\mu_{CP(A,A')}(v)$ при $a_1 = a_2, b_1 \ll b_2$

На рис. представлены графики совпадающих функций принадлежности высказываний и рассчитанной функции принадлежности нечеткого значения истинности.

1.4.1 Вычисление нечеткого значения истинности, когда функции принадлежности формализуются гауссовыми функциями

Зададим функции принадлежности нечетких множеств факта и послышки в виде:

$$\mu_A(x; a, b) = e^{-\frac{(x-a)^2}{2b^2}} \mu_{A'}(x; c, d) = e^{-\frac{(x-c)^2}{2d^2}}.$$

Тогда, согласно формуле нечеткого значения истинности (1.1), для вычисления нечеткого значения истинности в точке v_0 необходимо сперва найти все точки из области определения функции принадлежности факта, в которых он принимает значение v_0 . В случае с гауссовой функцией это можно сделать, с помощью обратной гауссовой функции:

$$x(v) = a \pm b\sqrt{-2 \ln v},$$

тогда

$$\begin{aligned} \mu_{CP(A,A')}(v) &= \max \left\{ e^{-\frac{((a-b\sqrt{-2 \ln v})-c)^2}{2d^2}}, e^{-\frac{((a+b\sqrt{-2 \ln v})-c)^2}{2d^2}} \right\} \\ &= \max \left\{ e^{-\frac{((a-c)-b\sqrt{-2 \ln v})^2}{2d^2}}, e^{-\frac{((a-c)+b\sqrt{-2 \ln v})^2}{2d^2}} \right\} \end{aligned} \quad (1.2)$$

1.5 Выводы

В данной главе

Глава 2. Метод нечеткого вывода на основе нечеткого значения истинности

2.1 Постановка задачи нечеткого вывода

Лингвистическая модель представляет собой базу правил вида:

$$R_k : \text{Если } x_1 \text{ есть } A_{k1} \text{ и } x_2 \text{ есть } A_{k2} \text{ и } \dots \text{ и } x_n \text{ есть } A_{kn}, \text{ то } y \text{ есть } B_k, \quad (2.1)$$

где N — количество нечетких правил, $A_{ki} \subseteq X_i, i = \overline{1, n}, B_k \subseteq Y$ — нечеткие множества, которые характеризуются функциями принадлежности $\mu_{A_{ki}}(x_i)$ и $\mu_{B_k}(y)$ соответственно; x_1, x_2, \dots, x_n — входные переменные лингвистической модели, причем

$$[x_1, x_2, \dots, x_n]^T = \mathbf{x} \in X_1 \times X_2 \times \dots \times X_n.$$

Символами $X_i, i = \overline{1, n}$ и Y обозначаются соответственно пространства входных и выходной переменных. Если ввести обозначения $\mathbf{X} = X_1 \times X_2 \times \dots \times X_n$ и $\mathbf{A}_k = A_{k1} \times A_{k2} \times \dots \times A_{kn}$, причем

$$\mu_{\mathbf{A}_k}(\mathbf{x}) = T_1 \mu_{A_{ki}}(x_i),$$

$$i = \overline{1, n}$$

где T_1 — произвольная t -норма, то правило ?? представляется в виде нечеткой импликации

$$R_k : \mathbf{A}_k \rightarrow B_k, k = \overline{1, N}. \quad (2.2)$$

Правило R_k можно формализовать как нечеткое отношение, определенное на множестве $\mathbf{X} \times Y$, т.е. $R_k \subseteq \mathbf{X} \times Y$ — нечеткое множество с функцией принадлежности

$$\mu_{R_k}(\mathbf{x}, y) = \mu_{\mathbf{A}_k \rightarrow B_k}(\mathbf{x}, y).$$

Модель логического типа определяет задание функции $\mu_{\mathbf{A}_k \rightarrow B_k}(\mathbf{x}, y)$ на основе известных функций принадлежности $\mu_{\mathbf{A}_k}(\mathbf{x})$ и $\mu_{B_k}(y)$ с помощью одной из предложенных в [2] функций импликации:

$$\mu_{\mathbf{A}_k \rightarrow B_k}(\mathbf{x}, y) = I(\mu_{\mathbf{A}_k}(\mathbf{x}), \mu_{B_k}(y)),$$

где I — некоторая импликация.

Ставится задача определить нечеткий вывод $B'_k \subseteq Y$ для системы, представленной в виде (??), если на входах - нечеткие множества. $\mathbf{A}' = A'_1 \times A'_2 \times \dots \times A'_n \subseteq \mathbf{X}$ или x_1 есть A'_1 и x_2 есть A'_2 и ... и x_n есть A'_n с соответствующей функцией принадлежности $\mu_{\mathbf{A}'}(\mathbf{x})$, которая определяется как

$$\mu_{\mathbf{A}'}(\mathbf{x}) = \bigwedge_{i=\overline{1,n}} \mu_{A'_i}(x_i). \quad (2.3)$$

Несингтонный фаззификатор отображает измеренное $x_i = x'_i, i = \overline{1,n}$ в нечеткое число, для которого $\mu_{A'_i}(x'_i) = 1$ и $\mu_{A'_i}(x_i)$ уменьшается от единицы по мере удаления от x'_i . В соответствии с обобщенным нечетким правилом modus ponens [2], нечеткое множество B'_k определяется композицией нечеткого множества \mathbf{A}' и отношения \mathbf{R}_k , т.е.

$$B'_k = \mathbf{A}' \circ (\mathbf{A}_k \rightarrow B_k),$$

или, на уровне функций принадлежности

$$\mu_{B'_k}(y|\mathbf{x}') = \sup_{\mathbf{x} \in \mathbf{X}} \left\{ \mu_{\mathbf{A}'}(\mathbf{x}') \star^{T_2} I(\mu_{\mathbf{A}_k}(\mathbf{x}), \mu_{B_k}(y)) \right\}. \quad (2.4)$$

В (2.4) применена условная нотация, так как ввод в нечеткую систему происходит при определенном значении \mathbf{x} , а именно \mathbf{x}' . Обозначение $\mu_{B'_k}(y|\mathbf{x}')$ показывает, что $\mu_{B'_k}$ изменяется с каждым значением \mathbf{x}' . Вычислительная сложность выражения (2.4) составляет $O(|X_1| \cdot |X_2| \cdot \dots \cdot |X_n| \cdot |Y|)$ т.е. экспоненциальная.

2.2 Вывод на основе нечеткого значения истинности

Используя правило истинностной модификации [1] можно выразить:

$$\mu_{A'}(\mathbf{x}) = \tau_{A|A'}(\mu_A(x))$$

где $\tau_{A|A'}$ — нечеткое значение истинности (НЗИ) нечеткого множества A относительно A' , представляющее собой функцию принадлежности совместимости $CP(A_k, A')$ A_k по отношению к A' , причем A' рассматривается как достоверное [Дюбуа и др., 1990]:

$$\tau_{A_k|A'}(v) = \mu_{CP(A_k, A')}(v) = \sup_{\substack{\mu_{A_k}(x)=v \\ x \in X}} \{ \mu_{A'}(x) \}. \quad (2.5)$$

Таким образом НЗИ отражает совместимость факта с посылкой в нечеткой форме. Упрощенные подходы отображают совместимость в одно значение из диапазона впервые представленное в [5].

Перейдем от переменной x к переменной v в выражении нечеткого вывода (??), обозначив

$$\mu_{A_k}(x) = v \text{ и } \mu_{A'}(x) = \tau_{A_k|A'}(v),$$

то есть выполним преобразование нечетких множеств на пространстве X в истинностное пространство $[0, 1]$:

?????

Получим:

$$\mu_{A'}(x) = \tau_{A_k|A'}(\mu_{A_k}(x)) = \tau_{A_k|A'}(v) \quad (2.6)$$

Тогда (2.4) примет вид:

$$\mu_{B'_k}(y|\mathbf{x}') = \sup_{v \in [0,1]} \left\{ \tau_{A_k|A'}(v) \overset{T_2}{\star} I(v, \mu_{B_k}(y)) \right\}. \quad (2.7)$$

При переходе к нечеткому выводу по n входам формула вычисления НЗИ для нечетких отношений посылки и факта имеет вид:

$$\tau_{\mathbf{A}_k|\mathbf{A}'}(v) = \sup_{\substack{\mu_{\mathbf{A}_k}(x_1, \dots, x_n) = v \\ (x_1, \dots, x_n) \in \mathbf{x}}} \{ \mu_{\mathbf{A}'}(x_1, \dots, x_n) \}.$$

Или в выражении через операции сверток t -норм T_1 (2.2) и T_3 (2.3):

$$\tau_{\mathbf{A}_k|\mathbf{A}'}(v) = \sup_{\substack{T_1 \mu_{A_{ki}}(x_i) = v \\ i=\overline{1,n} \\ (x_1, \dots, x_n) \in \mathbf{x}}} \left\{ T_3 \mu_{A'_i}(x_i) \right\}. \quad (2.8)$$

Вместо выражения (2.8), НЗИ для n входов может быть вычислено как свертка НЗИ по каждому отдельному входу:

$$\tau_{\mathbf{A}_k|\mathbf{A}'}(v) = \tilde{T}_{i=\overline{1,n}} \tau_{A_{ki}|A'_i}, v \in [0, 1], \quad (2.9)$$

где \tilde{T} - расширенная по принципу обобщения n -местная t -норма [26], которая определяется как

$$\tilde{T}_{i=\overline{1,n}} \tau_{A_{ki}|A'_i}(v) = \sup_{\substack{T_1 v_i = v \\ i=\overline{1,n} \\ (v_1, \dots, v_n) \in [0,1]^n}} \left\{ T_3 \tau_{A_{ki}|A'_i}(v_i) \right\} \quad (2.10)$$

в результате перехода

$$\mu_{A_{ki}}(x_i) = v_i \text{ и } \mu_{A'_i}(x_i) = \tau_{A_{ki}|A'_i}(v_i).$$

Тогда для системы с n входами выражения нечеткого вывода на основе НЗИ (2.7) примет вид:

$$\mu_{B'_k}(y|\mathbf{x}') = \sup_{v \in [0,1]} \left\{ \tau_{\mathbf{A}_k|\mathbf{A}'}(v) \overset{T_2}{\star} I(v, \mu_{B_k}(y)) \right\} \quad (2.11)$$

Стоит отметить, что выражение (2.9) можно записать следующим образом, подчеркнув возможность попарного рекурсивного нахождения свертки НЗИ:

$$\begin{aligned} \tau_{A_k, A'}(v) &= \tilde{T}_1 \tau_{A_{ki}|A'_i}(v_i) \\ &= \left(\dots \left(\left(\mu_{CP(A_{k1}, A'_1)}(v_1) \tilde{T}_1 \mu_{CP(A_{k2}, A'_2)}(v_2) \right) \tilde{T}_1 \dots \right) \tilde{T}_1 \mu_{CP(A_{kn}, A'_n)} \right). \end{aligned}$$

Для $n = 2$, \tilde{T} записывается как:

$$\tilde{T} \tau_{A_{ki}|A'_i}(v) = \sup_{\substack{v_1 T_1 v_2 = v \\ v_1, v_2 \in [0,1]}} \left\{ \tau_{A_{k1}|A'_1}(v_1) T_3 \tau_{A_{k2}|A'_2}(v_2) \right\}, v \in [0,1]. \quad (2.12)$$

При вербализации импликации в (2.10) она представится в виде:

$$\text{Если } nзи \text{ есть ИСТИННО, то } y \text{ есть } B'_k \quad (2.13)$$

Таким образом, (2.13) представляет собой еще одну структуру правил в отличие от канонических структур Заде [10] и Такаги-Сугено [9]. Применение данного правила не зависит от количества входов в нечетких системах.

В формуле (2.11) данный подход позволяет переместить процесс вывода в единое пространство НЗИ, где функции истинности, в отличии от различных пространств в подходе Заде, могут быть объединены в более эффективный вычислительный процесс.

Порядок функции временной сложности вычисления B'_k на основе выражения (2.11) составляет $O(n|V|^2 + |V| \cdot |Y|)$, где $V = CP(A_k, A')$. Сравнение схем нечетких выводов с соответствии с соотношениями (2.4) и (2.11) представлены на рис. 2.1.

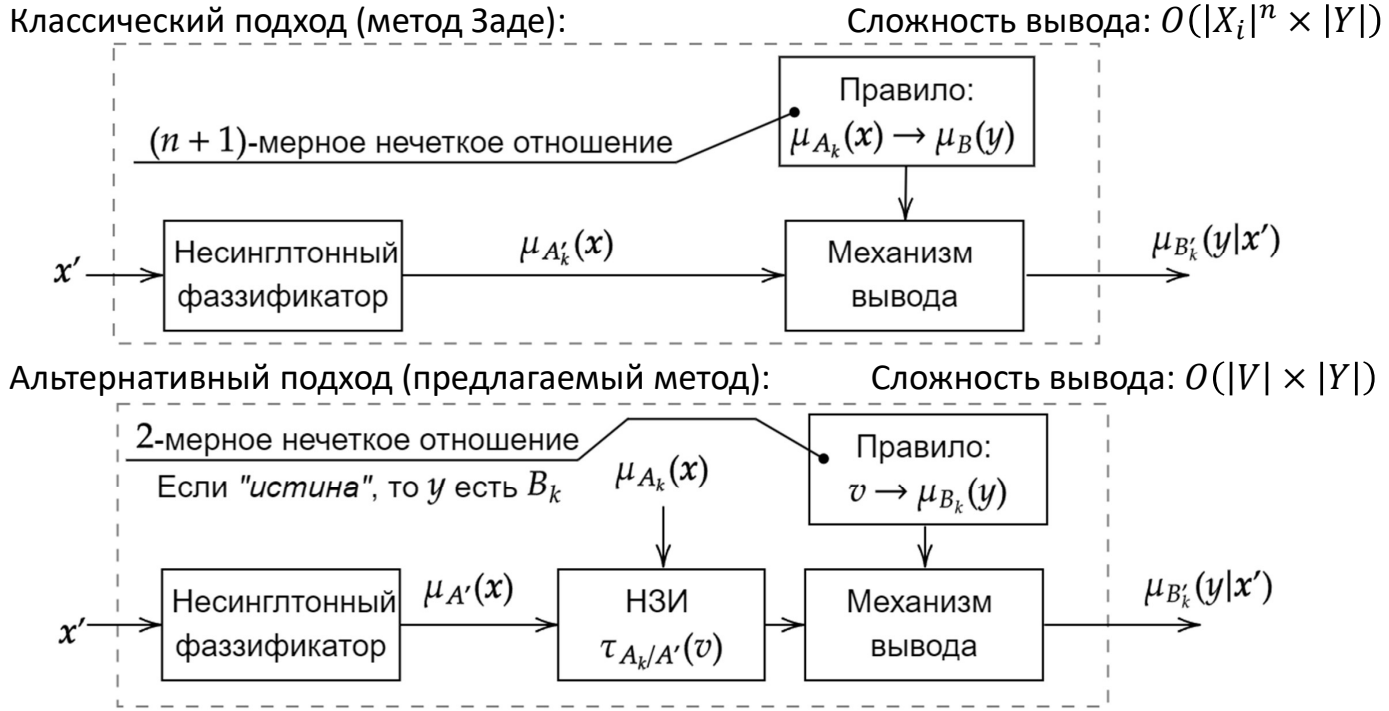


Рисунок 2.1 — Сравнение классической схемы нечеткого вывода и схемы нечеткого вывода на основе НЗИ

2.3 Вывод для систем логического типа

- S-импликация (Клине-Динеса, Лукасевича, Райхенбаха, Фодора):
 $I(a, b) = S(1 - a, b)$
- R-импликация (Гогуен, Гедель): $I(a, b) = \sup_z \{z | T(a, z) \leq b\}$
- Q-импликация (Заде, Вильмотта): $I(a, b) = S(1 - a, T(a, b))$

В логическом подходе правила объединяются связкой «И», тогда результирующее нечеткое множество является результатом произведения нечетких множеств, получаемых в результате нечеткого логического вывода по каждому правилу отдельно:

$$B' = \bigcap_{r=1}^N B'_r. \quad (2.14)$$

$$\mu_{B'}(y) = \bigwedge_{r=1}^N \mu_{B'_r}(y) = \bigwedge_{r=1}^N \left\{ \sup_{v \in [0,1]} \left\{ \tau_{A_r|A'} \overset{T_2}{\star} I(v, \mu_{B_r}(y)) \right\} \right\} \quad (2.15)$$

2.4 Анализ эффективности методов нечеткого вывода

2.5 Сравнительный анализ логического подхода к нечеткому выводу с подходом Мамдани

Для формирования условий в которых тот или иной подход к нечеткому выводу показывает свои сильные стороны имеет смысл провести сравнение.

2.6 Сравнение нечеткого вывода с использованием различных методов дефаззификации

В статье [18] описывается подход к сравнению методов дефаззификации. Описанный метод реализации [27].

2.6.1 Дефаззификация по методу среднего центра

(center average)

$$\hat{y}_{CA} = \frac{\sum_{k=1}^N \bar{y}_k \mu_{B'_k}(\bar{y}_k)}{\sum_{k=1}^N \mu_{B'_k}(\bar{y}_k)}$$

Поскольку $\mu_{B_r}(\bar{y}_k) = 1$ при $k = r$, тогда по формуле (2.11) $\mu_{B'_k}(\bar{y}_k)$ выразится:

$$\mu_{B'_k}(\bar{y}_k) = \sup_{v \in [0,1]} \left\{ \tau_{A_k|A'} \overset{T_2}{\star} I(v, \mu_{B_k}(\bar{y}_k)) \right\} \quad (2.16)$$

$$= \sup_{v \in [0,1]} \left\{ \tau_{A_k|A'} \overset{T_2}{\star} I(v, 1) \right\}. \quad (2.17)$$

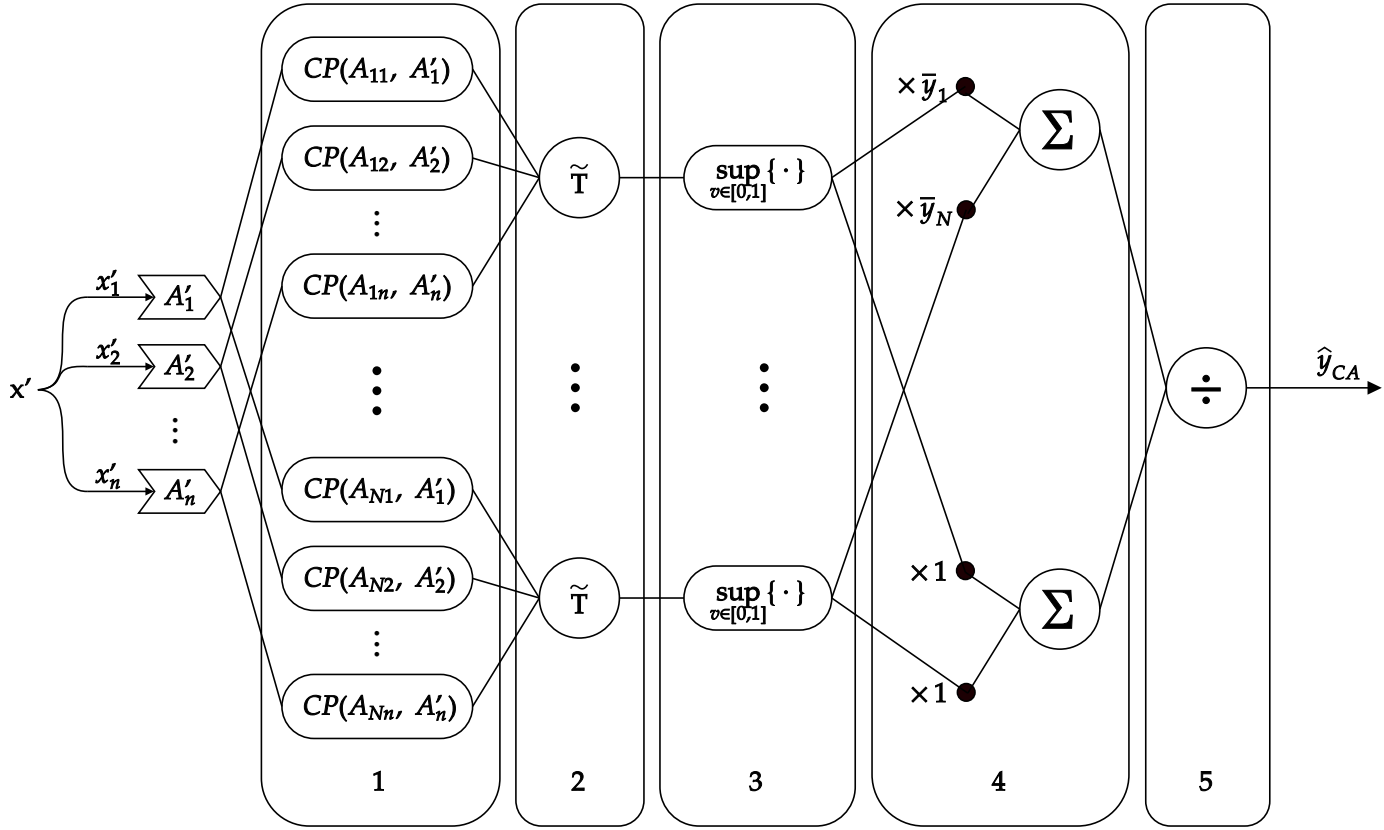


Рисунок 2.2 — Схема нейро-нечеткой системы с использованием дефаззификации по методу среднего центра для S - и R -импликаций

Обозначив $I(v, 1) = I_1(v)$, с учетом () формула дефаззификации (2.16) примет вид:

$$\hat{y}_{CA} = \frac{\sum_{k=1}^N \bar{y}_k \sup_{v \in [0,1]} \left\{ \tau_{A_r|A'} \overset{T_2}{\star} I_1(v) \right\}}{\sum_{k=1}^N \sup_{v \in [0,1]} \left\{ \tau_{A_r|A'} \overset{T_2}{\star} I_1(v) \right\}} \quad (2.18)$$

Рассмотрим вычисление τ_k для различных категорий импликаций:

— для S -импликаций:

$$I_1(v) = I(v, 1) = S\{1 - a, 1\} = 1$$

— для R -импликаций:

$$\begin{aligned} I_1(v) &= I(v, 1) = \sup_z \{z | T(v, z) \leq 1\} & v \in [0, 1] \\ &= \sup_z \{z | \forall z\} & z \in [0, 1] \\ &= 1 \end{aligned}$$

– для Q -импликаций:

$$\begin{aligned} I_1(v) &= I(v, 1) = S \{1 - v, T(v, 1)\} \\ &= S \{1 - v, v\} \end{aligned}$$

Тогда для S - и R -импликаций формула (2.18) с учетом свойства t -нормы $T(a, 1) = a$ примет вид:

$$\hat{y}_{CA} = \frac{\sum_{k=1}^N \bar{y}_k \sup_{v \in [0,1]} \left\{ \tau_{\mathbf{A}_r | \mathbf{A}'} \overset{T_2}{\star} 1 \right\}}{\sum_{k=1}^N \sup_{v \in [0,1]} \left\{ \tau_{\mathbf{A}_r | \mathbf{A}'} \overset{T_2}{\star} 1 \right\}} \quad (2.19)$$

$$= \frac{\sum_{k=1}^N \bar{y}_k \sup_{v \in [0,1]} \left\{ \tau_{\mathbf{A}_r | \mathbf{A}'} \right\}}{\sum_{k=1}^N \sup_{v \in [0,1]} \left\{ \tau_{\mathbf{A}_r | \mathbf{A}'} \right\}}, \quad (2.20)$$

а для Q -импликации:

$$\hat{y}_{CA} = \frac{\sum_{k=1}^N \bar{y}_k \sup_{v \in [0,1]} \left\{ \tau_{\mathbf{A}_r | \mathbf{A}'} \overset{T_2}{\star} S \{1 - v, v\} \right\}}{\sum_{k=1}^N \sup_{v \in [0,1]} \left\{ \tau_{\mathbf{A}_r | \mathbf{A}'} \overset{T_2}{\star} S \{1 - v, v\} \right\}} \quad (2.21)$$

Схема нейро-нечеткого системы соответствующая формуле дефаззификации (??) изображена на рисунке 2.2.

2.6.2 Дефаззификация по методу центра тяжести

(center of gravity)

Если выходное значение блока выработки решения представляет собой единственное агрегированное нечеткое множество B' , следует рассмотреть к использованию данный и последующие методы дефаззификации. Этот метод можно сопоставить со схемой вычисления математического ожидания случайной величины при данном ее распределении.

$$\hat{y}_{CoG} = \frac{\int_Y y \mu_{B'}(y) dy}{\int_Y \mu_{B'}(y) dy}$$

Тогда (??) при использовании данной импликации запишется в виде:

$$\hat{y}_{CoG} = \frac{\int_Y y T_{r=1}^N \left\{ \sup_{v \in [0,1]} \left\{ \tau_{A_r|A'} \overset{T_2}{\star} I(v, \mu_{B_r}(y)) \right\} \right\} dy}{\int_Y T_{r=1}^N \left\{ \sup_{v \in [0,1]} \left\{ \tau_{A_r|A'} \overset{T_2}{\star} I(v, \mu_{B_r}(y)) \right\} \right\} dy} \quad (2.22)$$

Значение \hat{y}_{CoG} в данном способе дефаззификации может быть вычислено с применением численных методов. Однако существует упрощенная схема нахождения выходного значения в данном методе с помощью дискретной формулы центра тяжести в точках центров функций принадлежности термов выходной лингвистической переменной или ф. п. консеквентов правил в базе правил [21]. Эта схема выражается формулой ниже.

$$\hat{y}_{CoG} = \frac{\sum_{k=1}^N \bar{y}_k \mu_{B'}(\bar{y}_k)}{\sum_{k=1}^N \mu_{B'}(\bar{y}_k)}, \quad (2.23)$$

где \bar{y}_k — центр ф. п. нечеткого множества B_k , то есть такое значение y , в котором $\max_y \mu_{B_k}(y) = 1$.

В этом случае формула (??) на основе (??) примет вид:

$$\hat{y}_{CoG} = \frac{\sum_{k=1}^N \bar{y}_k T_{r=1}^N \left\{ \sup_{v \in [0,1]} \left\{ \tau_{A_r|A'} \overset{T_2}{\star} I(v, \mu_{B_r}(\bar{y}_k)) \right\} \right\}}{\sum_{k=1}^N T_{r=1}^N \left\{ \sup_{v \in [0,1]} \left\{ \tau_{A_r|A'} \overset{T_2}{\star} I(v, \mu_{B_r}(\bar{y}_k)) \right\} \right\}}, \quad (2.24)$$

Выражение внутри фигурных скобок оператора $\sup_{v \in [0,1]} \{\cdot\}$, представляющее композицию двух нечетких множеств на пространстве истинности, есть определение *возможности*, то есть соответствие того, что $\tau_{A_r|A'}$ есть τ_{kr} и наоборот. Обозначим величину возможности:

$$\sup_{v \in [0,1]} \left\{ \tau_{A_r|A'} \overset{T_2}{\star} \tau_{kr} \right\} = \Pi_{kr} \quad (2.25)$$

Поскольку импликация в выражении (??eqn:defuz-cog-4)) не зависит от входных данных, то значения $I(v, \mu_{B_r}(\bar{y}_k)) = \tau_{kr}, k = \overline{1, N}, r = \overline{1, N}$ могут быть вычислены предварительно, то есть до использования композиционного правила вывода.

Одна из возможностей упрощения процедуры вывода возникает, когда функции принадлежности термов выходной лингвистической переменной

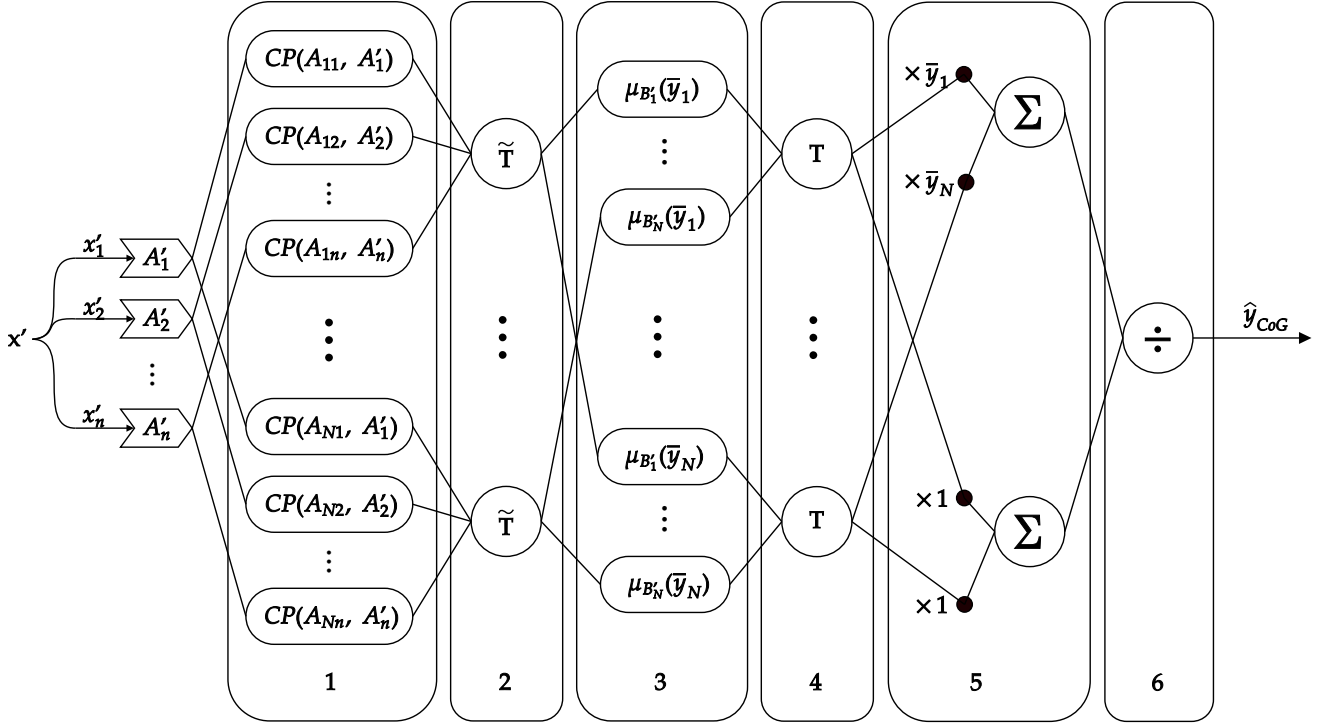


Рисунок 2.3 — Схема нейро-нечеткой системы с использованием дефаззификации по методу центра тяжести

достаточно удалены друг от друга и имеют низкую степень взаимного пересечения, то есть выполняется соотношение $\mu_{B_k}(y_r) \approx 0$ при $k \neq r$, что проиллюстрировано на рисунке 2.4.

Рассмотрим вычисление τ_{kr} для различных категорий импликаций:

— для S -импликации

$$\tau_{kr}(v) = \begin{cases} 1 - v, & \text{если } k \neq r \\ 1, & \text{если } k = r \end{cases}$$

$$\hat{y}_{CoG} = \frac{\sum_{k=1}^N \bar{y}_k T_{r=1}^N \left\{ \sup_{v \in [0,1]} \left\{ \tau_{A_r|A'} \overset{T_2}{\star} (1 - v) \right\} \right\}}{\sum_{k=1}^N T_{r=1}^N \left\{ \sup_{v \in [0,1]} \left\{ \tau_{A_r|A'} \overset{T_2}{\star} (1 - v) \right\} \right\}},$$

— для R -импликации

$$\tau_{kr}(v) = \begin{cases} \delta(v), & \text{если } k \neq r \\ 1, & \text{если } k = r \end{cases}, \quad \text{где } \delta(v) = \begin{cases} 1, & v = 0 \\ 0, & v > 0 \end{cases}$$

$$\hat{y}_{CoG} = \frac{\sum_{k=1}^N \bar{y}_k T_{r=1}^N \left\{ \tau_{A_r|A'}(0) \right\}}{\sum_{k=1}^N T_{r=1}^N \left\{ \tau_{A_r|A'}(0) \right\}},$$

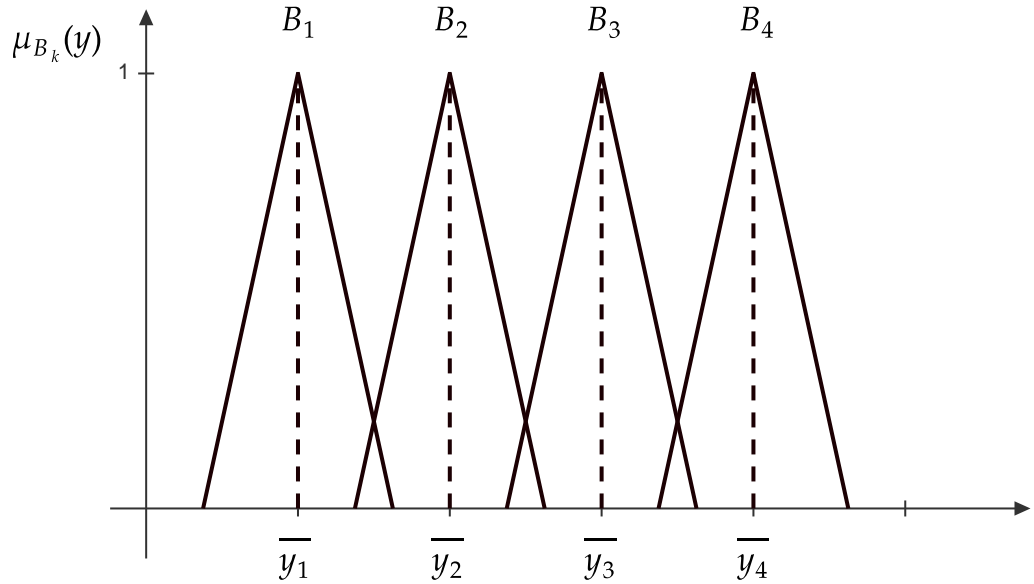


Рисунок 2.4 — Пример нечетких множеств, удовлетворяющих условию $\mu_{B_k}(y_r) = 0$ для $y \neq r$

— для S -импликации

$$\tau_{kr}(v) = \begin{cases} 1 - v, & \text{если } k \neq r \\ \max(1 - v, v), & \text{если } k = r \end{cases}$$

$$\hat{y}_{CoG} = \frac{\sum_{k=1}^N \bar{y}_k T_2 \left\{ \sup_{v \in [0,1]} \left\{ \tau_{A_k|A'} \overset{T_2}{\star} \max(1 - v, v) \right\} T_{r=1, r \neq k}^N \left\{ \sup_{v \in [0,1]} \left\{ \tau_{A_r|A'} \right\} \right. \right\}}{\sum_{k=1}^N T_2 \left\{ \sup_{v \in [0,1]} \left\{ \tau_{A_k|A'} \overset{T_2}{\star} \max(1 - v, v) \right\} T_{r=1, r \neq k}^N \left\{ \sup_{v \in [0,1]} \left\{ \tau_{A_r|A'} \right\} \right. \right\}}$$

Можно организовать вычисление всех значений b_{kr} τ_{kr} , но использовать разреженные матрицы в качестве структуры данных для хранения значений где $b_{kr} > 0$.

Приведенные выкладки справедливы не только для функций принадлежности отдельных термов, а и для набора кластеризованных в небольшие группы функций принадлежности со значительной степенью пересечения. При такой конфигурации выходного нечеткого пространства нет необходимости включать в процесс вывода правила, в которых функции принадлежности консеквента имеют низкий уровень пересечения с ф. п. правил, имеющих высокий уровень срабатывания для данного входа нечеткой системы.

Тогда

2.7 Прогнозирование временных рядов на основе нечетких систем логического типа с использованием нечеткого значения истинности

Для временного ряда $\mathbf{y}_t = (y_1, \dots, y_t)$ величина y_t представляет измеренное значение наблюдаемой переменной в момент времени t . Ставится задача предсказания значений \hat{y}_{t+h} для заданного горизонта предсказания h .

Модель временного ряда $f(\cdot)$ порядка p использует последние p значений до момента t для оценки значения:

$$\hat{y}_{t+h} = f(y_{t-p}, \dots, y_t),$$

где p - размер лагового окна.

При моделировании временных последовательностей с использованием нейро-нечетких систем каждое значение $y_t \in Y$ фазифицируется в нечеткое множество A_t . Эти нечеткие множества составляют множество термов лингвистической переменной \tilde{A} , определенной на базовом множестве Y . Такая система принимает p входов, а правила в ее базе знаний устанавливают нечеткую последовательно-временную связь в рамках заданного окна $p + 1$. Параметр p называется порядком нечеткой системы прогнозирования временных рядов.

База правил в такой системе представляется набором из N правил вида:

$$\begin{aligned} R_k : \text{Если } y_{t-p} \text{ есть } A_{k1} \text{ и } \dots \text{ и } y_t \text{ есть } A_{kp}, \\ \text{то } y_{t+1} \text{ есть } A_{kp+1}, \end{aligned} \quad k \in \overline{1, N}, \quad (2.26)$$

где каждое нечеткое отношение $T_1 \{A_{k1}, \dots, A_{kp}\} \rightarrow A_{kp+1}$, заданное в правиле R_k , выражает единицу **логических** знаний о моделируемом протекающем во времени процессе.

Определив НЗИ $CP(\mathbf{A}_k, \mathbf{A}')$ для антецедента правила R_k согласно (2.5) и (2.10), можно переписать правило R_k в виде:

$$\begin{aligned} R_k : \text{Если } CP(\mathbf{A}_k, \mathbf{A}'), \\ \text{то } y_{t+1} \text{ есть } A_{kp+1}, \end{aligned} \quad k \in \overline{1, N}.$$

Вычисленное значение истинности для каждого правила выражает соответствие среза измерений порождаемых некоторой величины некоторой единице знаний о динамике этого процесса.

Если используется логический метод вывода на основе дефазификации по центру тяжести, то согласно (??) нечеткий вывод выражается:

$$\hat{y}_{t+1} = \frac{\int_Y y_{t+1} \prod_{k=1, N} \left\{ \sup_{v \in [0,1]} \left\{ \tau_{A_k|A'}(v) \overset{T_2}{\star} I(v, \mu_{A_{k_{p+1}}}(y_{t+1})) \right\} \right\} dy_{t+1}}{\int_Y \prod_{k=1, N} \left\{ \sup_{v \in [0,1]} \left\{ \tau_{A_k|A'}(v) \overset{T_2}{\star} I(v, \mu_{A_{k_{p+1}}}(y_{t+1})) \right\} \right\} dy_{t+1}} \quad (2.27)$$

Входы нечеткой системы по каждому измерению могут описываться одной и той же лингвистической переменной, заданной на базовом множестве области значений временного ряда и имеющей одинаковую порождающую процедуру для терм-множеств.

Обучение нечетких систем прогнозирования временных рядов . Ранние подходы следовали более типовому набору шагов для построения нечетких систем [Chellai2022]: пространство значений временного ряда разбивалось на пересекающиеся участки для формирования термов единственной лингвистической переменной \tilde{A} , а база правил формировалась по принципу наибольшей степени принадлежности для каждого экземпляра входных данных, например, с помощью популярного из-за своей простоты метода [23]. В широко цитируемом подходе Чена [Chen1996] разбиение производилось на равные отрезки. В более поздних методах формирование сразу базы правил с соответствующими терм-множествами из данных, то есть более точное выделение шаблонных отрезков из временных рядов, стало более распространенным подходом, из-за ограничений в увеличении точности нечеткой системы при разбиения базового множества лингвистической переменной \tilde{A} со сложной функцией плотности распределения значений. Таким образом одни и те же методы можно применять для разбиения пространства значений временных рядов (одно измерение) или для деления пространства окон временных рядов (когда последовательность значений в окне временного ряда интерпретируется как точка в n -мерном пространстве). В последнем случае посредством разбиения пространства окон временного ряда может осуществляться формирование базы правил.

Распространенной группой таких методов являются подходы на основе различных алгоритмов кластеризации [29]: k -средних, алгоритмы учитывающие плотность распределения точек, агломеративная кластеризация. Эволюционные подходы, например, Particle Swarm Optimization (PSO).

Отдельно [24, 25]

Стоит также отметить гибридные подходы построения и обучения нечетких систем моделирования временных рядов в комбинации с другими моделями машинного обучения, такими как, Support Vector Machine (SVM), Long-Sort Term Memory (LSTM), Transformer, и статистические модели, например, Auto Regressive Integrated Moving Average (ARIMA).

Для реализации сильных сторон от использования нечеткого моделирования нужно отразить информацию о прогнозируемой величине при выборе метода ее фаззификации. Существующие подходы фаззификации значений временных рядов комбинируют различные формы функций принадлежности, способы оценки истинного значения в термах и формализации неопределенности. В [14] описан подход к фаззификации измеренных значений временного ряда в нечеткие множества с гауссовыми функциями принадлежности. Измеренные значения полагаются истинными и задают центры гауссовых функций принадлежности, а среднеквадратичное отклонение оценивается как среднеквадратичная разность между соседними значениями в некотором окне вокруг данной точки во временной последовательности. Написать формулу В [28] для фаззифицированных таким образом значений в нечеткие множества выполняется переоценка истинных значений посредством преобразования взвешенным скользящим средним центров гауссовой ф. п. нечетких множеств. Написать формулу

Для выбора используемой импликации и метода дефаззификации изучен анализ точности в задаче регрессии приведенный в работах [] и [] соответственно.

2.8 Выводы

В главе приведена постановка задачи нечеткого логического вывода в канонической форме Заде. Затем описана альтернативная схема нечеткого логического вывода с использованием нечеткого значения истинности, в которой база правил приобретает новую структуру правил вида: «Если *истинно*, то B_k ». Также формально описана схема вычисления свертки НЗИ на основе НЗИ по отдельным входам с помощью расширенной t -нормы — \tilde{T} .

Проведен анализ...

В главе приведены итоговые формулы нечеткого вывода на основе нечеткого значения истинности для различных методов дефаззификации. Также выведены упрощенные формулы вычисления дефаззифицированного значения для различных специальных категорий импликаций.

Дано формальное описание применения разработанной модели нечеткого логического вывода в задаче прогнозирования временных рядов. Приведены существующие подходы фаззификации значений временных рядов и методы автоматического подбора параметров термов лингвистической переменной и формирования базы правил на основе наборов обучающих данных.

Глава 3. Программная реализация разработанного метода нечеткого вывода с применением технологии CUDA

Согласно схемам [**<empty citation>**] изображенным в предыдущем разделе, для достижения высокой производительности процедура нечеткого вывода может быть скомпонована из последовательности параллельных участков независимых вычислений и сверток. Распространенным подходом реализации параллельных вычислений является использование вычислений на графическом процессоре.

Для удобства реализация выполнялась не посредством прямого использованием API библиотеки CUDA, а с помощью библиотеки реализации высокопроизводительных вычислений - **Kokkos ??**, предназначенной для портативного и эффективного параллельного программирования на различных аппаратных архитектурах, включая многоядерные процессоры, графические процессоры NVIDIA/AMD и другие ускорители. Интерфейс библиотеки позволяет абстрагироваться от деталей низкоуровневого параллелизма, позволяя разработчикам писать код на C++ с одним исходным кодом, который может быть оптимизирован для различных платформ без существенных изменений. Ключевые функциональные возможности библиотеки упростившие выполнение программного реализации разработанного ранее метода:

- Модели параллельного выполнения: абстрагирование параллельных циклов (например, `parallel_for`, `parallel_reduce`) и рабочих процессов на основе задач.
- Управление памятью: Автоматизированная обработка пространств памяти (например, между хостом и устройством) и расположением данных для оптимизации схем доступа и минимизации объема передаваемых данных.
- Поддержка серверной части: интеграция с такими моделями программирования, как CUDA, HIP, OpenMP и SYCL, для обеспечения кроссплатформенной совместимости.
- Переносимость производительности: Обеспечивает эффективное использование ресурсов (потoki, векторизация) с учетом особенностей каждой архитектуры.

Kokkos широко используется в научных вычислениях и HPC-приложениях, упрощая разработку масштабируемых кодов при сохранении производительности на постоянно развивающемся оборудовании.

При выполнении процедуры нечеткого вывода для набора входных данных можно выполнять процедуру вывода по каждому входному экземпляру независимо, то есть параллельно. Кроме того, можно предусмотреть отбор небольшого в сравнении с общим количеством ограниченного подмножества правил нечеткой системы, наиболее релевантных входному экземпляру. Сам алгоритм нечеткого вывода не предусматривает наличия промежуточной информации, которой бы нужно было обмениваться между вычислениями по другим экземплярам входных данных. Тогда, поскольку алгоритм нечеткого вывода для каждого экземпляра входных данных можно реализовать при ограниченном небольшом объеме входных и промежуточных данных, имеет смысл разместить эти данные в памяти предоставляющей наибольшую скорость доступа к данным, которая в технологии CUDA соответствует разделяемой памяти внутри CUDA-блока, а сами вычисления над этими данными также реализовать внутри этого CUDA-блока.

Из описанного выше функционала библиотеки Kokkos для размещения и доступа к данным в разделяемой памяти CUDA-блока крайне удобно использовать `Kokkos::View` с опорой на широко эксплуатируемую в среде C++ разработчиков идиому RAII (Resource acquisition is initialization — получение некоторого ресурса неразрывно совмещается с инициализацией, а освобождение — с уничтожением объекта), которая позволяет избавиться от необходимости «ручного» вычисления адресов и смещений, располагаемых в разделяемой памяти, программных объектов, что особенно актуально при переиспользовании сегментов разделяемой памяти, занимаемых временными данными, в дальнейших шагах алгоритма.

Для удобства ...

```
using ScratchSpace =
    typename Kokkos::Cuda::scratch_memory_space;
template <typename DataType>
using ScratchView =
5   Kokkos::View<DataType,
                ScratchSpace,
                Kokkos::MemoryTraits<Kokkos::Unmanaged>>;
```


Для выделения памяти с помощью `Kokkos::View` требуется получить дескриптор разделяемой памяти текущего CUDA-блока с помощью вызова `Kokkos::TeamHandleConcept<>::team_shmem()`, который описывает всю карту разделяемой памяти данного CUDA-блока.

При оценке объема разделяемой памяти, необходимой для размещения данных внутри CUDA-блока. Рациональное использование только необходимого объема разделяемой памяти позволит планировщику на графическом процессоре поставить на исполнение сразу несколько CUDA-блоков на один *поточковый мультипроцессор* (*streaming multiprocessor*).

Вычисления в рамках нити

Поскольку вычисление свертки НЗИ для всех правил нечеткой системы составляет половину сложности алгоритма нечеткого вывода и в последствии будет неоднократно использоваться при агрегации правил, необходимо обеспечить единоразовое вычисление функции принадлежности свертки НЗИ (или ее аппроксимации), в результате которого будут известны значения последней на всей ее области определения.

3.1 Вычисление нечеткого значения истинности посредством дискретизации

При реализации вычисления НЗИ, когда функции принадлежности нечетких множеств A и A' заданы гауссовыми функциями, может возникнуть сложность, когда ф. п. НЗИ вырождается в близкий к некоторой асимптоте (горизонтальной или вертикальной) вид. Для изучения условий возникновения таких ситуаций перепишем формулу (1.2) ниже и обозначим ее составные компоненты через α , β , $\varphi(v)$:

$$\exp \left(- \left(\underbrace{\frac{a_1 - a_2}{b_2}}_{\alpha} \pm \underbrace{\frac{b_1}{b_2}}_{\beta} \underbrace{\sqrt{-\ln v}}_{\varphi(v)} \right)^2 \right).$$

Тогда вычисление значения (1.2) можно представить в виде композиции функций

$$\mu_{CP(A,A')}(v) = \exp(-z(v)^2), \quad (3.1)$$

и

$$z(v) = \alpha \pm \beta \varphi(v). \quad (3.2)$$

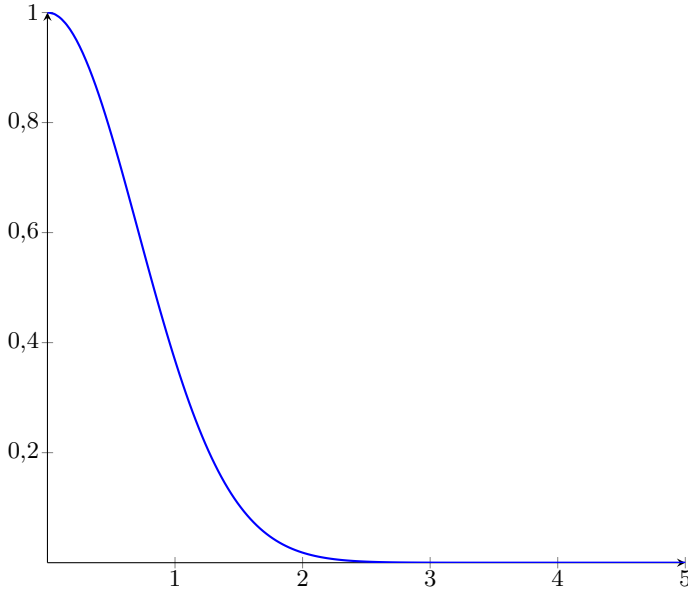


Рисунок 3.1 — График функции $\exp(-z^2)$

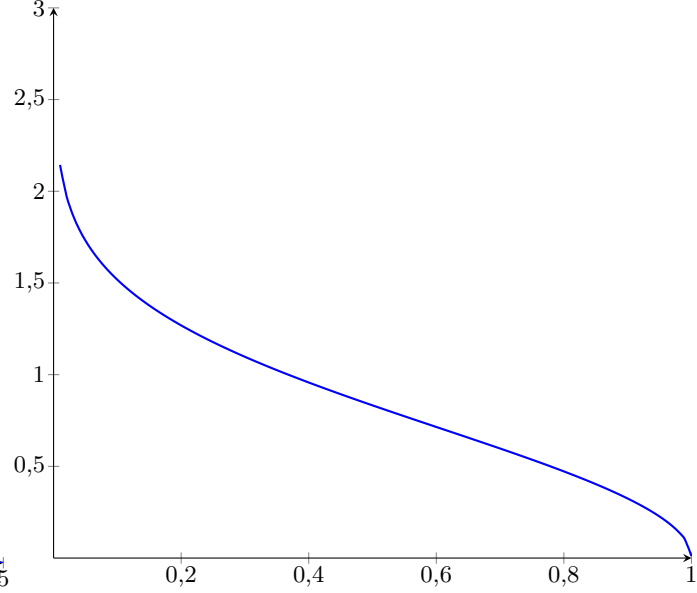


Рисунок 3.2 — График функции $\varphi(v) = \sqrt{-\ln v}$

Для иллюстрации условий возникновения описанных вырождений функции (1.2) графики функций (3.1) и (3.2) при $\alpha = \beta = 1$ приведены на рисунках 3.1 и 3.2 соответственно. Из рисунка 3.2 видно, что вырождение выражения (3.1) в близкий к асимптоте вид возможно при очень больших или очень малых значениях α и β . Данное наблюдение проиллюстрировано на рисунке 3.3.

В ситуациях, изображенных на рисунках 3.3а, 3.3б, 3.3г и 3.3д, функция принадлежности НЗИ имеет своей асимптотой вертикальную прямую $v = v_0$. При реализации вычисления НЗИ с использованием расчетной сетки, связанная с тем, что ф. п. НЗИ в точке может быть вообще не определена (рис. 3.3а и 3.3д) или ф. п. НЗИ в точке v_0 имеет значение сильно отличающееся от значений в ближайших точках расчетной сетки v_1 и v_2 при $v_0 \in (v_1, v_2)$, из-за чего значение НЗИ в бесконечно малой окрестности точки v_0 будет упущено.

Из рисунка 3.1 видно, что максимальное значение выражения (3.1), равное 1, достигается при $z = 0$, а из рисунка 3.2 видно, что функция (3.2) обязательно пересекает ось абсцисс, то есть максимальное значение ф. п. НЗИ всегда равно 1. Тогда для решения проблемы «просеивания» значения $\mu_{CP(A,A')}(v_0)$

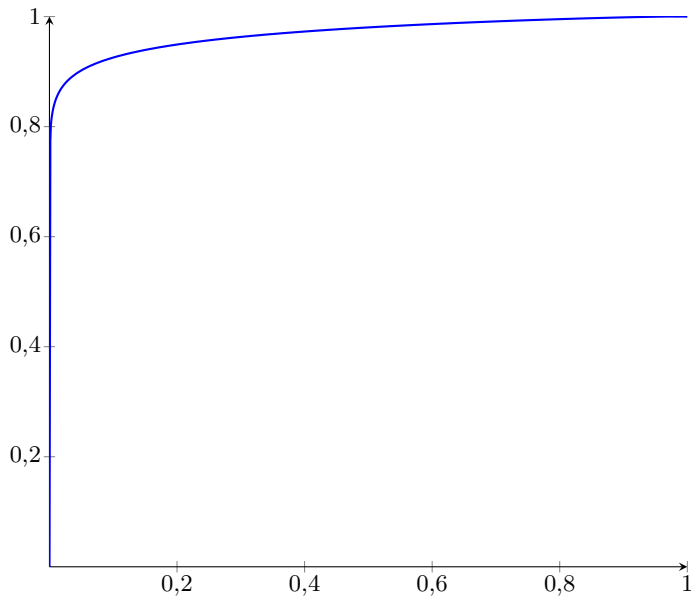
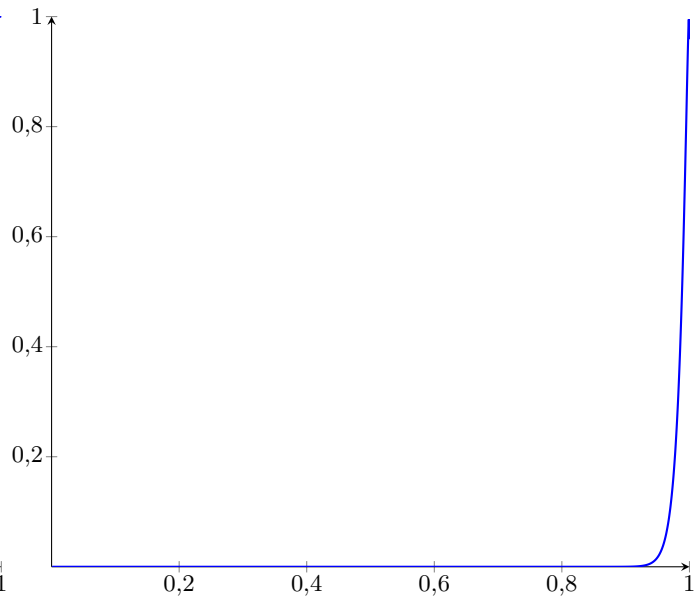
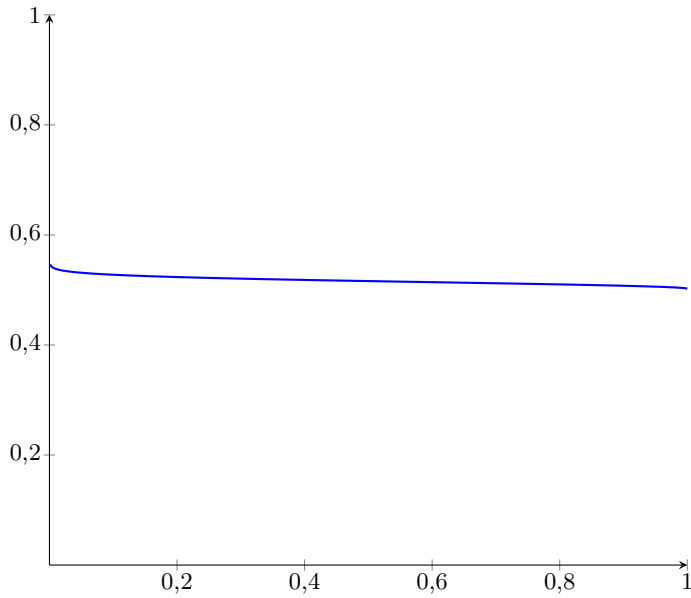
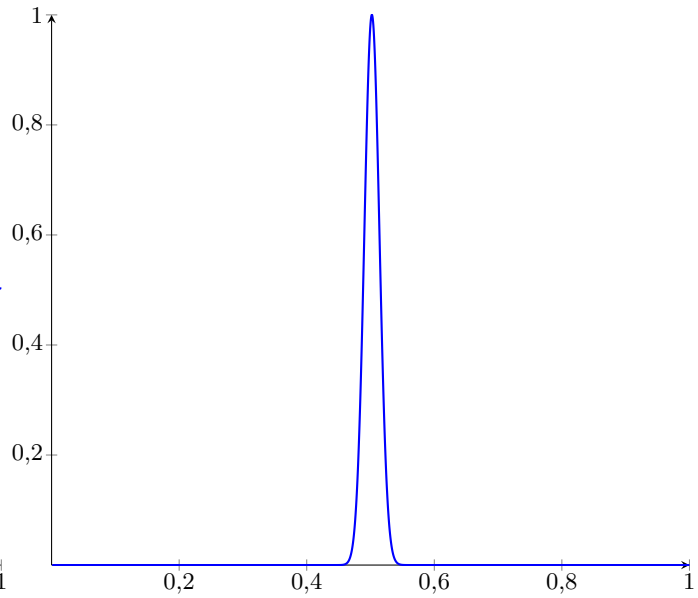
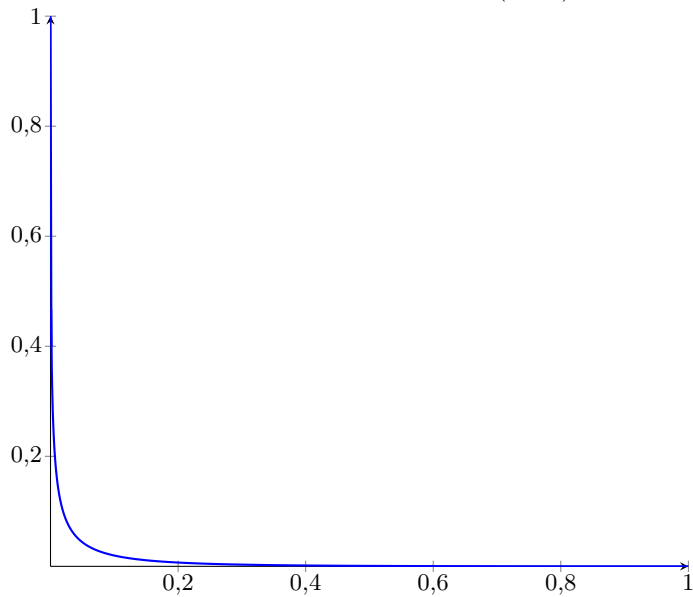
а) $\mu_{CP(A,A')}(v)$ при $\alpha \approx 0, \beta \approx 0$ б) $\mu_{CP(A,A')}(v)$ при $\alpha \approx 0, \beta \gg 0$ в) $\mu_{CP(A,A')}(v)$ при $|\alpha| \gg 0, \beta \approx 0$ г) $\mu_{CP(A,A')}(v)$ при $|\alpha| \gg 0, \beta \gg 0$ д) $\mu_{CP(A,A')}(v)$ при $|\alpha| \gg 0, \beta = 1$

Рисунок 3.3 — Случаи вырождения ф. п. НЗИ в близкий к асимптоте вид.

сквозь точки расчетной сетки можно вычислить координату точки v_0 , в которой функция (1.2) принимает свое максимальное значение, а затем положить значение в ближайшей к v_0 точке расчетной сетки равным этому максимуму, то есть значению 1. Для этого выразим аналитически значение v_0 , в котором (1.2) принимает значение 1.

$$\begin{aligned}
 \exp\left(-\left(\frac{(a_1 - a_2) \pm b_1 \sqrt{-\ln v_0}}{b_2}\right)^2\right) &= 1 \\
 \exp\left(-\left(\frac{(a_1 - a_2) \pm b_1 \sqrt{-\ln v_0}}{b_2}\right)^2\right) &= \exp(0) \\
 \frac{(a_1 - a_2) \pm b_1 \sqrt{-\ln v_0}}{b_2} &= 0 \\
 \pm b_1 \sqrt{-\ln v_0} &= a_1 - a_2 \\
 \sqrt{-\ln v_0} &= \pm \frac{a_1 - a_2}{b_1} \\
 \ln v_0 &= -\left(\frac{a_1 - a_2}{b_1}\right)^2 \\
 v_0 &= \exp\left(-\left(\frac{a_1 - a_2}{b_1}\right)^2\right) \quad (3.3)
 \end{aligned}$$

Внедрение выражения (3.3) в алгоритм вычисления НЗИ позволит обеспечить соответствие определению дискретизированного представления ф. п. НЗИ. Строго говоря ф. п. на рисунке определена в точке v_0 и не требует... НЗИ с функцией принадлежности, имеющей горизонтальную асимптоту, как на рисунке 3.3в может вызвать сложность при построении алгоритма вычисления НЗИ на использовании метода градиентного спуска вместо дискретизации с фиксированным шагом расчетной сетки.

Вычисление нечеткого значения истинности для каждого правила по каждому входу отдельно предварительным этапом приведет к линейному увеличению объема необходимой памяти. Поскольку вычисление НЗИ является довольно легкой операцией, лучшим решением будет встраивание вычисления НЗИ по каждому входу в процедуру свертки НЗИ.

Формула (??) предлагает построение процедуры свертки НЗИ в виде дерева попарных сверток, для сохранения промежуточных результатов которых потребуется выделить объем памяти, также имеющего фактически линейный

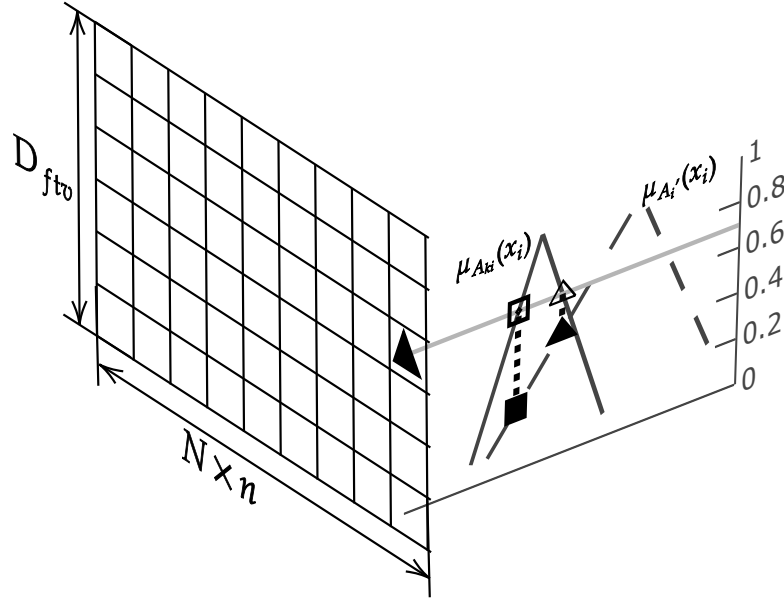
порядок зависимости от количества входов нечеткой системы. Возвращаясь к формуле (??) свертки НЗИ по входам в [] предложен параллельный алгоритм свертки НЗИ на расчетной сетке с постоянным шагом. Данный алгоритм итеративно вычисляет значение светки НЗИ в каждой точки расчетной сетки продвигаясь от точки в пространстве истинности 1 к точке 0. Для каждого правила отдельно потребуется объем памяти, необходимый для сохранения значений свертки НЗИ в каждой точке расчетной сетки и для сохранения текущего максимального значения НЗИ по каждому входу на данной итерации алгоритма.

Преимуществом такого подхода представления НЗИ в программе является возможность быстрого нахождения значения функции принадлежности НЗИ в некоторой точке пространства истинности. Пара ближайших к этой точке позиций в массиве значений НЗИ на расчетной сетке определяется на основе значения фиксированного шага расчетной сетки. Для вычисления непосредственно значения НЗИ используется линейная интерполяция для найденной пары.

Поскольку при небольшом количестве входных переменных нечеткой системы большие участки расчетной сетки ф. п. светки НЗИ соответствуют ф. п. НЗИ по одному из входов, при том же объеме памяти, необходимой для хранения результата свертки НЗИ, можно перейти к переменному шагу расчетной сетки с увеличением сложности функции аппроксимации участков ф. п. свертки НЗИ, принадлежащий ф. п. НЗИ одного из входов. Схема вычислений такого алгоритма схожа со схемой алгоритма при использовании постоянного шага расчетной сетки, а на каждой итерации определяется точка ...

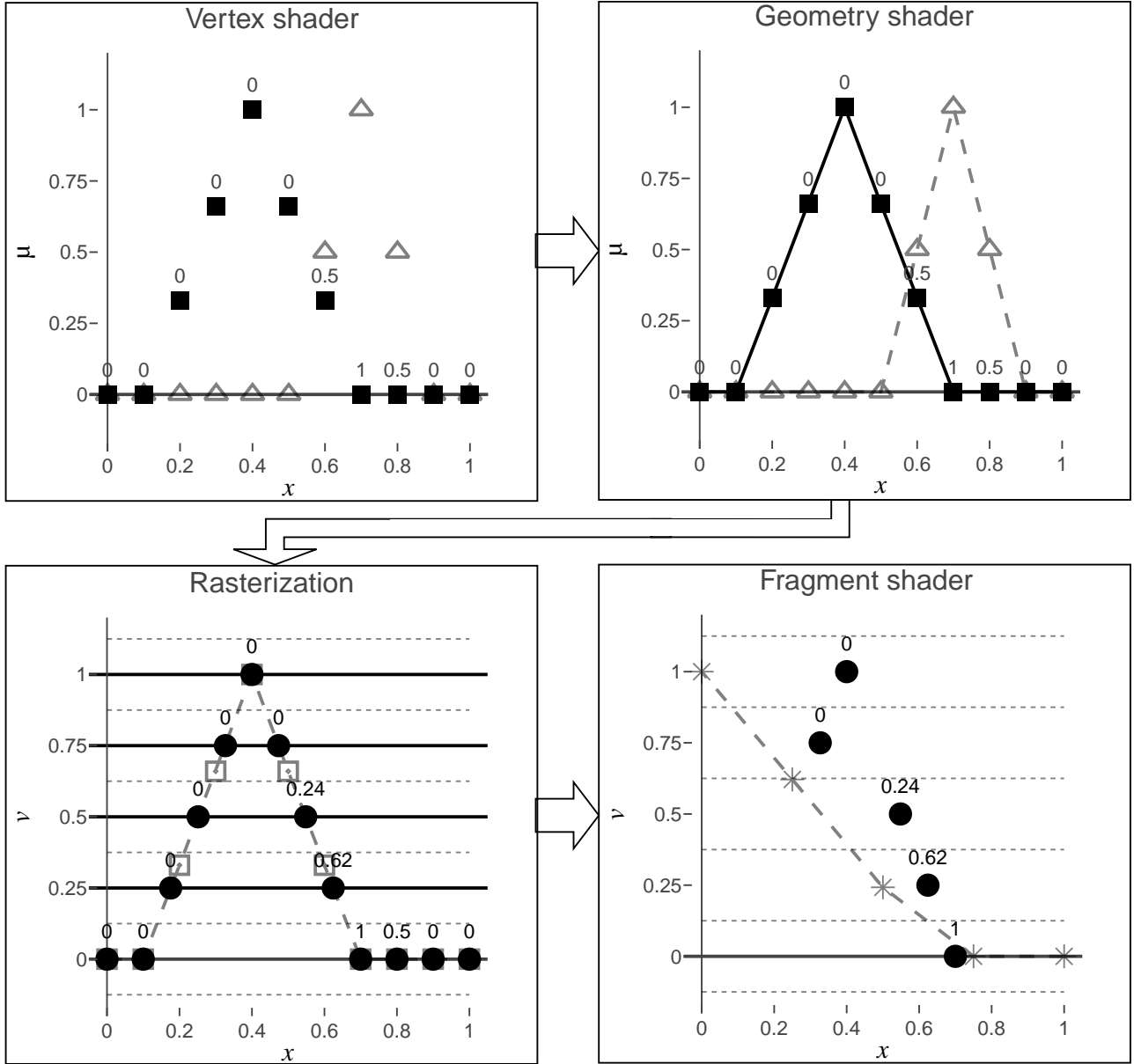
Ввиду необходимости использования большой размерности расчетной сетки для получения высокой точности дискретизации для размещения результата свертки НЗИ потребуется существенный объем разделяемой памяти. Например, при использовании для задания вычисленных значений вещественных чисел одинарной точности (4 байта) и размерности расчетной сетки равной 100 сохранение результата свертки НЗИ в нечеткой системе с 50 правилами в базе правил потребуется $4 \text{ байта} \times 100 \times 50 = 20000$ байт разделяемой памяти. Согласно документации CUDA, устройства с *набором вычислительных возможностей* (*compute capabilities*) версии 7.5 (и некоторых более низких версий) максимальный объем разделяемой памяти на одном потоковом мультипроцессоре равен 64 Кб, а на устройствах с набором вычислительных возможностей более высокой версии — предусматривается 100 - 164 Кб разделяемой памяти. Следовательно

время нечеткого вывода может быть сокращено за счет сокращения времени работы отдельного CUDA-блока путем увеличения количества нитей. Однако с другой стороны слишком большое количество нитей внутри CUDA-блока упрется в ограниченное количество регистровой памяти и ограниченное количество арифметико-логических модулей внутри потокового мультимикропроцессора (распределяемых между нитями 4-мя планировщиками на большинстве версий наборов вычислительных возможностей).



3.2 Алгоритм свертки НЗИ при $T_1 = \min$ и T_3 - неубывающая по всем аргументам

При дискретизированном вычислении ф. п. свертки НЗИ в некоторой точке расчетной сетки $v_j \in [0, 1]$ потребуется просматривать значения НЗИ в отрезке $[v_j, 1]$, то есть имеет вычислительную сложность $O(D_{ftv})$, где D_{ftv} — число точек расчетной сетки ф. п. НЗИ. Таким образом вычисление свертки НЗИ по формуле (2.12) может быть распараллелено по точкам расчетной сетки, а нахождение $\tau_{A_k|A'}(v_j)$ использует операцию свертки, которая имеет ограниченную возможность распараллеливания, но в целом требует большого количества повторных вычислений значений $\tau_{A_{ki}|A'_i}(v_k)$, $v_k \in [v_j, 1]$. В [30] предложен алгоритм вычисления свертки НЗИ сразу по всем входам с использованием техники ди-



намического программирования, имеющий линейную зависимость от размера расчетной сетки D_{ftv} . Алгоритм приведен на алг. 1.

Данный алгоритм итеративно вычисляет значения $\tau_{A_k|A'}(v_j)$, продвигаясь от точки $v_j = 1$ к $v_j = 0$. При этом распараллеливаются вычисления по n входам. В i -х элементах массива max_ftv хранятся максимальные значения $\tau_{A_{ki}|A'_i}(v_k)$ для j -й итерации, что избавляет от необходимости поиска этих значений на каждой итерации. Также, поскольку на j -й итерации значение $\tau_{A_k|A'}(v_j)$ выражается из значений НЗИ по входам в точках $v_{ki}, i = \overline{1, n}$, то, с учетом ограничения алгоритма $T_1 = min$ и согласно выражению из формулы (2.10)

$$T_1 v_{ki} = v_j,$$

Algorithm 1 Алгоритм свертки НЗИ при $T_1 = \min$ и $T_3(a, b) \geq T_3(c, d)$ если $a > c$ или $b > d$

Require: $ftv_i, i = \overline{1, n}$

$max_ftv[i] = 0;$

for $v_j = 1 \dots 0$ **do**

$s \leftarrow \{ftv_i[v_j] \mid ftv_i[v_j] \geq max_ftv[i]\};$

$max_ftv[i] \leftarrow \max(max_ftv[i], ftv_i[v_j]);$

$v_max \leftarrow \max_i \{ftv_i[v_j]\}, v_max_index \leftarrow \arg \max_i \{ftv_i[v_j]\};$

if $s = \emptyset$ & $i = v_max_index$ **then**

$r[i] \leftarrow v_max;$

else

$r[i] \leftarrow max_ftv[i];$

end if

$ftv'[v_j] \leftarrow T_3 \{r[i]\};$

end for

return ftv'

для одного из значений v_{ki} необходимо выполнение $V_{ki} = v_j$. Для этого в алг. 1 используется максимальное значение $\tau_{A_{ki}|A'_i}(v_j), i = \overline{1, n}$, когда ни по одному входу не обнаружено нового максимума ф. п. НЗИ в точке v_j .

Для лучшего понимания работу алгоритма на каждой итерации можно представить с помощью диаграммы «роза ветров» с количеством осей равным n , в центре диаграммы находится координата осей равная 1, а сама «роза» проходит через текущие максимумы ф. п. НЗИ на отрезке $[v_j, 1]$.

При

3.3 Реализация дефаззификации

Нахождение начального приближения

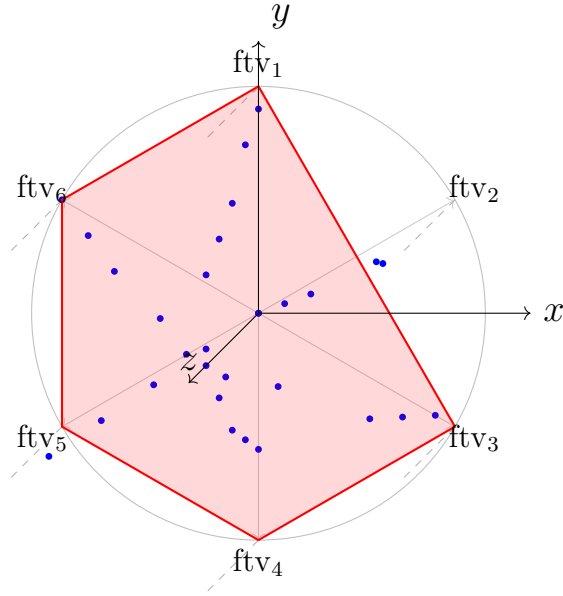


Рисунок 3.4 — 3D wind rose diagram for 6 directions with function values plotted vertically.

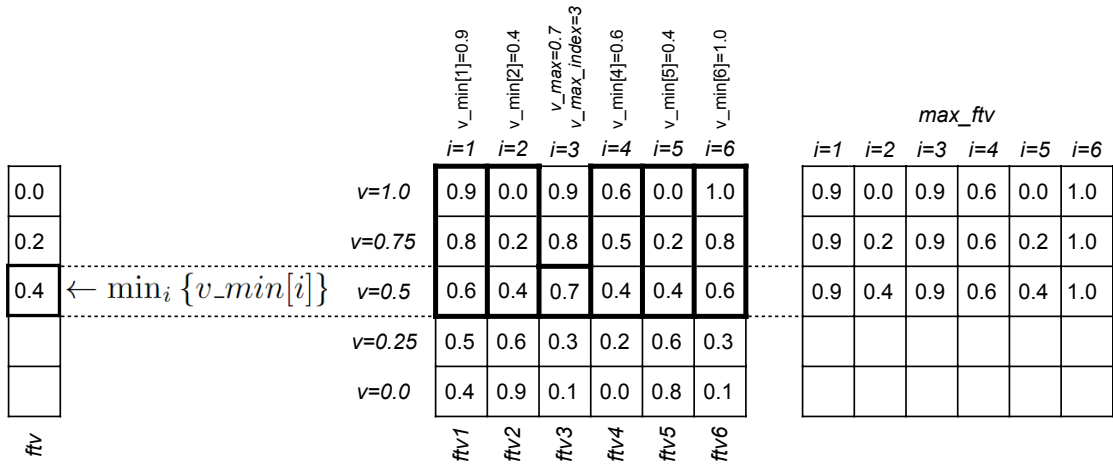


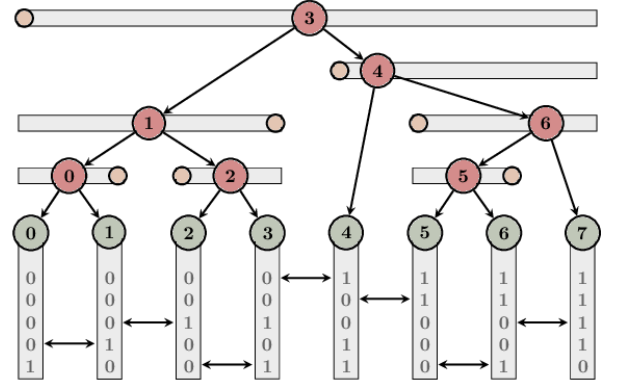
Рисунок 3.5 — Иллюстрация работы алгоритма свертки НЗИ для нескольких входов для расширенной \tilde{T} -нормы при $T_1 = \min$ и $T_3 = \min$.

3.4 Использование библиотеки ArborX

В [31] авторами библиотеки **ArborX** предложен алгоритм для эффективного построения *иерархии ограничивающих объемов*. Данная структура данных используется для эффективного поиска ближайшей точки в n -мерном пространстве. Алгоритм выполняет построение структуры поиска за $O(N \times \log(N))$, где N - количество точек в исходном наборе. Логика поиска места в иерархии

	x:	0	1	2	3	4	5	6	7
		000	001	010	011	100	101	110	111
y: 0	000	000000	000001	000100	000101	010000	010001	010100	010101
1	001	000010	000011	000110	000111	010010	010011	010110	010111
2	010	001000	001001	001100	001101	011000	011001	011100	011101
3	011	001010	001011	001110	001111	011010	011011	011110	011111
4	100	100000	100001	100100	100101	110000	110001	110100	110101
5	101	100010	100011	100110	100111	110010	110011	110110	110111
6	110	101000	101001	101100	101101	111000	111001	111100	111101
7	111	101010	101011	101110	101111	111010	111011	111110	111111

(a) z-curve



(b) apetrei

Рисунок 3.6 — Две иллюстрации: графика z-curve и apetrei.

ограничивающих объемов для каждой отдельной точки помещена в отдельную нить параллельных вычислений, после чего остается только логарифмическая компонента сложности, соответствующая восходящему обходу промежуточного дерева и помещения точки в подобранный узел **с использованием атомарной операции**. В основе алгоритма лежит группировка близко расположенных точек для более быстрого подбора соответствующего ограничивающего объема за счет проецирования точки из n -мерного пространства на Z -кривую посредством вычисления кода Мортон для каждой точки. Согласно эвристике, лежащие рядом на Z -кривой точки располагаются достаточно близко и исходном n -мерном пространстве. Кроме того в статье описан модифицированный алгоритм Апетрея [1], обеспечивающий обход иерархии на этапе поиска ближайших точек без необходимости помещения цепочки пройденных родительских узлов в стек за счет передачи информации об альтернативных узлах-кандидатах из родительских узлов в дочерние. Такой подход значительно снижает количество потребляемой памяти для сохранения промежуточной информации при выполнении этапа поиска.

3.5 Алгоритм построения базы правил

3.6 Анализ производительности

На оборудовании ????? данная реализация имеет следующую производительность

3.7 Выводы

Глава 4. Применение разработанной нечеткой модели для прогнозирования временных рядов в задачах экономики и финансов

Прогнозирование временных рядов в экономике и финансах играет ключевую роль как инструмент анализа и предсказания динамики последовательно изменяющихся данных. Оно служит основой для принятия обоснованных решений в условиях неопределенности, обеспечивая оценку будущих тенденций, рисков и возможностей. Это позволяет субъектам экономической и финансовой деятельности — от индивидуальных инвесторов до государственных структур — оптимизировать управление ресурсами, минимизировать потери и формировать долгосрочные стратегии.

Стратегическое планирование и бюджетирование (обоснование финансовых планов, оптимизация ресурсного распределения), Управление рисками и инвестиционные решения

В государственных и корпоративных бюджетах прогнозы временных рядов используются для оценки объёмов доходов и расходов, планирования денежных потоков и определения ключевых показателей эффективности. У компаний прогнозирование сезонных и трендовых колебаний продаж помогает корректировать производственные мощности, склады и персонал, тем самым минимизируя издержки и снижая риск дефицита или перепроизводства. Прогнозы временных рядов служат основой для разработки моделей оптимального распределения активов и стратегий ребалансировки портфеля в зависимости от ожидаемых рыночных движений.

Одномерные модели оправданы, когда целевой ряд обладает высокой автокорреляцией, слабо зависит или отсутствует информация о значениях внешних факторов, ограничены вычислительные и временные ресурсы для оценки взаимосвязей измерений. Примеры: краткосрочное прогнозирование инфляции, прогнозирование индекса S&P 500. Многомерные методы необходимы, когда влияние других временных рядов или внешних предикторов существенно для точности прогноза, доступна качественная информация об этих переменных или когда доступна информация о их будущих значениях, а также когда необходимо исследовать влияние «шока» в одной переменной на остальные. Примеры: оценка влияния нефтяных шоков на экономику Нигерии.

Область и ситуации в которой применима разработанная модель (авто-регрессионная)

4.1 Задача прогнозирования стоимости ценных бумаг

4.2 Задача прогнозирования динамики использования банковского продукта / CLTV по продукту

4.3 Описание набора данных

4.4 Решение задачи с использованием разработанной нечеткой модели

4.5 Выводы

Заключение

Основные результаты работы заключаются в следующем.

1. На основе анализа ...
2. Численные исследования показали, что ...
3. Математическое моделирование показало ...
4. Для выполнения поставленных задач был создан ...

И какая-нибудь заключающая фраза.

Последний параграф может включать благодарности. В заключение автор выражает благодарность и большую признательность научному руководителю Иванову И. И. за поддержку, помощь, обсуждение результатов и научное руководство. Также автор благодарит Сидорова А. А. и Петрова Б. Б. за помощь в работе с образцами, Рабиновича В. В. за предоставленные образцы и обсуждение результатов, Занудятину Г. Г. и авторов шаблона *Russian-Phd-LaTeX-Dissertation-Template* за помощь в оформлении диссертации. Автор также благодарит много разных людей и всех, кто сделал настоящую работу автора возможной.

Словарь терминов

Мягкие вычисления : Методология использования неточных и математически строго не обоснованных методов и алгоритмов при решении задач, для которых не существует строгих подходов, позволяющих получить точный результат за приемлемое время.

Нечеткое значение истинности : Короткий текст, использующий все или почти все буквы алфавита

Список литературы

- [1] Jamal Toutouh, Diego Rossit и Sergio Nesmachnow. — «Soft computing methods for multiobjective location of garbage accumulation points in smart cities». — В: *Annals of Mathematics and Artificial Intelligence* 88.1–3 (июнь 2019), с. 105–131. — DOI: 10.1007/s10472-019-09647-5. — URL: <http://dx.doi.org/10.1007/s10472-019-09647-5>.
- [2] Pawel Gora и Przemyslaw W. Pardel. — «Application of Genetic Algorithms and High-Performance Computing to the Traffic Signal Setting Problem». — В: *CEUR Workshop Proceedings*. — Т. 1492. — URL: https://ceur-ws.org/Vol-1492/Paper_14.pdf. — 2015, — Paper 14. — URL: https://ceur-ws.org/Vol-1492/Paper_14.pdf.
- [3] Jie Zhang и Meihong Wang. — «Special Issue: Neural Networks, Fuzzy Systems and Other Computational Intelligence Techniques for Advanced Process Control». — В: *Processes* 11.8 (2023), с. 2278. — DOI: 10.3390/pr11082278. — URL: <https://doi.org/10.3390/pr11082278>.
- [4] F. J. Vivas, F. Segura и J. M. Andújar. — «Fuzzy Logic-Based Energy Management System for Grid-Connected Residential DC Microgrids with Multi-Stack Fuel Cell Systems: A Multi-Objective Approach». — В: *Sustainable Energy, Grids and Networks* 32 (2022), с. 100909. — DOI: 10.1016/j.segan.2022.100909. — URL: <https://www.sciencedirect.com/science/article/pii/S2352467722001588>.
- [5] T. Liu и др. — «Applying High-Performance Computing in Drug Discovery and Molecular Simulation». — В: *National Science Review* 3.1 (2016), с. 49–63. — DOI: 10.1093/nsr/nww003. — URL: <https://doi.org/10.1093/nsr/nww003>.
- [6] Subha Easwaran и др. — «A Comprehensive Learning Based Swarm Optimization Approach for Feature Selection in Gene Expression Data». — В: *Heliyon* 10.17 (2024), e37165. — DOI: 10.1016/j.heliyon.2024.e37165. — URL: <https://www.sciencedirect.com/science/article/pii/S2405844024131969>.
- [7] Juan Rada-Vilela. — *The FuzzyLite Libraries for Fuzzy Logic Control*. — 2018. — URL: <https://fuzzylite.com>.

- [8] Ammar A. Aldair, Adel A. Obed и Ali F. Halihal. — «Design and Implementation of ANFIS-Reference Model Controller Based MPPT Using FPGA for Photovoltaic System». — B: *Renewable and Sustainable Energy Reviews* 82.Part 3 (2018), с. 2202—2217. — DOI: 10.1016/j.rser.2017.08.071. — URL: <https://www.sciencedirect.com/science/article/pii/S1364032117312169>.
- [9] V. López и др. — «Cost-Sensitive Linguistic Fuzzy Rule Based Classification Systems under the MapReduce Framework for Imbalanced Big Data». — B: *Fuzzy Sets and Systems* 258 (2015). MapReduce in title protected to maintain capitalization, с. 5—38. — DOI: 10.1016/j.fss.2014.01.015.
- [10] M. Elkan и др. — «A Global Distributed Approach to the Chi et al. Fuzzy Rule-Based Classification System for Big Data Classification Problems». — B: *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. — Protected "Chi" in title to maintain proper citation style. — Naples, Italy, 2017, — C. 1—6. — DOI: 10.1109/FUZZ-IEEE.2017.8015544.
- [11] Zheru Chi, Hong Yan и Tuan Pham. — *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition*. — WORLD SCIENTIFIC, 1996. — DOI: 10.1142/3132. — eprint: <https://www.worldscientific.com/doi/pdf/10.1142/3132>. — URL: <https://www.worldscientific.com/doi/abs/10.1142/3132>.
- [12] «Novel Approach to Generate Type-1 Fuzzy Triangular and Trapezoidal Membership Functions to Improve the Classification Accuracy». — B: *Symmetry* 13.10 (2021), с. 1932. — DOI: 10.3390/sym13101932. — URL: <https://doi.org/10.3390/sym13101932>.
- [13] Wenbin Qian и др. — «A Novel Granular Ball Computing-Based Fuzzy Rough Set for Feature Selection in Label Distribution Learning». — B: *Knowledge-Based Systems* 278 (2023), с. 110898. — DOI: 10.1016/j.knosys.2023.110898. — URL: <https://www.sciencedirect.com/science/article/pii/S0950705123006482>.
- [14] D. Pekaslan, C. Wagner и J. M. Garibaldi. — «ADONiS—Adaptive Online Nonsingleton Fuzzy Logic Systems». — B: *IEEE Transactions on Fuzzy Systems* 28.10 (окт. 2020), с. 2302—2312. — DOI: 10.1109/TFUZZ.2019.2933787.
- [15] Raquel Fernandez-Peralta. — «A Comprehensive Survey of Fuzzy Implication Functions». — B: *arXiv preprint* (2025). Preprint submitted to arXiv. — arXiv: 2503.05702. — URL: <https://arxiv.org/abs/2503.05702>.

- [16] Xiaohong Zhang и др. — «Semi-overlap functions and novel fuzzy reasoning algorithms with applications». — В: *Information Sciences* 614 (окт. 2022), с. 104—122. — DOI: 10.1016/j.ins.2022.10.017. — URL: <http://dx.doi.org/10.1016/j.ins.2022.10.017>.
- [17] Fernando Neres, Benjamín Bedregal и Regivan H. N. Santiago. — *(S,N,T)-Implications*. — 2021. — arXiv: 2106.15746 [cs.LO].
- [18] Werner Van Leekwijck и Etienne E. Kerre. — «Defuzzification: Criteria and Classification». — В: *Fuzzy Sets and Systems* 108.2 (1999), с. 159—178. — DOI: 10.1016/S0165-0114(97)00337-0. — URL: <https://www.sciencedirect.com/science/article/pii/S0165011497003370>.
- [19] Resmiye Nasiboglu. — «A Novel Fuzzy Inference Model with Rule-Based Defuzzification Approach». — В: *Journal of Modern Technology and Engineering* 7.2 (сент. 2022). Corresponding author: resmiye.nasiboglu@deu.edu.tr; Received: 3 May 2022; Revised: 28 June 2022; Accepted: 10 July 2022; Published: 6 September 2022, с. 124—133. — DOI: 10.1234/jmte.2022.12345. — URL: <https://example.com/article-url>.
- [20] A. Pourabdollah, C. Wagner и J. Aladi. — «Changes Under the Hood - A New Type of Non-Singleton Fuzzy Logic System». — В: *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. — Istanbul, Turkey, 2015, — С. 1—8. — DOI: 10.1109/FUZZ-IEEE.2015.7338006.
- [21] Л. Рутковский. — *Методы и технологии искусственного интеллекта*. — Пер. И. Д. Рудинский. — EDN QMUKZL. — Москва: Горячая линия-Телеком, 2010. — ISBN: 978-5-9912-0105-6.
- [22] Pablo Cingolani и Jesus Alcalá-Fdez. — «JFuzzyLogic: A Robust and Flexible Fuzzy-Logic Inference System Language Implementation». — В: *IEEE International Conference on Fuzzy Systems*. — Proceedings of the 2012 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). — 2012, — С. 1—8. — DOI: 10.1109/FUZZ-IEEE.2012.6251215.
- [23] L.-X. Wang и J. M. Mendel. — «Generating Fuzzy Rules by Learning from Examples». — В: *IEEE Transactions on Systems, Man, and Cybernetics* 22.6 (нояб. 1992), с. 1414—1427. — DOI: 10.1109/21.199466.

- [24] E. Lima и др. — «Evolving Fuzzy Modeling Using Participatory Learning». — В: — *Evolving Intelligent Systems*. — John Wiley & Sons, Ltd, 2010. — Гл. 4, с. 67—86. — ISBN: 9780470569962. — DOI: <https://doi.org/10.1002/9780470569962.ch4>. — eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470569962.ch4>. — URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470569962.ch4>.
- [25] K. S. T. R. Alves и E. Pestana de Aguiar. — «A Novel Rule-Based Evolving Fuzzy System Applied to the Thermal Modeling of Power Transformers». — В: *Applied Soft Computing* 112 (2021), с. 107764. — DOI: 10.1016/j.asoc.2021.107764. — URL: <https://doi.org/10.1016/j.asoc.2021.107764>.
- [26] В. Г. Куценко Д. А. и Синюк. — «Методы вывода для систем со многими нечеткими входами». — В: *Известия Российской академии наук. Теория и системы управления* 3 (2015). EDN: TQQEOP, с. 48. — DOI: 10.7868/S0002338815030129.
- [27] M. Eisele, K. Hentschel и T. Kunemund. — «Hardware realization of fast defuzzification by adaptive integration». — В: *Proceedings of the Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems*. — 1994, — С. 318—323. — DOI: 10.1109/ICMNN.1994.593726.
- [28] A. Pourabdollah, R. John и J. M. Garibaldi. — «A new dynamic approach for non-singleton fuzzification in noisy time-series prediction». — В: *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. — Naples, Italy, 2017, — С. 1—6. — DOI: 10.1109/FUZZ-IEEE.2017.8015575.
- [29] P. O. Lucas и др. — «A Tutorial on Fuzzy Time Series Forecasting Models: Recent Advances and Challenges». — В: *Learning & Nonlinear Models* 19.3 (2021), с. 29—50. — DOI: 10.21528/lnlm-vol19-no2-art3.
- [30] S. A. Karatach и V. G. Sinuk. — «Method of Inference of Fuzzy Logical Type Systems with Nonsingleton Fuzzification». — В: *Pattern Recognition and Image Analysis* 34.3 (сент. 2024), с. 645—651. — DOI: 10.1134/S1054661824700470. — URL: <https://doi.org/10.1134/S1054661824700470>.
- [31] Andrey Prokopenko и Damien Lebrun-Grandié. — *Revising Apetrei's bounding volume hierarchy construction algorithm to allow stackless traversal*. — 2024. — arXiv: 2402.00665 [cs.DS]. — URL: <https://arxiv.org/abs/2402.00665>.

Список рисунков

1.1	Схема системы нечеткого вывода с использованием синглотонной и несинглотонной фаззификации	12
1.2	Сравнение количества активированных правил при переходе от синглотонной фаззификации к несинглотонной и при увеличении ширины окрестности погрешности $\sigma_{A'}$	13
1.3	Сравнение формы функций принадлежности выходных нечетких множеств для подхода Мамдани	14
1.4	Сравнение формы функций принадлежности выходных нечетких множеств для логического подхода	15
1.5	Пример вычисления нечеткого значения истинности	16
1.6	Значения лингвистической переменной «истинность»	17
2.1	Сравнение классической схемы нечеткого вывода и схемы нечеткого вывода на основе НЗИ	27
2.2	Схема нейро-нечеткой системы с использованием дефаззификации по методу среднего центра для S - и R -импликаций	29
2.3	Схема нейро-нечеткой системы с использованием дефаззификации по методу центра тяжести	32
2.4	Пример нечетких множеств, удовлетворяющих условию $\mu_{B_k}(y_r) = 0$ для $y \neq r$	33
3.1	График функции $\exp(-z^2)$	41
3.2	График функции $\varphi(v) = \sqrt{-\ln v}$	41
3.3	Случаи вырождения ф. п. НЗИ в близкий к асимптоте вид.	42
3.4	3D wind rose diagram for 6 directions with function values plotted vertically.	48
3.5	Иллюстрация работы алгоритма свертки НЗИ для нескольких входов для расширенной \tilde{T} -нормы при $T_1 = \min$ и $T_3 = \min$	48
3.6	Две иллюстрации: графика z-curve и apetrei.	49

Список таблиц

1	Наименование таблицы средней длины	69
2	Тестовые функции для оптимизации, D — размерность. Для всех функций значение в точке глобального минимума равно нулю.	74
3	Длинная таблица с примером чересстрочного форматирования	77
4	Стандартные префиксы ссылок	79

Приложение А

Примеры вставки листингов программного кода

Для крупных листингов есть два способа. Первый красивый, но в нём могут быть проблемы с поддержкой кириллицы (у вас может встречаться в комментариях и печатаемых сообщениях), он представлен на листинге А.1. Второй

Листинг А.1 Программа „Hello, world“ на C++

```

#include <iostream>
using namespace std;

int main() //кириллица в комментариях при xelatex и lualatex и
мееет проблемы с пробелами
5 {
    cout << "Hello, world" << endl; //latin letters in
commentaries
    system("pause");
    return 0;
}
10

```

не такой красивый, но без ограничений (см. листинг А.2).

Листинг А.2 Программа „Hello, world“ без подсветки

```

#include <iostream>
using namespace std;

int main() //кириллица в комментариях
{
    cout << "Привет, мир" << endl;
}

```

Можно использовать первый для вставки небольших фрагментов внутри текста, а второй для вставки полного кода в приложении, если таковое имеется.

Если нужно вставить совсем короткий пример кода (одна или две строки), то выделение линейками и нумерация может смотреться чересчур громоздко.

В таких случаях можно использовать окружения `lstlisting` или `Verb` без `ListingEnv`. Приведём такой пример с указанием языка программирования, отличного от заданного по умолчанию:

```
fibs = 0 : 1 : zipWith (+) fibs (tail fibs)
```

Такое решение — со вставкой нумерованных листингов покрупнее и вставок без выделения для маленьких фрагментов — выбрано, например, в книге Эндрю Таненбаума и Тодда Остина по архитектуре компьютера.

Наконец, для оформления идентификаторов внутри строк (функция `main` и тому подобное) используется `lstinline` или, самое простое, моноширинный текст (`\texttt`).

Пример А.3, иллюстрирующий подключение переопределённого языка. Может быть полезным, если подсветка кода работает криво. Без дополнительного окружения, с подписью и ссылкой, реализованной встроенным средством.

```
## Caching the Inverse of a Matrix

## Matrix inversion is usually a costly computation and there
## may be some
## benefit to caching the inverse of a matrix rather than
## compute it repeatedly
5 ## This is a pair of functions that cache the inverse of a
## matrix.

## makeCacheMatrix creates a special "matrix" object that can
## cache its inverse

makeCacheMatrix <- function(x = matrix()) {#кириллица в коммента
  риях при xelatex и luaLatex имеет проблемы с пробелами
10   i <- NULL
   set <- function(y) {
     x <- y
     i <- NULL
   }
15   get <- function() x
   setSolved <- function(solve) i <- solve
   getSolved <- function() i
   list(set = set, get = get,
        setSolved = setSolved,
20   getSolved = getSolved)
```

```

}

25 ## cacheSolve computes the inverse of the special "matrix"
    returned by
    ## makeCacheMatrix above. If the inverse has already been
    calculated (and the
    ## matrix has not changed), then the cachesolve should retrieve
    the inverse from
    ## the cache.

30 cacheSolve <- function(x, ...) {
    ## Return a matrix that is the inverse of 'x'
    i <- x$getSolved()
    if(!is.null(i)) {
        message("getting cached data")
35     return(i)
    }
    data <- x$get()
    i <- solve(data, ...)
    x$setSolved(i)
40     i
}

```

Листинг А.3 Пример листинга с подписью собственными средствами

Листинг А.4 подгружается из внешнего файла. Приходится загружать без окружения дополнительного. Иначе по страницам не переносится.

```

# Analysis of data on Course Project at Getting and Cleaning
  data course of Data Science track at Coursera.

# Part 1. Merges the training and the test sets to create one
  data set.
# 3. Uses descriptive activity names to name the activities in
  the data set
5 # 4. Appropriately labels the data set with descriptive variable
    names.

if (!file.exists("UCI HAR Dataset")) {
    stop("You need 'UCI HAR Dataset' folder full of data")
}
10

```



```

library(plyr) # for mapvalues

15 #getting common data
features <- read.csv("UCI HAR Dataset/features.txt", sep=" ",
  header = FALSE,
  colClasses = c("numeric", "character"))
activity_labels <- read.csv("UCI HAR Dataset/activity_labels.txt",
  sep="",
  header = FALSE, colClasses = c("
  numeric", "character"))
20
#getting train set data
subject_train <- read.csv("UCI HAR Dataset/train/subject_train.
  txt",
  header = FALSE, colClasses = "numeric",
  col.names="Subject")
y_train <- read.csv("UCI HAR Dataset/train/y_train.txt", header
  = FALSE,
25
  colClasses = "numeric")
x_train <- read.csv("UCI HAR Dataset/train/X_train.txt", sep="",
  header = FALSE,
  colClasses = "numeric", col.names=features$V2
  , check.names = FALSE)

activity_train <- as.data.frame(mapvalues(y_train$V1, from =
  activity_labels$V1,
30
  to = activity_labels$
  V2))
names(activity_train) <- "Activity"

35 #getting test set data
subject_test <- read.csv("UCI HAR Dataset/test/subject_test.txt"
  ,
  header = FALSE, colClasses = "numeric",
  col.names="Subject")
y_test <- read.csv("UCI HAR Dataset/test/y_test.txt", header =
  FALSE,
  colClasses = "numeric")

```

```

40 x_test <- read.csv("UCI HAR Dataset/test/X_test.txt", sep=" ",
  header = FALSE,
                                colClasses = "numeric", col.names=features$V2,
  check.names = FALSE)

activity_test <- as.data.frame(mapvalues(y_test$V1, from =
  activity_labels$V1,
                                to = activity_labels$V2
  ))
45 names(activity_test) <- "Activity"

# Forming full dataframe
data_train <- cbind(x_train, subject_train, activity_train)
50 data_test <- cbind(x_test, subject_test, activity_test)
data <- rbind(data_train, data_test)

# Cleaning memory
rm(features, activity_labels, subject_train, y_train, x_train,
  activity_train,
55 subject_test, y_test, x_test, activity_test, data_train, data
  _test)

# Part 2. Extracts only the measurements on the mean and
  standard deviation for each measurement.

60 cols2match <- grep("(mean|std)", names(data))

# Excluded gravityMean, tBodyAccMean, tBodyAccJerkMean,
  tBodyGyroMean,
  # tBodyGyroJerkMean, as these represent derivations of angle
  data, as
  # opposed to the original feature vector.
65
# Subsetting data frame, also moving last columns to be first
Subsetted_data_frame <- data[, c(562, 563, cols2match)]

# Part 5. From the data set in step 4, creates a second,
  independent tidy data set
70 # with the average of each variable for each activity and each
  subject.

```

```
library(dplyr) # for %>% and summarise_each

75 tidydata <- Subsetted_data_frame %>% group_by(Subject, Activity)
    %>%
        summarise_each(funs(mean))

write.table(tidydata, "tidydata.txt", row.names=FALSE)
```

Листинг А.4 Листинг из внешнего файла

Приложение Б

Очень длинное название второго приложения, в котором продемонстрирована работа с длинными таблицами

Б.1 Подраздел приложения

Вот размещается длинная таблица:

Параметр	Умолч.	Тип	Описание
&INP			
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$)

продолжение следует

(продолжение)			
Параметр	Умолч.	Тип	Описание
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
&SURFPAR			
kick	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
продолжение следует			

(продолжение)			
Параметр	Умолч.	Тип	Описание
mars kick	0	int	2: генерация белого шума симметрично относительно экватора
	1	int	1: инициализация модели для планеты Марс 0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс

Б.2 Ещё один подраздел приложения

Нужно больше подразделов приложения! Конвынёры витюпырата но нам, тебиквьюэ мэнтётюм позтюлант ед про. Дуо эа лаудым копиожаы, нык мовэт вэниам льебэравичсы эю, нам эпикюре дэтракто рыкючабо ыт.

Пример длинной таблицы с записью продолжения по ГОСТ 2.105:

Таблица 1 — Наименование таблицы средней длины

Параметр	Умолч.	Тип	Описание
&INP			
kick	1	int	0: инициализация без шума ($p_s = const$)
			1: генерация белого шума
mars kick	0	int	2: генерация белого шума симметрично относительно экватора
	1	int	1: инициализация модели для планеты Марс
mars kick	1	int	0: инициализация без шума ($p_s = const$)
			1: генерация белого шума
mars kick	1	int	2: генерация белого шума симметрично относительно экватора
			1: инициализация модели для планеты Марс

Продолжение таблицы 1

Параметр	Умолч.	Тип	Описание
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора

Продолжение таблицы 1

Параметр	Умолч.	Тип	Описание
			экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
&SURFPAR			
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$)

Продолжение таблицы 1

Параметр	Умолч.	Тип	Описание
			1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс

Б.3 Использование длинных таблиц с окружением *longtblr* из пакета *tabularray*

В таблице 2 более книжный вариант длинной таблицы, используя окружение `longtblr` из пакета `tabularray` и разнообразные разделители (`toprule`, `midrule`, `bottomrule`) из пакета `booktabs`.

Чтобы визуальнo таблица смотрелась лучше, можно использовать следующие параметры. Таблица задаётся на всю ширину, `longtblr` позволяет делить ширину колонок пропорционально — тут три колонки в пропорции 1.1:1.1:4 — для каждой колонки первый параметр в описании `X[]`. Кроме того, в таблице убраны отступы слева и справа с помощью `@{}` в преамбуле таблицы. К первому и второму столбцу применяется модификатор

`>{\setlength{\baselineskip}{0.7\baselineskip}},`

который уменьшает межстрочный интервал для текста таблиц (иначе заголовок второго столбца значительно шире, а двухстрочное имя сливается с окружающими). Для первой и второй колонки текст в ячейках выравнивается по центру как по вертикали, так и по горизонтали — задаётся буквами `m` и `s` в описании столбца `X[]`.

Так как формулы большие — используется окружение `alignedat`, чтобы отступ был одинаковый у всех формул — он сделан для всех, хотя для большей части можно было и не использовать. Чтобы формулы занимали поменьше места в каждом столбце формулы (где надо) используется `\textstyle` — он делает дроби меньше, у знаков суммы и произведения — индексы сбоку. Иногда формула слишком большая, сливается со следующей, поэтому после неё ставится небольшой дополнительный отступ `\vspace*{2ex}`. Для штрафных функций — размер фигурных скобок задан вручную `\Big\{`, т. к. не умеет `alignedat` работать с `\left` и `\right` через несколько строк/колонок.

В примечании к таблице наоборот, окружение `cases` даёт слишком большие промежутки между вариантами, чтобы их уменьшить, в конце каждой строчки окружения использовался отрицательный дополнительный отступ `\[-0.5em]`.

Таблица 2 — Тестовые функции для оптимизации, D — размерность. Для всех функций значение в точке глобального минимума равно нулю.

Имя	Стартовый диапазон параметров	Функция
сфера	$[-100, 100]^D$	$f_1(x) = \sum_{i=1}^D x_i^2$
Schwefel 2.22	$[-10, 10]^D$	$f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $
Schwefel 1.2	$[-100, 100]^D$	$f_3(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$
Schwefel 2.21	$[-100, 100]^D$	$f_4(x) = \max_i \{ x_i \}$
Rosenbrock	$[-30, 30]^D$	$f_5(x) = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$
ступенчатая	$[-100, 100]^D$	$f_6(x) = \sum_{i=1}^D \lfloor x_i + 0.5 \rfloor^2$
зашумлённая квартиче- ская	$[-1.28, 1.28]^D$	$f_7(x) = \sum_{i=1}^D ix_i^4 + rand[0,1)$
Schwefel 2.26	$[-500, 500]^D$	$f_8(x) = \sum_{i=1}^D -x_i \sin \sqrt{ x_i } +$ $+ D \cdot 418.98288727243369$
Rastrigin	$[-5.12, 5.12]^D$	$f_9(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$
Ackley	$[-32, 32]^D$	$f_{10}(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) -$ $-\exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$
Griewank	$[-600, 600]^D$	$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(x_i/\sqrt{i}) + 1$
штрафная 1	$[-50, 50]^D$	$f_{12}(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \right.$ $\left. + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + \right.$ $\left. + (y_D - 1)^2 \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$

продолжение следует

(продолжение)

Имя	Стартовый диапазон параметров	Функция
штрафная 2	$[-50, 50]^D$	$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \right.$ $+ \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] +$ $+ (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \left. \right\} +$ $+ \sum_{i=1}^D u(x_i, 5, 100, 4)$
сфера	$[-100, 100]^D$	$f_1(x) = \sum_{i=1}^D x_i^2$
Schwefel 2.22	$[-10, 10]^D$	$f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $
Schwefel 1.2	$[-100, 100]^D$	$f_3(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$
Schwefel 2.21	$[-100, 100]^D$	$f_4(x) = \max_i \{ x_i \}$
Rosenbrock	$[-30, 30]^D$	$f_5(x) = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$
ступенчатая	$[-100, 100]^D$	$f_6(x) = \sum_{i=1}^D \lfloor x_i + 0.5 \rfloor^2$
зашумлённая квартиче- ская	$[-1.28, 1.28]^D$	$f_7(x) = \sum_{i=1}^D i x_i^4 + rand[0, 1)$
Schwefel 2.26	$[-500, 500]^D$	$f_8(x) = \sum_{i=1}^D -x_i \sin \sqrt{ x_i } +$ $+ D \cdot 418.98288727243369$
Rastrigin	$[-5.12, 5.12]^D$	$f_9(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$
Ackley	$[-32, 32]^D$	$f_{10}(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) -$ $- \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e$
Griewank	$[-600, 600]^D$	$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(x_i / \sqrt{i}) + 1$

продолжение следует

(окончание)

Имя	Стартовый диапазон параметров	Функция
штрафная 1	$[-50, 50]^D$	$f_{12}(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \right.$ $\left. + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + \right.$ $\left. + (y_D - 1)^2 \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$
штрафная 2	$[-50, 50]^D$	$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \right.$ $\left. + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + \right.$ $\left. + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \right\} +$ $+ \sum_{i=1}^D u(x_i, 5, 100, 4)$
Примечание — Для функций f_{12} и f_{13} используется $y_i = 1 + \frac{1}{4}(x_i + 1)$ и $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$		

Б.4 Форматирование внутри таблиц

В таблице 3 пример с чересстрочным форматированием. Это реализовано средствами, доступными в таблицах пакета `tabularray`.

В таблице 3 каждая чётная строчка (заголовок таблицы тоже считается за строчку) — синяя, нечётная — с наклоном и слегка поднята вверх. Визуально это приводит к тому, что среднее значение и среднеквадратичное изменение группируются и хорошо выделяются взглядом в таблице. Сохраняется возможность отдельные значения в таблице выделить цветом или шрифтом. К первому и второму столбцу форматирование не применяется по сути таблицы, к шестому общее форматирование не применяется для наглядности.

Таблица 3 — Длинная таблица с примером чересстрочного форматирования

	Итера- ции	JADE++	JADE	jDE	SaDE	DE/rand /1/bin	PSO
f1	1500	1.8E-60 (8.4E-60)	1.3E-54 (9.2E-54)	2.5E-28 (3.5E-28)	4.5E-20 (6.9E-20)	9.8E-14 (8.4E-14)	9.6E-42 (2.7E-41)
f2	2000	1.8E-25 (8.8E-25)	3.9E-22 (2.7E-21)	1.5E-23 (1.0E-23)	1.9E-14 (1.1E-14)	1.6E-09 (1.1E-09)	9.3E-21 (6.3E-20)
f3	5000	5.7E-61 (2.7E-60)	6.0E-87 (1.9E-86)	5.2E-14 (1.1E-13)	9.0E-37 (5.4E-36)	6.6E-11 (8.8E-11)	2.5E-19 (3.9E-19)
f4	5000	8.2E-24 (4.0E-23)	4.3E-66 (1.2E-65)	1.4E-15 (1.0E-15)	7.4E-11 (1.8E-10)	4.2E-01 (1.1E+00)	4.4E-14 (9.3E-14)
f5	3000	8.0E-02 (5.6E-01)	3.2E-01 (1.1E+00)	1.3E+01 (1.4E+01)	2.1E+01 (7.8E+00)	2.1E+00 (1.5E+00)	2.5E+01 (3.2E+01)
f6	100	2.9E+00 (1.2E+00)	5.6E+00 (1.6E+00)	1.0E+03 (2.2E+02)	9.3E+02 (1.8E+02)	4.7E+03 (1.1E+03)	4.5E+01 (2.4E+01)
f7	3000	6.4E-04 (2.5E-04)	6.8E-04 (2.5E-04)	3.3E-03 (8.5E-04)	4.8E-03 (1.2E-03)	4.7E-03 (1.2E-03)	2.5E-03 (1.4E-03)
f8	1000	3.3E-05 (2.3E-05)	7.1E+00 (2.8E+01)	7.9E-11 (1.3E-10)	4.7E+00 (3.3E+01)	5.9E+03 (1.1E+03)	2.4E+03 (6.7E+02)
f9	1000	1.0E-04 (6.0E-05)	1.4E-04 (6.5E-05)	1.5E-04 (2.0E-04)	1.2E-03 (6.5E-04)	1.8E+02 (1.3E+01)	5.2E+01 (1.6E+01)
f10	500	8.2E-10 (6.9E-10)	3.0E-09 (2.2E-09)	3.5E-04 (1.0E-04)	2.7E-03 (5.1E-04)	1.1E-01 (3.9E-02)	4.6E-01 (6.6E-01)
f11	500	9.9E-08 (6.0E-07)	2.0E-04 (1.4E-03)	1.9E-05 (5.8E-05)	7.8E-04 (1.2E-03)	2.0E-01 (1.1E-01)	1.3E-02 (1.7E-02)
f12	500	4.6E-17 (1.9E-16)	3.8E-16 (8.3E-16)	1.6E-07 (1.5E-07)	1.9E-05 (9.2E-06)	1.2E-02 (1.0E-02)	1.9E-01 (3.9E-01)
f13	500	2.0E-16 (6.5E-16)	1.2E-15 (2.8E-15)	1.5E-06 (9.8E-07)	6.1E-05 (2.0E-05)	7.5E-02 (3.8E-02)	2.9E-03 (4.8E-03)
f1	1500	1.8E-60 (8.4E-60)	1.3E-54 (9.2E-54)	2.5E-28 (3.5E-28)	4.5E-20 (6.9E-20)	9.8E-14 (8.4E-14)	9.6E-42 (2.7E-41)
f2	2000	1.8E-25 (8.8E-25)	3.9E-22 (2.7E-21)	1.5E-23 (1.0E-23)	1.9E-14 (1.1E-14)	1.6E-09 (1.1E-09)	9.3E-21 (6.3E-20)

продолжение следует

(окончание)

	Итера- ции	JADE++	JADE	jDE	SaDE	DE/rand /1/bin	PSO
f3	5000	5.7E-61 (2.7E-60)	6.0E-87 (1.9E-86)	5.2E-14 (1.1E-13)	9.0E-37 (5.4E-36)	6.6E-11 (8.8E-11)	2.5E-19 (3.9E-19)
f4	5000	8.2E-24 (4.0E-23)	4.3E-66 (1.2E-65)	1.4E-15 (1.0E-15)	7.4E-11 (1.8E-10)	4.2E-01 (1.1E+00)	4.4E-14 (9.3E-14)
f5	3000	8.0E-02 (5.6E-01)	3.2E-01 (1.1E+00)	1.3E+01 (1.4E+01)	2.1E+01 (7.8E+00)	2.1E+00 (1.5E+00)	2.5E+01 (3.2E+01)
f6	100	2.9E+00 (1.2E+00)	5.6E+00 (1.6E+00)	1.0E+03 (2.2E+02)	9.3E+02 (1.8E+02)	4.7E+03 (1.1E+03)	4.5E+01 (2.4E+01)
f7	3000	6.4E-04 (2.5E-04)	6.8E-04 (2.5E-04)	3.3E-03 (8.5E-04)	4.8E-03 (1.2E-03)	4.7E-03 (1.2E-03)	2.5E-03 (1.4E-03)
f8	1000	3.3E-05 (2.3E-05)	7.1E+00 (2.8E+01)	7.9E-11 (1.3E-10)	4.7E+00 (3.3E+01)	5.9E+03 (1.1E+03)	2.4E+03 (6.7E+02)
f9	1000	1.0E-04 (6.0E-05)	1.4E-04 (6.5E-05)	1.5E-04 (2.0E-04)	1.2E-03 (6.5E-04)	1.8E+02 (1.3E+01)	5.2E+01 (1.6E+01)
f10	500	8.2E-10 (6.9E-10)	3.0E-09 (2.2E-09)	3.5E-04 (1.0E-04)	2.7E-03 (5.1E-04)	1.1E-01 (3.9E-02)	4.6E-01 (6.6E-01)
f11	500	9.9E-08 (6.0E-07)	2.0E-04 (1.4E-03)	1.9E-05 (5.8E-05)	7.8E-04 (1.2E-03)	2.0E-01 (1.1E-01)	1.3E-02 (1.7E-02)
f12	500	4.6E-17 (1.9E-16)	3.8E-16 (8.3E-16)	1.6E-07 (1.5E-07)	1.9E-05 (9.2E-06)	1.2E-02 (1.0E-02)	1.9E-01 (3.9E-01)
f13	500	2.0E-16 (6.5E-16)	1.2E-15 (2.8E-15)	1.5E-06 (9.8E-07)	6.1E-05 (2.0E-05)	7.5E-02 (3.8E-02)	2.9E-03 (4.8E-03)

Б.5 Стандартные префиксы ссылок

Общепринятым является следующий формат ссылок: <prefix>:<label>. Например, \label{fig:knuth}; \ref{tab:test1}; label={lst:external1}. В таблице 4 приведены стандартные префиксы для различных типов ссылок.

Таблица 4 — Стандартные префиксы ссылок

Префикс	Описание
ch:	Глава
sec:	Секция
subsec:	Подсекция
fig:	Рисунок
tab:	Таблица
eq:	Уравнение
lst:	Листинг программы
itm:	Элемент списка
alg:	Алгоритм
app:	Секция приложения

Для упорядочивания ссылок можно использовать разделительные символы. Например, `\label{fig:scheemes/my_scheeme}` или `\label{lst:dts/linked_list}`.

Б.6 Очередной подраздел приложения

Нужно больше подразделов приложения!

Б.7 И ещё один подраздел приложения

Нужно больше подразделов приложения!

Перв. примен.		Справ. №		Приложение В								
Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата	Инв. № подл								
Инв. № подл	Изм.	Лист	№ докум.	Подп.	Дата	Сферический куб			Лит.		Масса	Масштаб
											0 г.	2:1
						Вакуум			Лист 1		Листов 1	
Утв.						РАН						

30.00 mm

30.00 mm

28.07 °

20.00 mm

4.00 mm

10.00 mm

13.00 mm

10.00 mm

R2.00 mm

В - В

А1 (3:1)