

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Тамбовский государственный технический университет»



На правах рукописи

ХОРОХОРИН МИХАИЛ АЛЕКСАНДРОВИЧ

МОДЕЛИ И АЛГОРИТМЫ ПОЛУЧЕНИЯ ОЦЕНКИ ЖИВУЧЕСТИ
СИСТЕМ С НЕЧЕТКОЙ ИНФОРМАЦИОННОЙ СТРУКТУРОЙ,
ОБЕСПЕЧИВАЮЩИЕ СОКРАЩЕНИЕ ВРЕМЕНИ РАСЧЕТА

Специальность 05.13.17

Теоретические основы информатики

Диссертация на соискания ученой степени кандидата технических наук

Научный руководитель:
доктор технических наук,
профессор
Ю.Ю. Громов

ТАМБОВ 2014

СОДЕРЖАНИЕ

Введение.....	5
1 Проблема оценки живучести информационных структур и ее изложение в научной литературе.....	14
1.1 Понятие информационных структур	14
1.2 Анализ понятия живучести в различных прикладных областях.....	16
1.3 Анализ существующих подходов к оценке живучести информационных структур.....	20
1.4 Матричное представление информационных структур.....	23
1.5 Оценка живучести информационных структур на основе нечеткой продукционной модели.....	25
1.6 Классификация негативных внешних воздействий на информационную структуру.....	28
1.7 Выводы по первой главе. Постановка цели и задач исследования.....	30
2 Продукционная и нейросетевая модели расчета оценки живучести информационных структур	32
2.1 Типы нейронных сетей	32
2.1.1 Однонаправленные многослойные сети сигмоидального типа	32
2.1.2 Сети с самоорганизацией на основе конкуренции	33
2.1.3 Радиальные нейронные сети	35
2.1.4 Нечеткая нейронная продукционная сеть Такаги-Сугено-Канга.....	36
2.1.5 Нечеткие нейронные сети. Сеть Ванга-Менделя.....	38
2.2 Анализ нейронных сетей для построения нечеткой нейронной модели оценки живучести информационных структур.....	40
2.3 Построение продукционной и нейросетевой модели расчета оценки живучести информационных структур.....	43
2.4 Выводы по второй главе.....	55
3 Алгоритмы, реализующие продукционную и нейросетевую модели расчета оценки живучести информационных структур.....	57

3.1 Алгоритм перехода от нечеткой продукционной модели к соответствующей ей нечеткой нейросетевой модели	57
3.2 Построение диаграммы вариантов использования моделей и алгоритмов оценки живучести информационных структур.....	66
3.3 Построение диаграммы классов моделей и алгоритмов оценки живучести информационных структур.....	68
3.4 Построение диаграммы деятельности моделей и алгоритмов оценки живучести информационных структур.....	69
3.5 Алгоритм обратного распространения ошибки для нечеткой нейронной сети.....	74
3.6 Метод Фибоначчи поиска экстремума.....	82
3.7 Выводы по третьей главе.....	84
4 Программная реализация моделей и алгоритмов оценки живучести информационных структур	85
4.1 Анализ и выбор языка программирования.....	85
4.2 Выбор средства разработки программной реализации моделей и алгоритмов оценки живучести информационных структур.....	90
4.3 Программная реализация моделей и алгоритмов оценки живучести информационных структур	92
4.4 Тестирование программного обеспечения оценки живучести информационных структур	97
4.5 Выводы по четвертой главе	114
Заключение	115
Список используемых источников.....	117
Приложение А. Тестирование обученной нечеткой нейронной сети в разработанном программном обеспечении	133
Приложение Б. Сравнительный анализ рекомендаций.....	134
Приложение В. Примеры сгенерированных топологий информационных структур.....	135
Приложение Г. Таблица истинности для различных видов импликаций	136

Приложение Д. Графики зависимости функций принадлежности.....	137
Приложение Ж. Копии свидетельств о государственной регистрации программы для ЭВМ.....	
Приложение З. Акты об использовании результатов исследования	146

ВВЕДЕНИЕ

Актуальность темы исследования. Современное общество характеризуется широким использованием и непрерывным совершенствованием информационных систем (ИС), которые в большинстве своем имеют развитую структуру и обеспечивают эффективное принятие решений в различных сферах деятельности. Такими системами являются, например, информационные системы, обеспечивающие поддержку принятия управленческих решений, мониторинга, прогнозирования в сферах жилищно-коммунального хозяйства, транспорта, муниципального управления, связи и др. Совершенствование ИС приводит к усложнению их топологической структуры и в большинстве случаев она является комбинаторной т.е. состоит из множества узлов связанных между собой информационными потоками и характеризуется экстремальностью нагрузок и влиянием негативных внешних воздействий(НВВ). Это приводит к изменению или уничтожению информационных потоков, которые характеризуются соответствующими наборами параметров. Информационная структура может не выполнять возложенные на нее функции или может быть разрушена вследствие сбоев, разрывов информационных потоков, перегрузок, выхода из строя ее элементов и возникновения дефектов, имеющих место как в процессе эксплуатации, так вследствие воздействия НВВ. В связи с этим необходимо построить математическое описание информационных структур, позволяющее проанализировать работу информационных структур в различных условиях, а именно: как в условиях эксплуатации так действия НВВ. Использование детерминированного и стохастического подходов сопряжено значительными трудностями, которые определяются отсутствием экспериментальных данных, недостаточным объемом статистической информации, отсутствием жестких функциональных зависимостей, связывающие между собой

параметры информационных процессов, протекающих в информационных структурах и их характеристики. В следствии этого математическим аппаратом, позволяющим проводить анализ информационных структур современных ИС является теория нечетких множеств, позволяющая рассматривать эти структуры как нечеткие. Нечеткость информационных структур (НИС) этих систем определяется территориальной распределенностью, количеством элементов и многообразием связей между ними, дальностью передачи между узлами, возможностью разрыва связей, сложностью системы, действием трудноформализуемых НВВ. Развитие ИС, имеющих НИС увеличивает возможность их уязвимости за счет влияния негативных внешних воздействий (НВВ), как непреднамеренного, так и преднамеренного характера. Поэтому исключительно важной является задача получения оценок живучести НИС за определенное время. Требования к времени получения оценки живучести информационных структур определены ГОСТом Р 53111-2008[1] и обусловлены необходимостью быстрого перестроения структуры для парирования влияния НВВ.

В связи с этим актуальной является задача сокращения времени расчета оценки живучести систем с НИС с целью повышения эффективности их функционирования в условиях НВВ.

Результаты анализа методов получения оценок живучести систем с НИС доказывают необходимость их развития в условиях действия НВВ, приводящих к изменению, как самих структур, так и параметров их характеризующих.

Степень разработанности темы исследования. Значительный вклад в разработку вопросов теории живучести структур различного рода назначения и применения внесли работы, как отечественных авторов (Д.Л. Белоцерковский, Л. С. Бернштейн, И.Н. Боженюк, А.Н. Борисов, В.М. Вишневецкий, Ю.Ю.Громов, А.Г. Додонов, В.Ф. Крапивина, В.В. Круглов, М.Г. Кузнецова, Д.В. Ландэ, Ю.Е. Малашенко, А.Н. Мелихов, Ю.Е.

Мельникова, Ю.М. Парфенова, А.В. Поспелов, В.М. Розенберг, И.А. Рябина, И.Ю. Стекольников, Б.С. Флейшмана), так и зарубежных авторов (С.Л. Colbourn, К. Sekine, Н. Imai, S. Tani, А.Е. Smith, Y. Tutte, Д. Дюбуа, А. Прад, и др.), в которых разработаны математические модели и методы, с помощью которых можно получить оценки живучести информационных структур. Однако эти методы имеют ограниченные применимы для получения оценок живучести систем с НИС сокращению времени получения оценок живучести в силу ряда особенностей, а именно вычислительной трудоемкости и невозможности формализации ряда параметров и характеристик систем с НИС.

Системы с НИС характеризуются множеством параметров и характеристик, таких как территориальной распределенностью, количеством элементов и многообразием связей между ними, дальностью передачи между узлами, возможностью разрыва связей, сложностью системы, которые влияют на работоспособность и живучесть, как отдельных элементов, так и всей информационной структуры. В результате НВВ возникают первичные последствия[3], которые выражаются в нарушении работоспособности элементов, связей, а также работы функционирования параметров и характеристик информационной структуры. Вследствие чего параметры и характеристики НИС достаточно сложно формализовать[4], используя детерминированный, статистический или стохастический подходы, поэтому в этом случае единственным возможным и наиболее хорошо себя зарекомендовавшим является подход, основанный на использовании качественной информации и как следствие применение методов теории нечетких множеств.

Использование качественных данных и сведений об информационной структуре обуславливает целесообразность применения нечетких продукционных моделей, которые позволяют наглядно отобразить информационные процессы и их взаимодействие. Однако данный тип

модели приводит к существенным вычислительным затратам и как следствие снижению эффективности.

В связи с этим актуальным является применение нечетких продукционных моделей (НПМ) с использованием нечетких нейронных продукционных сетей (ННПС), которые позволяют повысить эффективность проводимых расчетов.

Вышесказанное определяет **практическую задачу** – сокращения времени расчета оценки живучести систем с НИС с целью повышения эффективности их функционирования в условиях НВВ и соответствующую ей **научную задачу**, которая заключается в построении моделей и алгоритмов оценки живучести системы с НИС, связывающих ее качественные характеристики и количественные переменные для сокращения времени расчета.

Объект исследования: нечеткие информационные структуры систем.

Предмет исследования: модели и алгоритмы, обеспечивающие сокращение времени расчета оценки живучести систем с НИС.

Цели и задачи. Целью исследования является сокращение времени расчета оценки живучести НИС при НВВ на основе использования разработанных моделей и алгоритмов, основанных на совместном использовании качественной и количественной информации о ее характеристиках. Для достижения цели и решения научной задачи были поставлены следующие частные задачи:

1. Провести анализ существующих методов и подходов к оценке живучести информационных структур.

2. Построить нечеткую продукционную модель оценки живучести информационных структур.

3. Построить нечеткую нейросетевую модель, алгоритм перехода от нечеткой продукционной модели к соответствующей ей нечеткой

нейросетевой модели и алгоритм обучения нечеткой нейронной продукционной сети.

4. Провести вычислительный эксперимент с использованием разработанных моделей и алгоритмов оценки живучести систем с НИС с оценкой достоверности полученных результатов.

Научная новизна исследования заключается в разработке:

1. Нечеткой продукционной модели оценки живучести НИС, отличающейся использованием лингвистических переменных (территориальная распределенность, количество элементов и многообразие связей между ними, дальность передачи между узлами, возможность разрыва связи, сложность системы), характеризующих ее структуру и параметры для свертки которых используются однопараметрические T-нормы, параметры которых уточняются вследствие решения оптимизационной задачи, для решения которой применяется разработанный алгоритм, использующий метод чисел Фибоначчи совместно с правилом Голдстейна.

2. Алгоритма перехода от построенной нечеткой продукционной модели к соответствующей ей нечеткой нейронной продукционной сети, отличающегося наличием пяти этапов каждый из которых представлен набором правил для свертки которых применяется однопараметрическая T-импликация, модифицирующей вывод Мамдани.

3. Алгоритма обучения нечеткой нейронной продукционной сети, основанный на использовании предложенного квазидискретного Ньютоновского метода с немонотонной стабилизацией, в котором для решения задачи одномерной оптимизации используется модификация метода чисел Фибоначчи, основанного на применении правила Голдстейна.

Положения, выносимые на защиту:

1. Нечеткая продукционная модель оценки живучести НИС, в которой используются лингвистические переменные (территориальная распределенность, количество элементов и многообразие связей между

ними, дальность передачи между узлами, возможность разрыва связи, сложность системы), характеризующие ее структуру и параметры, позволяющая получить оценку живучести НИС в условиях НВВ.

2. Алгоритм перехода от нечеткой продукционной модели к соответствующей ей нечеткой нейросетевой продукционной сети, позволяющий снизить вычислительные затраты расчета оценки живучести НИС.

3. Алгоритм обучения нечеткой нейронной сети, использующий разработанный квазидискретный Ньютоновский метод с немонотонной стабилизацией, который позволяет сократить время обучения.

Методология и методы исследования. Методология исследования основывается на принципах системного анализа и общей теории систем. При решении поставленных задач в работе были использованы методы: теории систем, системного анализа, теории нечётких множеств, теории нейронных сетей, теории графов и имитационного моделирования.

Теоретическая и практическая значимость работы.

Теоретическая значимость результатов исследования состоит в развитии математического аппарата теории нечетких множеств, нейронных сетей для решения задачи повышения эффективности функционирования НИС за счет сокращения времени расчета оценки живучести НИС при НВВ с помощью разработанных моделей и алгоритмов, обеспечивающих перестроение структуры.

Практическая значимость работы заключается в применении программной реализации разработанных моделей и алгоритмов для исследования систем с НИС при действии НВВ.

Степень достоверности и апробация результатов.

Достоверность результатов работы основана на корректном применении математического аппарата теории графов, нечетких множеств, эволюционного моделирования; использования современных методов распределения информационных потоков и ресурсов информационных

структур; на результатах вычислительного эксперимента, подтверждающих повышение живучести и эффективности функционирования информационных структур вследствие применения разработанных моделей.

Основные результаты работы представлены и обсуждены на следующих конференциях: XI Международной научно-методической конференции «Информатика: проблемы, методология, технологии» г. Воронеж, 10-11 февраля 2011 г.; XIV Международной научно-практической конференции-выставки «Актуальные проблемы информатики и информационных технологий», г. Тамбов, 22 марта 2010 г.; II-ой международной кластерной научно-практической конференции «Аспекты ноосферной безопасности в приоритетных направлениях деятельности человека», Москва, 11 февраля 2011 г., VIII Всероссийской научно-практической конференции «Математические методы и информационно-технические средства», г. Краснодар, 22-23 июня 2012 г., XIII Международной научно-методической конференции «Информатика: проблемы, методология, технологии», г. Воронеж, 7-8 февраля 2013 года, а также на семинарах кафедры «Информационные системы и защита информации» ФГБОУ ВПО «ТГТУ».

В 2012 году результаты диссертационной работы использованы в прикладной НИР по программе Фонда содействия развитию малых форм предприятий в научно-технической сфере Старт 12 по направлению Информационных технологий, контракт № 10647р/19106. В 2012 году результаты диссертационного исследования использованы в заявках 13-08-00285 и 13-07-00118, поданных на конкурс инициативных научно-исследовательских проектов по программе РФФИ.

Внедрение результатов исследования. Основные положения диссертационной работы использованы при обучении студентов кафедры «Информационные системы и защита информации» в Институте автоматизации и информационных технологий ФГБОУ ВПО «ТГТУ». Результаты диссертационной работы приняты к внедрению на кафедре

«Информационные системы и защита информации» ФГБОУ ВПО «ТГТУ», в ВЧ 61460, ОАО «Медтехника» (г. Тамбов), ООО «КОНУС-ИТ» (г. Тамбов), что подтверждено актами о внедрении результатов исследований.

Публикации. По теме диссертации опубликовано 29 работ, в том числе 7 статей в изданиях, рекомендованных ВАК Минобрнауки России, получено 6 свидетельств о государственной регистрации программы для ЭВМ.

Объём и структура работы. Диссертация, общий объём которой составляет 149 страниц, состоит из введения, четырёх глав, заключения, списка использованной научной литературы, включающего 148 наименований научных трудов на русском и иностранных языках. Основной текст диссертации содержит 59 иллюстраций и 3 таблицы.

Во **введении** обоснована актуальность выбранной темы, сформулирована цель работы, поставлены задачи, решение которых позволит достичь цели исследования.

В **первой главе** рассмотрена задача оценки живучести и сокращение времени расчета систем с НИС и их изложение в научной литературе. На основе изученных литературных источников проведен анализ информационных структур (ИС), проведена систематизация и анализ термина «живучесть», выделены основные свойства живучести определяющих качество функционирования ИС. Также были выделены основные методы и подходы к обеспечению живучести ИС для разработки аналитического и программного обеспечения нечеткая продукционная модель оценки живучести информационных структур, которая является наиболее неизученным направлением обеспечения живучести ИС, и является актуальным решением для выполнения поставленной задачи.

Во **второй главе** проведен сравнительный анализ нейронных сетей с выбор оптимальной нейросетевой модели для решения поставленной задачи. Разработана нечеткая продукционная модель оценки живучести ИС. Так же

разработана модель перехода от нечеткой продукционной модели к соответствующей ей нечеткой нейронной сети

В третьей главе представлен алгоритм перехода от нечеткой продукционной модели к соответствующей ей нечеткой нейросетевой модели, который использует T- импликации в алгоритм нечеткого вывода Мамдани, позволяющие снизить вычислительные затраты расчета оценки живучести систем с НИС и алгоритм обучения нечеткой нейронной сети, использующий разработанный квазидискретный Ньютоновский метод с немонотонной стабилизацией, который позволяет сократить количество вычисляемых операции по сравнению с другими градиентными методами и тем самым сократить время обучения.

В четвертой главе проведены имитационные исследования оценки живучести информационных структур (ИС), оценена достоверность результата разработанных моделей и алгоритмов; проведен вычислительный эксперимент программного комплекса оценки живучести информационных структур.

В заключении сформулированы основные результаты работы.

Работа соответствует п. 2 «Исследование информационных структур, разработка и анализ моделей информационных процессов и структур» паспорта специальности 05.13.17 «Теоретические основы информатики».

1 Проблема оценки живучести информационных структур и ее изложение в научной литературе

1.1 Понятие информационных структур

В современном информационном обществе информационные структуры (ИС) занимают одно из важных мест и представляют собой сложные, распределенные структуры[6,7]. Все эти структуры составляют базу для управленческой инфраструктуры предприятия, организации, региона и государства. От их работоспособности и качества функционирования зависит не только существующий уровень экономического развития, но и дальнейшего его роста, как экономики государства, так и отдельного предприятия. Поддержание таких структур в работоспособном состоянии предполагает большого внимания и связано с большими материальными затратами и человеческими ресурсами[5,8,136].

Массовое использование ИС потребовало решения вопросов повышения качества функционирования на каждом этапе их жизненного цикла, связанном с решением различного типа задач, возникающими перед аналитиками и эксплуатационщиками таких как моделирование распределения информационных потоков внутри структуры при воздействии на ее структуру негативных внешних воздействий, выявление «узких мест», характеризующихся избыточной информационной нагрузкой узлов системы, которые являются ключевыми и их выход из строя приводит к возникновению дорогостоящих последствий[10,11].

Важнейшим фактором, определяющее качество функционирования ИС, является ее живучесть. Понятие "живучесть" согласно ГОСТ Р 50922-96 [12] – это способность ИС выполнять свои основные функции, несмотря на происходящие аппаратные или программные сбои и отказы, изменения характеристик функционирования ИС и влияние внешних неблагоприятных воздействий (НВВ) [13,14]. В состав ИС входит множество связанных между

собою узлов и представляющих собой многоуровневую иерархическую структуру[15]. При действии НВВ, эта структура очень уязвима, в которой за счет многочисленных узлов и связей между ними часто появляется «каскадный эффект»[10]. При этом эффекте сбой или разрушение, как одного узла или связи в одном месте происходят перегрузки и выход из строя других узлов и связей ИС и даже может привести к разрушению всей системы.

Информационная структура состоит из множества узлов и множества дуг, соединяющих их [10]. Узлы иногда называют вершинами или точками, а дуги – ребрами или звеньями. Информационная структура может быть представлена в виде графа $G = (V, E)$, где V –множество узлов структуры G , E – множество дуг структуры G [16,17].

Узлом может быть точка, являющаяся источником или стоком некоторого информационного потока, а так же точка, в которой величина этого потока изменяется. Поэтому узел можно рассматривать как точку ветвления структуры. Примерами узлов информационной структуры могут быть пересечения автострад, электростанции, телефонные узлы, водохранилища, вычислительные машины.

Примерами дуг информационной структуры могут быть дороги, линии электропередачи, телефонные линии, водные магистрали или обобщенные каналы через которые протекает некоторый поток.

При графическом представлении информационной структуры узлы обозначаются кружками, дуги – отрезками прямой, а направление указывается стрелками[18].

На рисунке 1.1 изображены неориентированная, ориентированная и биориентированная дуги соответственно.

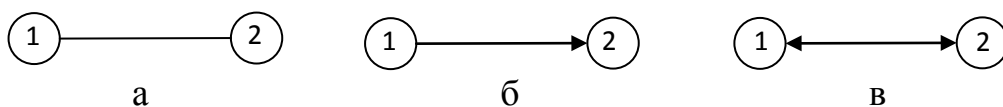


Рисунок 1.1 – Графическое представление дуг и узлов информационной структуры.

а – неориентированная; б – ориентированная; в – биориентированная.

Обычно узлам приписываются номера или буквенные обозначения, чтобы различать и иметь возможность ими оперировать в процессе анализа[20,21], и проектирования информационных структур. Нумерацию узлов можно осуществлять произвольным образом, однако, исследования упрощаются, если их пронумеровать в определенном порядке.

Дуги информационной структуры можно рассматривать как каналы, по которым протекает поток некоторого обобщенного продукта. Величина этого продукта в заданном узле равна разности величин потока, входящего в узел и потока, вытекающего из него. Если эта разность положительна, то узел называется источником, а если отрицательная – стоком или конечным узлом. Сеть может содержать более одного стока или источника[19].

1.2 Анализ понятия живучести в различных прикладных областях

Проблеме оценки живучести ИС посвящен ряд работ, как отечественных авторов (А.Г. Додонов, М.Г. Кузнецова, В.М. Вишневский, В.Ф. Крапивина, Ю.Ю. Громова, И.А. Рябинина, Ю.М. Парфенова, Б.С. Флейшмана, Д.В. Ландэ, И.Ю. Стекольников, Д.Л. Белоцерковского, Ю.Е. Мельникова, Ж.С. Сарыпбекова, Ю.Е. Малашенко) так и зарубежных авторов (S. Tani , C.J. Colbourn, A.E. Smith, K. Sekine, H. Imai и др.).

Важное значение научных разработок данных авторов – формирование математических основ анализа и разработка аналитических моделей оценки живучести информационных структур, которые адекватно описывают процесс построения и расчет оценки живучести ИС. Большое разнообразие предлагаемых моделей, методов, алгоритмов оценки живучести свидетельствует об отсутствии единого подхода. Поэтому решение задачи разработки аналитического описания оценки живучести информационных структур [9] большой размерности и сложной структуры, а так же

разработка новых методов и подходов проектирования и анализа ИС является актуальной в настоящее время.

Прежде чем формулировать общетехническое определение понятия "живучесть", проследим за его эволюцией в различных областях техники. Г.Н. Черкесов сформулировал ряд определений. В судостроении живучесть судна определена, как способность противостоять воздействию стихийных сил ветра и волн, пожаров, оружия противника, сохранять и восстанавливать, как полностью, так и частично мореходность и боевые качества при наносимых повреждениях [11].

В электроэнергетике под живучестью понимается свойство объекта противостоять возмущениям, не допуская их каскадного развития с массовым нарушением питания потребителей[22,23]. Здесь следует обратить внимание на требование к системе, заключающееся в том, что она должна противостоять переводу ее частей в нерабочее состояние вследствие технологически связанных отказов, вызванных нарушением внешних (по отношению к некоторой подсистеме) условий функционирования[24].

В вычислительных системах с живучестью связывается отсутствие потерь любой задачи (функции) из-за отказов элементов. Это свойство обеспечивается развитыми средствами технического диагностирования ее восстановления и реконфигурации[11].

Общетехнические определения живучести информационных структур приведены в работах[11,25]. В работе [11] под живучестью понимается способность системы сохранять основные свои свойства и функций при некоторой потере качества выполнения всех функций путем воздействия НВВ. В работах[11,14] живучесть определена как свойство объекта, заключающееся в его способности выполнять заданное назначение в процессе неблагоприятных воздействий на весь объект или отдельные его компоненты, поддерживая в допустимых пределах свои эксплуатационные показатели[26].

Для некоторых информационных структур важны лишь прямые последствия ННВ, а именно состояние системы непосредственно после завершения НВВ. В этом случае в определенный промежуток времени оценивается уровень работоспособности и функционирования элементов структуры при этом не важна судьба структуры в дальнейшей ее эксплуатации (оценка живучести по состоянию структуры)[27]. Для других информационных структур выполнение заданных функций происходит в течение определенного и продолжительного интервала времени после завершения НВВ. В таких структурах успешное выполнение задания определяется состоянием системы в начальный момент времени и эксплуатацией и функционированием в дальнейшем. Помимо того, на структуру начинают воздействовать другие факторы, влияющих на отказ, как отдельного элемента, так и всех структуры в целом. К таким факторам относят: остаточный уровень избыточности, эффективность системы восстановления, исправность и неисправность элементов и пр[28]. В данном случае оценка живучести информационной структуры должна оцениваться по результатам выполнения задания.

В дополнение существующих обзоров определений необходимо отметить большой вклад в формирования понятия живучести для ИС А.Г. Додонова и Ю.И. Стекольников. В работе А.Г. Додонова [8,29] сформулировано следующее определение живучести: «под живучестью ИС будем понимать понимаем способность структуры сохранять и восстанавливать выполнение основных свойств и функций в заданном объеме и на протяжении определенного времени в случае изменения информационной структуры и/или алгоритмов и условий ее функционирования при воздействии НВВ». Ю.И. Стекольников сформулировал следующее определение: «живучесть – свойство, характеризующее способность системы эффективно функционировать при получении повреждений (разрушений) или восстанавливать данную способность в течении заданного времени» [5].

С учетом вышеизложенного живучесть определяется, как свойство структуры сохранять и восстанавливать работоспособность выполнения основных свойств и функций структуры в заданном объеме в течение заданной наработки при изменении алгоритмов и условий ее функционирования вследствие воздействия на нее НВВ. Основные функции и заданная наработка могут определяться для одного и нескольких, различных по тяжести НВВ, причем для общего случая они различны на разных уровнях структуры [23]. Данные определения допускают любые последствия НВВ, влияющие на выполнение задания, а в частности потери работоспособности узлов и связей информационной структуры после физического разрушения или нарушения целостности структуры, изменения технических характеристик (скорости, производительности, пропускной способности и пр.), искажения алгоритмов функционирования, уменьшения структурной избыточности, ухудшения безотказности элементов, уровня запасов продукции, управляемости системы, изменения внешних условий функционирования (резкое уменьшение или увеличение нагрузки, перераспределение нагрузки, изменение динамических характеристик нагрузки) [11,30].

В этих определениях следует обратить внимание на следующее:

- живучесть информационных структур рассматривается как внутреннее свойство структуры, обладающая независимо от возникающих в данный момент времени условий функционирования. Она обладает им всегда и в определенной мере может проявляться при нормальных условиях функционирования, когда возникают отказы элементов, вызванные производственными дефектами, старением, уходом параметров и пр. Но в полной мере живучесть проявляется при крупных НВВ, не предусмотренных условиями нормальной эксплуатации и поэтому трудно прогнозируемых, так как они создают в системе экстремальные условия функционирования.

- живучесть ИС проявляется в сохранении не всех своих свойств и функций, выполняющихся при нормальной работе системы, а лишь основные

свойства и функции, с возможным понижением качества их выполнения. Это означает, что возможно изменение стратегии функционирования системы по мере увеличения тяжести НВВ.

- ИС должна обладать свойством постепенной деградации по мере увеличения тяжести на каждом этапе НВВ и способна максимально эффективно и оперативно использовать сохранившиеся ресурсы системы для сохранения и выполнения основных своих свойств и функций с учетом изменения стратегии функционирования (целевой функции), а в дальнейшем реализовать оптимальную стратегию восстановления с учетом возникающих ограничений.

Живучесть систем анализируют и оценивают на различных уровнях проектирования, моделирования и функционирования информационных систем. Во время исследования функциональной живучести могут использоваться теоретико-игровые, вероятностные, графовые, матричные модели.

1.3 Анализ существующих подходов к оценке живучести информационных структур

При исследовании функциональной живучести информационных структур особенности топологии структур межкомпонентных связей учитываются опосредовано. Предполагается, что в информационных системах обеспечивается необходимая связность работоспособных компонент.

При анализе живучести функционирования информационная структура характеризуется[10,137]:

- целью функционирования (информирование, дезинформирование, информационное воздействие и т.п.);

- множеством задач $Q = \{q_1, \dots, q_m\}$, решение которых обеспечивается с ее помощью;
- множеством компонент (информационных ресурсов) $\{S_1, S_2, \dots, S_p\}$, являющихся составными частями системы.

В процессе функционирования информационной структуры ее компоненты могут находиться в одном из состояний: работоспособном, не работоспособном, частично работоспособном, т.е. работоспособном, но при частичном снижении (в допустимых пределах) значения каких-либо показателей качества функционирования[31].

На основе теоретико-игровых моделей исследуют живучесть систем, которые функционируют в условиях целенаправленного влияния противника, внешних и внутренних деструктивных воздействий, когда компенсировать нештатные ситуации, потоки отказов и сбоев можно лишь за счет внутренних резервов системы и воздействия на источник деструктивных воздействий[32,138].

Вероятностные, графовые, матричные модели анализа и оценки живучести достаточно разнообразны. В каждом конкретном случае для разных моделей, учитывая различные цели функционирования, а также условия работоспособности системы возможно нахождение количественных оценок живучести. Показатели живучести различных систем можно сравнивать, если цели их функционирования совпадают[33,34].

Структурная живучесть рассматривается как возможности реконструкции, реорганизации, реконфигурации при НВВ [37], которые позволят создать структуру, обеспечивающую выполнение критического подмножества функций для достижения цели функционирования системы.

При рассмотрении структурной живучести учитывается топология сети межкомпонентной связи и надеждностные характеристики компонент [1,11]. Задачи, связанные с анализом структурной живучести, можно свести к

задачам надежности[35,36] и связности топологических структур, в зависимости от введения понятия «разрушение».

Анализ структурной живучести требует определения:

- структуры для выполнения цели функционирования системы в некоторый момент времени, когда возникают нежелательные влияния на систему;

- требований к отдельным видам ресурсов системы и их взаимосвязи;
- требований к функциональным возможностям компонент системы;
- особенностей характера нежелательных влияний или их последствий.

Живучесть информационной структуры оценивается при некоторых допущениях, которые позволяют упростить задачу оценки и свести ее к задаче анализа связности графов, оценки вероятности формирования работоспособной структуры в случае нежелательных влияний и т.п.

При рассмотрении информационной структуры необходимо также учитывать количество узлов и связей в этой системе, которые могут выполнять критическое подмножество информационных функций. Для подсчета этого количества необходимо выделять такие структуры, т.е. находить в информационных структурах множество связанных определенным образом компонент (не обязательно «сильно связанных»), что связи формируются за счет смысловых зависимостей. Переход к рассмотрению информационных моделей должен базироваться также на оценке характеристик структур, их максимальных и минимальных сечениях, потоках в этих структурах, стоимостях и т.д.

При исследовании структурной живучести с помощью графовых моделей [38,39] совокупность компонент системы $G(V, R)$ изображают в виде вершин графа $v \in V$, а ребра графа $r \in R$ соответствуют связям между ними [31]. Систему, которая моделируется с помощью графа, считают разрушенной, если в случае удаления вершин или ребра граф будет удовлетворять одному или большему количеству условий:

- граф состоит минимум из двух компонент;
- не существует направленных путей для определенных множеств вершин;
- количество вершин в наибольшей компоненте графа $G(V, R)$ меньше некоторого заранее заданного числа;
- кратчайший путь превышает некоторую заданную величину[41].

Соответственно структура считается живучей, если эти условия не выполняются. Структурную живучесть систем обычно характеризуют различными показателями связности. Расчет таких показателей, как например, возможные значения функции принадлежности ребра графа, на практике ограничивается вычислительной сложностью подобных задач[40]. Одновременно, используя пути и разрывы графа, моделирующего систему, можно получить достаточно простые предельные оценки необходимых показателей.

1.4 Матричное представление информационных структур

Информационные структуры, которые графически представлены различным образом, соответствуют одним и тем же элементам некоторой системы и выражают одно и то же отношение. Установить эквивалентность двух информационных структур большой размерности бывает чрезвычайно сложно. В теории графов две структуры, между дугами и узлами которых может быть установлено взаимно однозначное соответствие, называется изоморфными[17,42,139].

Если два графа изоморфны, то любое уравнение, справедливое для одного из них, справедливо и для другого графа. Некоторые общие формы представления информационных структур позволяют упростить расчет или установить такие свойства исследуемого объекта, благодаря которым отпадает необходимость в выполнении части вычислений или проведения дальнейшего анализа[31,42].

Количественные характеристики [43] информационной структуры могут быть представлены с помощью матрицы расстояний или матрицы смежностей [31]. Взаимосвязь между узлами задается с помощью матрицы смежности или матрицы инцидентий узлы-дуги.

Рассмотрим информационную структуру $G=(V,E)$, в которой узлы из множества V занумерованы числами $1,2,\dots,n$, а каждой дуге (i, j) из множества E поставлен в соответствие количественный параметр C_{ij} , который называют обобщенной стоимостью или длиной дуги[31,44]. Матрица стоимостей задается массивом $C=[c_{ij}]$, где $(i, j) \in E$.

Элементы матрицы C определяют возможности рассматриваемой системы или естественные ограничения, накладываемые на нее. Они могут соответствовать расстоянию, транспортным затратам, пропускным способностям водных магистралей, нагрузкам на линии электропередачи, живучести компонент системы и т.п[19,45].

Если G – неориентированная структура, то $c_{ij}=c_{ji}$, для всех $(i, j) \in E$. В этом случае матрица является C симметричной.

В [17,31] матрица смежности ориентированной структуры задается массивом $X=[x_{ij}]$, где

$$x_{ij} = \begin{cases} 1, & \text{если дуга } (i, j) \in E \text{ направлена от узла } i \in V \text{ к узлу } j \in E \\ 0, & \text{в противном случае} \end{cases}$$

Если информационная структура G является неориентированной, то каждую дугу $(i, j) \in E$ можно заменить двумя противоположно направленными дугами. При этом матрица X будет симметричной.

В [17,31] матрица инцидентий узлы-дуги ориентированной структуры задается массивом $Z=[z_{ik}]$, где

$$z_{ij} = \begin{cases} 1, & \text{если узел } i \in E \text{ является начальным узлом дуги } a_k \in V \\ -1, & \text{если узел } i \in E \text{ является конечным узлом дуги } a_k \in V \\ 0, & \text{в противном случае} \end{cases}$$

Если информационная структура G является неориентированной, то величины z_{ik} определяются следующим образом [17,31]:

$$z_{ij} = \begin{cases} 1, & \text{если узел } i \in E \text{ принадлежит дуге } a_k \in V \\ 0, & \text{в противном случае} \end{cases}$$

С помощью матричного представления информационных структур могут быть получены важные свойства как неориентированных, так и ориентированных структур и определены отношения предпочтения или установления эквивалентности информационных моделей и структур [17,31,42].

1.5 Оценка живучести информационных структур на основе нечеткой продукционной модели

Информационная структура характеризуется множеством параметров и характеристик, таких как дальность передачи, территориальная распределенность, количество элементов и связей, сложность информационной структуры, которые влияют на работоспособность и живучесть, как отдельных элементов, так и всю структуру в целом. В результате НВВ возникают первичные последствия, которые выражаются в нарушении работоспособности элементов, связей, а также в искажении алгоритмов функционирования параметров и характеристик ИС [46,47]. Вследствие чего параметры и характеристики ИС достаточно сложно формализовать, используя детерминированный, статистический или стохастический подходы, поэтому в этом случае единственным возможным и наиболее хорошо себя зарекомендовавшим является подход, основанный

на использовании качественной информации и как следствие применение методов теории нечетких множеств.

На практике вычисление живучести ИС, которую можно представить в виде простого, связного, неориентированного графа $G = (V, E)$ с количеством вершин V и количеством ребер E - тесно связана с полиномом Татта [10,17,140,142]. Полином Татта графа G – это полином с двумя переменными $T(G; x, y)$ [10,17,142].

Оценка живучести ИС $R(G, p)$ предполагает вероятность того, что информационная структура G будет связанной после разрыва связи (ребра) b с одинаковой вероятностью p и можно подсчитать остовы информационной структуры G [17,143].

При добавлении числа узлов и связей в информационную структуру возрастают экспоненциально вычислительные затраты и ресурсы для оценки живучести с помощью полиномиального расчета [141], поэтому целесообразно использовать нечеткую продукционную модель [48] с использованием нечетких нейронных продукционных сетей (ННПС) к тому же она наиболее отказоустойчива [48,49]. Это значит, что при НВВ производительность падает незначительно, а нечеткая нейронная продукционная сеть более адекватна при обучении.

Использование качественной информации о ИС обуславливает целесообразность применения нечетких продукционных моделей, которые позволяют наглядно отобразить информационные процессы и их взаимодействие. Однако данный тип модели приводит к существенным вычислительным затратам и как следствие снижению эффективности.

В связи с этим актуальным является использование нечетких нейронных продукционных сетей (ННПС) построенных на основе нечетких продукционных моделей, которые позволяют повысить эффективность проводимых расчетов.

Нечеткая продукционная модель в отличие от подходов рассмотренных ранее дают возможность решать неформализованные задачи [50]. При этом

можно моделировать различные ситуации, подавая на вход сети различные данные и оценивая выдаваемый результат.

Нечеткие нейронные сети характеризуются введением нечеткости в различные компоненты традиционных нейронных сетей[49,51,52] на основе положений нечетких множеств и нечеткой логики[53,144,145].

Введение нечеткости в компоненты нейронных сетей представляется следующим образом[48,49]:

1. Введение нечеткости в структуры нейронных сетей путем приписывания нечеткости значениям обучающих примеров.

2. Наделение нейронов, из которых состоят нейронные сети, нечеткостью. Входы и выходы и веса таких нейронов могут быть нечеткими, кроме того, активизация нейронных сетей, состоящих из этих нейронов, является нечетким процессом.

3. Изменения базисных характеристик нейронов таким образом, что они выполняют операции, используемые в теории нечетких множеств (подобно нечеткому объединению, пересечению, агрегированию), вместо стандартного сложения и умножения.

4. Моделирование ошибки, нестабильности или функции энергии нейронной сети с использованием нечетких (неопределенных) значений.

5. Использование нечетких продукционных сетей в нейронных сетях.

Благодаря своим свойствам: нелинейности, адаптивности, отказоустойчивости, масштабируемости, параллельности, самоорганизации, надежности и способности к полной обработке информации ННПС имеют преимущество по сравнению с другими подходами и методами и остаются универсальным и перспективным инструментом для решения задач оценки живучести[48,49,54,55].

1.6 Классификация негативных внешних воздействий на информационную структуру

Современные информационные структуры приобретают глобальный характер, охватывая все сферы жизнедеятельности человека, формируя информационное единство всей человеческой цивилизации [56,57]. С широким использованием информационных структур увеличивается возможность их уязвимости за счет влияния НВВ как непреднамеренного, так и преднамеренного характера.

Непреднамеренные угрозы могут быть вызваны авариями и стихийными бедствиями, сбоями и отказами в работе компьютерных систем, ошибками, допущенными при их разработке, а также ошибочными действиями пользователей и обслуживающего персонала. В настоящее время такие угрозы достаточно хорошо изучены статистическими методами, что позволяет вполне адекватно противодействовать им [56,58].

Множество непреднамеренных угроз, связанных с внешними факторами, обусловлено влиянием воздействий, неподдающихся предсказанию. К ним относят угрозы, связанные со стихийными бедствиями, техногенными, политическими, экономическими, социальными факторами, развитием информационных и коммуникационных технологий, другими внешними воздействиями.

К внутренним непреднамеренным относят угрозы, связанные с отказами вычислительной и коммуникационной техники, ошибками программного обеспечения, персонала, другими внутренними непреднамеренными воздействиями.

Более опасными с точки зрения характера и последствий проявления являются преднамеренные угрозы, вызванные злоумышленными действиями людей. Арсенал и изощренность таких угроз неуклонно расширяются, поэтому для их нейтрализации необходимо предугадать не

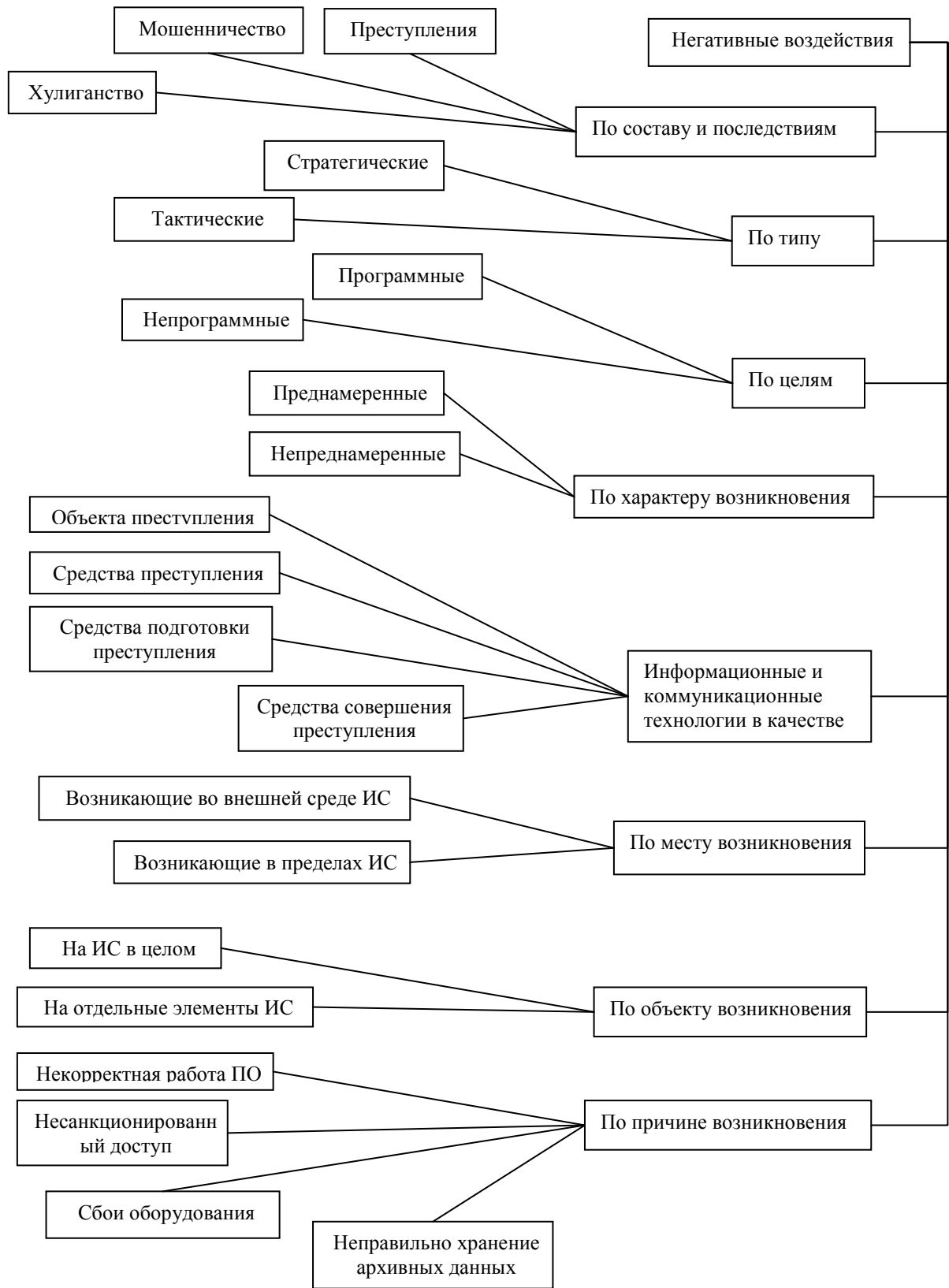


Рисунок 1.2– Классификация негативных воздействий на информационные структуры

только возможные цели злоумышленника, но и его квалификацию, и техническую оснащенность.

Классификация негативных воздействий на информационные структуры представлена на рисунке 1.2.

В результате НВВ на информационную структуру, возникают первичные последствия, которые выражаются в нарушении работоспособности элементов, связей, а также работы функционирования параметров и характеристик информационной структуры [58,59].

1.7 Выводы по первой главе. Постановка цели и задач исследования

Анализ предметной области показал, что недостаточно исследованы существующие методы оценки живучести с точки зрения сокращения времени расчета оценки живучести систем с НИС при НВВ и отсутствует единый универсальный подход к оценке живучести таких систем, параметры и характеристики которых достаточно сложно формализовать, используя детерминированный, статистический или стохастический подходы [60].

В качестве недостатков выделено:

- при вероятностном подходе вероятность разрыва канала ИС, как показано на практике, невозможно получить. Это требует больших материальных и временных затрат, и не является целесообразным.
- потоковые модели требуют определения тяготеющих пар логического ориентированного графа ИС, что на практике является достаточно трудной задачей.
- точный полиномиальный расчет оценки живучести ИС требует больших вычислительных затрат, т.к. вычисления возрастают экспоненциально с увеличением числа узлов и связей.

Поэтому в этом случае единственным возможным и наиболее хорошо себя зарекомендовавшим является подход, основанный на использовании качественной информации и как следствие применение методов теории нечетких множеств.

Использование качественной информации о информационной структуре обуславливает целесообразность применения НПМ, которые позволяют наглядно отобразить информационные процессы и их взаимодействие. Однако данный тип модели приводит к существенным вычислительным затратам и как следствие снижению эффективности.

В связи с этим актуальным является использование ННПС для сокращения времени получения оценки живучести информационных структур, так как она позволяет решать неформализованные задачи, что более эффективно по сравнению с традиционными методами, применяемыми при оценке живучести НИС.

Использование ННПС, построенных на основе НПМ позволяют повысить эффективность проводимых расчетов. Таким образом, результаты исследований, проведенные на основе изучения предметной области, позволили сформулировать цель и задачи исследования.

Цель исследования заключается в построении моделей и алгоритмов оценки живучести системы с нечеткой информационной структурой, связывающих ее качественные характеристики с переменными модели для оценки живучести, обеспечивающие сокращение времени расчета. Для достижения цели поставлены следующие задачи:

1. Провести анализ существующих методов и подходов к оценке живучести информационных структур.
2. Построить нечеткую продукционную модель оценки живучести информационных структур.
3. Построить нечеткую нейросетевую модель, алгоритм перехода от нечеткой продукционной модели к соответствующей ей нечеткой нейросетевой модели и алгоритм обучения нечеткой нейронной продукционной сети.
4. Провести вычислительный эксперимент с использованием разработанных моделей и алгоритмов оценки живучести систем с НИС с оценкой достоверности полученных результатов.

2 Продукционная и нейросетевая модели расчета оценки живучести информационных структур

2.1 Типы нейронных сетей

2.1.1 Однонаправленные многослойные сети сигмоидального типа

Однонаправленные многослойные сети сигмоидального типа имеют устоявшееся название многослойный перцептрон MLP (MultiLayer Perceptron) [48,51,61]. На рисунке представлен двухслойный перцептрон.

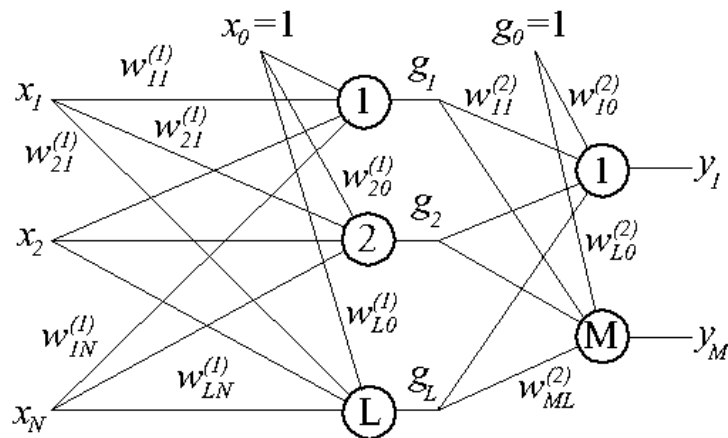


Рисунок 2.1 – Двухслойный перцептрон

Здесь $g_l, l = 1, 2, \dots, L$ – выходные сигналы первого слоя нейронов. Верхние индексы в скобках (m), $m = 1, 2, \dots, n$ означают номер слоя нейрона [62,146].

На рисунке 2.1 представлена трехслойная нейронная сеть, где входные сигналы x_1, x_2, \dots, x_N сети составляют входной слой, выходные сигналы первого нейронного слоя g_1, g_2, \dots, g_L образуют первый скрытый слой, а выходные сигналы y_1, y_2, \dots, y_M – выходной слой.

Выходные сигналы нейронных слоев рассчитываются по следующим формулам:

$$g_l = f \left(\sum_{j=0}^N w_{lj}^{(1)} x_j \right), l = 1, 2, \dots, L$$

$$y_i = f\left(\sum_{l=0}^N w_{il}^{(2)} g_l\right) = f\left(\sum_{l=0}^L w_{il}^{(2)} f\left(\sum_{j=0}^N w_{lj}^{(1)} x_j\right)\right), i = 1, 2, \dots, M$$

Необходимо подчеркнуть, что функции активации, полученные в литературных источниках [63,147], всех нейронов сети идентичны.

Обучение нейронной сети заключается в том, что необходимо правильно подобрать такие значения весовых коэффициентов сети $w_{lj}^{(1)}$ и $w_{il}^{(2)}$, которые смогут обеспечить максимальное совпадение выходного вектора Y^k и эталонного вектора ожидаемых значений D^k при входном векторе X^k .

В случае единичной обучающей выборки $\langle X, D \rangle$ целевая функция согласно работам [48] имеет вид:

$$E(W) = \frac{1}{2} \sum_{i=1}^N (y_i - d_i)^2$$

В случае множества обучающих пар $\langle X^k, D^k \rangle, k=1, 2, \dots, p$ целевая функция, представленная в работах [48], сумма по всем парам имеет вид:

$$E(W) = \frac{1}{2} \sum_{k=1}^p \sum_{i=1}^M (y_i - d_i)^2$$

2.1.2 Сети с самоорганизацией на основе конкуренции

Нейронные сети с самоорганизацией называются сети, не требующие для своего обучения "учителя" и самостоятельно адаптирующие свои веса под обучающие данные [48,51,61]. Как правило, это однослойные сети, в которых каждый нейрон получает все компоненты входного вектора X размерностью N . На рисунке 2.2 представлена структурная схема такой нейронной сети.

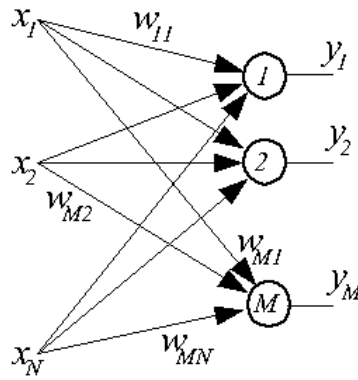


Рисунок 2.2 - Структурная схема сети с самоорганизацией на основе конкуренции

Веса входных связей i -ого нейрона образуют вектор $W_i = [w_{i1}, w_{i2}, \dots, w_{iN}]^T$. Кроме связей, явно представленных в схеме, на этапе обучения имеют место связи между нейронами, позволяющие судить о степени "соседства" нейронов друг с другом. Такие сети часто требуют нормализации значений входного вектора.

Процесс обучения сети выглядит следующим образом. На вход сети подается обучающий вектор X^k , для каждого нейрона определяется расстояние $d(X^k, W_i)$ между векторами X^k и W_i . Определяется первый нейрон, для которого это расстояние оказывается наименьшим и образуется окрестность S_w^k из соседних нейронов с известным "расстоянием" до первого нейрона. Веса первого нейрона и веса соседних из окрестности S_w^k уточняются по правилу Кохонена согласно работам [64,65]:

$$W_i^{k+1} = W_i^k + \eta_i^k (X^k - W_i^k)$$

где η_i^k – коэффициент обучения, значение которого уменьшается с увеличением расстояния от i -ого нейрона до первого. Веса нейронов вне S_w^k не изменяются. Размер окрестности S_w^k и величина η_i^k с течением времени обучения уменьшаются.

2.1.3 Радиальные нейронные сети

Радиальные нейронные сети строятся с использованием радиальных нейронов, функция активации которых имеет ненулевые значения только в окрестностях своего центра. [48,51,61].

Радиальная сеть имеет двухслойную структуру[49,51,54,63], первый слой составляют радиальные нейроны, выходной — один или несколько линейных. На рисунке 2.3 представлена структурная схема радиальной сети с одним выходным нейроном.

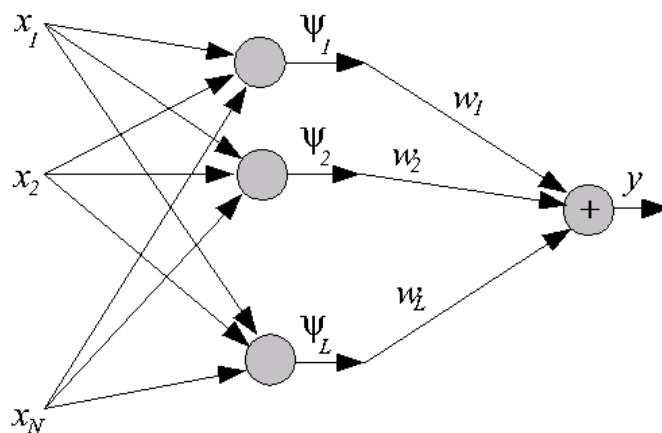


Рисунок 2.3 – Структурная схема радиальной сети с одним выходным нейроном

Здесь $\psi_i (\| X_i - C_i \|)$ – функция активации i -ого радиального нейрона, называемая базисной функцией.

Сеть выполняет аппроксимацию входных данных согласно выражению

$$y = \sum_{i=1}^L w_i \psi_i (\| X - C_i \|), \text{ в соответствии с [49,51,54,63].}$$

Задача обучения сети состоит в соответствии с [49,51,54,63] состоит в подборе таких значений L, C_i и w_i , которые доставляют минимум целевой функции

$$E = \frac{1}{2} \sum_{k=1}^p \left(\sum_{i=1}^M w_i \psi (\| X^k - C_i \|) - d^k \right)^2,$$

где p – количество обучающих выборок.

2.1.4 Нечеткая нейронная продукционная сеть Такаги-Сугено-Канга

Нечеткие нейронные сети дают возможность решать неформализованные задачи, подавая на вход сети различные данные и оценивая выдаваемый результат.

Выходной сигнал в нечеткой продукционной сети Такаги-Сугено-Канга (TSK) [48] рассчитывается следующим образом:

$$y(X) = \frac{\sum_{i=1}^M w_i y_i(X)}{\sum_{i=1}^M w_i},$$

где $y_i(X) = p_{i0} + \sum_{j=1}^N p_{ij} x_j$ – i -ый полиномиальный компонент аппроксимации.

Весовые коэффициенты w_i компонентов сети можно рассчитать по следующей формуле (в данном случае используется рациональная форма функции Гаусса, но можно использовать и другие функции)

$$w_i = \prod_{j=1}^N w_{ij}(x_j) = \prod_{j=1}^N \frac{1}{1 + \left(\frac{|x_j - c_{ij}|}{\sigma_{ij}} \right)^{2b_{ij}}}$$

Данным выражениям соответствует пятислойная нечеткая нейронная сеть, которая представлена на рисунке 2.4.

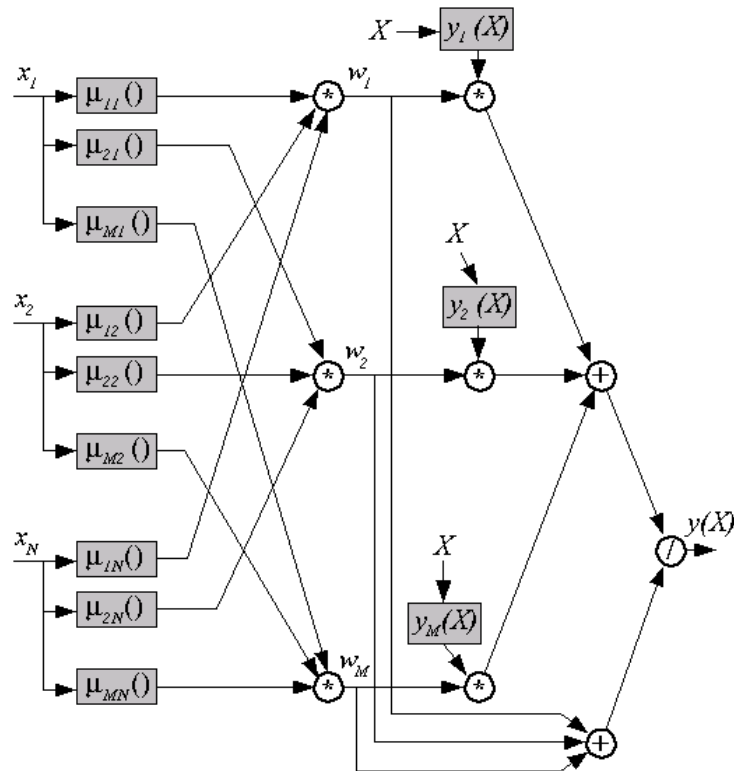


Рисунок 2.4. Структурная схема нечеткой продукционной сети сети Такаги-Сугено-Канга

Первый слой содержит $N \times M$ узлов, где каждый реализует расчет функции Гаусса с параметрами c_{ij} , σ_{ij} и b_{ij} . В этом слою происходит фаззификация входных переменных сети, где изменяются параметры c_{ij} , σ_{ij} и b_{ij} в процессе обучения сети и он является параметрическим слоем

Второй слой сети не содержит параметров и в нем происходит агрегирование левых частей продукций.

В третьем слою происходит генерация полиномиальных функций TSK $y_i(X)$ и умножение их на весовой коэффициент w_i . В процессе обучения сети изменению и настройке подвергаются коэффициенты p_{ij} , $i = 1, 2, \dots, M$; $j = 0, 1, \dots, N$ и он является параметрическим слоем. Общее количество коэффициентов p_{ij} в сети равно $M \times (N + 1)$.

В четвертом слое находятся два нейронных сумматора. При помощи первого сумматора рассчитывается взвешенная сумма выходных сигналов $y_i(X)$, а второй рассчитывает сумму весовых коэффициентов w_i , $i = 1, 2, \dots, M$. Данный слой не является параметрическим т.к. в нем нет настройки и изменения параметров сети.

В пятом слое производится нормализация весовых коэффициентов. Этот слой также не является параметрическим.

Сеть Такаги-Сугено-Канга[48,51,61] содержит два параметрических слоя (первый и третий), параметры, которых подлежат изменению и настройке в процессе обучения сети. Параметры первого слоя относятся к нелинейным функциям и называются нелинейными, а параметры третьего слоя – линейными.

Общее количество линейных и нелинейных параметров сети TSK рассчитывается следующим образом $M \cdot 3N + M \cdot (N + 1) = M \cdot 4(N + 1)$.

Во многих практических приложениях это достаточно большая величина, поэтому для входных переменных x_j используют ограниченный набор функций $\mu(x_j)$, что уменьшает количество нелинейных параметров[48,51,61].

2.1.5 Нечеткие нейронные сети. Сеть Ванга-Менделя

В нечеткой нейронной сети Ванга-Менделя[48,51,61] выходной сигнал сети можно представить следующим образом:

$$y(X) = \frac{\sum_{i=1}^M c_i w_i}{\sum_{i=1}^M w_i} = \frac{\sum_{i=1}^M c_i \prod_{j=1}^N \mu_{ij}(x_j)}{\sum_{i=1}^M \prod_{j=1}^N \mu_{ij}(x_j)},$$

где c_i – весовой коэффициент, являющиеся центром функции принадлежности правой части продукции, μ_{ij} – функция Гаусса

представленная в экспоненциальном или рациональном виде с параметрами центра c_{ij} , ширины σ_{ij} и формы b_{ij} .

Выражение выходного сигнала $y(X)$ в сети Ванга-Менделя является частным случаем аналогичного выражения в сети TSK, если в последней принять $y_i(X) = c_i$. Поэтому сеть Ванга-Менделя проще и имеет следующую трехслойную структуру (рисунок 2.5).

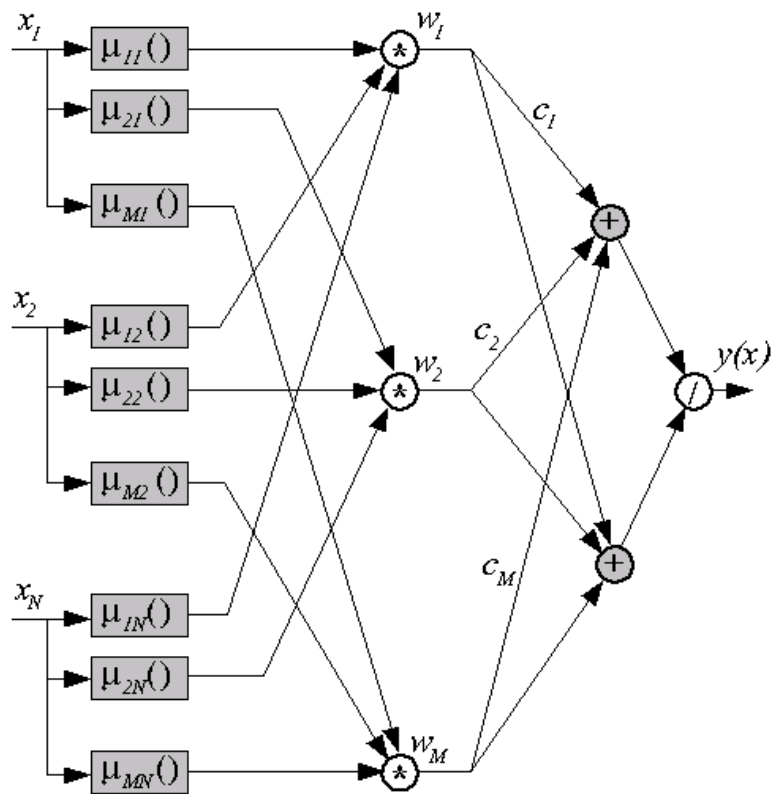


Рисунок 2.5 Структурная схема сети Ванга-Менделя

В данной сети параметры изменяются в первом и третьем слое, которые являются параметрическими. Первый слой сети содержит $3M \cdot N$ нелинейных параметров функции Гаусса, а третий M – линейных параметров c_i [48,51,61].

2.2 Анализ нейронных сетей для построения нечеткой нейронной модели оценки живучести информационных структур

Анализ нейронных сетей рассмотренных ранее показал, что для решения поставленных задач в главе 1 целесообразно использовать нечеткие нейронные продукционные сети в ряде их достоинств и преимуществ (рисунок 2.4).

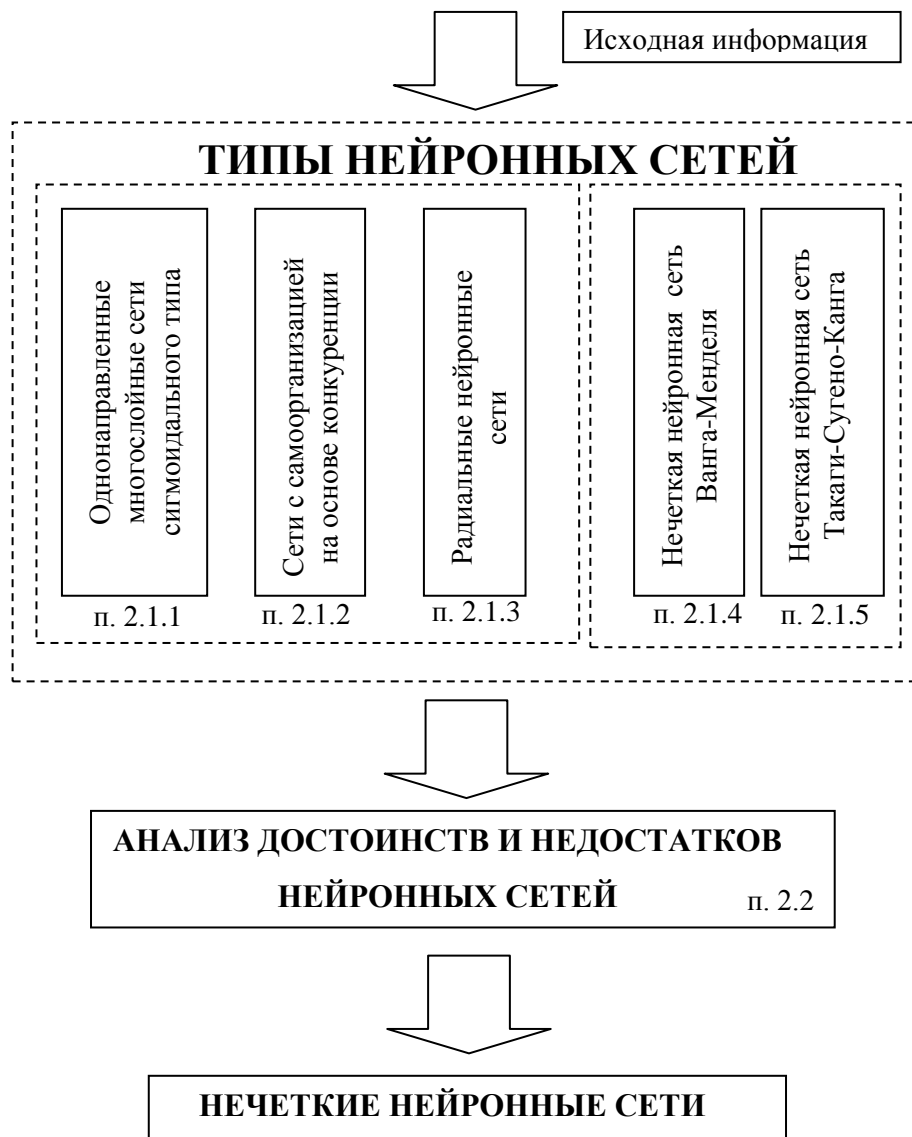


Рисунок 2.6 – Анализ нейронных сетей

Нечеткие нейронные сети наделены всеми достоинствами обычных нейронных сетей, а так же системой нечеткого вывода и нечеткой

логикой[50,66]. Они позволяют разрабатывать и представлять модели систем с помощью нечетких правил, которые обладают наглядностью и простотой представления.

Преимуществом нечетких нейронных сети заключается в распараллеливании[67-72] обработки информации и способности самообучаться, создавая обобщения, которые способны получать обоснованный результат на основании данных, которые не участвовали и не встречались в процессе обучения сети. Это преимущество нечетких нейронных сетей позволяет решать сложные, а порой и масштабные задачи, которые на сегодняшний день являются трудноразрешимыми. К таким задачам можно отнести оценку живучести информационных структур, где при добавлении определенного количества узлов и связей в информационную структуру экспоненциально возрастают вычислительные затраты и ресурсы.

Перечислим основные преимущества нечетких нейронных сетей, что обуславливает целесообразность их использования для решения поставленных задач в главе 1:

1. Возможность решения поставленной задачи при неизвестных данных и закономерностях.

На множестве сгенерированных топологий информационной структуры, обученная нечеткая нейронная сеть способна решать задачи, где неизвестны данные и закономерности развития определенной ситуации в зависимости между входными и выходными данными.

2. Устойчивость нечеткой нейронной сети к шумам во входной информации.

Выполнение работы всей сети при большом количестве неинформативных шумов во входных данных и сигналах. В этом случае нечеткая нейронная сеть определит малозначимость и непригодность входных данных и сигналов для решения поставленной задачи и сама отбросит их.

3. Адаптация к окружающей среде и ее изменениям.

Нечеткие нейронные сети способны адаптироваться к окружающей среде и ее изменениям. Если нечеткие нейронные сети, обученные работать в определенной окружающей среде, то они могут быть переучены в условиях несущественных колебаний параметров среды для дальнейшей работы.

4. Нечетким нейронным сетям присуще высокое быстродействие.

Использование массового параллелизма обработки данных информации позволяет нечетким нейронным сетям обладать высоким быстродействием.

5. Отказоустойчивость нечеткой нейронной сети при реализации аппаратной составляющей.

При воздействии неблагоприятных воздействий на нечеткие нейронные сети, производительность которых падает незначительно, и являются отказоустойчивыми.

С помощью алгоритмов нечеткого вывода характеристики ИС представлены в виде нечетких продукционных моделей. По структуре нечеткие продукционные сети похожи с многослойными нейронными сетями с прямым распространением сигнала, где элементы каждого слоя нечеткой продукционной модели реализуют отдельный этап нечеткого вывода:

- в первом слое реализуется функция фазификации т.е. приведение четких данных в нечеткие входной переменной;
- последующие слои – скрытые. Они представляют совокупность всех нечетких правил, реализующие алгоритм нечеткого вывода
- последние слои реализуют функцию дефазификации т.е. приведения нечеткости к четным данным выходной переменной.

В связи имеющихся недостатков нечетких продукционных моделей и для их устранения, необходимо создавать адаптивные модели, в которых реализуется коррекция состава правил в базе, весовых коэффициентов и

параметров функций принадлежности в процессе и по результатам функционирования сети.

2.3 Построение продукционной и нейросетевой модели расчета оценки живучести информационных структур

Проведенный анализ нечетких нейронных сетей в параграфе 2.2 позволяет сделать вывод, что для решения поставленной задачи в главе 1 целесообразно использовать нечеткую нейронную продукционную сеть [48,49], т. к она наиболее отказоустойчива. Это значит, что при НВВ, производительность падает незначительно, и ННПС будет более адекватна при обучении.

Для нечеткого вывода воспользуемся алгоритмом Мамдани, в котором в качестве нечеткой импликации используем T -импликацию.

Операция нечеткой импликации является ключевым элементом в нечетких продукционных моделях [48]. Она определяет причинно-следственное отношение между предпосылками и заключениями правил.

Выделяют три основных класса операций нечеткой импликации [48]:

– S -импликации, определяемые как

$$\mu_R(x, y) = S(NOT(\mu_A(x), \mu_B(y))),$$

где S – операция S -нормы, NOT – оператор нечеткого дополнения на интервале значений $[0, 1]$ для $\mu_A(x)$, $\mu_B(y)$.

– T -импликации, определяемые как

$$\mu_R(x, y) = T(\mu_A(x), \mu_B(y)), \quad (2,1)$$

где T – операция T -нормы [76,77].

– R -импликации, образованные на основе сравнения с T -нормой:

$$\mu_R(x, y) = \sup\{z \in [0,1] \mid T(\mu_A(x), z) \leq \mu_B(y)\}.$$

Хотя эти операции импликации не удовлетворяют их общепринятым свойством импликации, они широко используются в различных

приложениях нечеткой логики. Типичными примерами операций T -импликации является импликация Мамдани и Ларсена.

Операции нечеткой импликации принято оценивать по двум критериям [48]:

1. максимальная ошибка при нечетком выводе, обусловленная операцией импликации, не должна превышать максимальной ошибки лингвистического представления результата;

2. нечеткая модель, построенная с использованием выбранной операции нечеткой импликации, должна быть робастной по отношению к выбору вида функции принадлежности.

Отметим, что большинство операций импликации, широко используемых в настоящее время, не являются таковыми. В связи с этим, в зарубежной литературе [148] они получили новое название – инженерные импликации.

Примерами таких импликаций являются импликации Мамдани и Ларсена, которые используются практически во всех работах, посвященных вопросам моделирования объектов и систем, в том числе и информационных, в условиях качественной неопределенности. Доказательство того, что импликации Мамдани и Ларсена таковыми не являются в классическом смысле, представлено в Приложении Г.

При построении нечеткой продукционной модели введены лингвистические переменные [78,79,81]: дальность передачи между узлами (x_1), количество элементов и многообразие связей между ними (x_2), возможность разрыва связей (x_3), сложность системы (x_4), территориальная распределенность (x_5) и т.п.

Далее формализуем лингвистические переменным в следующем виде $\langle x_1, T_1, X_1, G_1, M_1 \rangle$, где

x_1 – «дальность передачи между узлами»; T_1 – {«малая», «средняя», «высокая»}; X_1 – [0, 1000]; G_1 – процедура образования новых термов с помощью различных связок и модификаторов; M_1 – процедура задания на

$X_1 = [0, 1000]$ нечетких подмножеств $A_{11} = \text{«малая»}$, $A_{12} = \text{«средняя»}$, $A_{13} = \text{«высокая»}$.

x_2 – «количество элементов и многообразие связей между ними»; T_2 – {«малое», «среднее», «большое»}; $X_2 = [0, 100]$; G_1 – процедура образования новых термов с помощью различных связок и модификаторов; M_2 – процедура задания на $X_2 = [0, 100]$ нечетких подмножеств $A_{21} = \text{«малое»}$, $A_{22} = \text{«среднее»}$, $A_{23} = \text{«большое»}$.

x_3 – «возможность разрыва связей»; T_3 – {«низкая», «средняя», «высокая»}; $X_3 = [0, 100]$; G_1 – процедура образования новых термов с помощью различных связок и модификаторов; M_3 – процедура задания на $X_3 = [0, 100]$ нечетких подмножеств $A_{31} = \text{«низкая»}$, $A_{32} = \text{«средняя»}$, $A_{33} = \text{«высокая»}$.

x_4 – «сложность системы»; T_4 – {«простая», «сложная», «очень сложная»}; $X_4 = [0, 100]$; G_1 – процедура образования новых термов с помощью различных связок и модификаторов; M_4 – процедура задания на $X_4 = [0, 100]$ нечетких подмножеств $A_{41} = \text{«простая»}$, $A_{42} = \text{«сложная е»}$, $A_{43} = \text{«очень сложная»}$.

x_5 – «территориальная распределенность»; T_5 – {«локальные», «региональные», «глобальные»}; $X_5 = [0, 1000]$; G_1 – процедура образования новых термов с помощью различных связок и модификаторов; M_5 – процедура задания на $X_5 = [0, 1000]$ нечетких подмножеств $A_{51} = \text{«малое»}$, $A_{52} = \text{«среднее»}$, $A_{53} = \text{«высокое»}$.

y_1 – «живучесть»; T_1 – {«низкая», «средняя», «высокая»}; $Y_1 = [0, 100]$; G_1 – процедура образования новых термов с помощью различных связок и модификаторов; M_1 – процедура задания на $Y_1 = [0, 100]$ нечетких подмножеств $B_{11} = \text{«низкая»}$, $B_{12} = \text{«средняя»}$, $B_{13} = \text{«высокая»}$.

Затем формируем базу нечетких [80,81]правил на основе алгоритма Мамдани из множества[149] отдельных нечетких продукционных правил вида «ЕСЛИ A , ТО B » (где A и B – предпосылка и заключение правила), который является достаточным для решения поставленной задачи.

Фрагмент правил нечеткой продукционной модели представлен следующим образом:

P_1 : ЕСЛИ «дальность передачи между узлами» есть «высокая» И «количество элементов и многообразие связей между ними» есть «большое» И «возможность разрыва связей» есть «высокая» И «сложность системы» есть «сложная» И «территориальная распределенность» есть «глобальная», ТО «живучесть» есть «низкая»

(P_1 : Если x_1 есть A_{13} И x_2 есть A_{23} И x_3 есть A_{31} И x_4 есть A_{43} И x_5 есть A_{53} то B_{11})

P_2 : ЕСЛИ «дальность передачи между узлами» есть «средняя» И «количество элементов и многообразие связей между ними» есть «большое» И «возможность разрыва связей» есть «высокая» И «сложность системы» есть «сложная» И «территориальная распределенность» есть «глобальная», ТО «живучесть» есть «низкая»

(P_2 : Если x_1 есть A_{12} И x_2 есть A_{23} И x_3 есть A_{31} И x_4 есть A_{43} И x_5 есть A_{53} то B_{11})

⋮

P_{25} : ЕСЛИ «дальность передачи между узлами» есть «средняя» И «количество элементов и многообразие связей между ними» есть «средняя» И «возможность разрыва связей» есть «низкая» И «сложность системы» есть «мало сложная» И «территориальная распределенность» есть «региональная», ТО «живучесть» есть «средняя»

(P_{25} : Если x_1 есть A_{12} И x_2 есть A_{22} и x_3 есть A_{31} И x_4 есть A_{41} И x_5 есть A_{52} то B_{32})

⋮

P_{30} : ЕСЛИ «дальность передачи между узлами» есть «низкая» И «количество элементов и многообразие связей между ними» есть «малое» И «разрыв связи» есть «низкая» И «сложность системы» есть «мало сложная» И «территориальная распределенность» есть «локальная», ТО «живучесть» есть «высокая»

(П₃₀: Если x_1 есть A_{11} И x_2 есть A_{21} и x_3 есть A_{31} И x_4 есть A_{41} И x_5 есть A_{51} то B_{33})

По каждому из правил производим активизацию [48] (определение степеней истинности) заключений.

На рисунке 2.7 представлена активизация заключений для двух лингвистических переменных x_1 и x_2 .

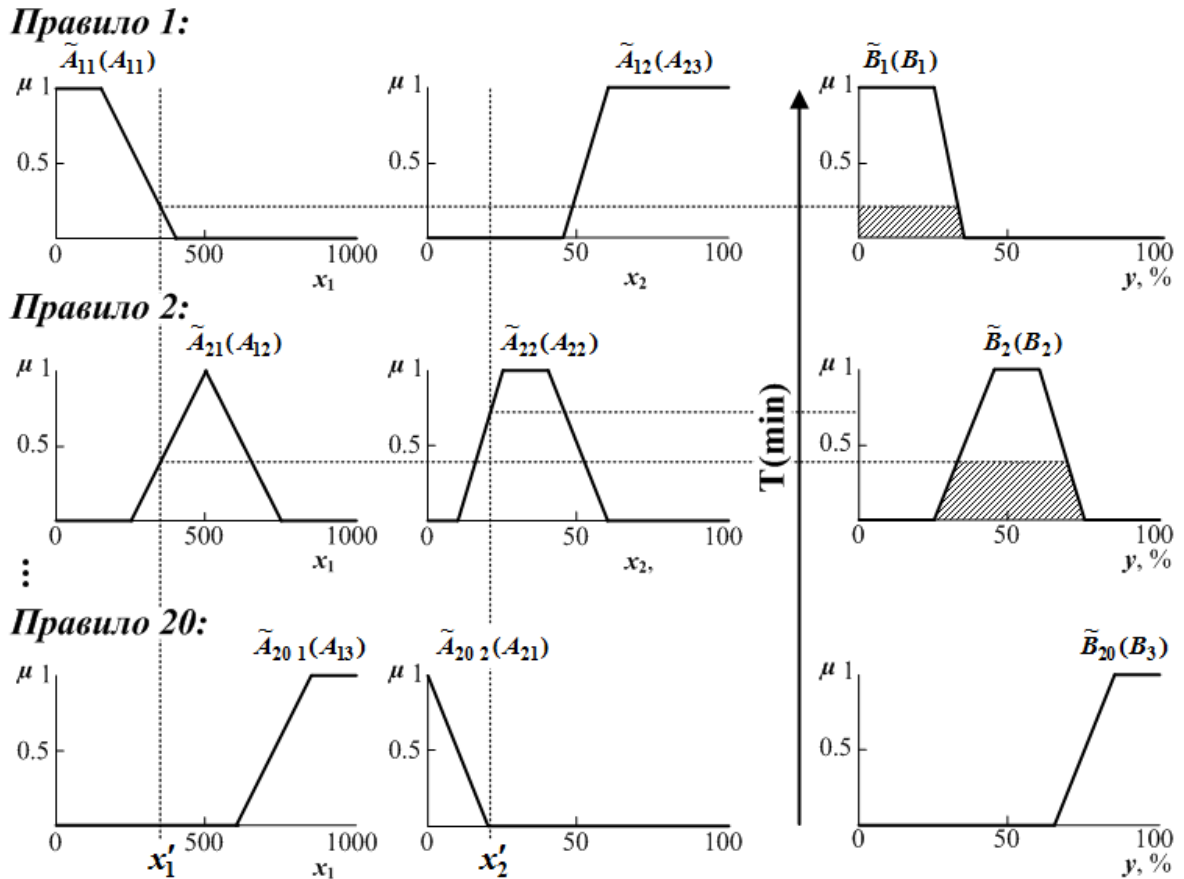


Рисунок 2.7 – Иллюстрация активизации заключений алгоритма нечеткого вывода

Использование построенной НПМ для построения оценок живучести НИС приводит к большим вычислительным затратам, поэтому целесообразно, как отмечается в различных литературных источниках [48,49], перейти к ННПС. Алгоритм перехода от НПМ к ННПС, обладающий элементами научной новизны, рассмотрен подробно в главе 3.

ННПС модель расчета оценки живучести НИС представлена на рисунке 2.8.

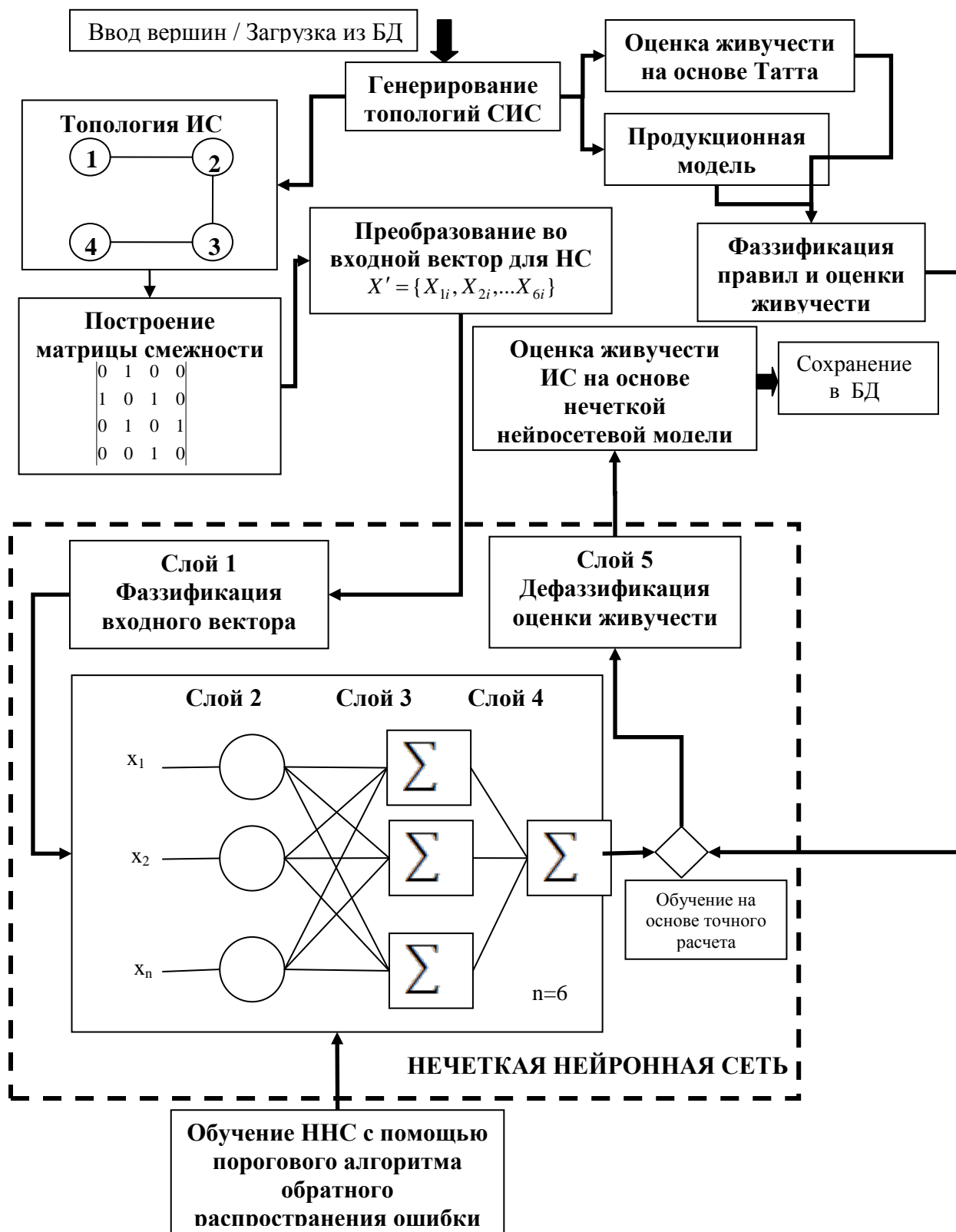


Рисунок 2.8 - ННПС модель оценки живучести ИС

Процесс создания ННПС является сложным, т.к при проектировании структуры нейронной сети возникают трудности с определением процедуры выбора количества нейронов и слоев в сети. Если в нечеткой нейронной сети большое количество нейронов и слоев, то сеть обучается и работает медленнее, чем небольшое количество и зависимость вход – выход может быть более нелинейной. Количество нейронов и количество скрытых слоев будет эмпирически уточнено в процессе разработки программного обеспечения оценки живучести информационных структур.

Функция активации выбираются в зависимости от требований к моделируемому процессу [83]. В таблице 2.1 представлены наиболее используемые функций активаций[49].

Таблица 2.1 – Функция активации для обучения нечеткой нейронной сети

Название	Формула	Область значений
Пороговая	$\Psi = F(\psi) = \begin{cases} 0, \psi < 0, \\ 1, \psi \geq 0 \end{cases}$	0, 1
Знаковая	$\Psi = F(\psi) = \begin{cases} 1, \psi > 0, \\ -1, \psi \leq 0 \end{cases}$	-1, 1
Сигмовидная	$\Psi = F(\psi) = \frac{1}{1 + e^{-\psi}}$	(0, 1)
Полулинейная	$\Psi = F(\psi) = \begin{cases} \psi, \psi > 0, \\ 0, \psi \leq 0 \end{cases}$	(0, ∞)
Линейная	$\Psi = F(\psi) = \psi$	($-\infty, \infty$)
Радиальная базисная	$\Psi = F(\psi) = e^{-\psi^2}$	(0, 1)
Полулинейная с насыщением	$\Psi = F(\psi) = \begin{cases} 0, \psi \leq 0, \\ \psi, 0 < \psi < 1, \\ 1, \psi \geq 1 \end{cases}$	(0, 1)

Окончание таблицы 2.1

Линейная с насыщением	$\Psi = F(\psi) = \begin{cases} -1, & \psi \leq -1, \\ \psi, & -1 < \psi < 1, \\ 1, & \psi \geq 1 \end{cases}$	(-1, 1)
Гиперболический тангенс	$\Psi = F(\psi) = \frac{e^\psi - e^{-\psi}}{e^\psi + e^{-\psi}}$	(-1, 1)
Треугольная	$\Psi = F(\psi) = \begin{cases} 1 - \psi , & \psi \leq 0, \\ 0, & \psi > 0 \end{cases}$	(0, 1)

Одними из наиболее распространенных функций являются:

1. линейная,
2. логистическая функция или сигмоид с насыщением, которая является не линейной,
3. гиперболический тангенс.

В качестве функции активации будем использовать логистическую функцию ($\frac{1}{1+e^{-x}}$, рисунок 2.5), т.к. она принадлежит к классу сигмоидальных функций, и ее аргумент может принимать любое значение в диапазоне от $-\infty$ до $+\infty$, а выход изменяется в диапазоне от 0 до 1.

Сигмоидная функция (рисунок 2.9) дифференцируема на всей оси абсцисс, что применительно во многих алгоритмах обучения нечеткой нейронной сети.

Кроме того, сигмоидная функция усиливает слабые сигналы лучше, чем сильные, и тем самым не дает насыщение от сильных сигналов, так как они соответствуют областям аргументов, где функция имеет пологий наклон (рисунок.2.9). Это делает логистическую функцию наиболее применимой.

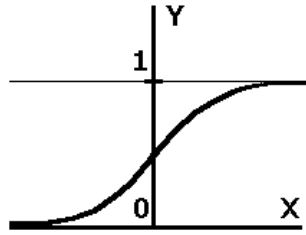


Рисунок 2.9 – Логистическая функция активации

При M правилах вывода агрегирование выходного результата сети производится по формуле предложенной в работе[48]:

$$y(x) = \frac{1}{\sum_{i=1}^M w_i} \sum_{i=1}^M w_i y_i(x), \quad (2.2)$$

где $y_i(x) = p_{i0} + \sum_{j=1}^N p_{ij} x_j$, агрегация импликации. Присутствующие в

этом выражении веса w_i интерпретируются как компоненты $\mu_A^{(i)}(x)$, определённые формулой (2.2).

В построенной ННПС используется пять слоёв:

1. Первый слой выполняет отдельную фазификацию входного вектора $x' = \{x_1, x_2, \dots, x_N\}$, определяя для каждого i -го правила вывода значение коэффициента принадлежности $\mu_A^{(i)}(x_j)$ заданных гауссовыми функциями принадлежности с параметрами (a_{ij}, b_{ij}) . В этом слое подлежат изменению параметры (a_{ij}, b_{ij}) которые изменяются в процессе обучения сети и этот слой является параметрическим.

2. Второй слой выполняет агрегирование отдельных переменных x' , определяя результирующее значение коэффициента функции принадлежности $w_i = \mu_A^{(i)}(x)$ для входного вектора x в соответствии с выше представленной формулой (2.2). Этот слой не меняется в процессе создания сети и не является параметрическим.

3. Третий слой рассчитывает значения по формуле $y_i(x) = p_{i0} + \sum_{j=1}^N p_{ij} x_j$

и представляет собой генератор функции. Также здесь происходит умножение выходных сигналов $y_i(x)$ на весовые значения w_i , сформированные в предыдущем слое. В этом слое изменения подлежат линейные веса p_{ij} для $i = 1, 2, \dots, M$ и $j = 1, 2, \dots, N$ и он является параметрическим слоем.

4. Четвёртый слой состоит из двух нейронов, которые называются сумматорами, которых один рассчитывает взвешенную сумму сигналов $y_i(x)$, а второй сумматор определяет сумму весовых коэффициентов $\sum_{i=1}^M w_i$.

Данный слой не изменяется в процессе обучения и не является параметрическим.

5. Пятый слой состоит из одного нормализующего элемента, который выполняет дефаззификацию выходной переменной. Это также непараметрический слой.

При уточнении функциональной зависимости (2.2) для нечеткой нейронной сети получаем:

$$y(x) = \frac{1}{\sum_{i=1}^M \left[\prod_{j=1}^N T(\mu_A^{(i)}(x), \mu_A^{(i+1)}(x)) \right]} \sum_{i=1}^M \left[\prod_{j=1}^N T(\mu_A^{(i)}(x), \mu_A^{(i+1)}(x)) \right] \times \left[p_{i0} + \sum_{j=1}^N p_{ij} x_j \right] \quad (2.3)$$

где $T(\mu_A^{(i)}(x), \mu_A^{(i+1)}(x))$ – Т норма, которая выбирается из таблицы 2.2

Таблица 2.2 – Примеры T- и S- норм

№ П/ п	$T(a, b)$ $a = \mu_A(x), b = \mu_B(x)$	$S(a, b)$ $a = \mu_A(x), b = \mu_B(x)$	Параметры
1	$\min\{a, b\}$	$\max\{a, b\}$	-
2	$a \cdot b$	$a + b - a \cdot b$	-
3	$\max\{0, a + b - 1\}$	$\min\{1, a + b\}$	-
4	$\begin{cases} a, \text{ если } b = 1, \\ b, \text{ если } a = 1, \\ 0, \text{ если } a, b < 1 \end{cases}$	$\begin{cases} a, \text{ если } b = 0, \\ b, \text{ если } a = 0, \\ 0, \text{ если } a, b > 0 \end{cases}$	-
5	$\frac{ab}{\max\{a, b, \gamma\}}$	$\frac{(1-a)(1-b)}{\max\{(1-a), (1-b), \gamma\}}$	$\gamma \in [0, 1]$
6	$\frac{ab}{\gamma + (1-\gamma)(a+b-ab)}$	$\frac{a+b-(2-\gamma)ab}{\gamma - (1-\gamma)ab}$	$\gamma > 0$
7	$\frac{1}{1 + \left(\left(\frac{1}{a} - 1 \right)^\gamma + \left(\frac{1}{b} - 1 \right)^\gamma \right)^{\frac{1}{\gamma}}}$	$\frac{1}{1 + \left(\left(\frac{1}{a} - 1 \right)^{-\gamma} + \left(\frac{1}{b} - 1 \right)^{-\gamma} \right)^{\frac{1}{\gamma}}}$	$\gamma > 0$
8	$\frac{1}{\left(\frac{1}{a^\gamma} + \frac{1}{b^\gamma} \right)^{\frac{1}{\gamma}} - 1}$	$1 - \frac{1}{\left(\frac{1}{(1-a)^\gamma} + \frac{1}{(1-b)^\gamma} - 1 \right)^{\frac{1}{\gamma}}}$	$\gamma > 1$
9	$1 - \left(\frac{(1-a)^\gamma + (1-b)^\gamma - 1}{-(1-a)^\gamma(1-b)^\gamma} \right)^{\frac{1}{\gamma}}$	$\left(a^\gamma + b^\gamma - a^\gamma b^\gamma \right)^{\frac{1}{\gamma}}$	$\gamma > 1$
10	$\max \left\{ 0, 1 - \left((1-a)^\gamma + (1-b)^\gamma - 1 \right)^{\frac{1}{\gamma}} \right\}$	$\min \left\{ 1, \left(a^\gamma + b^\gamma - a^\gamma b^\gamma \right)^{\frac{1}{\gamma}} \right\}$	$\gamma \geq 1$
11	$\log_\gamma \left(1 + \frac{(\gamma^a - 1)(\gamma^b - 1)}{\gamma - 1} \right)$	$1 - \log_\gamma \left(1 + \frac{\gamma^{1-a} + \gamma^{1-b}}{\gamma - 1} \right)$	$\gamma > 0,$ $\gamma \neq 1$
12	$\max\{0, (\gamma - 1)(a + b) - 1 - \gamma ab\}$	$\min\{1, a + b + \gamma ab\}$	$\gamma > -1$

Анализ Т- и S- норм, представленных в таблице 2.2, которые использовались при решении ряда практических задач, полученные результаты представлены в работах [84,85] и подтверждены актами внедрениями (Приложение 3), позволил сделать вывод, что наиболее эффективной является следующая Т-норма $\frac{ab}{\gamma + (1-\gamma)(a+b-ab)}$ (Приложение Д) и поэтому для уточнения функциональной зависимости (2.3) будем использовать ее, при этом необходимо настроить параметр γ .

Для нахождения γ необходимо поставить и решить задачу безусловной оптимизации. Анализ методов решения задачи одномерной оптимизации [86] позволяет сделать вывод, что наиболее эффективным по скорости сходимости является метод чисел Фибоначчи [87,88]. Однако для его использования необходимо определить или задать интервал измерения переменной, по которой проводится оптимизация. В нашем случае известна левая граница интервала, которая определяется условием $\gamma = 0$, для определения правой границы интервала используем правило Голдстейна [89]. Поставленная задача имеет вид:

$$\sigma \leq \frac{f(\gamma_k + x_k d_k) - f(\gamma_k)}{x_k f'(\gamma_k) d_k} \leq 1 - \sigma$$

где $\sigma \in (0, 1/2)$

При этом целевая функция $f(\circ)$ должна быть ограничена снизу. Данное условие выполняется т.к. в качестве $f(\circ)$ использовалась Т-норма №6 из таблицы 2.2, которая ограничена снизу значением ноль.

Оптимизационная задача выбора оптимального значения γ имеет следующий вид:

$$f(\gamma) \rightarrow \min_{\gamma}$$

$$f(\gamma) = \sum_{i=1}^n (T_i(a, b, \gamma) - T_{зад})^2$$

где $i = \overline{1, n}$ – номер правила в продукционной модели.

Поставленная задача решалась методом чисел Фибоначчи [87,88], применение которого стало возможным вследствие того, что использование правила Голдстейна [89] позволило определить правую границу изменения переменной γ .

Функция $y(x)$ является линейной относительно переменных x_j ($j = 1, 2, \dots, N$), Если в определенный момент времени параметры условия зафиксированы.

При наличии N входных переменных каждое правило формирует $N + 1$ переменных p_{ij} линейной зависимости. При M правилах вывода это даёт $M(N + 1)$ линейных параметров сети. В свою очередь каждая функция принадлежности использует три параметра, подлежащих адаптации. Так как каждая переменная x_j характеризуется собственной функцией принадлежности, то мы получим $3MN$ нелинейных параметров. В сумме это даёт $M(4N + 1)$ линейных и нелинейных параметров, значения которые подбираются в процессе обучения сети [48,89].

2.4 Выводы по второй главе

В результате исследования разработана продукционная и нейросетевая модели сокращения времени расчета оценки живучести систем с НИС. Были введены лингвистические переменные, которые определяют различные характеристики ИС, влияющие на ее живучесть (территориальная распределенность, количество элементов и многообразием связей между ними, дальность передачи между узлами, возможность разрыва связей, сложность информационной системы). С помощью правил продукции и введенных лингвистических переменных экспертами (либо на основе экспериментальных данных) формируется база

нечетких правил, в которой прописывается влияние определенного набора характеристик ИС на живучесть этой системы.

Разработана нечеткая продукционная модель оценки живучести информационных структур, позволяющая формализовать параметры, характеристики и получить оценку живучести информационной структуры в условиях неопределенности.

3 Алгоритмы, реализующие продукционную и нейросетевую модели расчета оценки живучести информационных структур

3.1 Алгоритм перехода от нечеткой продукционной модели к соответствующей ей нечеткой нейросетевой модели

Алгоритм перехода от НПМ, построенной в главе 2, к соответствующей ей ННПС представлен в виде диаграммы деятельности [90,91] на рисунке 3.1 и реализуется 5-ю этапами.

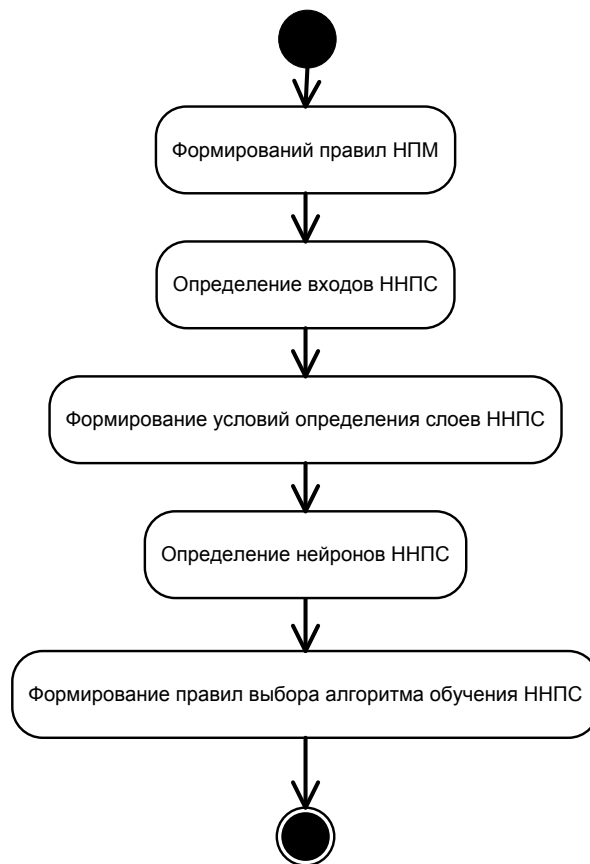


Рисунок –3.1 Алгоритм перехода от НПМ к соответствующей ей ННПС.

1 этап. Формирование правил НПМ рассмотрен в главе 2 и представлен НПМ (2.6).

2 этап. Определение входов ННПС включает следующие правила:

Π_1 : ЕСЛИ «дальность передачи между узлами» есть «высокая» И

«количество элементов и многообразие связей между ними» есть «большое» И «возможность разрыва связей» есть «высокая» И «сложность системы» есть «сложная» И «территориальная распределенность» есть «глобальная», ТО «живучесть» есть «низкая»

(П₁: Если x_1 есть A_{13} И x_2 есть A_{23} И x_3 есть A_{31} И x_4 есть A_{43} И x_5 есть A_{53} то B_{11})

.

П₂₅: ЕСЛИ «дальность передачи между узлами» есть «низкая» И «количество элементов и многообразие связей между ними» есть «малое» И «разрыв связи» есть «низкая» И «сложность системы» есть «мало сложная» И «территориальная распределенность» есть «локальная», ТО «живучесть» есть «высокая»

(П₂₅: Если x_1 есть A_{11} И x_2 есть A_{21} И x_3 есть A_{31} И x_4 есть A_{41} И x_5 есть A_{51} то B_{33})

Число нечетких правил с конъюнктивной формой предпосылки, необходимых для покрытия пространства входных переменных, определяется формулой[48]:

$$n = \prod_{j=1}^m K_j$$

где m - размерность пространства входных признаков; K_j - число лингвистических термов j -ой входной переменной.

После определения нечетких правил производим свертку нечетких множеств, которая определяется формулой

$$\mu_{A_{i1} \times A_{i2}}(x_1, x_2) = T(\mu_{A_{i1}}(x_1), \mu_{A_{i2}}(x_2))$$

Анализ существующих T - и S - норм приведенных в таблице 2.2 пункта 2.3 показал, что для решения поставленной задачи целесообразно

использовать T -норму вида: $\frac{ab}{\gamma + (1 - \gamma)(a + b - ab)}$

3 этап. Определения входов ННПС включает следующие правила:

Π_1 : ЕСЛИ «количество входов ННПС» есть «небольшое», ТО «количество слоев ННПС» есть «небольшое»

(Π_1 : Если x_1 есть A_{11} то B_{11})

Π_2 : ЕСЛИ «количество входов ННПС» есть «среднее», ТО «количество слоев ННПС» есть «среднее»

(Π_2 : Если x_1 есть A_{12} то B_{12})

Π_3 : ЕСЛИ «количество входов ННПС» есть «большое», ТО «количество слоев ННПС» есть «большое»

(Π_3 : Если x_1 есть A_{13} то B_{13})

В представленных правилах используются следующие лингвистические переменные вида $\langle x_1, T_1, X_1, G_1, M_1 \rangle$, где

x_1 – «количество входов ННПС»; T_1 – {«небольшое», «среднее», «большое»}; X_1 – $[0, 100]$; G_1 – процедура образования новых термов с помощью различных связей и модификаторов; M_1 – процедура задания на $X_1 = [0, 100]$ нечетких подмножеств $A_{11} = \text{«небольшое»}$, $A_{12} = \text{«среднее»}$, $A_{13} = \text{«большое»}$.

y_1 – «количество слоев ННПС»; T_1 – {«небольшое», «среднее», «большое»}; X_1 – $[0, 100]$; G_1 – процедура образования новых термов с помощью различных связей и модификаторов; M_1 – процедура задания на $X_1 = [0, 100]$ нечетких подмножеств $B_{11} = \text{«небольшое»}$, $B_{12} = \text{«среднее»}$, $B_{13} = \text{«большое»}$.

4 этап. Выбор алгоритма обучения ННПС осуществляется с помощью правил:

Π_1 : ЕСЛИ «время обучения ННПС» есть «малое» И «допустимая ошибка обучения ННПС» есть «малая», ТО «алгоритм обучения ННПС» есть «алгоритм 1»

(Π_1 : Если x_1 есть A_{11} И x_2 есть A_{21} то B_{11})

Π_2 : ЕСЛИ «время обучения ННПС» есть «малое» И «допустимая ошибка обучения ННПС» есть «средняя», ТО «алгоритм обучения ННПС» есть «алгоритм 1»

(П₂: Если x_1 есть A_{11} И x_2 есть A_{22} то B_{11})

П₃: ЕСЛИ «время обучения ННПС» есть «малое» И «допустимая ошибка обучения ННПС» есть «большая», ТО «алгоритм обучения ННПС» есть «алгоритм 3»

(П₃: Если x_1 есть A_{11} И x_2 есть A_{23} то B_{13})

П₄: ЕСЛИ «время обучения ННПС» есть «среднее» И «допустимая ошибка обучения ННПС» есть «средняя», ТО «алгоритм обучения ННПС» есть «алгоритм 2»

(П₄: Если x_1 есть A_{12} И x_2 есть A_{22} то B_{12})

П₅: ЕСЛИ «время обучения ННПС» есть «высокое» И «допустимая ошибка обучения ННПС» есть «высокая», ТО «алгоритм обучения ННПС» есть «алгоритм 3»

(П₅: Если x_1 есть A_{13} И x_2 есть A_{23} то B_{13})

В представленных правилах используются следующие лингвистические переменные[48,78] вида $\langle x_1, T_1, X_1, G_1, M_1 \rangle$, где

x_1 – «время обучения ННПС»; T_1 – {«малое», «среднее», «большое»}; X_1 – [0, 100]; G_1 – процедура образования новых термов с помощью различных связок и модификаторов; M_1 – процедура задания на $X_1 = [0, 100]$ нечетких подмножеств $A_{11} =$ «малое», $A_{12} =$ «среднее», $A_{13} =$ «большое».

x_2 – «допустимая ошибка обучения ННПС»; T_1 – {«малая», «средняя», «большая»}; X_1 – [0, 100]; G_1 – процедура образования новых термов с помощью различных связок и модификаторов; M_1 – процедура задания на $X_1 = [0, 100]$ нечетких подмножеств $A_{21} =$ «малая», $A_{22} =$ «средняя», $A_{23} =$ «большая».

y_1 – «алгоритм обучения ННПС»; T_1 – {«алгоритм 1», «алгоритм 2», «алгоритм 3»}; X_1 – [0, 3]; G_1 – процедура образования новых термов с помощью различных связок и модификаторов; M_1 – процедура задания на $X_1 = [0, 3]$ нечетких подмножеств $B_{11} =$ «обучение алгоритмом 1», $B_{12} =$ «обучение алгоритмом 2», $B_{13} =$ «обучение алгоритмом 3».

Правила выбора алгоритма обучения ННПС имеют следующий вид:

Фаззификация определяется степенью истинности каждой предпосылки каждого правила для заданных входных переменных $\mu_{A_{ij}}(x'_j)$, ($i, j = 1, 2$).

Агрегирование результатов срабатывания предпосылок по каждому правилу и формирование уровней их пересечения осуществляется алгоритмом Мамдани[48].

Активизация заключений каждого из нечетких продукционных правил выполняется на операции *prod*-активизации:

$$\mu_{B'_1}(y) = \alpha_1 \mu_{B_1}(y),$$

$$\mu_{B'_2}(y) = \alpha_2 \mu_{B_2}(y).$$

Аккумуляция полученных заключений и результирующего нечеткого множества выполняется с использованием *max*-дизъюнкции:

$$\mu_{B'}(y) = \bigvee_{i=1}^n \mu_{B'_i}(y)$$

Приведение к четкости проводится на основе методов фаззификации.

Общая схема алгоритма обучения нечетких нейронных сетей с нечеткими входами и выходами и четкими весами представлен на рисунке 3.2.

В данных алгоритмах задаются нечеткие правила типа:

P_k : Если x_1 есть A_{k1} И ... x_j есть A_{kj} И ... И x_m есть A_{km} то y есть B_k ,

$$k=1, \dots, n,$$

где $A_{k1}, \dots, A_{km}, B_k$ – нечеткие числа.

Далее получаем обучающие образы:

$$\{(A_1, B_1), \dots, (A_n, B_n)\},$$

где $A_k = (A_{k1}, \dots, A_{km})$ обозначает антецедент, а B_k – консеквент k – го нечеткого правила. $A_k = (A_{k1}, \dots, A_{km})$

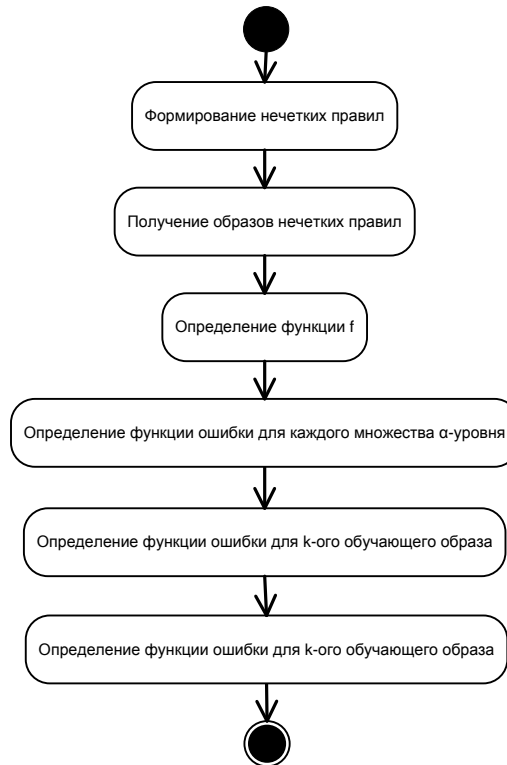


Рисунок 3.2 – Обучение ННС с помощью алгоритма обучения

АЛГОРИТМ 2 и АЛГОРИТМ 3

АЛГОРИТМ 2 реализуется следующим образом. Пусть $A_k = (A_{k1}, \dots, A_{km})$ – k -й обучающий образ. Выход i -го элемента скрытого слоя нечеткой нейронной сети вычисляется следующим образом:

$$o_{ki} = f \left(\sum_{j=1}^m w_{ij} \cdot A_{kj} \right).$$

Для выходного элемента

$$O_k = f \left(\sum_{i=1}^r v_i \cdot o_{ki} \right)$$

где $f(z) = 1/(1 + \exp(-z))$ - униполярная функция преобразования.

Отношение вход-выход каждого элемента такой нечеткой нейронной сети определяется на основе принципа обобщения.

Обозначим множества α -уровней вычисляемого выхода O_k следующим образом[48]:

$$[O_k]^\alpha = [O_k^L(\alpha), O_k^R(\alpha)], \alpha \in [0,1],$$

где $O_k^L(\alpha), O_k^R(\alpha)$ - соответственно значения левой и правой границы множества α -уровней вычисляемого выхода O_k .

Так как функция f является строго монотонно возрастающей, то справедливым является равенство:

$$[O_k]^\alpha = \left[f \left(\sum_{i=1}^r v_i o_{ki} \right) \right]^\alpha = \left[f \left(\sum_{i=1}^r [v_i o_{ki}]^L(\alpha) \right), f \left(\sum_{i=1}^r [v_i o_{ki}]^R(\alpha) \right) \right],$$

где

$$[O_k]^\alpha = \left[f \left(\sum_{j=1}^m w_{ij} A_{kj} \right) \right]^\alpha = \left[f \left(\sum_{j=1}^m [w_{ij} A_{kj}]^L(\alpha) \right), f \left(\sum_{j=1}^m [w_{ij} A_{kj}]^R(\alpha) \right) \right].$$

Множества α -уровней целевого выхода B_k обозначается как

$$[B_k]^\alpha = [B_k^L(\alpha), B_k^R(\alpha)], \alpha \in [0,1],$$

где $B_k^L(\alpha), B_k^R(\alpha)$ - соответственно значения левой и правой границ множества α -уровней целевого выхода B_k .

Минимизируемая целевая функция для каждого множества α -уровней определяется следующим образом[48,50]:

$$\varepsilon_k(\alpha) = \varepsilon_k^L(\alpha) + \varepsilon_k^R(\alpha),$$

где

$$\varepsilon_k^L(\alpha) = \frac{1}{2} \left(B_k^L(\alpha) - O_k^L(\alpha) \right)^2,$$

$$\varepsilon_k^R(\alpha) = \frac{1}{2} \left(B_k^R(\alpha) - O_k^R(\alpha) \right)^2,$$

где $\varepsilon_k^L(\alpha)$ - ошибка между левыми границами множеств α -уровней целевого и вычисленного значений выхода; $\varepsilon_k^R(\alpha)$ - ошибка между правыми границами множеств α -уровней целевого и вычисленного значений выхода.

Затем формируется функция ошибки для k -го обучающего образа:

$$\varepsilon_k = \sum_{\alpha} \alpha \varepsilon_k(\alpha).$$

Общая схема алгоритма обучения нечетких нейронных сетей с нечеткими входами, выходами и весами (АЛГОРИТМ 3), так же как и в предыдущем алгоритме и представлена на рисунке 3.1.

Задаются нечеткие правила. Из этих правил могут быть получены обучающие образы.

Выход i -го элемента скрытого слоя нечеткой нейронной сети вычисляется по формуле[48,50]:

$$o_{ki} = f \left(\sum_{j=1}^m W_{ij} A_{kj} \right).$$

Для выходного элемента

$$O_k = f \left(\sum_{i=1}^r V_i o_{ki} \right)$$

где $f(z)$ - униполярная передаточная функция вида $f(z) = 1/(1 + \exp(-z))$; W_{ij}, V_i - нечеткие веса элементов скрытого и выходного слоев соответственно; A_{kj} - нечетких вход.

Нечеткие выходы каждого элемента сети вычисляются в виде множеств α -уровней относительно нечетких входов и весов.

Обозначим множества α -уровней вычисляемого выхода O_k , целевого входа B_k , весов выходного элемента V_i и весов элементов скрытого слоя W_{ij} следующим образом:

$$[O_k]^\alpha = [O_k^L(\alpha), O_k^R(\alpha)], \quad [B_k]^\alpha = [B_k^L(\alpha), B_k^R(\alpha)],$$

$$[V_i]^\alpha = [V_i^L(\alpha), V_i^R(\alpha)], \quad [W_{ij}]^\alpha = [W_{ij}^L(\alpha), W_{ij}^R(\alpha)],$$

для $\alpha \in [0,1], i = 1, \dots, r, j = 1, \dots, m$.

Так как функция f является строго монотонно возрастающей, то справедливым является равенство:

$$[O_k]^\alpha = \left[f \left(\sum_{i=1}^r V_i o_{ki} \right) \right]^\alpha = \left[f \left(\sum_{i=1}^r [V_i o_{ki}]^L(\alpha) \right), f \left(\sum_{i=1}^r [V_i o_{ki}]^R(\alpha) \right) \right],$$

где

$$[o_k]^\alpha = \left[f \left(\sum_{j=1}^m W_{ij} A_{kj} \right) \right]^\alpha = \left[f \left(\sum_{j=1}^m [W_{ij} A_{kj}]^L(\alpha) \right), f \left(\sum_{j=1}^m [W_{ij} A_{kj}]^R(\alpha) \right) \right].$$

Минимизируемая целевая функция для каждого множества α -уровней определяется следующим образом:

$$\varepsilon_k(\alpha) = \varepsilon_k^L(\alpha) + \varepsilon_k^R(\alpha),$$

где

$$\varepsilon_k^L(\alpha) = \frac{1}{2} \left(B_k^L(\alpha) - O_k^L(\alpha) \right)^2,$$

$$\varepsilon_k^R(\alpha) = \frac{1}{2} \left(B_k^R(\alpha) - O_k^R(\alpha) \right)^2,$$

где $\varepsilon_k^L(\alpha)$ - ошибка между левыми границами множеств α -уровней целевого и вычисленного значений выхода; $\varepsilon_k^R(\alpha)$ - ошибка между правыми границами множеств α -уровней целевого и вычисленного значений выхода.

Затем формируется функция ошибки для k -го обучающего образа:

$$\varepsilon_k = \sum_{\alpha} \alpha \varepsilon_k(\alpha).$$

5 этап. Процесс создания ННПС является сложным, т.к. при проектировании структуры ННПС возникают трудности с определением процедуры выбора количества нейронов и слоев в сети. Если в ННПС большое количество нейронов и слоев, то сеть обучается и работает медленнее, чем небольшое количество и зависимость вход – выход может быть более нелинейной. Количество нейронов и количество скрытых слоев уточняется эмпирически в процессе программной разработки.

После определения структуры ННПС, выбора количества нейронов и слоев в сети выбирается алгоритм обучения ННПС.

3.2 Построение диаграммы вариантов использования моделей и алгоритмов оценки живучести информационных структур

Программную реализацию продукционной и нейросетевой модели сокращения времени расчета оценки живучести информационных структур [92,93] необходимо сначала смоделировать визуально, которое обеспечивает наглядное представление реализуемых модулей программного обеспечения, и позволяют представить и понять принципы работы разрабатываемой системы во всей ее полноте.

Язык проектирования и моделирования включает в себя [94,95,96]:

- элементы моделей, включающие в себя фундаментальную концепцию проектирования, моделирования и их семантику;
- нотацию, которая включает в себя визуальное изображение всех элементов моделирования;
- принципы использования, включающие в себя базу правил, которые применяются непосредственно к элементам моделирования.

С помощью диаграммы вариантов использования можно описывать функциональное назначение системы.

При моделировании диаграммы вариантов использования необходимо выполнить следующее:

- определение общих границ и содержание моделируемой и проектируемой предметной области;
- формирование общих требований к поведению моделируемой и проектируемой системы;
- разработка исходной концептуальной модели предметной области, которая в дальнейшем будет детализирована в форме логических и физических моделей;
- подготовка исходной документации информации, которая будет

взаимодействовать с разработчиками системы и ее пользователями.

Для построения разрабатываемого программного модуля, реализующего нечеткую продукционную модель оценки живучести структур необходимо представить разрабатываемый модуль в виде множества сущностей или актеров, которые взаимодействуют между собой с помощью вариантов использования (рисунок 3.3).

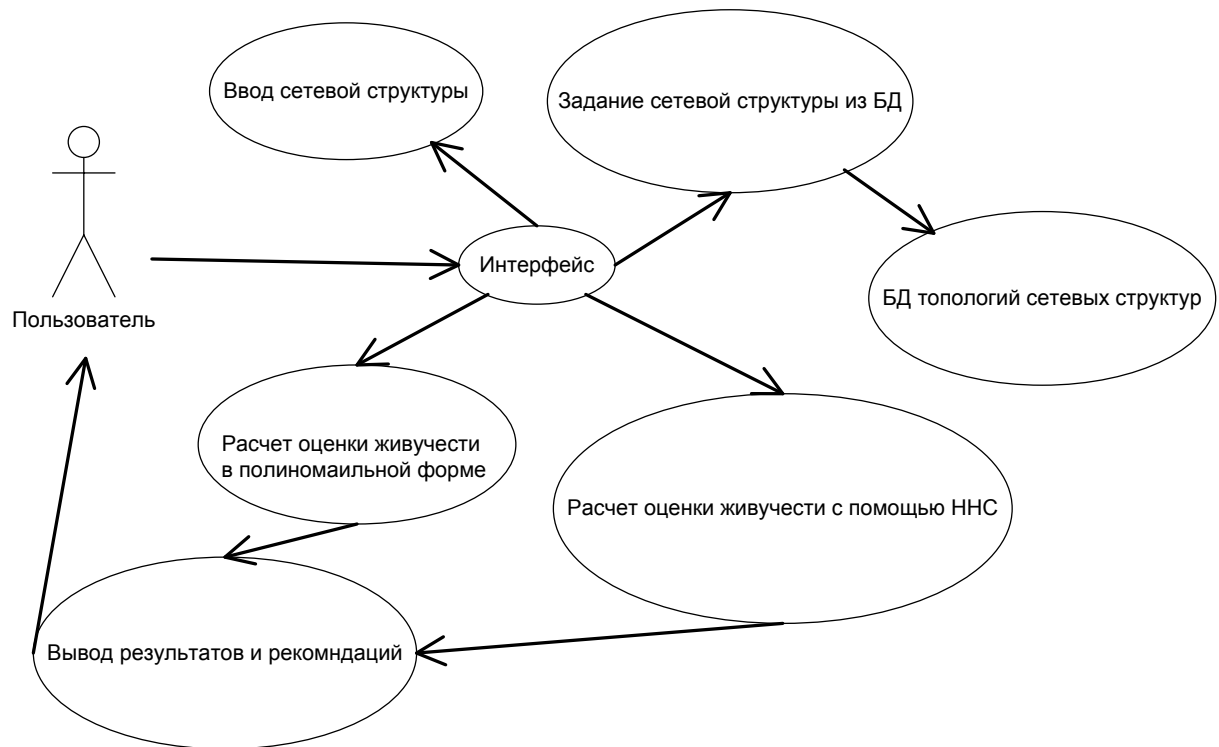


Рисунок 3.3 - Диаграмма прецедентов (вариантов использования) моделей и алгоритмов оценки живучести информационных структур

При взаимодействии с системой извне актером или действующим лицом может выступать как человек, так и программа, техническое устройство или любая другая система, которая служит источником воздействия на проектируемую систему и определяется самим разработчиком. Программные модули и сервисы, которые взаимодействуют между собой, описываются с помощью вариантов использования и предоставляются актеру. С помощью пояснительного текста, который раскрывает смысловое значение компонентов программного модуля, дополняется сама диаграмма вариантов использования[97,98].

3.3 Построение диаграммы классов моделей и алгоритмов оценки живучести информационных структур

При моделировании и проектировании разрабатываемой модели при помощи объектно-ориентированного программирования разработка логической модели программного модуля в виде диаграммы классов занимает одно из ключевых мест. С помощью диаграммы классов [95,96] можно представить статическую структуру проектируемой модели, которая отображает взаимодействие классов и объектов программного модуля и описывает внутреннюю структуру с типами отношений (рисунок 3.4)

Класс `fann_activation` содержит основные функции активации, выбираются в зависимости от требований к моделируемому процессу.

С помощью класса `fann_io` можно сохранить/загрузить созданную конфигурацию нечеткой нейронной сети.

Класс `fann_train` содержит способы обучения нейронных сетей.

Класс `fann_error` содержит объекты, функции и методы изменения ошибок при обучении, тестировании и работы нечеткой нейронной сети, которые сохраняются в файле.

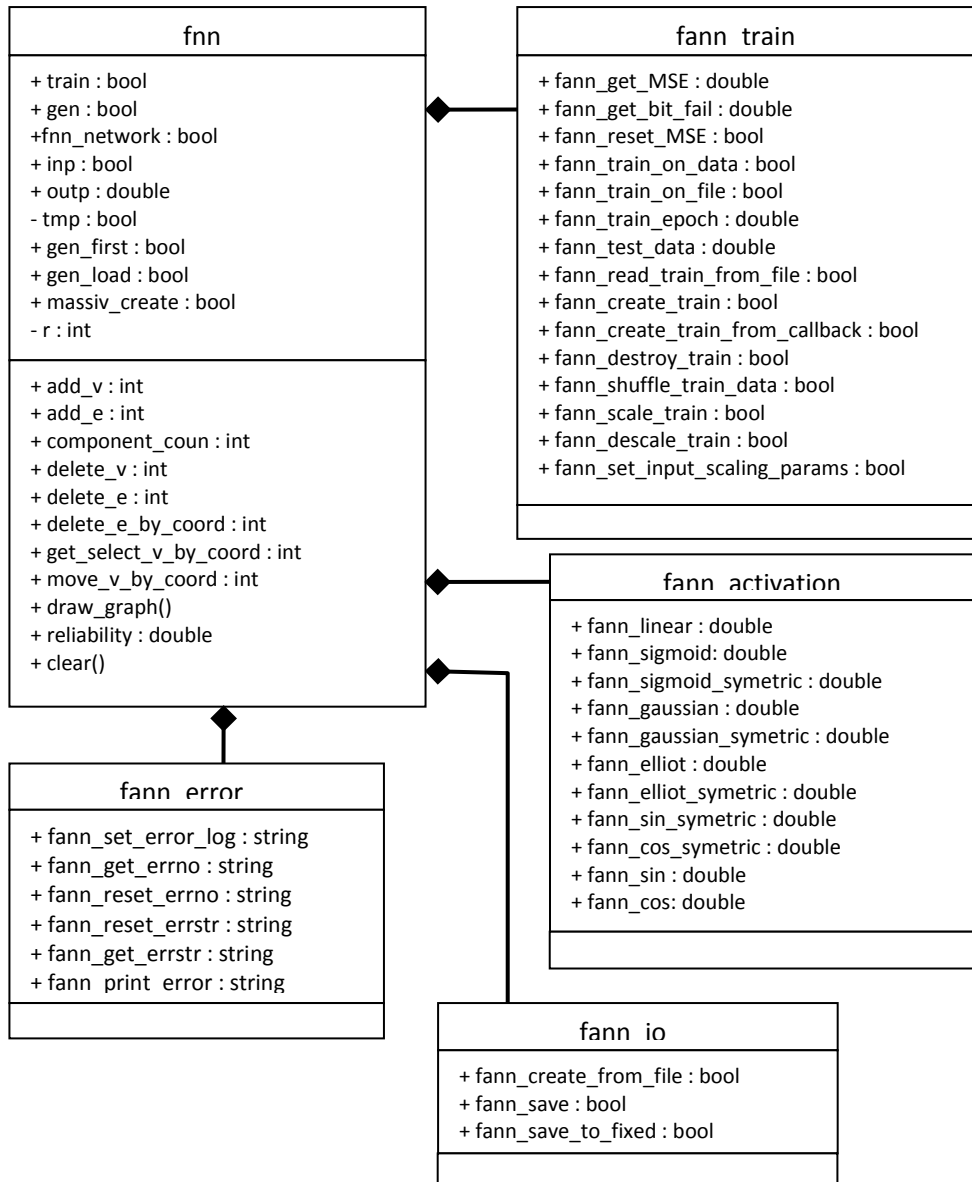


Рисунок 3.4 – Диаграмма классов программной реализации

3.4 Построение диаграммы деятельности моделей и алгоритмов оценки живучести информационных структур

При анализе и моделировании проектируемой нечеткой процедурной модели необходимо представить процесс изменения ее состояний и детализацию алгоритмических и логических особенностей выполняемых системой операций.

С помощью языка UML[95,96], в котором используется диаграмма деятельности, моделируется процесс выполнения. Графическая нотация

схожа с нотацией диаграммы состояний, поскольку на этих диаграммах имеются обозначения состояний и переходов системы. На диаграмме деятельности каждое состояние соответствует выполнению некоторой операции, а переход в следующее состояние выполняется только при завершении этой операции.

С помощью диаграммы деятельности можно представить процедурную и синхронную особенность управления, которая завершает внутреннюю деятельность и действия проектируемой модели[95].

Деятельность представляет собой совокупность отдельных вычислений, выполняемая автоматом, которая приводит к некоторому результату или действию. С помощью диаграммы деятельности можно представить всю логику и последовательность переходов, и состояние проектируемой системы.

Программное обеспечение, реализующий нечеткую процедурную и нейросетевую модель оценки живучести информационных структур[92], представим в виде диаграммы деятельности (рисунок 3.5).

- 1) Начало.
- 2) Ввод вершин и ребер графа G (графически), функцию принадлежности ребра p , количества обучающих топологий информационных структур (*count*), допустимой ошибки ННС (MSE).
- 3) Выбор способа создания массива (генерирование, загрузка из файла).
- 4) Генерирование обучающих топологий информационных структур
- 5) Загрузка из файла обучающих топологий информационных структур.
- 6) Определение количества входов ННС.
- 7) Создание ННС.
- 8) Выполнение одной эпохи обучения.
- 9) Получение ошибки ННС (MSE).
- 10) $MSE > Error$ и пользователь не прервал обучение?
- 11) Конец.

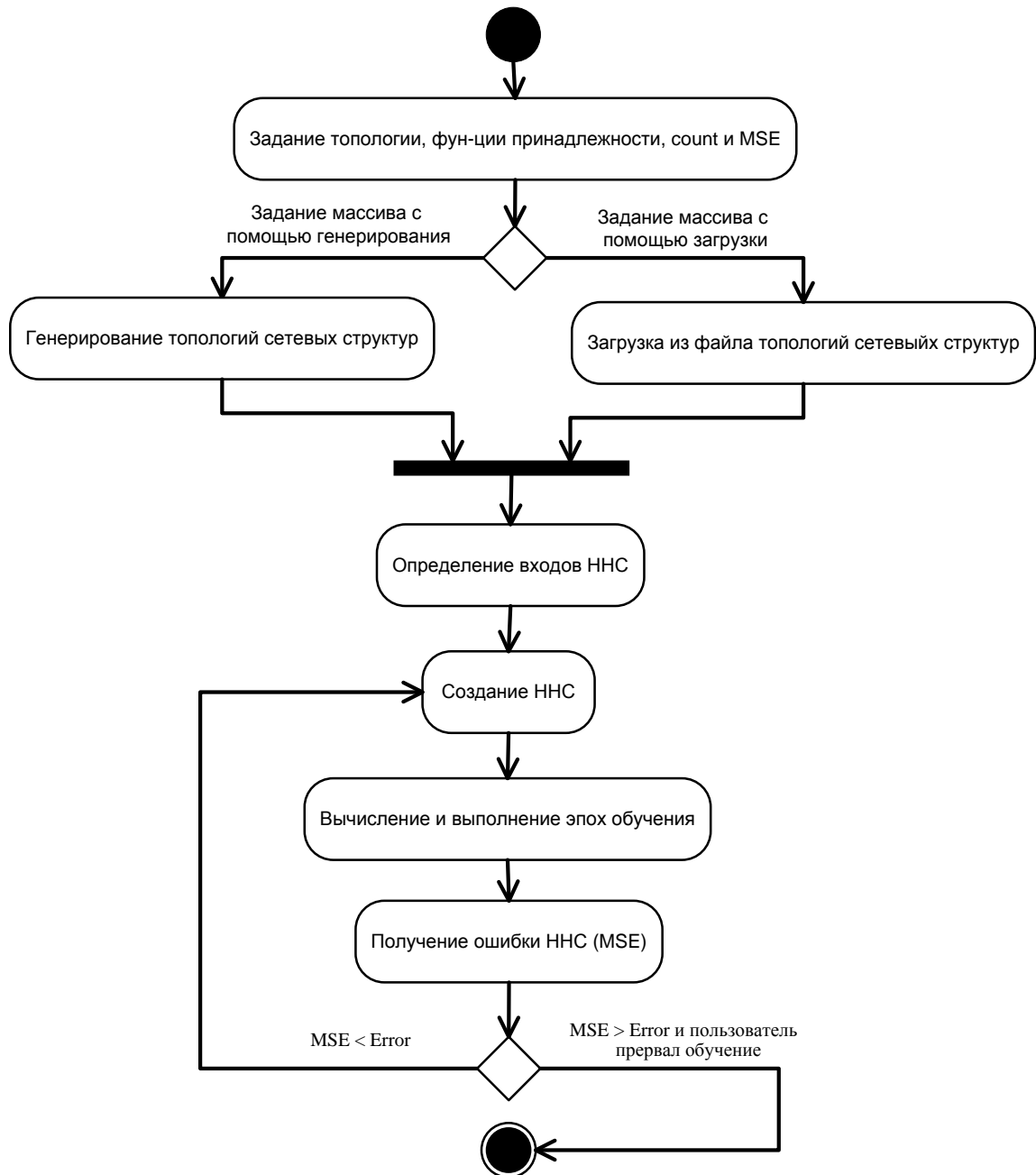


Рисунок 3.5 – Диаграмма деятельности программного обеспечения оценки живучести информационных структур

Процесс генерирования обучающих топологий представим в виде блок-схемы (рисунок 3.6).

- 1) Начало.
- 2) Существует ли массив для генерирования информационных структур?
- 3) Удаление массива.

- 4) Создание массива.
- 5) Существует ли выходной массив ННС?
- 6) Удаление массива.
- 7) Создание массива.
- 8) Определение максимально возможного количества ребер графа G ($\frac{N(N-1)}{2}$).
- 9) $i=1, \text{count}, 1$.
- 10) Генерирование случайной топологии информационной структуры (матрицы смежности графа G).[99,100]
- 11) Преобразование матрицы смежности во входной вектор $INP[i]$ сети ННС.
- 12) Расчет оценки живучести R сгенерированной топологии информационной структуры на основе полинома Татта.
- 13) Формирование выходного вектора $OUTP[i]=R$ сети ННС.
- 14) Конец.

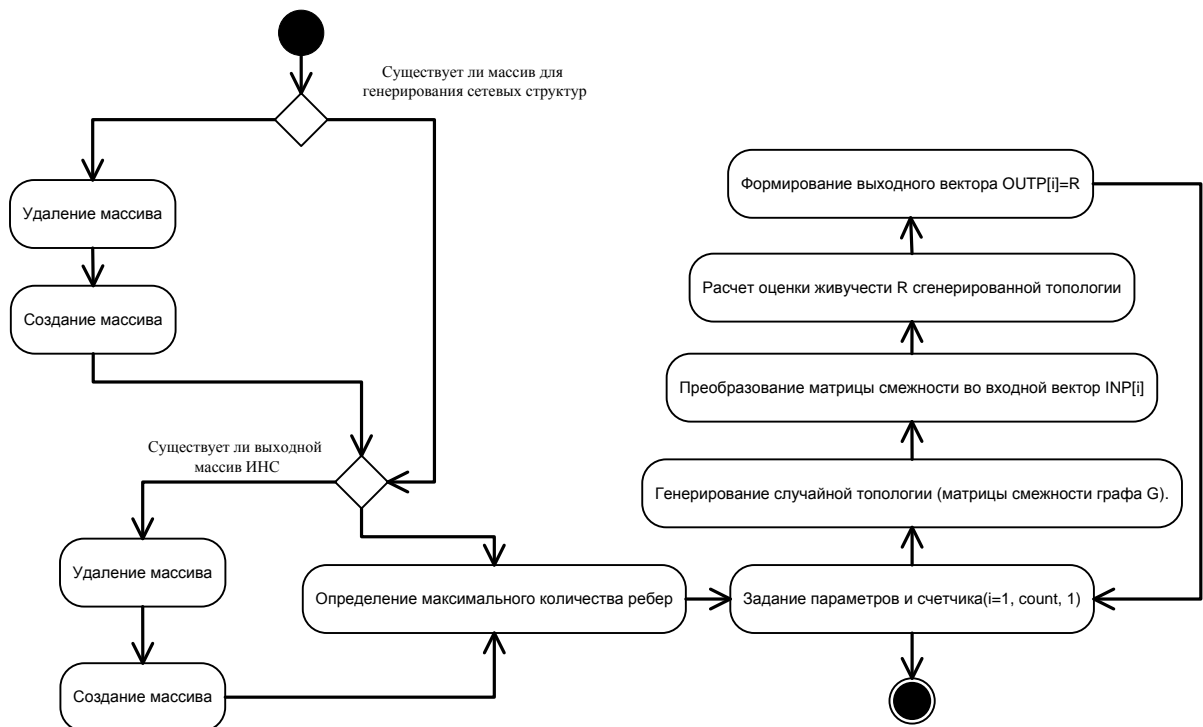


Рисунок 3.6 – Диаграмма деятельности генерирования обучающих топологий

Процесс тестирования обученной нечеткой нейронной сети представим в виде диаграммы деятельности (рисунок 3.7).

- 1) Начало.
- 2) Ввод вершин и ребер графа G (графически), разрыв ребра f
- 3) $i=1, 100, 1$.
- 4) Генерирование случайной топологии информационной структуры (матрицы смежности графа G).[99,100]
- 5) Преобразование матрицы смежности во входной вектор $INP[i]$ сети ННС.
- 6) Расчет оценки живучести R сгенерированной топологии информационной структуры на основе полинома Татта.
- 7) Оценка живучести информационной структуры с помощью ННС.
- 8) Отображение результатов оценки живучести в виде графика.
- 9) Конец.

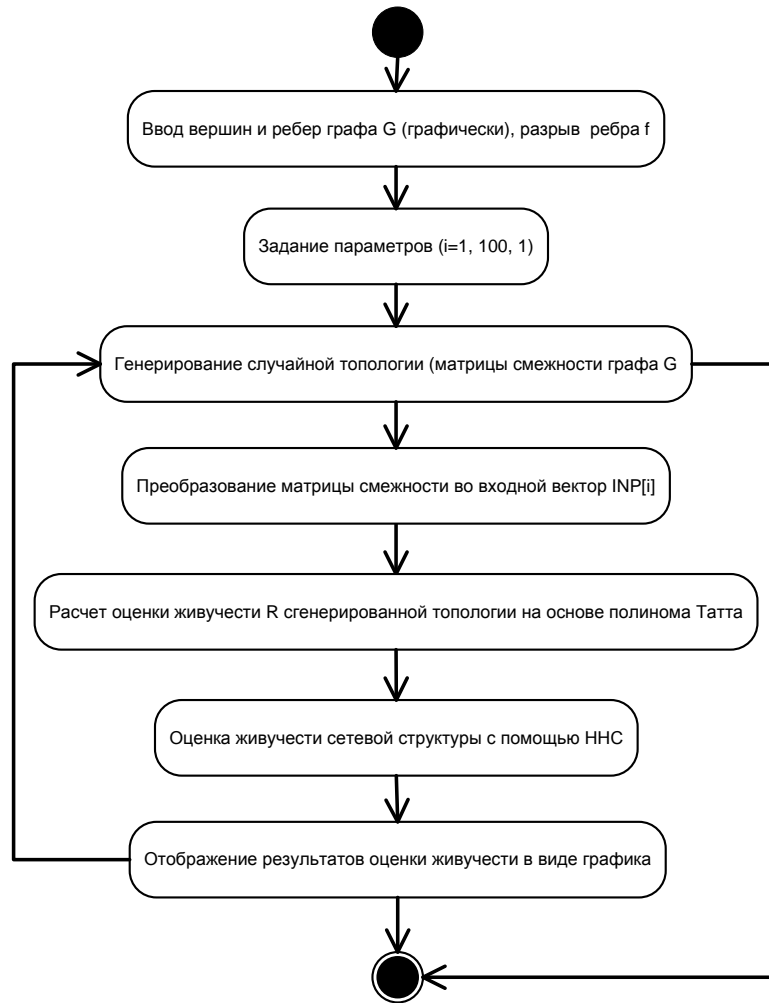


Рисунок 3.7 – Диаграмма деятельности тестирования обученной ННС

3.5 Алгоритм обратного распространения ошибки для нечеткой нейронной сети

Нечеткая нейронная сеть имеет многослойную структуру с прямым распространением сигнала, при этом выходы сети могут изменяться, внося изменения в параметры элементов слоев. Это позволяет использовать алгоритм обратного распространения ошибки для обучения сети.[48,49]

Данный алгоритм представлен в виде диаграммы деятельности на рисунке 3.8.

1. На первом этапе задаются начальные весовые коэффициенты.

Цель данного этапа заключается в нахождении более хорошего начального приближения таким образом, что бы сэкономить время обучения

и улучшить сходимость функции. Для всех весовых коэффициентов случайным образом выбираются малые значения, чтобы не было перенасыщения сигмоидальных элементов функции. Иногда такое приближение приводит к глобальному минимуму и уменьшит время сходимости.

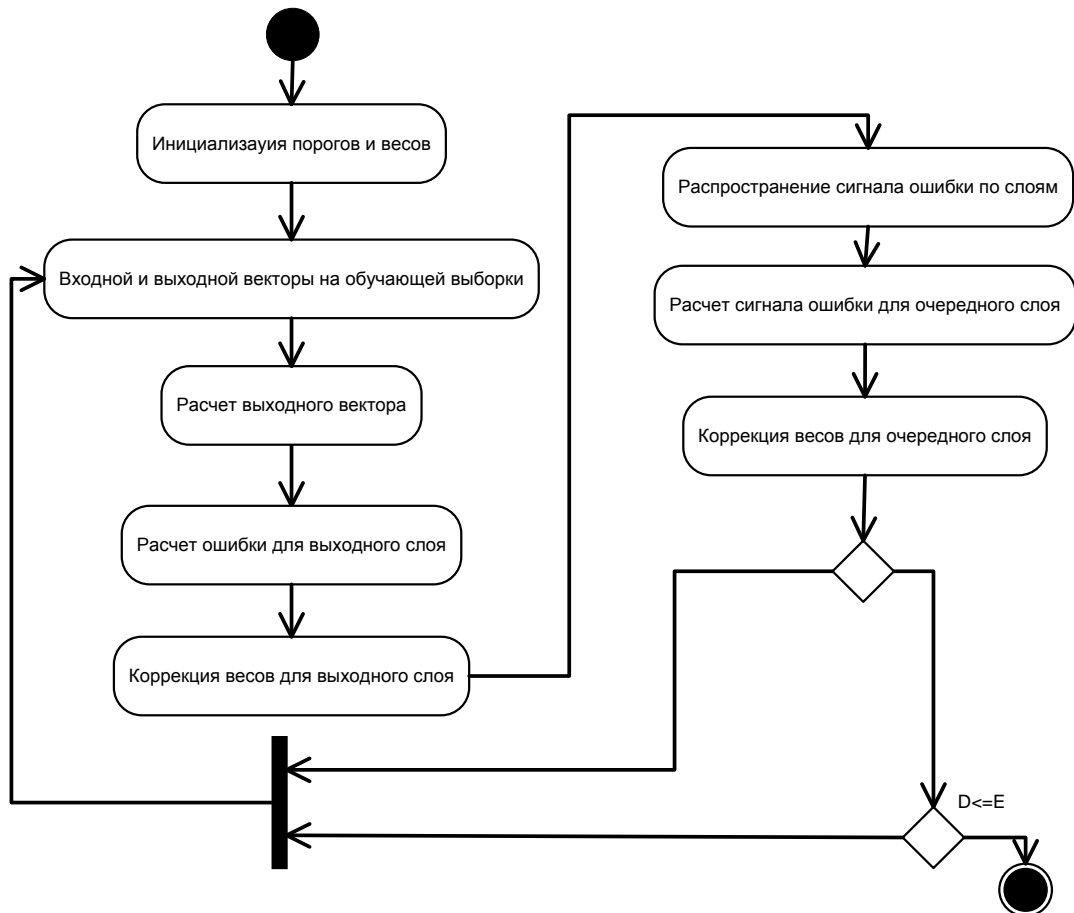


Рисунок 3.8 – Диаграмма деятельности алгоритм обратного распространения ошибки

2. На втором этапе происходит упорядочивание данных. При оценке живучести информационных структур, необходимо нужно перемешивать данные в случайном порядке, чтобы обучение двигалось в правильном направлении. Если этого не делать, то нечеткая нейронная сеть будет делать ошибку т.к. "выучит" последовательность случайно оказавшихся рядом значений как истинное правило.

3. На третьем этапе учитывается импульс. При изменении весовых

коэффициентов связей нейронов к текущему изменению веса прибавляется вектор смещения с предыдущего шага, взятый с некоторым коэффициентом, который задается пользователем в интервале $[0,1]$. В этом случае нужно учитывать предыдущий импульс. Изменение весовых коэффициентов связи можно представить следующим образом:

$$\Delta W_{ij}(t+1) = \mu * \Delta W_{ij}(t) - (1 - \mu)\epsilon \frac{\partial E}{W_{ij}}$$

4. На четвертом этапе происходит регулирование величины шага. При большом увеличении шага, происходит более грубое уменьшение суммарной ошибки нечеткой сети. Если шаг меньше, то требуется больше времени на обучение сети и становится более возможным попадание в окрестность локального минимума ошибки. Для улучшения сходимости нечеткой нейронной сети управление шагом имеет важное значение.

5. На пятом этапе происходит выбор оптимального количества скрытых слоев и нейронов в нечеткой нейронной сети. Выбор оптимального количества слоев и нейронов нечеткой нейронной сети весьма сложная задача, т.к. при недостаточном количестве нейронов произойдет потеря нелинейных связей и приведет к "переобучению" сети, которая "выучит" исходную информацию, а не распознает их структуру.

6. На шестом этапе производится масштабирование данных. Что бы улучшить работу нечеткой нейронной сети необходимо предварительно масштабировать данные обучающей выборки к интервалу $[0, 1]$, что приводит к наименьшим ошибкам при обучении и работе.

7. На седьмом этапе задается способ ускорения и обеспечения сходимости с помощью квазидискретного Ньютоновского алгоритма с методом немонотонной стабилизации.

8. На восьмом этапе происходит организация процесса обучения нечеткой нейронной сети. При обучении нечеткая нейронная сеть может подстраиваться под любые данные для минимизации суммарной

квадратичной ошибки. Чтобы этого не было используются следующий способ проверки нечеткой нейронной сети. Необходимо до начала обучения случайно разбить обучающую выборку на две подвыборки: обучающую и тестовую. Обучающая выборка используется в процессе обучения, в которой меняются весовые коэффициенты нейронов сети. В процессе обучения сети для проверки суммарной квадратичной ошибки используется тестовая подвыборка, но при этом не происходит изменение весовых коэффициентов. Если происходит улучшение аппроксимации как на обучающей, так и на тестовой выборках, то обучение нечеткой нейронной сети идет в правильном направлении. Если снижаться ошибка на обучающей подвыборке, а увеличивается на тестовой, то сеть "переобучилась" и не может быть использована в решении поставленной задачи. Что бы этого избежать необходимо изменение весовые коэффициенты нейронов, таким образом, нейронов, чтобы вывести нечеткую нейронную сеть из окрестности локального минимума ошибки.

Степень близости вектора-ответа сети y_i на i -м примере и соответствующего вектора указаний учителя \tilde{y}_i при текущем векторе весов нейронной сети $w \in R^W$, где W – количество весовых коэффициентов сети. Задача состоит в коррекции параметров сети, описанной выражением 2.5 чтобы мера погрешности, задаваемая выражением:

$$E = \frac{1}{2}(y(w) - d)^2, \quad (2.6)$$

была минимальной т.е. $E = \frac{1}{2}(y(w) - d)^2 \rightarrow \min$,

Для достижения высокой точности обучения нейронной сети в работах профессора Леденевой Т.М. [101] предлагается использовать алгоритмы Левенберга-Марквардта и Гаусса-Ньютона при этом отмечается их основной недостаток, который заключается, в том ,что данные алгоритмы требуют проведения сложных вычислений, занимающих много времени [101].

В связи с этим для решения оптимизационной задачи (2.6) с целью увеличения производительности и повышения эффективности обучения будем использовать предложенный квазидискретный Ньютонский алгоритм с методом немонотонной стабилизации [102], т.к. у этого метода наименьшее количество операций требуемых для нахождения экстремумов функции.

Анализ методов нахождения экстремумов функции представлен на рисунке 3.9.

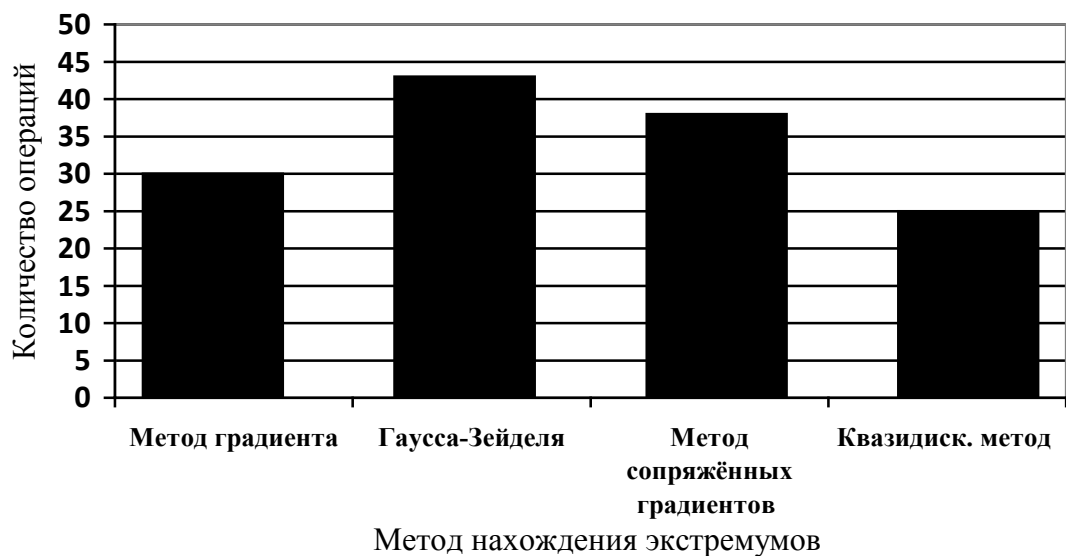


Рисунок 3.9 – Сравнительный анализ методов нахождения экстремумов функции.

Введем обозначение $f(x) \equiv E(w)$.

Для точки x_k выполняем проверку:

$$\|d_k\| \leq \alpha\beta^h \quad (3.1)$$

где $\alpha > 0$ и $\beta \in (0,1)$ заданные числа, h – целое число, увеличиваемое на единицу при выполнении условия. Если условие (3.1) выполняется, то $\alpha_k = 1$ и принимается временная точка $x_{k+1} = x_k + d_k$ без проверки целевой функции. А если условие (3.1) не выполняется, вычисляем f_k и определяем α_k путем выполнения немонотонного линейного поиска, где размер шага α_k –

выбирается так, чтобы $f(x_k + \alpha_k d_k)$ удовлетворяло условию (3.1) по отношению к максимальному значению функции из числа предыдущих контрольных точек.

$$f(x_k + \alpha_k d_k) < W_j + \gamma \alpha_k g_k^T d_k \quad (3.2)$$

где $\gamma \in (0,1)$ и

$$W_j = \max_{0 \leq i \leq m(j)} w_{j-1}$$

где $m(j)$ ограниченно заданным целым M

$$m(j) = \min[m(j-1) + 1, M] \quad (3.3)$$

Если x_k является контрольной точкой, вычисляем f_k и выполняем проверку $f_k < W_j$, где W_j значение целевой функции в предыдущей точке. Если проверка успешно проходит, сохраняем $w_j = f_k$, определяем x_k как стандартную точку и продолжаем проверять условие (3.2). Если проверка не проходит, происходит восстановление вектора переменных в предыдущей контрольной точке и выполняется алгоритм немонотонной стабилизации.

Этап 1. Установим $k = 0, j = 0, h = 0, l(0) = 0, m(0) = 0$. Вычислим $f(x_0)$ и установим $y_0 = x_0; w_0 = f(x_0), W_0 = w_0$.

Этап 2. Вычислим $g(x_k)$. Если $\|g(x_k)\| = 0$, то остановка вычислений. Если $k = l(j) + N$, переходим на Этап 4, иначе вычисляем направление d_k и, если $k = 0$, установим $v_0 = d_0$.

Этап 3. Если $\|d_k\| \leq \alpha \beta^h$, установим $x_{k+1} = x_k + d_k, k = k + 1, h = h + 1$ и переходим на Этап 2, иначе, если $k = l(j)$, переходим на Этап 5.

Этап 4. Вычислим $f(x_k)$. Если $f(x_k) \geq W_j$, установим $k = l(j)$ и заменяем x_k, d_k на y_j, v_j соответственно и переходим к Этапу 5, иначе берем x_k как контрольную точку путем установки $j = j + 1, w_j = f(x_k)$ и $y_j = x_k$, обновляя W_j и $m(j)$ соответственно с формулами (3.1 - 3.3). Если

$k \neq l(j) + N$, устанавливаем $l(j) = k$ и переходим к Этапу 5, иначе вычисляем d_k , устанавливаем $v_j = d_k$, и переходим к Этапу 3.

Этап 5. Устанавливаем $\alpha = 1$.

Этап 6. Вычислим $f_\alpha = f(x_k + \alpha d_k)$. Если $f_\alpha \leq W_j + \gamma \alpha g(x_k)^T d_k$, установим $\alpha_k = \alpha$, $x_{k+1} = x_k + \alpha_k d_k$, $k = k + 1$, $f(x_k) = f_\alpha$. Вычисляем $g(x_k)$; если $\|g(x_k)\| = 0$ остановка вычислений; иначе берем x_k как контрольную точку путем установки $j = j + 1$, $w_j = f(x_k)$ и $y_j = x_k$. Обновляем W_j и $m(j)$ согласно формулам (3.1 - 3.3); Вычисляем d_k , установив $v_j = d_k$, $l(j) = k$, и переходим на Этап 3.

Этап 7. Устанавливаем $\alpha = \sigma \alpha$ и переходим на Этап 6.

Блок схема квазидискретного Ньютоновского алгоритма с методом немонотонной стабилизации представлена на рисунке 3.10.

При этом подходе уточнение весов сети рассчитываются следующим образом:

$$w_{ij}(t+1) = w_{ij}(t) - \eta \cdot \frac{\partial E(t)}{\partial w_{ij}(t)} + \alpha (w_{ij}(t) - w_{ij}(t-1)) \quad (3.4)$$

где α - это коэффициент момента, принимающий значения в интервале $[0, 1]$.

Значение выражения (3.4) соответствует квазидискретному Ньютоновскому алгоритму и не зависит от фактического значения градиента и учитывает последнее изменение весов. Чем больше значение коэффициента α , тем большее значение оказывает показатель момента на подбор весов. При постоянном значении коэффициента обучения $\eta(t) = \eta$ приращение весов остается примерно одинаковым, то есть $\Delta w_{ij}(t) = \eta p(t) + \alpha \Delta w_{ij}(t)$, поэтому эффективное приращение весов можно писать формулой:

$$\Delta w_{ij}(t) = \frac{\eta}{1 - \alpha} p(t) \quad (3.5)$$

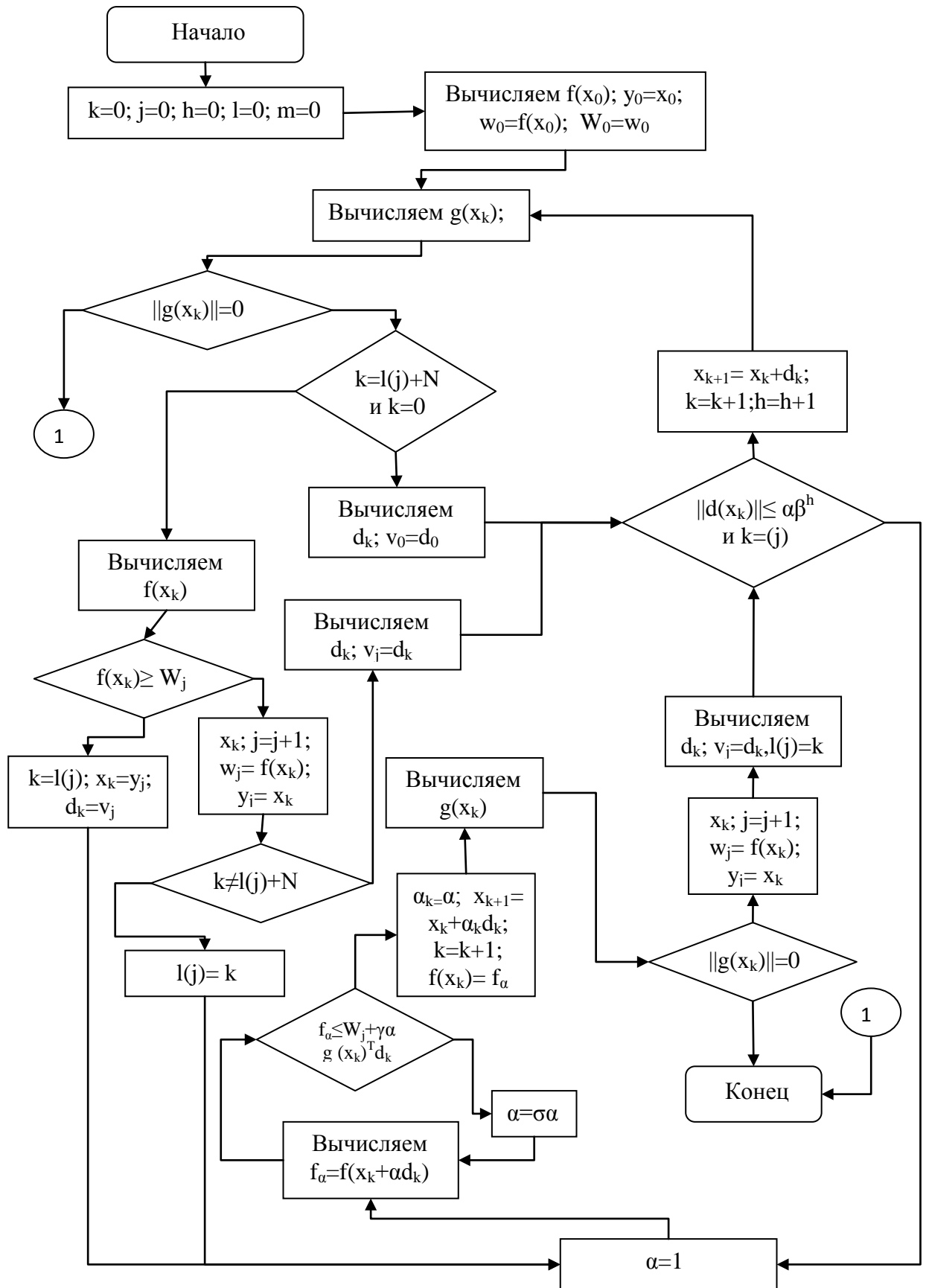


Рисунок 3.10 – Квазидискретный Ньютонский алгоритм с методом немонотонной стабилизации.

При значении $\alpha=0,9$ это соответствует десятикратному увеличению значения коэффициента обучения и, следовательно, десятикратному ускорению процесса обучения.

3.6 Метод Фибоначчи поиска экстремума

Алгоритм метода Фибоначчи [87,88,102] определяется следующим образом. На каждом $(i+1)$ -ом шаге $(i=3, \dots, N-2)$ точка очередного вычисления функции x_{i+1} выбирается симметрично относительно середины отрезка локализации $[a_i, b_i]$ к лежащей внутри этого отрезка точке уже проведенного вычисления \tilde{x}_i , где $f(\tilde{x}_i) = \min_{j=1, \dots, i} f(x_j)$.

После $(N-1)$ -го шага точка \tilde{x}_{N-1} оказывается в середине отрезка локализации и $x_N = \tilde{x}_{N-1} + \varepsilon, \varepsilon > 0$. Ясно, что гарантированное значение критерия эффективности (длина отрезка локализации точки минимума после N вычислений) в случае $D = [a, b]$ равно

$$Q(\alpha_3) = \sup_{f \in F_u} Q(f, \alpha_3) = \frac{b-a}{F_N} + \varepsilon.$$

Блок схема минимизации функции $f(\gamma)$ с использованием чисел Фибоначчи [87,88] представлен на рисунке 3.11.

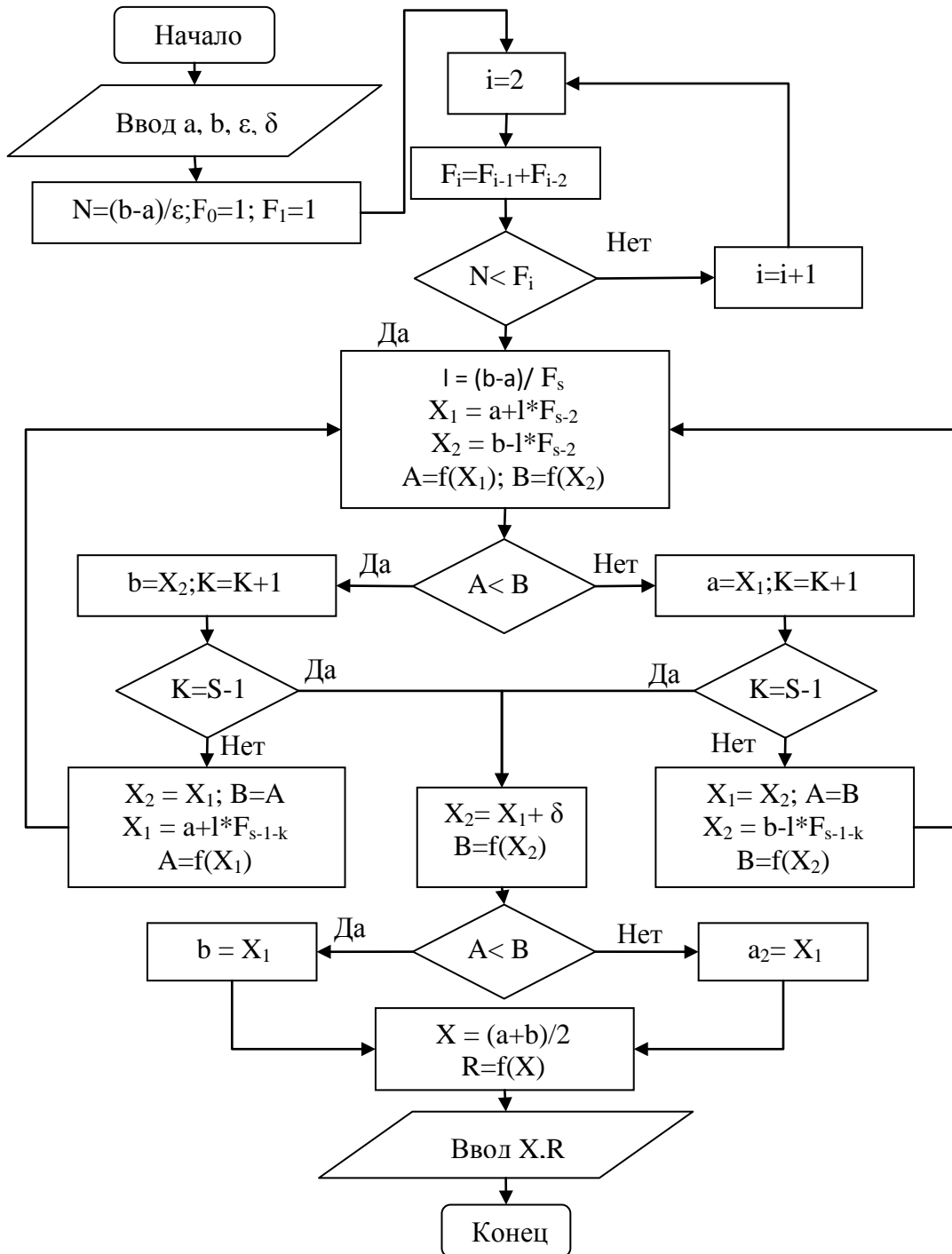


Рисунок 3.11 – Блок схема поиска минимума функции $f(x)$
использованием чисел Фибоначчи

3.7 Выводы по третьей главе

На основе нечеткой продукционной и нейросетевой модели сокращения времени расчета оценки живучести информационных структур, разработанной в главе 2, реализован алгоритм перехода от нечеткой продукционной модели к соответствующей ей нечеткой нейронной продукционной сети. Так же реализован алгоритм обучения нечеткой нейронной продукционной сети, который использует разработанный квазидискретный Ньютоновского метод с немонотонной стабилизацией и алгоритм генерирования обучающих структур нейросетевой модели.

В результате спроектированы диаграммы классов, прецедентов и деятельности программного обеспечения оценки живучести ИС, что позволяет приступить непосредственно к его программной реализации.

4 Программная реализация моделей и алгоритмов оценки живучести информационных структур

4.1 Анализ и выбор языка программирования

Для реализации алгоритма модуля оценки живучести возникает задача выбора языка программирования.

Рассмотрим рейтинг популярности и характеристики языков программирования за 2012-ый и 2013-ый год согласно проводимому рейтингу ТЮВЕ[103]. Специалисты ТЮВЕ определяют язык программирования как таковой, только если он является полным по Тьюрингу и измерения проводятся 10 раз в месяц.

Метод подсчета популярности языков программирования базируется на количестве запросов пользователей в наиболее популярных следующих поисковых системах и сайтах: Google, Blogger, Wikipedia, YouTube, Baidu, Yahoo, Bing, Amazon[103].

В таблице 4.1 представлены самые популярные языки программирования 2012-ый и 2013-ый год по рейтингу ТЮВЕ[103].

Таблица 4.1 – Рейтинг языков программирования согласно ТЮВЕ

№	Положение «Октябрь 2012»	Положение «Октябрь 2013»	Повышение (+) / понижение (-) положения	Язык программирования	Рейтинг, %	Прирост рейтинга, %	Статус
1	1	1	=	C	17.246	-2.58	A
2	2	2	=	Java	16.107	-1.09	A
3	3	3	=	Objective-C	8.992	-0.49	A
4	4	4	=	C++	8.664	-0.60	A
5	5	6	+	PHP	6.094	+0.43	A
6	6	5	-	C#	5.718	-0.81	A

Окончание таблицы 4.1

7	7	7	=	(Visual)Basic	4.819	-0.30	A
8	8	8	=	Python	3.107	-0.79	A
9	9	23	+++++	Transact-SQL	2.621	+2.13	A
10	10	11	+	JavaScript	2.038	+0.78	A
11	11	18	+++++	Visual Basic .NET	1.933	+1.33	A
12	12	9	---	Perl	1.607	-0.52	A
13	13	10	---	Ruby	1.246	-0.56	A
14	14	14	=	Pascal	0.753	-0.09	A
15	15	17	++	PL/SQL	0.730	+0.10	A
16	16	13	---	Lisp	0.725	-0.22	A
17	17	12	-----	Delphi/Object Pascal	0.701	-0.40	A
18	18	53	+++++	Groovy	0.658	+0.53	B
19	19	19	=	MATLAB	0.614	+0.02	B
20	20	26	+++++	COBOL	0.599	+0.15	B

Из представленных языков программирования выделим наиболее подходящие для реализации поставленной задачи: C, C++, C#, Delphi / Object Pascal.

Pascal является уникальным языком программирования, который имел самый быстрый компилятор в мире, да и в нем впервые была разработана технология создания платформенно независимых программ, которая нашла широкое применение в таком языке, как Java.

Delphi/Object Pascal является средой программирования, разработанный фирмой Borland, в которой используется язык программирования Object Pascal, являющиеся наследником языка Pascal, имеющие объектно-ориентированные расширения. Язык программирования

Pascal, является полностью процедурным языком. Расширения, внесенные в язык достигались за счет упрощения обработки структур, типов данных строки, файлов и реализации основных возможностей объектно-ориентированных языков программирования. После введения расширений необходимо было серьезно переработать и доработать синтаксис языка. В синтаксис добавили следующие изменения: ключевые слова, синтаксические конструкции и типы данных. Но все, же предложенный вариант языка программирования нельзя назвать удачным, т.к. в нем механизмы объектно-ориентированного программирования не полностью реализованы, а именно отсутствие таких мощных средств разработки, как исключения и шаблоны[104].

Следующий представитель объектно-ориентированного языка программирования – C#. Это полнофункциональный язык программирования, который включает в себя поддержку таких главных вещей как инкапсуляция, наследования и полиморфизм. Он имеет высокую надежность и устойчивость с большим набором благодаря обработки исключений и безопасности типов. При разработке язык C# унаследовал множество полезных свойств из следующих языков программирования: примерно 75% из Java, около 10% из C++, 5% из Visual Basic, Delphi, Pascal. Язык C# разработан исключительно для технологий Интернет и ASP.NET, как и Java.

Си является универсальным языком программирования, который наделен современными механизмами управления с компактным представлением записей, структурными данными и большим количеством операторов. Благодаря своей универсальности и широкими возможностями Си намного эффективнее, чем языки более высокого уровня для решения многих задач. Одно из главных преимуществ языка Си связано с тем что он разработан в UNIX системе и тесно связан с ней, т.к большинство программного обеспечения в этой операционной системе написано именно на нем, но он не к какой-то одной ЭВМ или операционной системе. Хотя он и

назван "языком системного программирования", поскольку удобен для написания компиляторов и операционных систем, оказалось, что на нем столь же хорошо писать большие программы другого профиля. Язык Си можно назвать языком "низкого уровня", потому что он работает с теми же объектами, классами, символами, числами и адресами что и большинство компьютеров.

Язык C++ один из ярких представителей объектно-ориентированного программирования высокого уровня, который является компилируемым, статическим и типизированным языком программирования общего значения. Язык C++ оснащен всеми парадигмами программирования, поддерживающий процедурное программирование и объектно-ориентированное программирование, которое включает в себя обобщенное программирование, обеспечение модульности, содержит виртуальные функции, отдельную компиляцию, обработку исключений, абстракцию данных, объявление типов и классов объектов[105,108].

Одно из главных преимуществ языка программирования C++ перед остальными языками заключается в адекватном отображении объектов реального мира и способности поддерживать эволюционное развитие программного обеспечения, которое достигается за счет сочетания наследования и виртуализации. Благодаря графическому интерфейсу снижается трудоемкость разработки программного модуля, что особенно важно при создании и проектировании нечетких нейронных сетей[106,107].

Язык программирования C++ содержит в себе весь инструментарий для создания эффективных приложений любого назначения, включая низкоуровневые утилиты и драйверы, сложные программные комплексы самого различного назначения, которые включают[105,108]:

– поддержку различных стилей и технологий программирования, которые включают в себя объектно-ориентированное программирование, традиционное директивное программирование, обобщенное

программирование, программирование с использованием шаблонов, макросов;

- предсказуемое выполнение программных модулей и построение систем в режиме реального времени, которое является важным достоинством. Измерение и расчет времени реакции программного обеспечения на внешнее событие строго определено в местах программы, в которых этот код выполняется.

- вызов конструктора и автоматический вызов деструктора объектов при их удалении, что значительно упрощает объявление переменных и делает более надёжным освобождение ресурсов таких как память, семафоры, файлы, а также позволяет выполнять переходы состояний программного обеспечения, которые не связаны с освобождением ресурсов таких как запись в журнал;

- пользовательские функции-операторы, позволяющие коротко и ёмко записывать выражения над пользовательскими типами данных в естественной алгебраической форме;

- поддержку физической (const) и логической (mutable) константы. С помощью этого достоинства программное обеспечение становится более надёжным, что позволяет компилятору диагностировать ошибочные изменения значения переменных.

- использование шаблонов, с помощью которых можно создавать обобщённые контейнеры и определенные алгоритмы для разных типов данных, которые на этапе компиляции программного обеспечения позволяют специализировать и вычислять;

- имитацию расширения языка программирования для поддержки парадигм, которые не поддерживаются компиляторами напрямую – это позволяет расширить кругозор решаемых задач.

- использование шаблонов и большое множество наследования элементов, которые имитируют классы-примеси и комбинаторную

параметризацию библиотек.

- использование кроссплатформенности языка программирования, которое накладывается с минимальными требованиями запуска скомпилированных программных модулей на ЭВМ.

- эффективность языка программирования дает максимальный контроль над всеми структурами и порядком исполнения программного модуля.

- возможность языка программировать на низком уровне с адресами и памятью.

- высокую совместимость с языком Си, что позволяет использование всего существующего кода Си с минимальными изменениями и доработками под компилятор C++. Также позволяет использовать библиотеки, которые написаны на Си, вызываемые из C++ непосредственно без каких-либо трудностей дополнительных затрат.

Анализ языков программирования показал, что целесообразно использовать для реализации программного модуля оценки живучести информационных структур на основе нечеткой нейронной продукционной сети язык высокого уровня C++[109,110].

4.2 Выбор средства разработки программной реализации моделей и алгоритмов оценки живучести информационных структур

В качестве средства программной реализации моделей и алгоритмов сокращения времени расчета оценки живучести информационных структур сети была выбрана библиотека Qt 4[111,112]. Qt является кросс платформенным инструментарием разработки программного обеспечения, которые использует мощнейший язык программирования C++. Написанное на Qt программное обеспечение выполняется и запускается на большинстве современных операционных систем, для которых необходимо просто

откомпилировать исходный программный код без его изменения. При разработке программного обеспечения в Qt есть все необходимые основные элементы: классы, классы для работы с сетью, элементы графического интерфейса, классы с базами данных и XML, которые требуются при разработке данного программного модуля.

Qt является полностью объектно-ориентированной библиотекой, в которую введены меж-объектные коммуникации, называемые "сигналами и слотами", заменяющую модель обратную модель вызовов, которая не надежна.

В Qt введена возможность систем расширений (plug-ins), позволяющая создавать модули с расширенными функциональными возможностями, созданных приложений.

Библиотека Qt, созданная для языка программирования C++, может работать с другими языками программирования в которых предусмотрена библиотека, а это Qt# в C#, PyQt в Python Qt Jambi в Java, PerlQt в Perl и PHP[113,114].

Программы, которые написаны с помощью библиотеки Qt, могут использовать язык сценариев, который называется Qt Script. С помощью этой технологии пользователи вашего приложения могут расширять возможности вашего приложения без перекомпиляции и изменений в исходном коде.

Для библиотеки Qt существует множество руководств и документов с помощью их посмотреть и изучить информацию по интересующей библиотеке, но в случае недостаточности информации можно открыть и посмотреть более детально все принципы работы библиотеки, как в целом так отдельных функций т.к. Qt является библиотекой с открытым исходным кодом (Open Source).

4.3 Программная реализация моделей и алгоритмов оценки живучести информационных структур

Общая структура программной реализации моделей и алгоритмов оценки живучести информационных структур представлена на рисунке 4.1.

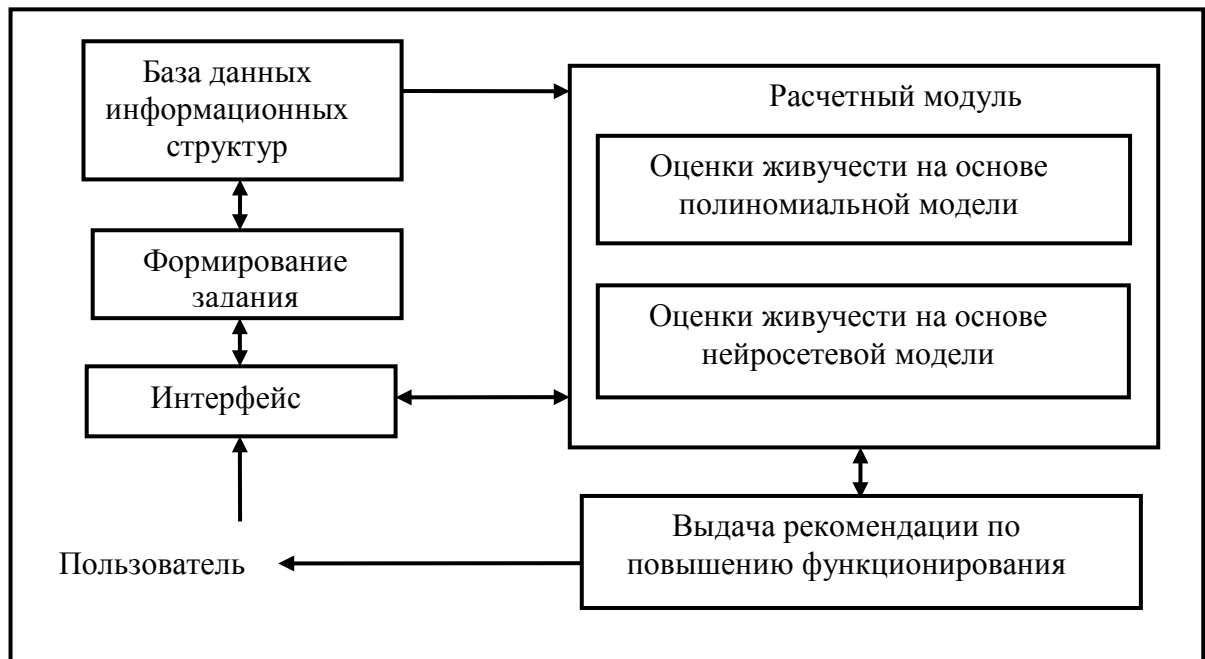


Рисунок 4.1 – Общая структура программной реализации моделей и алгоритмов оценки живучести информационных структур

Структуру исходных файлов разрабатываемого моделей и алгоритмов живучести на основе полиномиальной модели представим следующим образом:

1. ReliabilityCounter.pro – файл проекта. Используется в качестве файла описания проекта.

Заголовочные файлы:

2. mainwindow.h – основной заголовочный файл.

Исходные файлы:

3. main.cpp – файл главного окна.

4. `mainwindow.cpp` – основной исходный файл.

5. `mainwindow.ui` - файл интерфейса программы.

В файле `mainwindow.h` описаны структуры и методы двух основных классов: `Combinations` и `Graph`.

Класс «`Combinations`» служит для реализации алгоритма сочетаний без повторений, необходимого для обхода всех возможных остовных подграфов графа `G`.

Класс «`Graph`» – главный, включающий различные методы для наглядной работы с графами (рисование графа мышкой, в том числе добавление и удаление ребер/вершин, визуальное перемещение вершин без нарушения целостности графа; синхронное создание матрицы смежности для создаваемого графа; поиск матрицы связности с помощью алгоритма Уоршолла[115]; вычисление компонент связности; вычисление полинома Татта и оценки живучести для текущего заданного графа; сохранение и загрузка графа в файл/из файла).

Интерфейс программного обеспечения оценки живучести информационных структур представлен на рисунке 4.2. Компиляция произведена под операционной системой Ubuntu 11.10[116].

Информационная структура «рисует» в виде графа с помощью манипулятора мыши.

Теперь разработаем подмодуль оценки живучести на основе нейросетевых моделей.

Структура исходных файлов состоит из:

1. `ReliabilityCounter.pro` – файл проекта. Используется в качестве файла описания проекта.

2. Заголовочные файлы:

`mainwindow.h` – основной заголовочный файл.

`floatfann.h`, `fann_train.h`, `fann_io.h`, `fann_internal.h`, `fann_error.h`, `fann_data.h`, `fann_cascade.h`, `fann_activation.h`, `fann.h`, `compat_time.h` – заголовочные файлы библиотеки FANN.

3. Исходные файлы:

main.cpp – файл главного окна.

mainwindow.cpp – основной исходный файл.

fann_train_data.c, fann_train.c, fann_io.c, fann_error.c, fann_cascade.c, fann.c – исходные файлы библиотеки FANN.

4. mainwindow.ui - файл интерфейса программы.

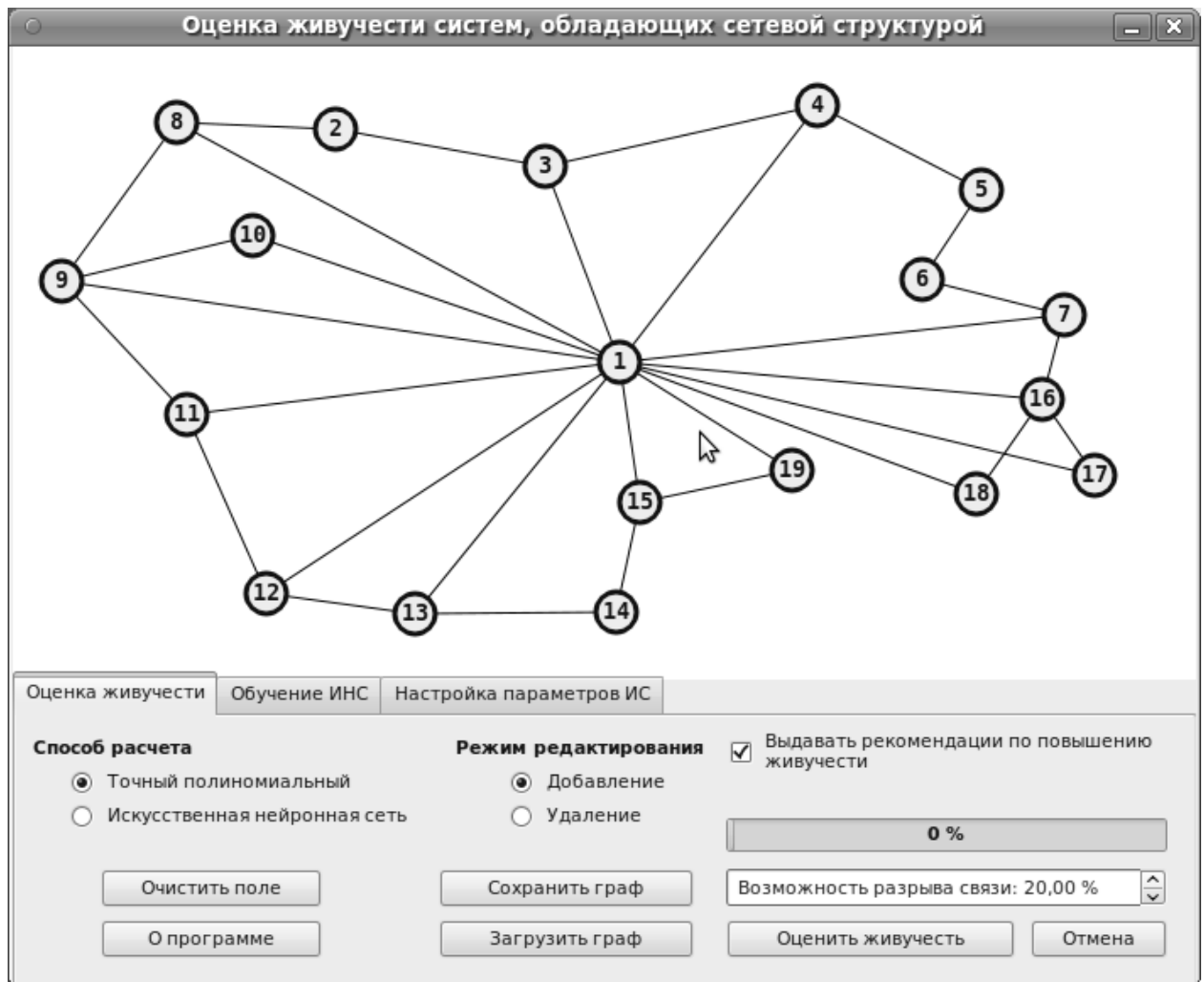


Рисунок 4.2— Интерфейс программного обеспечения оценки живучести систем, обладающих информационной структурой

В файле mainwindow.h описаны структуры и методы трех основных классов: Combinations, Graph и ANN.

Классы «Combinations» и «Graph» были рассмотрены ранее.

Класс «FNN» –полносвязная ННС с прямым распространением

сигнала, созданная с помощью библиотеки FANN (содержит входные, выходные, обучающие данные; архитектуру сети; методы загрузки, сохранения, экспорта в MATLAB обучающих данных[117]; методы оценки живучести на основе переданного входного графа G). Данная библиотека разработана для многих языков программирования, таких как C, C++, PERL, PHP, Java, Python, Pascal[104,105,111,113].

Библиотека FANN работает со следующими алгоритмами обучения:

- FANN_TRAIN_INCREMENTAL (стандартный алгоритм обратного распространения, где веса обновляются после каждой итерации);
- FANN_TRAIN_BATCH (стандартный алгоритм обратного распространения, где веса обновляются после вычисления средней квадратической ошибки для всего комплекса подготовки);
- FANN_TRAIN_RPROP (пороговый алгоритм обратного распространения ошибки, не использующий скорость обучения);
- FANN_TRAIN_QUICKPROP (пороговый алгоритм обратного распространения ошибки, использующий скорость обучения).

Нами был выбран алгоритм обучения сети FANN_TRAIN_RPROP.

Программное обеспечение оценки живучести информационных структур состоит из трех вкладок: «Оценка живучести», «Обучение ИНС» и «Настройка параметров ИС».

На вкладке «Оценка живучести» (рисунок 4.3) можно задать способ расчета оценки живучести ИС, саму информационную структуру с помощью манипулятора мыши, а так выдачу рекомендации.

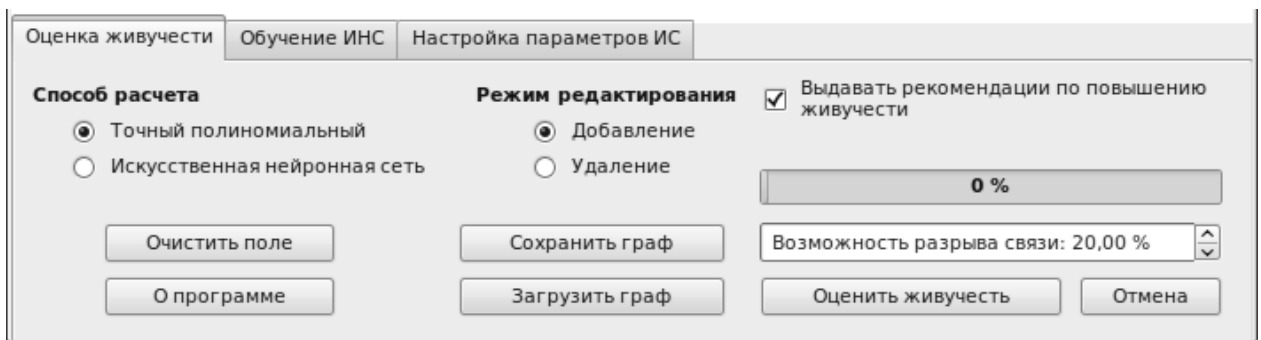


Рисунок 4.3 — Внешний вид вкладки «Оценка живучести»

На вкладке «Настройка параметров ИС» (рисунок 4.4) задаются параметры информационной структуры, такие как территориальная распределенность, количество элементов и связей между ними, дальность передачи между узлами, возможность разрыва связи и сложность системы.

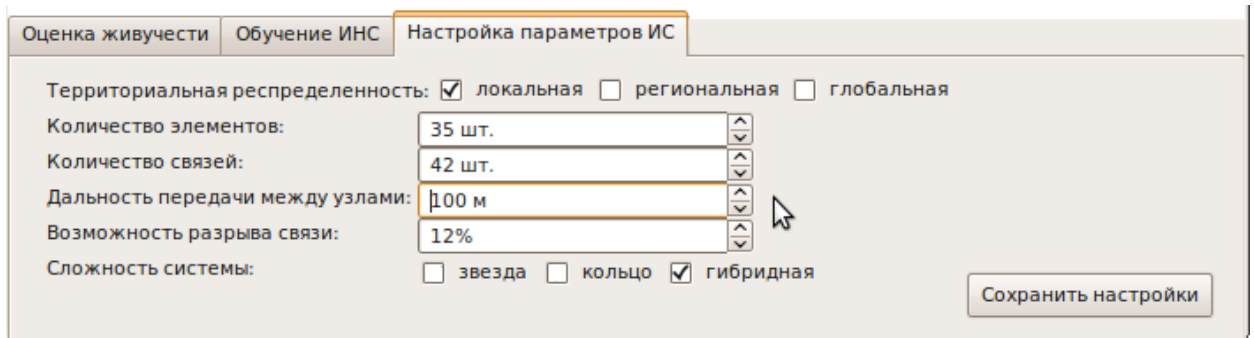


Рисунок 4.4 — Внешний вид вкладки «Настройка параметров ИС»

Внешний вид блока модуля для обучения искусственной нейронной сети (ИНС) представлен на рисунке 4.5, в котором генерируются структуры, задается точность обучения, тестируется обученная нейронная сеть а также сохранение/загрузка структур и нейронной сети.

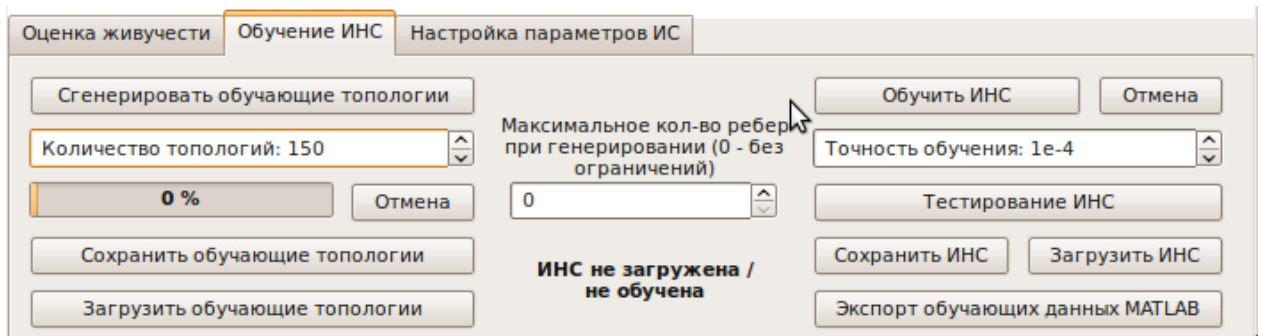


Рисунок 4.5 — Внешний вид вкладки «Обучение ИНС»

Обучение ИНС происходит следующим образом:

- манипулятором мыши задаются вершины топологии (здесь главное их количество, а не связи);
- задается количество топологий для обучения (по умолчанию 150);
- с помощью кнопки «Сгенерировать обучающие топологии» запускается процесс генерации, на каждом шаге производится точный полиномиальный расчет оценки живучести для сгенерированной топологии;

– после процесса генерации топологий с помощью кнопки «Обучить ННС» запускается процесс обучения ННС на основе сгенерированных топологий и рассчитанной оценки живучести для каждой топологии.

После обучения ННС с ее помощью рассчитывается оценка живучести в блоке модуля «Оценка живучести».

4.4 Тестирование программного обеспечения оценки живучести информационных структур

Далее проведем тестирование и апробацию программного обеспечения модуля оценки живучести.

Существует множество хозяйственных объектов, обладающих информационной структурой (сети объектов коммунального хозяйства, телекоммуникационные сети, транспортные сети и др.). Для тестирования разработанного программного комплекса оценки живучести, реализующего продукционную модель, произведем оценку живучести выделенной области информационной структуры ОАО «Ростелеком» (рисунок 4.6) и проведем сравнительный анализ с другими методами. Фрагменты информационной структуры ОАО «Ростелеком» представлены в Приложении В.

Выделим для тестирования фрагмент сети, т.к. он включает в себя наиболее проблемные районы Тамбовской области с точки зрения качества связи.

Граф, соответствующий данной информационной структуре (рисунок 4.7), состоит из 7 узлов и 9 ребер.

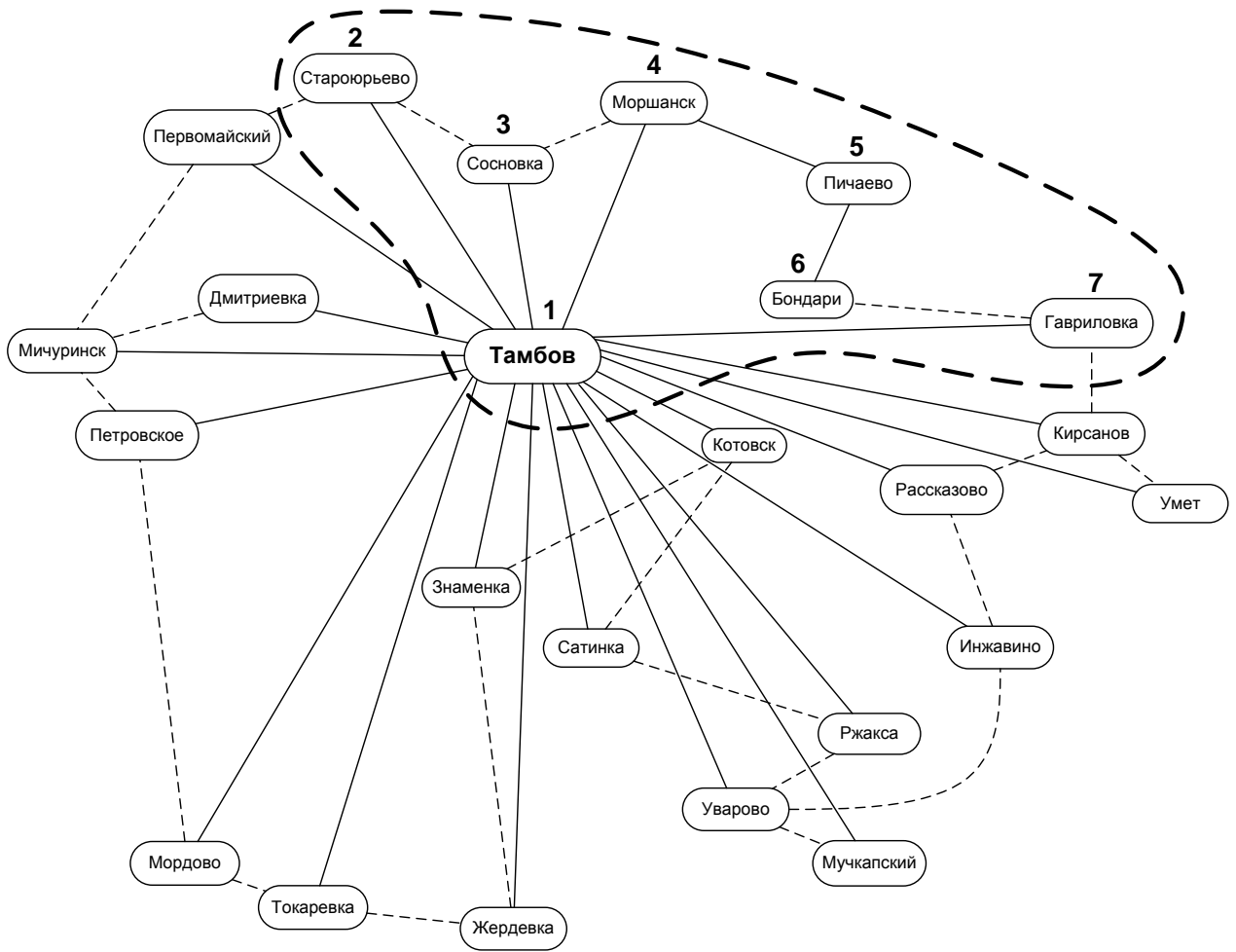


Рисунок 4.6 — Информационная структура Тамбовской области ОАО «Ростелеком»

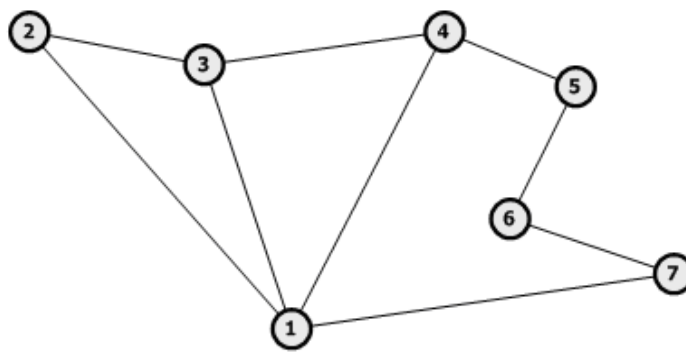


Рисунок 4.7 — Граф, соответствующий информационной структуре Тамбовской области ОАО «Ростелеком»

В технической документации ОАО «Ростелеком» задан порог живучести сети, равный 75%. Произведем расчет оценки живучести для функции принадлежности ребра 50%. Для данных условий оценка живучести исходной информационной структуры, полученная с помощью разработанного программного модуля, составляет всего 14,84%. Для повышения живучести добавим несколько связей в исходную топологию. Одна из новых возможных топологий сети (рисунок 4.8, а) удовлетворяет заданным условиям.

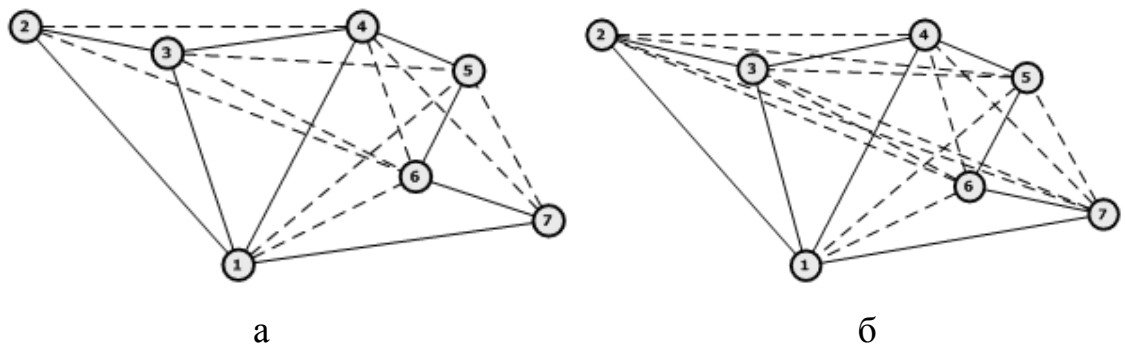


Рисунок 4.8 — Графы новой (а) и полносвязной (б) информационной структуры

Оценка живучести данной сети – 77,27%. Мы добавили 9 связей в исходную топологию. Результаты оценки живучести для возможных значений функции принадлежности представлены на рисунке 4.9 (б).

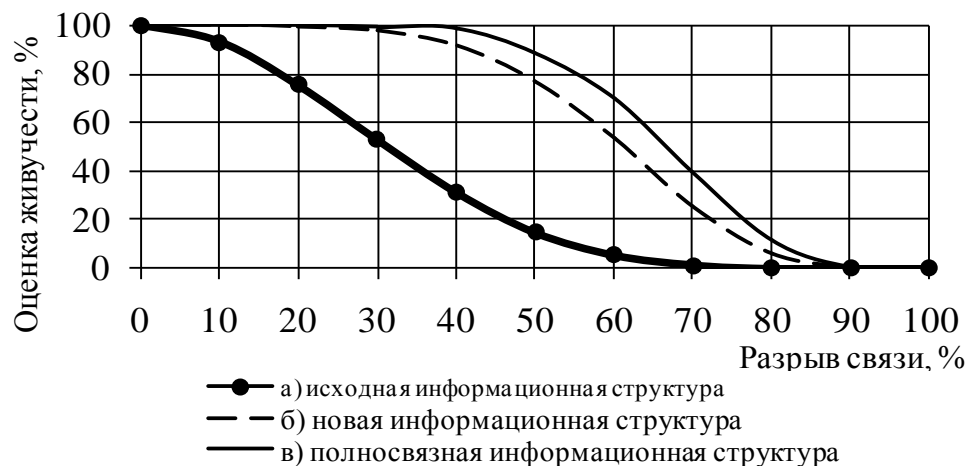


Рисунок 4.9 — Оценка живучести для возможных значений функции принадлежности ребра графа

Очевиден следующий факт, что оценка живучести является наибольшей для полносвязной информационной структуры (рисунок 4.9, б). Если связать каждый узел с каждым (на том же примере сети ОАО «Ростелеком»), то уже требуется 2097152 итерации для расчета. Из графика (рисунок 4.9, в) видно, что оценка живучести повысилась, но цена такой информационной структуры будет, скорее всего, неоправданно высокой. Поэтому наша задача сводится к тому, чтобы максимизировать живучесть и минимизировать затраты на изменение информационной структуры, при этом затраты ограничены снизу и сверху.

Реализованное программное обеспечение способно производить точный расчет оценки живучести информационной структуры на основе полиномиальной модели.

Произведем тестирование программного обеспечения оценки живучести, реализующего нечеткую нейросетевую модель (Приложение А). В настоящее время существуют готовые ППП для работы с нейронными сетями. Примером может служить NNT, функционирующий под управлением ядра системы MatLab [118], что делает его таким популярным. К недостаткам данного пакета можно отнести следующее: некорректное обучение сети в GUI режиме для некоторых алгоритмов обучения, достаточно сложный консольный вариант работы с ППП NNT для неподготовленных пользователей, высокая стоимость.

Создадим и обучим ННС с помощью системы MatLab R2010a с ППП NNT версии 6.0.4. Экспортируем ранее сгенерированные при исследовании полиномиальной модели обучающие топологии информационной структур в формат данных MatLab (данная возможность предусмотрена в разработанном программном обеспечении). [117]

Интерфейс модуля NNT в режиме обучения представлен на рисунке 4.10.

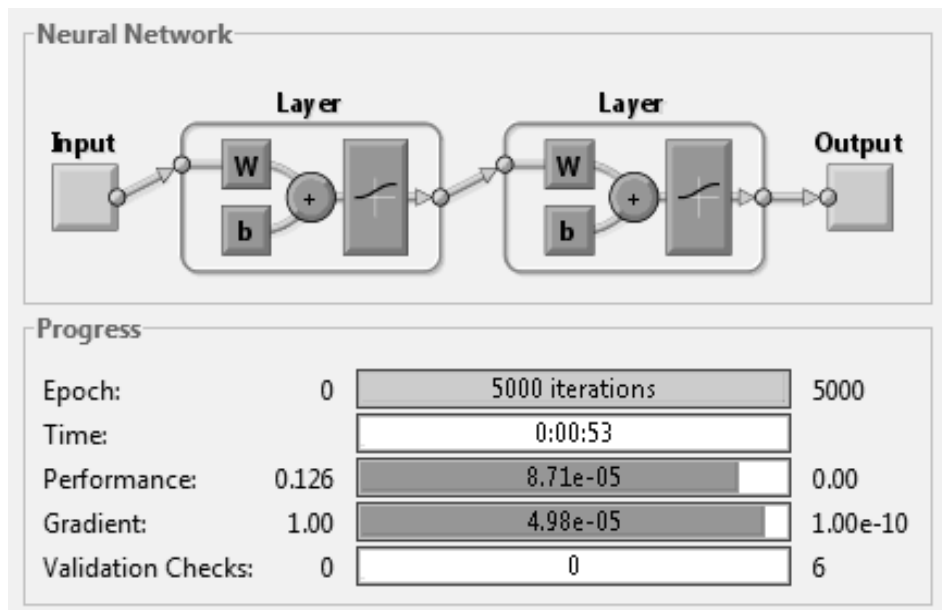


Рисунок 4.10 — Интерфейс модуля NNT в режиме обучения

Вводим обучающие данные в среду:

```
>> INP = [0 0 1 1 ... ];
```

Создаем ННС (сеть с прямой передачей сигнала, количество нейронов в скрытом слое – 21, выход – 1, сигмоидальная функция активации нейронов в скрытом и выходном слое, алгоритм обучения – пороговый обратного распространения ошибки):

```
>> net = newff(minmax(INP),[21,1],{'logsig','logsig'},'trainrp');
```

Задаем условие останова – достижение 5000 эпох (по аналогии с разработанным модулем):

```
>> net.trainParam.epochs = 5000;
```

Обучаем сеть:

```
>> net=train(net,INP,OUTP);
```

Процесс обучения занял 53 с (25 с в нашем модуле), среднеквадратическая ошибка (MSE) снизилась примерно до $10 \cdot 10^{-4}$.

Введем сгенерированный тестовый набор:

```
>> INP_TEST = [1 1 1 0 ... ];
```

Оценим живучесть тестового набора:

```
>> sim(net,INP_TEST);
```

В переменную ans система записала переменную, сформированную на выходе ННС, которая является оценкой живучести.

Произведем сравнительный анализ результатов оценки живучести [119] одного и того же тестового набора для ННС, полинома Татта и ППП NNT (рисунок 4.11 -4.14).

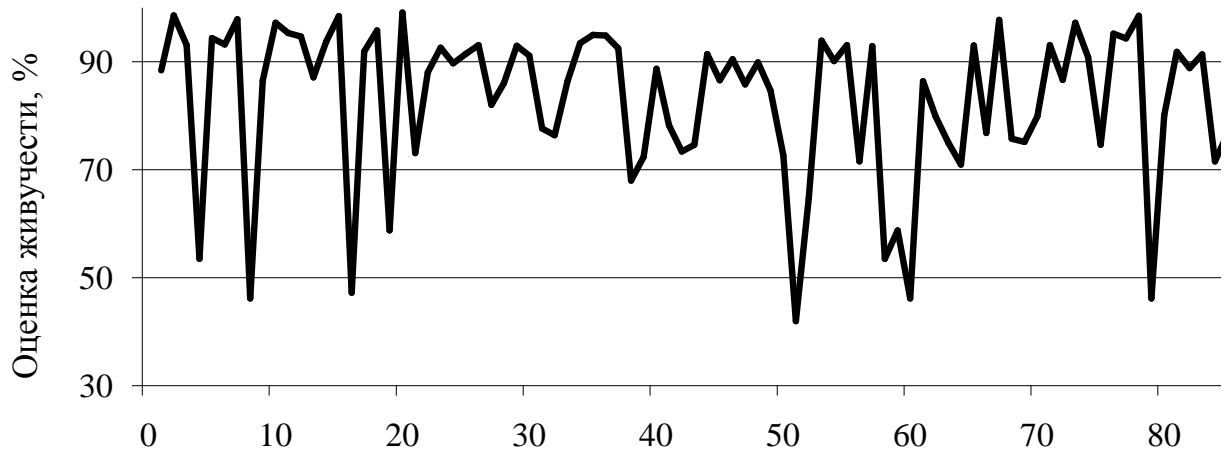


Рисунок 4.11 — Оценки живучести в полиномиальной форме



Рисунок 4.12 — Оценки живучести с помощью ННС



Рисунок 4.13 — Оценка живучести с помощью ППП NNT

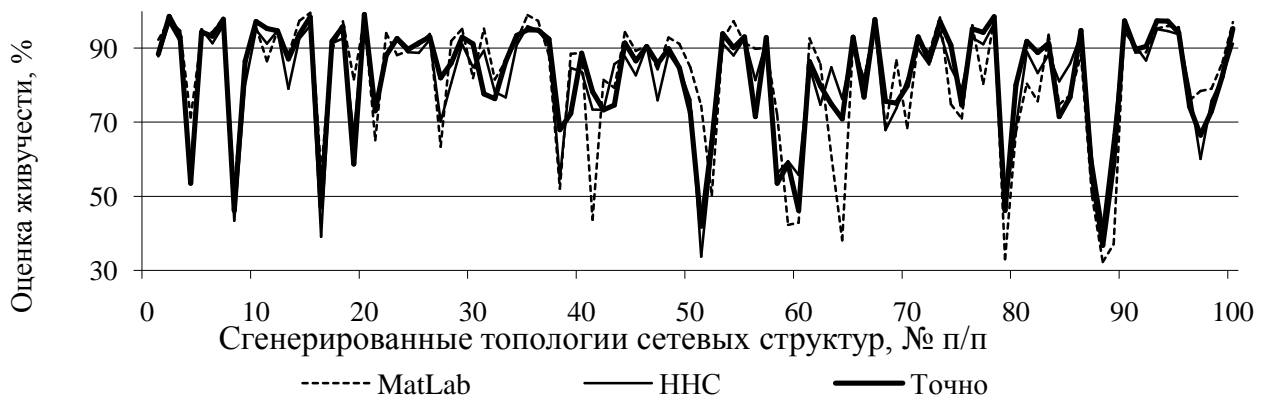


Рисунок 4.14 — Сравнительный анализ результатов оценки живучести

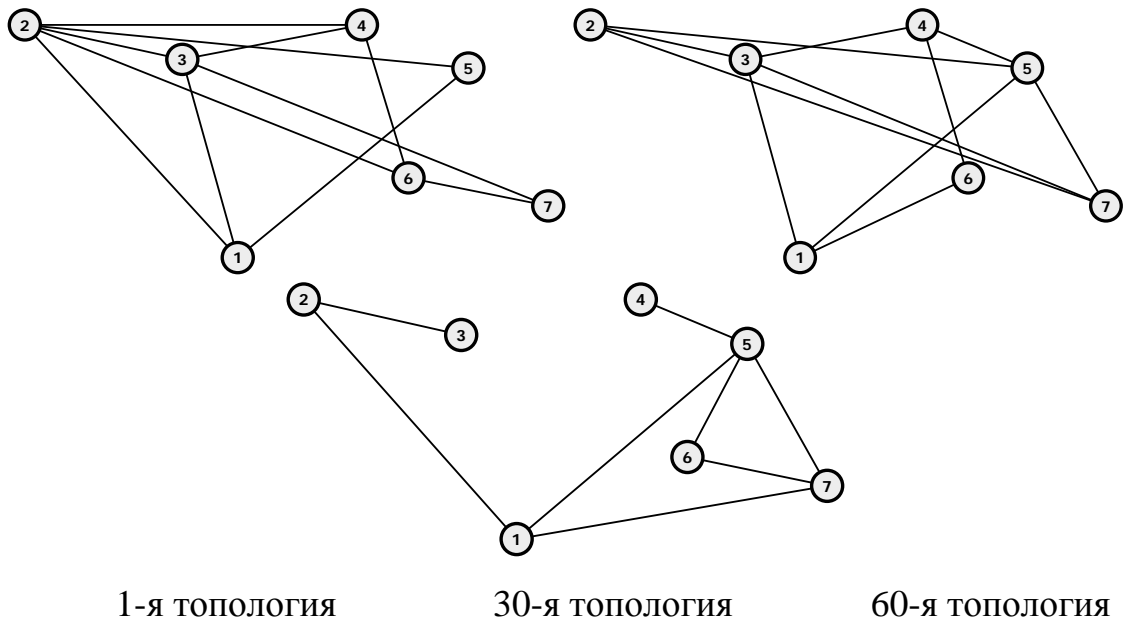


Рисунок 4.15 — Примеры тестовых топологий

Примеры сгенерированных тестовых топологий информационных структур с порядковыми номерами 1, 30 и 60 представлены на рисунке 4.15 и

в Приложении В.

Несмотря на то, что MSE в ППП NNT на порядок меньше, на основе анализа графика (рисунок 4.14), ННС более адекватна (максимальные отклонения от точного значения на некоторых информационных структурах в MatLab достигают ~30%, ННС ~15%) для одинакового тестового набора. Поэтому можно сделать вывод, что программное обеспечение на основе продукционное и нейросетвой модели способен точнее и быстрее оценивать живучесть информационных структур.

Для проведения вычислительного эксперимента программной реализации оценки живучести информационных структур была использована структура сети ООО «Тамбовские мультимедийные сети», которая представлена на рисунке 4.16.

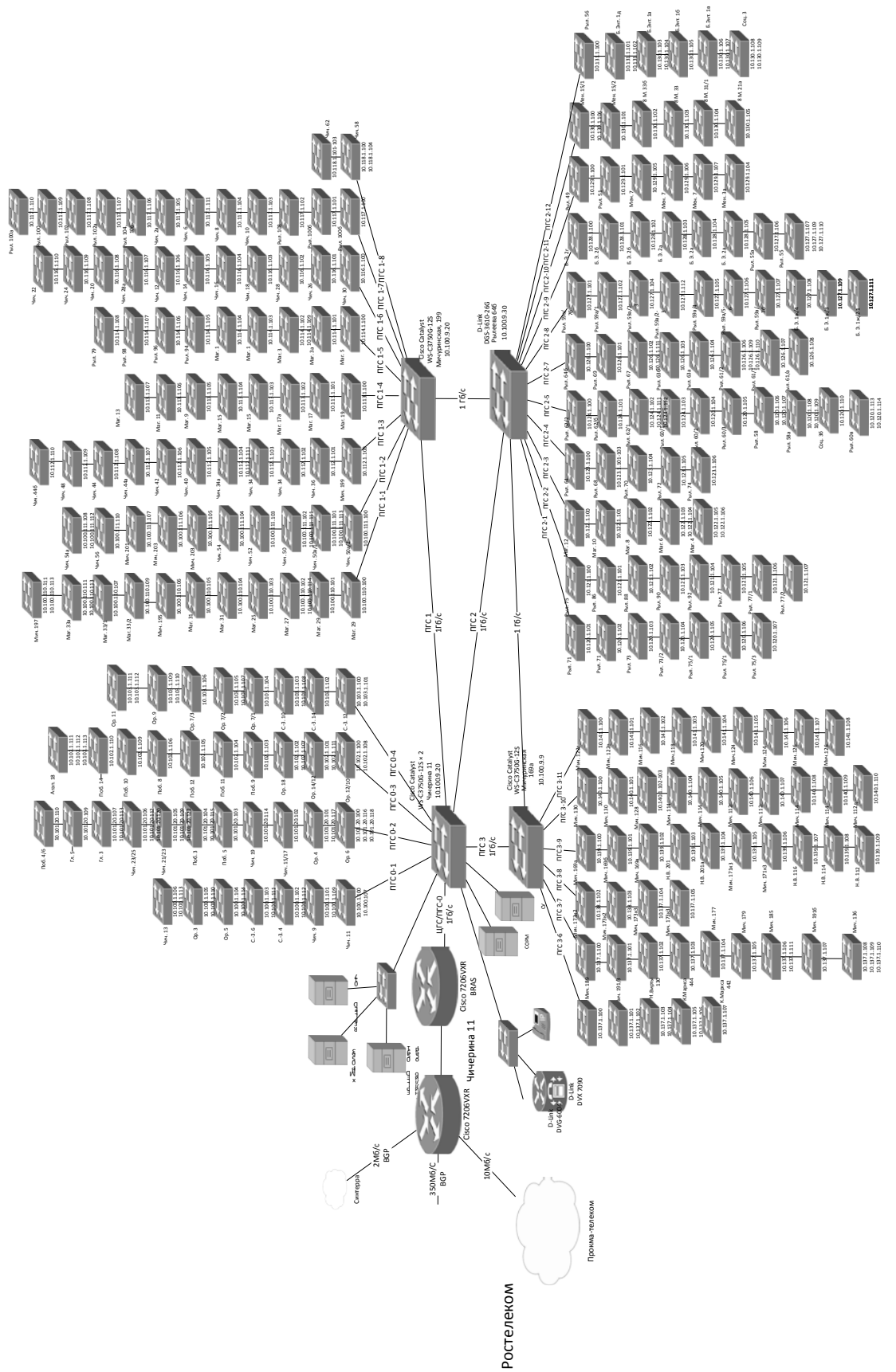


Рисунок 4.16 — Топология сети ООО «Тамбовские мультимедийные сети»

Выделим особо значимые узлы сети, выход из строя которых приведет к отказу части или всей сети. В данной сети их 8 (рисунок 4.17).

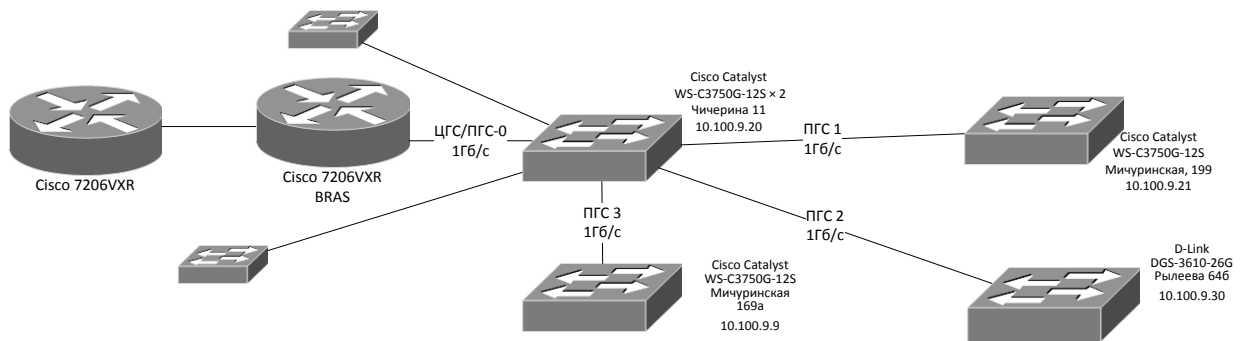


Рисунок 4.17 — Значимые узлы сети

Информационная структура в программном обеспечении выглядит следующим образом (рисунок 4.18).

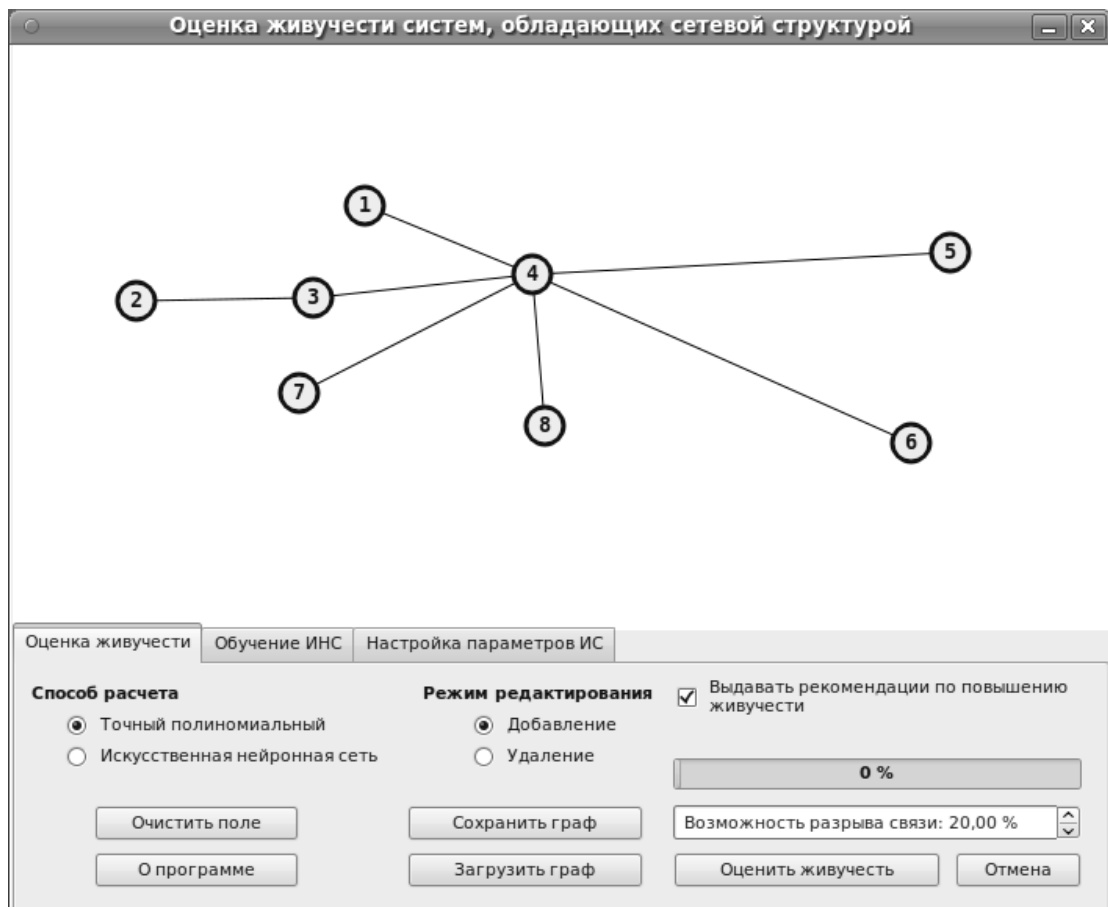


Рисунок 4.18 — Информационная структура в программном обеспечении

Данная структура содержит 7 связей, которые соединяют 8 узлов, которые могут быть подвержены уязвимости. Результаты оценки живучести для возможных значений разрыва связи представлены на рисунке 4.19.

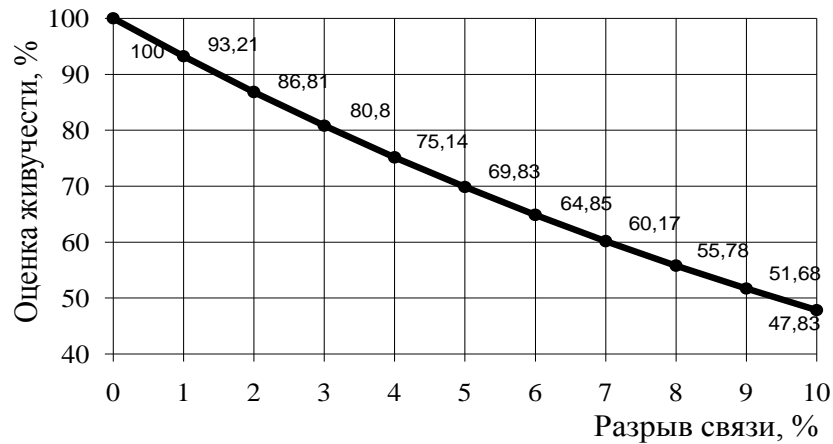


Рисунок 4.19 — Оценка живучести для возможных значений функции принадлежности ребра графа

Для повышения живучести добавим три связи в исходную топологию (рисунок 4.20).

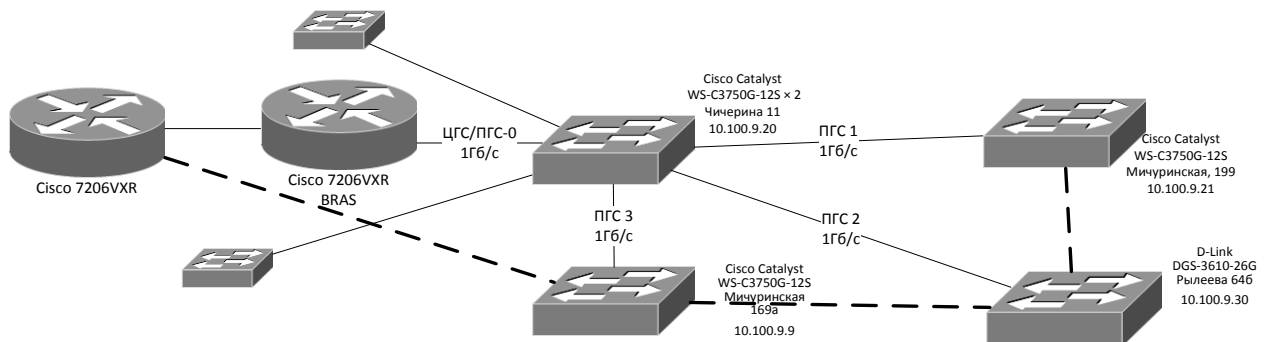


Рисунок 4.20 — Топология сети с добавлением новых связей

Проведем вычислительный эксперимент и сравнительный анализ, показывающий эффективность разработанных алгоритмов и моделей оценки живучести одного и того же тестового набора фрагмента структуры тамбовской области с помощью разработанного программного

обеспечения, полинома Тагга[121,122,123], ППП NNT и нечеткой нейросетевой моделью[124,125,126], которая описана в книге А.В. Боженюк, И.Н. Розенберг, Д.Н. Ястребнитская[127] (рисунок. 4.21).

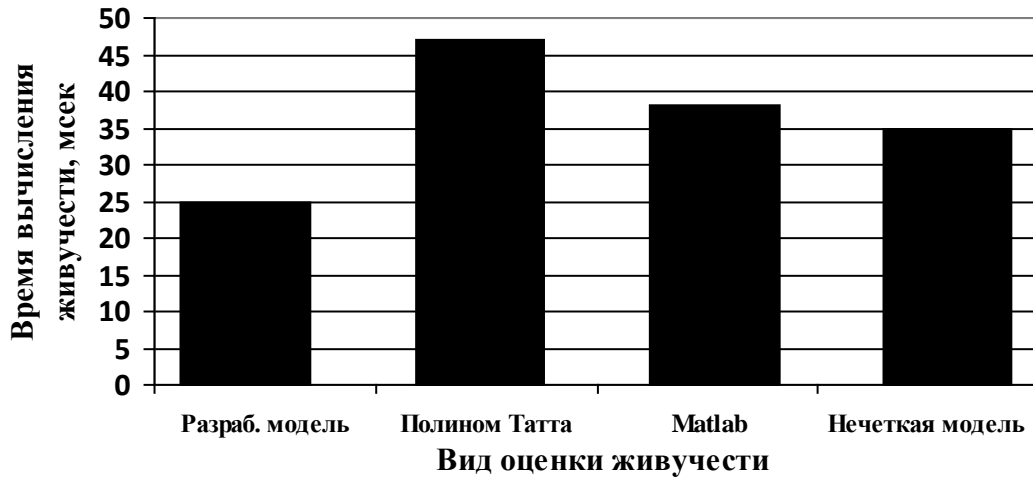


Рисунок 4.21 – Сравнительный анализ времени для оценки живучести одной и той же сетевой структуры Тамбовской области ОАО «Ростелеком»

Результаты оценки живучести исходной и новой топологий с добавлением связей для возможных значений разрыва связи после выдачи рекомендации по повышению функционирования информационной структуры представлены на рисунке 4.22.

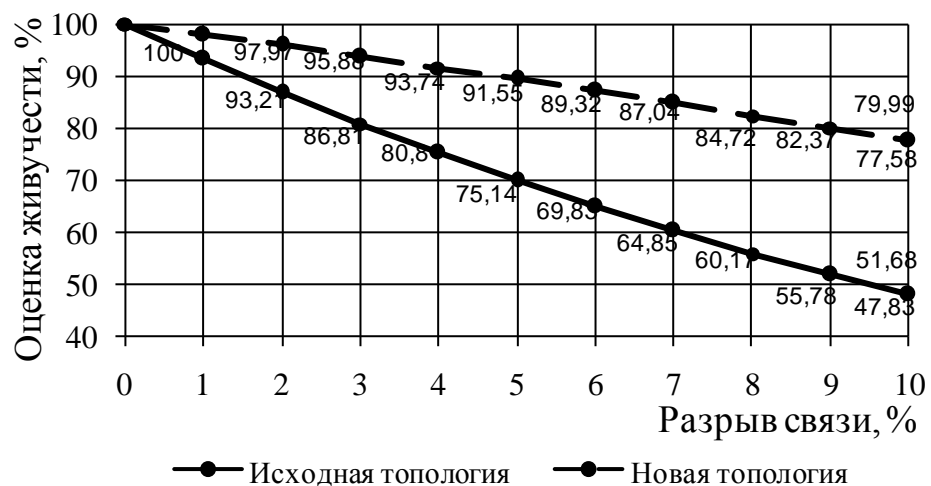


Рисунок 4.22 — Сравнение оценок живучести исходной (рисунок 4.17) и новой (рисунок 4.20) топологий информационных структур

Можно сделать вывод, что при добавлении связей оценка живучести увеличивается, из чего следует корректность работы разработанных моделей и алгоритмов оценки расчета живучести.

После добавления связей в исходную информационную структуру необходимо выдать рекомендации по повышению эффективности функционирования всей системы.

Алгоритм выбора способа оценки живучести интеллектуального модуля можно представить следующим образом[128]:

1. Проверка на возможность представления графа, соответствующего заданной сетевой структуре, в виде дерева. Если данная возможность существует, то переходим к этапу 2, иначе – к этапу 3.

2. Запрос системы о представлении сетевой структуры в виде дерева. Если пользователь выбрал ответ «Да», то переходим к этапу 4, иначе – к этапу 3.

3. Оценка живучести сетевой структуры с помощью полинома Татта. Переходим к этапу 8.

4. Определение системой главного узла дерева. Если пользователь согласен с выбором данного узла, то переходим к этапу 6, иначе – к этапу 5.

5. Запрос системы на ввод главного узла дерева.

6. Построение дерева сетевой структуры (рисунок 4.23).

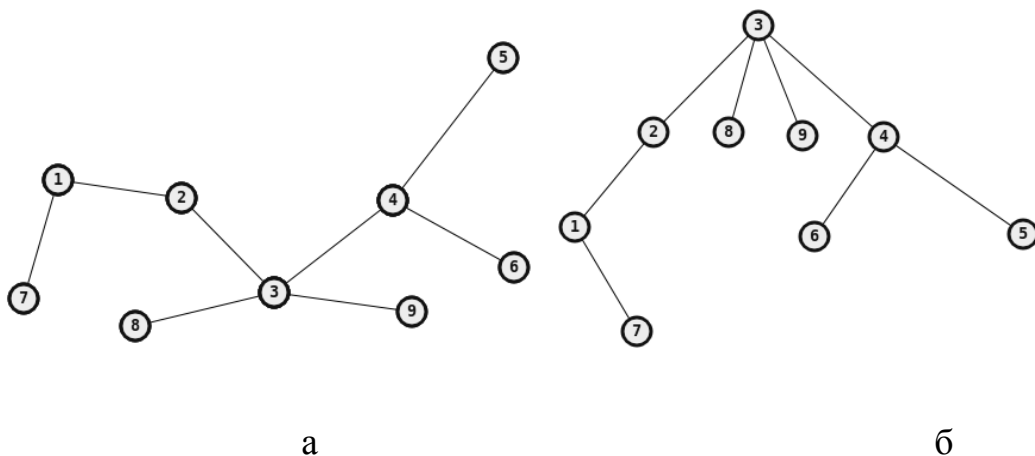


Рисунок 4.23 — Пример информационной структуры

а) исходная топология; б) топология (а), представленная в виде дерева

7. Оценка живучести сетевой структуры с помощью аппарата цепных дробей.

8. Выдача рекомендаций по повышению оценки живучести.

Алгоритм выдачи рекомендаций интеллектуального модуля по повышению оценки живучести представим следующим образом:

1. Если живучесть сетевой структуры была оценена с помощью аппарата цепных дробей, то переходим к этапу 2, иначе к этапу 3.

2. Запрос системы о максимальном количестве переносимых узлов при генерировании новых топологий.

3. Генерирование новых возможных топологий сетевых структур (либо перемещение узлов (аппарат цепных дробей), либо добавление новых связей (полином Татта).

4. Расчет оценок живучести для сгенерированных топологий.

5. Сортировка топологий и оценок живучести в порядке убывания.

6. Вывод отсортированных топологий на экран для выбора пользователем подходящей структуры.

7. Если пользователя устраивает оценка живучести, то переходим к этапу 8, иначе – к этапу 1.

8. Получена конечная топология сетевой структуры.

Для программной связи интеллектуального модуля оценки живучести с существующими был добавлен соответствующий элемент управления на вкладку «Оценка живучести» (рисунок 4.24).

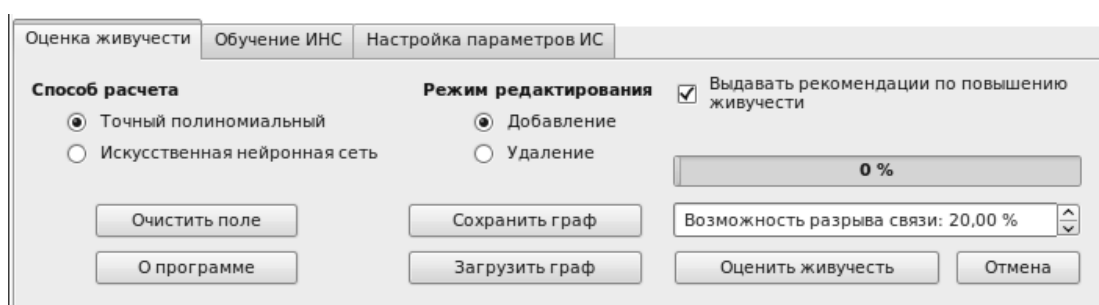


Рисунок 4.24 — Интеграция управляющего элемента «Выдавать рекомендации по повышению живучести»

Программная реализация алгоритма выбора способа оценки живучести интеллектуального модуля включает в себя интерфейс общения с пользователем, представленный на рисунок 4.25.

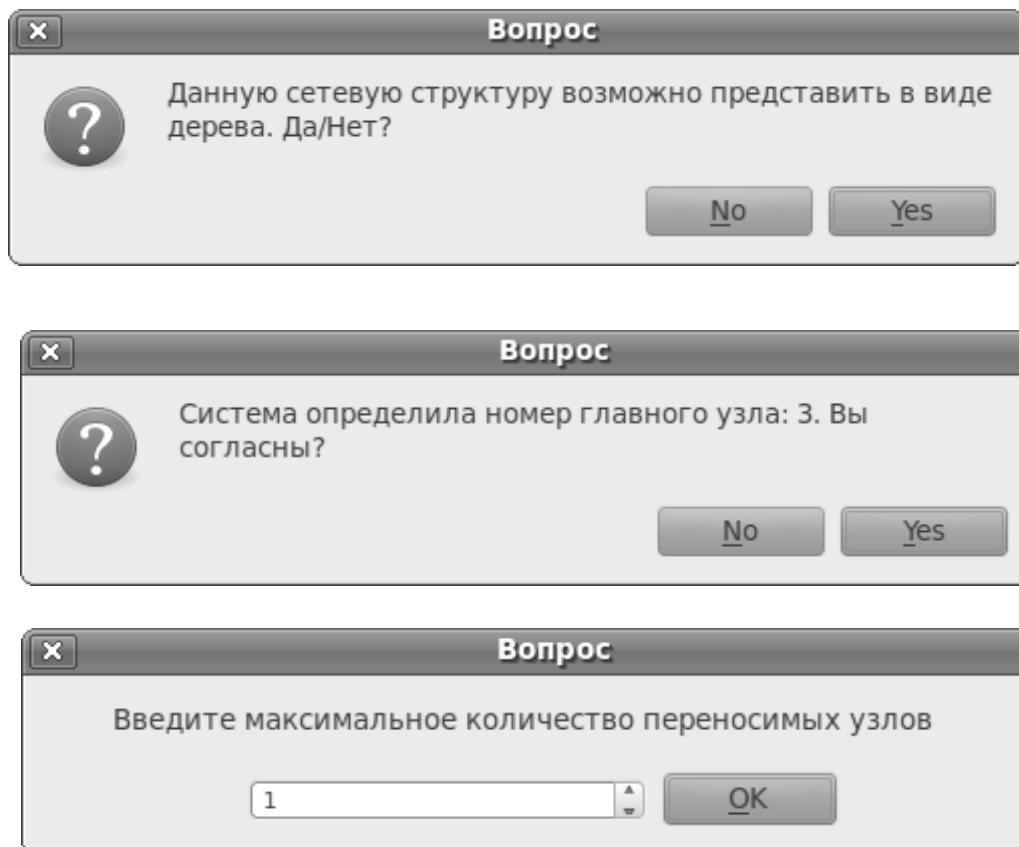


Рисунок 4.25 — Вопросы интеллектуального модуля

После ответов на предложенные системой вопросы выдаются рекомендации по повышению оценки живучести (рисунок 4.26). Система генерирует новые сетевые топологии и выводит их в порядке убывания оценок живучестей.

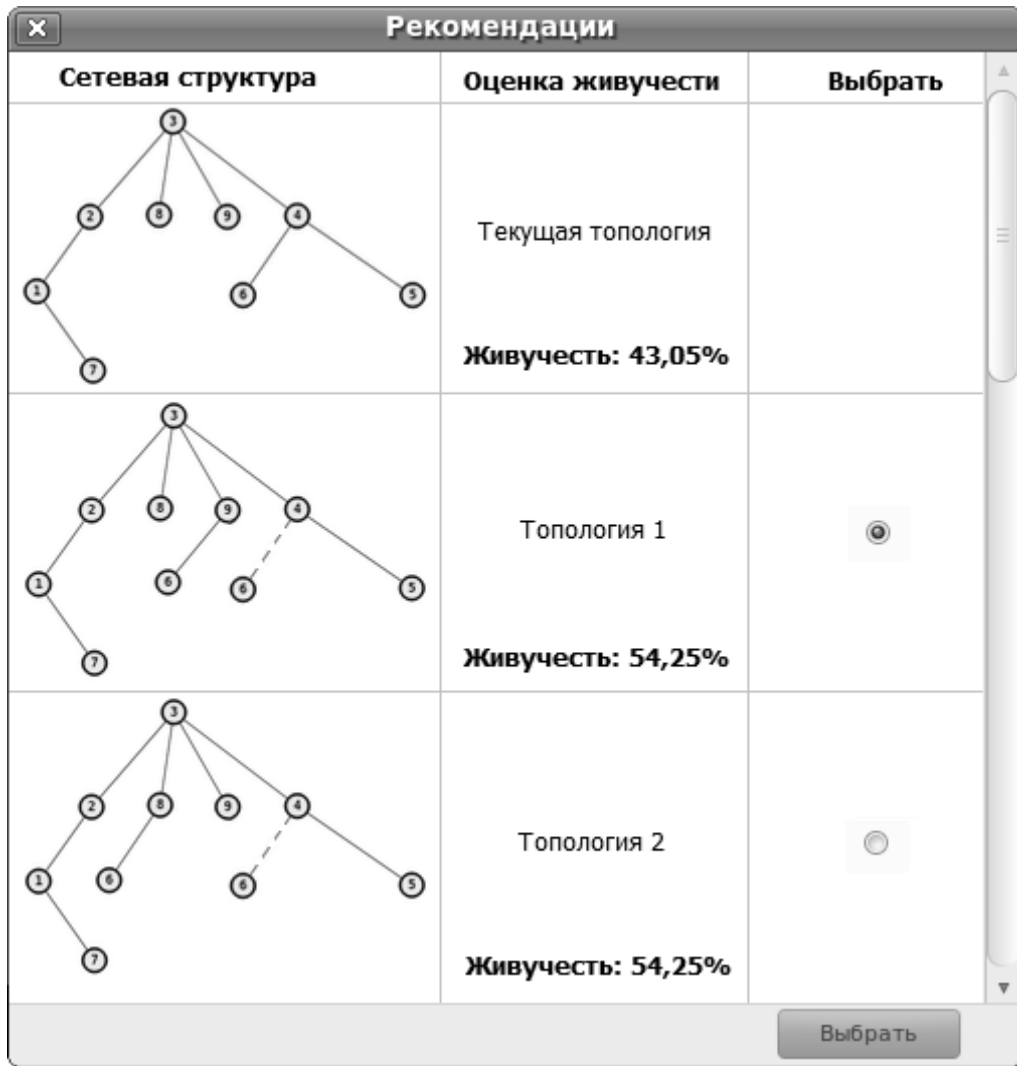


Рисунок 4.26 — Выдача рекомендаций для древовидных сетевых структур

Рекомендации, выдаваемые экспертами в данной области, которые представлены в Приложении Б, оказались достаточно близкими к рекомендациям, выдаваемым интеллектуальным модулем, что обуславливает корректность работы программы. В качестве тестовых были выбраны простые сетевые структуры, на более же сложных топологиях экспертам будет сложнее выдавать рекомендации по повышению живучести. Кроме того модуль в отличие от экспертов при сравнении топологий использует числовые значения оценок живучестей.

Далее произведем тестирование интеллектуального модуля оценки живучести на примере ранее рассмотренной сетевой структуры ООО «Тамбовские мультимедийные сети». Результат работы интеллектуального модуля выдачи рекомендаций по повышению оценки живучести представлен на рисунке 4.27.

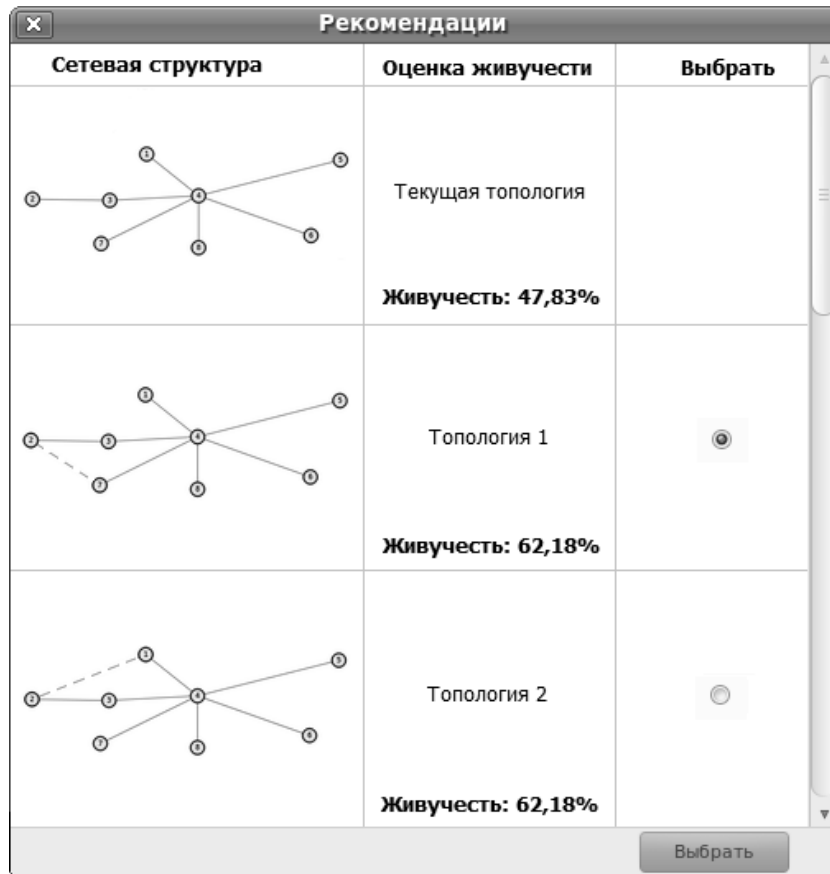


Рисунок 4.27 — Выдача рекомендаций для прочих сетевых структур

Также в интеллектуальном модуле реализован учет длин связей (экономическая эффективность), что показано на рисунке 4.27 (топологии с одинаковыми оценками живучестей сортируются в порядке увеличения длин добавляемых связей).

4.5 Выводы по четвертой главе

В результате разработано программное обеспечение оценки живучести НИС, которое включает в себя несколько модулей, обучение нечеткой нейронной продукционной сети, интерфейс задания качественных характеристик НИС, полиномиальный расчет оценки живучести, оценку живучести с помощью нечеткой нейронной продукционной сети. Проведен вычислительный эксперимент расчет оценки живучести информационной структуры Тамбовской области ОАО «Ростелеком» и ООО «Тамбовские мультимедийные сети» г. Тамбова с дальнейшей выдачей рекомендации по повышению оценки живучести НИС.

ЗАКЛЮЧЕНИЕ

Анализ исследования известных методов и подходов к оценке живучести информационных структур позволяет сделать заключение о недостаточной изученности вопроса о живучести и свидетельствует, о том, что нет единого подхода к оценке. С точки зрения технико-экономического аспекта, как на федеральном, так и на региональном уровне необходимо осваивать и внедрять новые методы, подходы и алгоритмы оценки живучести систем, обладающей информационных структурой, в теоретическую и практическую деятельность.

Точный полиномиальный расчет и другие методы оценки живучести систем, обладающий информационной структурой показал, что требуется больше затрачиваемых вычислительных мощностей и экспоненциально увеличивается время вычислений при увеличении числа узлов и связей информационной структуры, что не выгодно экономически. Поэтому предложенный нами подход для оценки живучести информационных структур на основе нечеткой нейронной продукционной модели и нечеткой нейронной продукционной сети является актуальным и оправданным решением.

Таким образом, можно считать, что цель исследования достигнута и получены следующие результаты:

1. Выполнен анализ существующих методов и подходов к оценке живучести информационных структур, который показал, что существующие модели и методы не могут обеспечить оценку живучести за требуемое время.

2. Построена нечеткая продукционная модель оценки живучести НИС, отличающейся использованием лингвистических переменных (территориальная распределенность, количество элементов и многообразие связей между ними, дальность передачи между узлами, возможность разрыва связи, сложность системы), характеризующих ее структуру и параметры для свертки которых используются однопараметрические Т-нормы, параметры

которых уточняются вследствие решения оптимизационной задачи, для решения которой применяется разработанный алгоритм, использующий метод чисел Фибоначчи совместно с правилом Голдстейна., позволяющая получить оценку живучести систем с НИС в условиях НВВ.

3. Разработан алгоритм перехода от построенной нечеткой продукционной модели к соответствующей ей нечеткой нейронной продукционной сети, отличающегося наличием пяти этапов каждый из которых представлен набором правил для свертки которых применяется однопараметрическая T-импликация, модифицирующей вывод Мамдани., позволяющий снизить вычислительные затраты расчета оценки живучести систем с НИС.

4. Разработан алгоритм обучения нечеткой нейронной продукционной сети, основанный на использовании предложенного квазидискретного Ньютоновского метода с немонотонной стабилизацией, в котором для решения задачи одномерной оптимизации используется модификация метода чисел Фибоначчи, основанного на применении правила Голдстейна., позволяющий сократить на 10 – 15% количество вычисляемых операции по сравнению с другими методами и тем самым сократить время обучения на 12 – 15 %.

В диссертации решена научная задача – построена модель и алгоритмы оценки живучести системы с НИС, связывающие ее качественные характеристики и количественные переменные для сокращения времени расчета., а поставленная цель достигнута.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. ГОСТ Р 53111-2008. Устойчивость функционирования сети связи общего пользования. Требования и методы проверки. – М.: Стандартинформ 2009.
2. Хорохорин, М.А. Выбор видов средств парирования негативных внешних воздействий на информационную систему / М.А. Хорохорин, А.А. Долгов, А.И. Елисеев, Ю.В. Минин, Д.В. Поляков // Информатика: проблемы, методология, технологии: материалы XIII Международной научно-методической конференции, Воронеж, 7-8 февраля 2013 г.: в 4 т. - Воронеж: Издательско-полиграфический центр Воронежского государственного университета. - Т. 2. - 2013. - С.353-357.
3. Силов, В. Б. Принятие стратегических решений в нечеткой обстановке. / В. Б. Силов. – М.: ИНПРО-РЕС, 1995.
4. Борисов, А. Н. Обработка нечеткой информации в системах принятия решений / А. Н. Борисов, А. В. Алексеев, Г. В. Меркурьева.– М.: Радио и связь, 1989.
5. Стекольников, Ю.И. Живучесть систем. / Ю.И. Стекольников. – СПб.: Политехника, 2002. – 155 с.
6. Использование теории возможностей при оценке живучести сетевых информационных структур / А.А. Долгов, Ю.Ю. Громов, М.А. Хорохорин, Ю.В. Минин // Информация и безопасность. – Воронеж: Изд. ВГУ. – 2014.
7. Применение кластерных вычислений для оценки живучести сетевых информационных структур / М.А. Хорохорин, А.А. Долгов, Ю.Ю. Громов, Ю.В. Минин // Информация и безопасность. – Воронеж: Изд. ВГУ. – 2014.
8. Додонов, А.Г. Введение в теорию живучести вычислительных систем / А.Г. Додонов, М.Г. Кузнецова, Е.С. Горбачик. – Киев: Наукова

думка, 1990. – 184 с.

9. Хорохорин, М.А. Системный анализ в обеспечении живучести сетевых информационных систем / М.А. Хорохорин, А.А. Долгов, А.И. Елисеев, М. Ауад // Математические методы и информационно-технические средства: Труды VIII Всероссийской научно-практической конференции, 22-23 июня 2012 г. - Краснодар: Краснодарский университет МВД России, 2012. - 278 с.

10. Синтез и анализ живучести сетевых систем: монография / Ю.Ю. Громов, В.О. Драчев, К.А. Набатов, О.Г. Иванова. – М.: Машиностроение-1, 2007. – 152 с.

11. Черкесов, Г.Н. Методы и модели оценки живучести сложных систем / Г.Н. Черкесов. – М.: Знание, 1987.

12. ГОСТ Р 50922-96. Защита информации. Основные термины и определения. – М.: Госстандарт, 1996.

13. Хорохорин, М.А. Выбор видов средств парирования при различных типах негативных внешних воздействий на информационную систему / М.А. Хорохорин, А.А. Долгов, А.И. Елисеев, Ю.В. Минин, Д.В. Поляков // Информатика: проблемы, методология, технологии: материалы XIII Международной научно-методической конференции, Воронеж, 7-8 февраля 2013 г.: в 4 т. - Воронеж: Издательско-полиграфический центр Воронежского государственного университета. - Т. 2. - 2013. - С.357-361.

14. Волик, Б.Г. Эффективность, надежность и живучесть управляющих систем / Б.Г. Волик, Ё.А. Рябинин // Автоматика и телемеханика. – 1984. – № 12.

15. Воронин, А.А. Оптимальные иерархические структуры. / А.А. Воронин, С.П. Мишин. – М.: ИПУ РАН, 2003. – 214 с.

16. Уилсон, Р. Введение в теорию графов: Пер. с англ. И.Г. Никитиной / Р. Уилсон. – М.: Мир, 1977. – 202 с.

17. Татт, У. Теория графов: Пер. с англ. / У. Татт. – М.: Мир, 1988. – 424 с.

18. Берж, К. Теория графов и её применение. / К. Берж. – М., 1962. – 319 с.
19. Харари, Ф. Теория графов, 2-е издание: Пер. с англ. и предисл. В.П. Козырева. Под ред. Г.П. Гаврилова / Ф. Харари. – М.: Едиториал УРСС, 2003. – 296 с.
20. Лысенко И. В. Анализ и синтез сложных технических систем. Часть 1. / И.В. Лысенко. – М: Воениздат, 1995. – 397 с.
21. Лысенко И. В. Анализ и синтез сложных технических систем. Часть 2. / И.В. Лысенко. – М: Воениздат, 1995. – 260 с.
22. Касти, Дж. Большие системы. Связность, сложность и катастрофы. / Дж. Касти. – М.: Мир, 1982. – 216 с.
23. Ковалев, А.П. О живучести объектов энергетики. / А.П. Ковалев, В.И. Чурсинов, В.В. Якимишина. – Донецк: ДонНТУ, 2004. – 221 с.
24. Руденко, Б.Н. Надежность систем энергетики. / И.Н. Ушаков. – М.: Наука, 1986. - 252 с.
25. Крапивин, В. Ф. О теории живучести сложных систем./ В.Ф. Крапивин. – М.: Наука, 1978. – 248 с.
26. Хорохорин, М.А. К вопросу о показателях живучести сетевых информационных систем / М.А. Хорохорин, А.А. Долгов, А.И. Елисеев, Хак. Д. Лыонг // Математические методы и информационно-технические средства: Труды VIII Всероссийской научно-практической конференции, 22-23 июня 2012 г. - Краснодар: Краснодарский университет МВД России, 2012. - 278 с. - С.74.
27. Перельмутер, А.В. Об оценке живучести несущих конструкций. / А.В. Перельмутер. – М.: Стройиздат, 1969. – 190 с.
28. Хорохорин, М.А. К вопросу исследования живучести информационных систем в рамках механики катастроф / М.А. Хорохорин, А.А. Долгов, А.И. Елисеев, Аль-Балуши // Прикладная математика, управление и информатика: сборник научных работ, 3 - 5 октября 2012 г. - Белгород: БелГУ. - 2012.- Т.1. - С. 122-125.

29. Додонов А.Г. Проблемы и тенденции создания живучих вычислительных систем: Метод. Разработки. / А.Г. Додонов, М.Г. Кузнецова. – К.: Наук. думка, 1981. – 55 с.
30. Вишневский, В.М. Теоретические основы проектирования компьютерных сетей / В.М. Вишневский. – М.: Техносфера, 2003. – 512 с.
31. Филлипс, Д. Методы анализа сетей: Пер. с англ. / Д. Филлипс, А. Гарсиа-Диас. – М.: Мир, 1984. – 496 с., ил.
32. Чхартишвили, А.Г. Теоретико-игровые модели информационного управления. А.Г. Чхартишвили / – М.: ЗАО «ПМСОФТ», 2004. – 227 с.
33. Горшков, В.В. Логико-вероятностный метод расчета живучести сложных систем / В.В. Горшков. – Кибернетика АН УССР, 1982. – С. 104-107.
34. Рябинин, И.А. Логико-вероятностные методы исследования надежности структурно-сложных систем. / И.А. Рябинин, Г.Н. Черкесов. - М.: Радио и связь, 1984. - 238 с.
35. Острейковский, В.А. Теория надежности. / В.А. Острейковский – М.: Высшая школа, 2003. – 463 с.
36. Барлоу, Р. Статистическая теория надежности и испытание на безотказность. / Р. Барлоу, Ф. Прошан. – М.: Наука, 1984. – 328 с.
37. Куюнджич, С.М. Разработка и анализ моделей надежности и безопасности систем. / С.М. Куюнджич. – М.: Физматлит, 2001. – 463 с.
38. Мелентьев, В. А. Скобочная форма описания графов и ее использование в структурных исследованиях живучих вычислительных систем / В. А. Мелентьев // Автометрия. – 2000. – № 4. – С. 36-52.
39. Райншке, К. Оценка надежности систем с использованием графов. / К. Райншке, И.А. Ушаков // – М.: Радио и связь, 1988. – 208 с.
40. Хорохорин, М.А. Решение задачи оценки живучести локальных вычислительных сетей с целью повышения устойчивости их функционирования / М.А. Хорохорин, А.А. Долгов, А.И. Елисеев, Ю.В. Минин // Информатика: проблемы, методология, технологии: материалы XIII

Международной научно-методической конференции, Воронеж, 7-8 февраля 2013 г.: в 4 т. - Воронеж: Издательско-полиграфический центр Воронежского государственного университета. - Т. 1. - 2013. - С.387-390.

41. Оре, О. Графы и их применение. / О. Оре. – М., 2002. – 171 с.

42. Алгоритмы и программы решения задач на графах и сетях / М.И. Нечепуренко, В.К. Попков, С.М. Майнагашев, С.Б. Кауль и др. – Новосибирск: Наука: Сиб. отд-ние, 1990.

43. Хорохорин, М.А. Структуризация характеристик живучести сетевых информационных систем / М.А. Хорохорин, А.А. Долгов, А.И. Елисеев, Аль-Балуши // Математические методы и информационно-технические средства: Труды VIII Всероссийской научно-практической конференции, 22-23 июня 2012 г. - Краснодар: Краснодарский университет МВД России, 2012. - 278 с. С.72

44. Камерон, П. Теория графов, теория кодирования и блок-схемы / П. Камерон, ван Линт Дж. – М.: Наука, 1980. – 140 с.

45. Алексеев, В.В. Использование графов в моделировании структуры при внешних воздействиях / М.А. Никанкин, В.В. Алексеев // Научно-теоретический и прикладной журнал широкого профиля Вестник ТВВАИУРЭ (ВИ), 2008 г. – Тамбов – №2(6) С. 67–71

46. Цвиркун, А.Д. Структура сложных систем. / А.Д. Цвиркун. – М.: «Сов. Радио», 1975. – 200 с.

47. Губко, М.В. Математические модели оптимизации иерархических структур. / М.В. Губко. – М.: Ленанд, 2006. – 264 с.

48. Борисов, В.В. Нечеткие модели и сети / В.В. Борисов, В.В. Круглов, А.С. Федулов. – М.: Горячая линия-Телеком, 2007. – 284с.

49. Круглов, В.В. Искусственные нейронные сети. Теория и практика / В.В. Круглов, В.В. Борисов. – М.: Горячая линия-Телеком, 2002. – 382 с.

50. Пospelова, Д. А. Нечеткие множества в системах управления и искусственного интеллекта / Д. А. Пospelова. – М.: Наука. Гл. ред. физ.-мат. лит. 1986.

51. Хайкин, С. Нейронные сети: полный курс, 2-е издание: Пер. с англ. / С. Хайкин. – М.: Вильямс, 2006. – 1104 с.: ил.
52. Заенцев, И.В. Нейронные сети: основные модели / И.В. Заенцев. – 1999. – 76 с.
53. Хофман, А. Введение в теорию нечетких множеств./ А. Хофман. – М.: Радио и связь, 1982.
54. Галушкин, А.И. Теория нейронных сетей / А.И. Галушкин. – М.: ИПРЖР, 2000. – 416 с.
55. Борисов, В. В. Компьютерная поддержка сложных организационно-технических систем. / В. В. Борисов, И. А. Бычков, А. С. Федулов. – М.: Горячая линия - Телеком, 2002.
56. Белов, С. В. Безопасность жизнедеятельности и защита окружающей среды (техносферная безопасность) / С. В Белов. // - 2-е изд., испр. и доп. - М. : Юрайт, 2011. – 679с.
57. Загоруйко, Н. Г. Прикладные методы анализа данных и знаний. / Н. Г. Загоруйко. – Новосибирск: Изд-во Ин-та мат., 1999.
58. Блинов, А.М. Информационная безопасность: Учебное пособие. Часть 1. А.М. Блинов // – СПб.:Изд-во СПбГУЭФ, 2010. – 96 с.
59. Павлов, А.И. Безопасность жизнедеятельности. Учебное пособие. /А.И. Павлов, В.Н. Тушонков, В.В. Титаренко // – М.: МИЭМП, 2006. — 302 с.
60. Городецкий, А.Я. Информационные системы. Вероятностные модели и статистические решения, учеб. пособие / А .Я. Городецкий. – СПб.: Изд-во СПбГПУ, – 2003. – 326 с. Data Processing Vocabulary. Section 14. Reliability, Maintenance and Availability. – Geneva: ISO 2382, 1976. – 16 p.
61. Осовский, С. Нейронные сети для обработки информации / С. Осовский. – М.: Финансы и статистика, 2002. – 344 с.
62. Мкртчян, С.О. Нейроны и нейронные сети. (Введение в теорию формальных нейронов) / С.О. Мкртчян – М.: Энергия, 1971. – 232 с.
63. Суровцев, И.С. Нейронные сети / И.С. Суровцев, В.И. Клюкин,

Р.П. Пивоварова. – Воронеж: ВГУ, 1994. – 224 с.

64. Кохонен, Т. Самоорганизующиеся карты / Т. Кохонен. // - М.: БИНОМ. Лаборатория знаний, 2008. - 665с.

65. Уоссермен, Ф. Нейрокомпьютерная техника: Теория и практика: Пер. с англ. / Ф. Уоссермен. – 1992.

66. Яхьяева, Г.Э. Нечёткие множества и нейронные сети / Г.Э. Яхьяева. – М.: Интернет-Университет Информационных Технологий. 2006. – 316 с.

67. Хорохорин, М.А. Применение распределенных информационных систем для оценки живучести локальных вычислительных сетей / М.А. Хорохорин, Ю.Ю. Громов, Ю.В. Минин, В.В. Родин // Управление, технологии и безопасность в информационных системах специального назначения: Сборник научных трудов. – 2012. – С.66-69.

68. Хорохорин, М.А. К вопросу оценки живучести сетевых структур с использованием Grid-технологий / М.А. Хорохорин, А.А. Долгов, Ю.В. Минин, А.Б.М.П.Б. Шихук // Информация и безопасность. - Воронеж: Издательско-полиграфический центр Воронежского государственного университета. - 2012. - № 2. Том 15. - С. 249-252.

69. Хорохорин, М.А. Применение распределенных информационных систем для оценки живучести нечетких графов / М.А. Хорохорин, А.А. Долгов, М. Ауад, Х.Д. Лыонг // Информация и безопасность. - Воронеж: Издательско-полиграфический центр Воронежского государственного университета. - 2012. - № 2. Том 15. - С. 245-248.

70. Хорохорин, М.А. Применение распределенных вычислений для оценки живучести систем, обладающих сетевой структурой / М.А. Хорохорин, А.А. Долгов // Сборник материалов открытой Всероссийской научно-практической конференции «Актуальные проблемы деятельности подразделений УИС». Воронеж: ООО ИПЦ «Научная книга», 2012. с.61.

71. Хорохорин, М.А. Применение распределенных систем для оценки живучести сетевых структур в условиях неопределенности / М.А.

Хорохорин, А.А. Долгов // Сборник материалов открытой Всероссийской научно-практической конференции «Актуальные проблемы деятельности подразделений УИС». Воронеж: ООО ИПЦ «Научная книга», 2012. с.62.

72. Хорохорин, М.А. Использование системы Х-СОМ для оценки живучести сетевых структур / М.А. Хорохорин, А.А. Долгов, А.И. Елисеев, М.К Силла // Прикладная математика, управление и информатика: сборник научных работ, 3 - 5 октября 2012 г. - Белгород: БелГУ. - 2012.- Т.2. - С. 72-76.

73. Хорохорин, М.А. Применение распределенных систем для оценки живучести сетевых структур в условиях неопределенности / М.А. Хорохорин, А.А. Долгов, А.И. Елисеев // Математические методы и информационно-технические средства: Труды VIII Всероссийской научно-практической конференции, 22-23 июня 2012 г. - Краснодар: Краснодарский университет МВД России, 2012. - 278 с. - С.257.

74. Хорохорин, М.А. К вопросу оценки живучести сложных систем с использованием распределенных систем на основе графовой модели / М.А. Хорохорин, А.А. Долгов, А.И. Елисеев, Хак.Д. Лыонг // Прикладная математика, управление и информатика: сборник научных работ, 3 - 5 октября 2012 г. - Белгород: БелГУ. - 2012.- Т.2. - С. 287-292.

75. Борисов, В.В. Основы нечеткого логического вывода. Учебное пособие для вузов / В.В. Борисов, А.С. Федулов, М.М. Зернов. – М.: Горячая линия Телеком, 2014. – 122 с.

76. Ягер, Р.Р. Нечеткие множества и теория возможностей: Последние достижения / Р.Р. Ягер. – М.: Радио и связь, 1986.

77. Леденева, Т.М. Обработка нечеткой информации: учебное пособие / Т.М.Леденева. – Воронеж: Воронежский государственный университет, 2006. – 233 с.

78. Заде, Л. А. Понятие лингвистической переменной и его применение к принятию приближенных решений. / Л. А. Заде. – М.: Мир, 1976.

79. Борисов, А.Н. Модели принятия решений на основе лингвистической переменной / А.Н. Борисов, А.В. Алексеев, О.А. Крумберг О.А. – Рига: Зинатне, 1982. – 256 с.

80. Хорохорин, М.А. Многокритериальная задача оптимизации на нечетком множестве альтернатив / М.А. Хорохорин, Ю.В. Удодов, Ю.А. Губсков, А.А. Долгов // Приборы и системы. Управление, контроль, диагностика. - 2012. - №8 - Стр. 8-12.

81. Хорохорин, М.А. Решение задачи проектирования распределенной базы данных с использованием генетических алгоритмов / М.А. Хорохорин, Д.В. Ю.В. Удодов, Ю.А. Губсков, А.А. Долгов // Приборы и системы. Управление, контроль, диагностика. - 2012. - №10 - Стр. 18-22.

82. Хорохорин, М.А. Построение нечетких запросов к реляционным базам данных / М.А. Хорохорин, Ю.В. Удодов, Ю.А. Губсков, А.А. Долгов // Приборы и системы. Управление, контроль, диагностика. - 2012. - №7 - Стр. 6-10.

83. Математические постановки задач восстановления и обеспечения живучести для многопродуктовых сетей / М.Р. Давидсон, Ю.Е. Малашенко, Н.М. Новикова и др. – М.: ВЦ РАН, 1993.

84. Хорохорин, М.А. Генетический алгоритм для построения нечеткого множества Парето / М.А. Хорохорин, Ю.В. Удодов, Ю.А. Губсков, А.А. Долгов // Приборы и системы. Управление, контроль, диагностика. - 2012. - №7 - Стр. 27-31.

85. Хорохорин, М.А. Методы обеспечения и повышения живучести сетевых информационных систем / М.А. Хорохорин, А.А. Долгов, А.И. Елисеев, К.М. Копылов // Математические методы и информационно-технические средства: Труды VIII Всероссийской научно-практической конференции, 22-23 июня 2012 г. - Краснодар: Краснодарский университет МВД России, 2012. - 278 с. - С.75.

86. Черноруцкий, И.Г. Методы оптимизации в теории управления / И.Г. Черноруцкий - Учебное пособие. — СПб.: Питер, 2004. — 256 с.

87. Воробьёв, Н. Н. Числа Фибоначчи / Н. Н. Воробьёв. - Издательство: Наука, 1978. — 144с.
88. Рудаков А. Н. Числа Фибоначчи и простота числа 2127-1 / А. Н. Рудаков - Математическое Просвещение, третья серия. — 2000. — Т. 4.
89. Максимов, Ю.Я. Алгоритмы решения задач нелинейного программирования: учебное пособие / Ю. Я. Максимов, Е. А. Филипповская - Москва: МИФИ.
90. Краснощеков, П.С. Принципы построения моделей / П.С. Краснощеков, А.А. Петров. – М.: Фазис, 2000. – 412 с.
91. Братищенко, В.В. Проектирование информационных систем. / В.В. Братищенко. – Иркутск: Изд-во БГУЭП, 2004. – 84 с.
92. Хорохорин, М.А. Исследование программного обеспечения моделирования и оценки нечетких сетевых информационных систем / М.А. Хорохорин // Гаудеамус. 2010. №2 (16). Материалы XIV Международной научно-практической конференции-выставки "Актуальные проблемы информатики и информационных технологий". Тамбов: Издательство ТГУ им. Г.Р. Державина, 2010. С. 323-326.
93. Хорохорин, М.А. Синтез программного обеспечения моделирования и оценки живучести сетевых информационных систем / М.А. Хорохорин, А.А. Долгов // Материалы II-ой международной кластерной научно-практической конференции "Аспекты ноосферной безопасности в приоритетных направлениях деятельности человека". – Тамбов: Изд-во «ТР-принт». – 2011. – С. 92-94.
94. Вендров, А.М. Практикум по проектированию программного обеспечения экономических информационных систем: Учеб. пособие, 2-е изд., перераб. и доп. / А.М. Вендров. – М.: Финансы и статистика, 2006. – 192 с.
95. Вендров, А.М. Проектирование программного обеспечения экономических информационных систем, 2-е изд., перераб. и доп. / А.М. Вендров. – М.: Финансы и статистика, 2006. – 544 с.

96. Иванов, Д.Ю. Основы моделирования на UML: Учеб. пособие / Д.Ю.Иванов, Ф.А. Новиков. – СПб.: Изд-во Политехн. ун-та, 2010. – 249с.
97. Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, А.Джекобсон. – М., СПб.: ДМК Пресс, Питер, 2004. – 432 с.
98. Бабич, А.В. Введение в UML. Лекция 5: Диаграмма активностей: крупным планом [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/1007/229/lecture/5958>.
99. Хорохорин, М.А. Синтез программного обеспечения моделирования и процедур поиска оптимальных путей в нечетких графах / М.А. Хорохорин // Информатика: проблемы, методология, технологии: материалы XI Международной научно-методической конференции, Воронеж, 10-11 февраля 2011 г.: в 3 т. – Воронеж: Издательско-полиграфический центр Воронежского государственного университета. -2011. - Т. 2. – С.430-434.
100. Хорохорин, М.А. Синтез моделирования и оценки оптимальных путей в направленных нечетких графах / М.А. Хорохорин // Сборник материалов Всероссийской конференции с элементами научной школы для молодежи «Математическое моделирование в технике и технологии». Воронеж: Издательско-полиграфический центр Воронежского государственного технического университета, 2011. С.77-80.
101. Пархоменко, С.С. Обучение нейронных сетей методом Левенберга-Маркварда в условиях большого количества данных / С.С. Пархоменко, Т.М. Леденёва // Вестник Воронежского государственного университета; Серия: Системный анализ и информационные технологии, 2014, № 2.С. 98-106.
102. Бахвалов, Н.С. Численные методы. Н.С. Бахвалов,Н.П.Жидков, Г.Г.Кобельков — 8-е изд.. — М.: Лаборатория Базовых Знаний, 2000.
103. TIОBE Programming Community Index for April 2013. – Режим доступа: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>.
104. Федоренко, Ю. Алгоритмы и программы на Turbo Pascal. Учебный курс /Ю.Федоренко - Издательство:Питер,2001-240с.

105. Страуструп, Б. Язык программирования C++ / Б. Страуструп. – М.: Бином, 2008.
106. Фленов, М.Е. Программирование на C++ глазами хакера. / М.Е. Фленов – СПб.: ВХБ-Петербург, 2005. – 336 с.
107. Иванова, Г.С. Технология программирования: Учебник для вузов. / Г.С. Иванова. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 320 с.
108. Седжвик, Р. Фундаментальные алгоритмы на C++ / Р. Седжвик. – М.: ДиаСофт, 2002.
109. Страуструп, Б. Язык программирования C++. Специальное издание / Б. Страуструп. – М.: Бином-Пресс, 2007. – 1104 с.
110. Шилдт, Г. Полный справочник по C++ / Г. Шилдт. – М.: Вильямс, 2011. – 800 с.
111. Бланшет, Ж. QT 4: программирование GUI на C++ / Ж. Бланшет, М. Саммерфилд. – М.: Кудиц-Пресс, 2007.
112. Шлее М. Qt 4.5. Профессиональное программирование на C++ / М. Шлее. – СПб.: БХВ-Петербург, 2010. – 896 с.
113. Чеботарев, А. Библиотека Qt 4. Создание прикладных приложений в среде Linux / А. Чеботарев. – М.: Диалектика, 2006. – 256 с.
114. Земсков, Ю.В. Qt 4 на примерах / Ю.В. Земсков. – СПб.: БХВ-Петербург, 2008. – 608 с.
115. Хорохорин, М.А. Моделирование и оценка нечетких сетевых информационных систем с нахождением оптимальных путей в направленных нечетких графах / М.А. Хорохорин, А.А. Долгов // Проблемы управления, информатизации и моделирования. - М: НПЦ «Модуль», 2012. - №3 – С.51-54.
116. Операционная система Ubuntu Linux / Б. Мако Хилл, Д. Бейкон, К. Бургер, Д. Джесси, И. Крстик. – М.: Триумф, 2008. – 384 с.
117. Хорохорин, М.А. Сравнительный анализ нейросетевых моделей оценки живучести сетевых структур, построенных с помощью Matlab и Qt / М.А. Хорохорин, А.А. Долгов // Проблемы управления, информатизации и

моделирования. - М: НПЦ «Модуль», 2012. - №3 – С.42-50.

118. Медведев, В.С. Нейронные сети. MATLAB 6 / В.С. Медведев, В.Г. Потемкин. – М.: ДИАЛОГ-МИФИ, 2002. – 496 с.

119. Хорохорин, М.А. Оценка живучести нечетких сетевых информационных систем / М.А. Хорохорин // Вестник Воронежского института высоких технологий. -2010. -№7. –С. 255-258.

120. Свидетельство о государственной регистрации программы для ЭВМ 2011610917 Российская Федерация. Программа: Алгоритм оценки живучести сетевых информационных систем с использованием полинома Татта / А.А. Долгов, М.А. Хорохорин, Ю.Ю. Громов; № 2010617463; заявление 29.11.10;зарегистрировано в реестре программ для ЭВМ 24.01.11.

121. Свидетельство о государственной регистрации программы для ЭВМ 2011610919 Российская Федерация. Программа: Алгоритм оценки живучести нечетких сетевых информационных систем с использованием полинома Татта / А.А. Долгов, М.А. Хорохорин, Ю.Ю. Громов; № 2010617465; заявление 29.11.10;зарегистрировано в реестре программ для ЭВМ 24.01.11.

122. Свидетельство о государственной регистрации программы для ЭВМ 2011614063 Российская Федерация. Программа: Аудит сетевых систем с использованием полиномиальной оценки живучести / А.А. Долгов, М.А. Хорохорин; № 2011612184; заявление 04.04.11; зарегистрировано в реестре программ для ЭВМ 25.15.11.

123. Свидетельство о государственной регистрации программы для ЭВМ 2011610916 Российская Федерация. Программа: Алгоритм оценки живучести нечетких сетевых информационных систем с использованием нейронных сетей / А.А. Долгов, Ю.Ю. Громов, М.А. Хорохорин; № 2010617462; заявление 29.11.10;зарегистрировано в реестре программ для ЭВМ 24.01.2011.

124. Свидетельство о государственной регистрации программы для ЭВМ 2011610918 Российская Федерация. Программа: Алгоритм оценки

живучести сетевых информационных систем с использованием нейронных сетей / А.А.Долгов, Ю.Ю. Громов, М.А. Хорохорин; № 2010617464; заявление 29.11.10; зарегистрировано в реестре программ для ЭВМ 24.01.2011.

125. Свидетельство о государственной регистрации программы для ЭВМ 2011614062 Российская Федерация. Программа: Аудит сетевых систем с использованием искусственных нейронных сетей для оценки живучести / А.А.Долгов, М.А. Хорохорин; № 2011612183; заявление 01.04.11; зарегистрировано в реестре программ для ЭВМ 25.05.2011.

126. Боженюк, А.В. Нахождение живучести нечетких транспортных сетей с применением геоинформационных систем / А.В. Боженюк, И.Н. Розенберг, Д.Н.Ястребинская. – М.: Научный мир, 2012. – 176 с.

127. Хорохорин, М.А. Применение интеллектуальной экспертной системы для задачи выбора оптимальной схемы численного решения уравнения переноса / М.А. Хорохорин, Д.В. Сулов, Ю.В.Кулаков, А.А.Долгов // Приборы и системы. Управление, контроль, диагностика. – 2012. - №3 – С.51-54.

128. Deeter, D.L. Economic design of reliable networks / D.L. Deeter, A.E. Smith // NE Transactions. – 1998. – V. 30.

129. Konak, A. An improved general upperbound for all-terminal network reliability / A. Konak, A.E. Smith // NE Transactions. – 1998. – V. 30.

130. Funahashi, K. On the approximate realization of continuous mappings by neural networks / K. Funahashi // Neural Networks. – 1989. – V. 2. – P. 183 – 192.

131. Konak, A. A general upperbound for all-terminal network reliability and its uses / A. Konak, A.E. Smith // Proceedings of the Industrial Engineering Research Conference. – Banff, Canada, 1998. – May. – CD-Rom format.

132. Cheng, B. Neural networks: A review from a statistical perspective / B. Cheng, D.M. Titterington // Statistical Science. – 1994. – V. 9. – P. 2 – 54.

133. Gusfield, D. Optimal mixed graph augmentation / D. Gusfield //

SIAM Journal on Computing. – 1987. – V. 16, № 4. – P. 599 – 613.

134. Cottrell, G.W. Image compressions by backpropagation: An example of extensional programming / G.W Cottrell, P. Munro, D. Zipser // Advances in cognitive science. – NJ: Ablex, 1987. – V. 3.

135. Deeter, D.L. Heuristic optimization of network design considering all-terminal reliability / D.L. Deeter, A. E. Smith // Proceedings of the Reliability and Maintainability Symposium. – 1997. – P. 194 – 199.

136. Jan, R.-H. Topological optimization of a communication network subject to a reliability constraint / R.-H. Jan, F.-J. Hwang, S.-T. Chen // IEEE Transactions on Reliability. – 1993. – V. 42. – P. 63 – 70.

137. Cheng M.X., Li Y., Du D.-Z. Combinatorial Optimization in Communication Networks. Springer, 2006. – 685 p.

138. Provan, J.S. The Complexity of Reliability Computations in Planar and Acyclic Graphs / J.S. Provan // SIAM Journal on Computing. – 1986. – V. 15, № 3. – P. 694 – 702.

139. Tani, S. An Extended Framework of Ordered Binary Decision Diagrams for Combinatorial Graph Problems / S. Tani. – Master's thesis, University of Tokyo, 1995.

140. Tutte, W.T. A Contribution to the Theory of Chromatic Polynomials / W.T. Tutte // Canadian Journal of Mathematics. – 1954. – V. 6. – P. 80 – 91.

141. Sekine, K. Computing the Tutte Polynomial of a Graph of Moderate Size / K. Sekine, H. Imai, S. Tani // Proceedings of the 6th International Symposium on Algorithms and Computation (ISAAC'95), Lecture Notes in Computer Science. – 1995. – V. 1004. – P. 224 – 233.

142. Tutte, W.T. A Contribution to the Theory of Chromatic Polynomials / W.T. Tutte // Canadian Journal of Mathematics. – 1954. – Vol. 6. – P. 80-91.

143. Zimmermann, H.J. Fuzzy set theory and its applications (2nd edition) / H.J.Zimmermann. – Boston/Dordrecht/London: Kluwer Academic Publishers, 1991. – 435p.

144. Pedrycz, W. Fuzzy Control and Fuzzy Systems. / W. Pedrycz. N. Y.:

John Wiley and Sons, 1993.

145. Geman, S. Neural networks and the bias/variance dilemma / S. Geman, E. Bienenstock, R. Doursat // *Neural Computation*. – 1992. – V. 4. – P. 1 – 58.

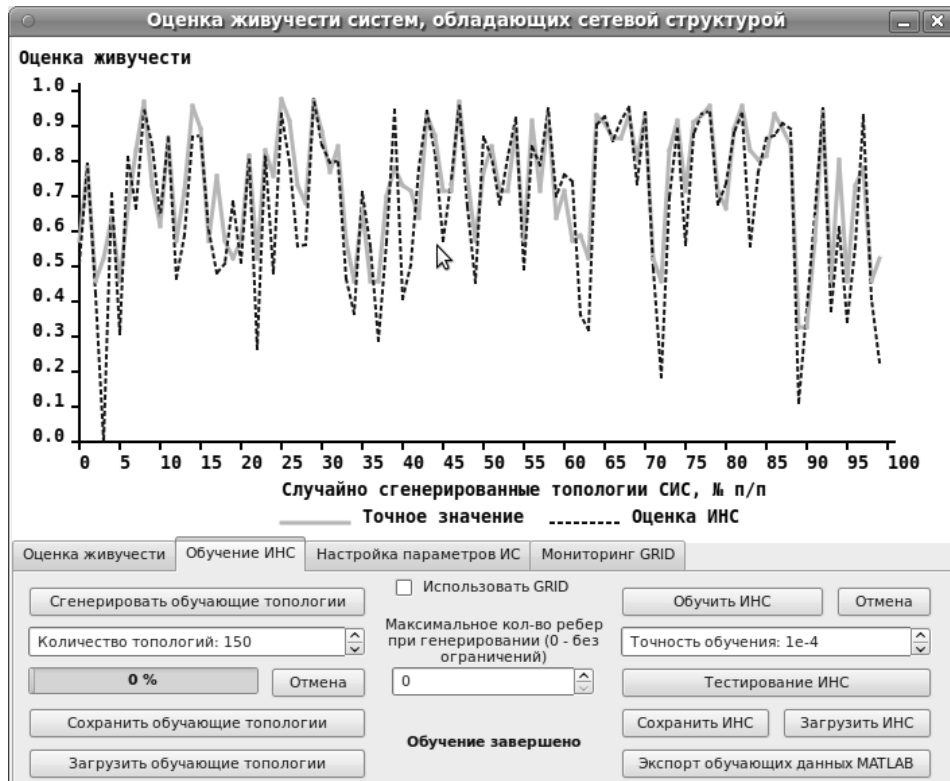
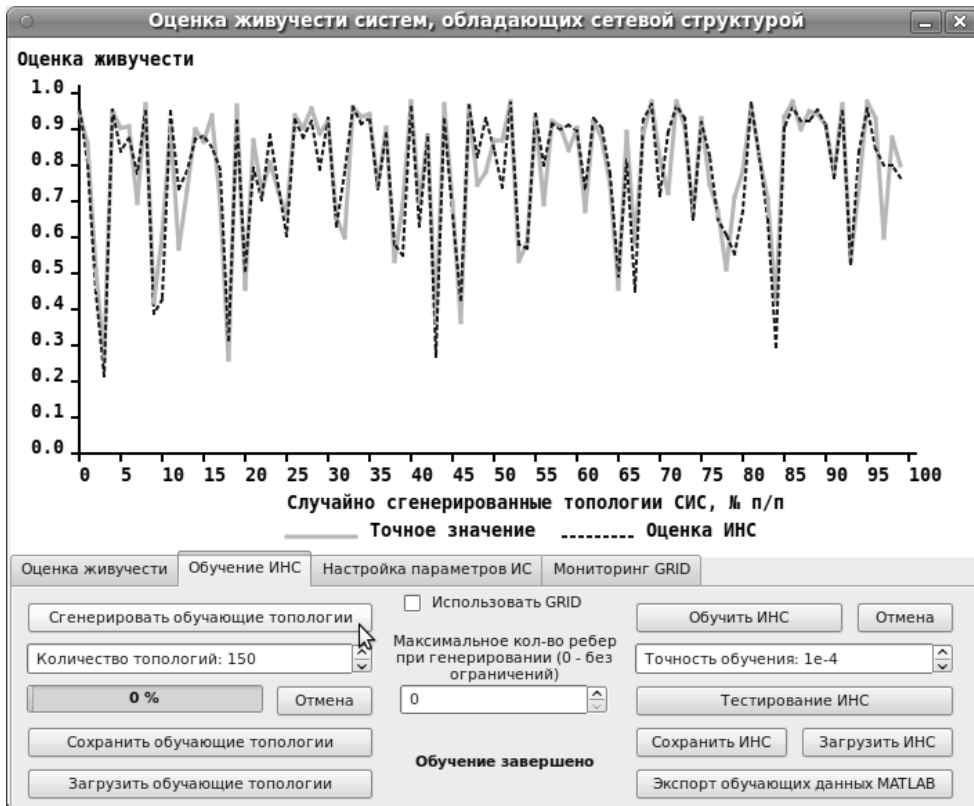
146. Burr, D.J. Experiments with a connectionist text reader. In *Proceedings of the First International on Neural Networks / D.J. Burr*. – CA: SOS Printing, 1987. – V. 4. – P. 24-717.

147. Fuller, R. *Neural Fuzzy Systems / R. Fuller*. – Publishing House: AboAkademi University, 1995.

148. Pedrycz, W. An identification algorithm in fuzzy relational systems. / W. Pedrycz. – *Fuzzy Sets and Systems*. 1984. V. 13.

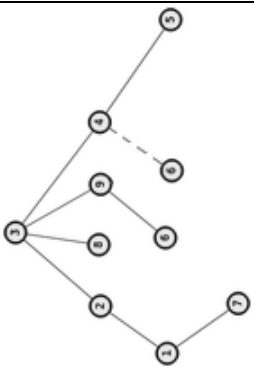
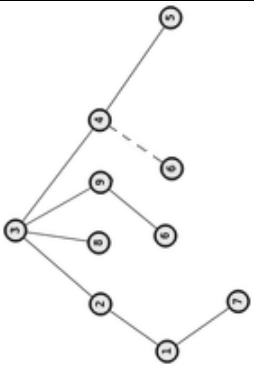
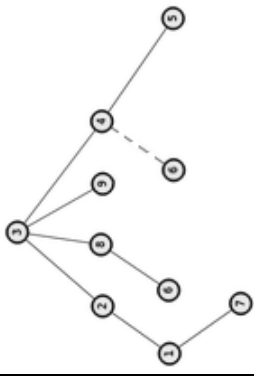
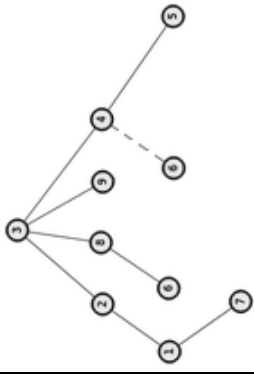
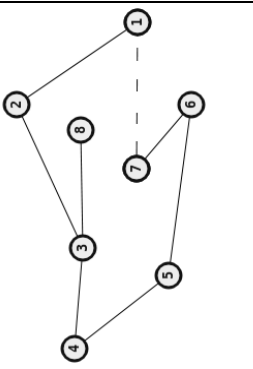
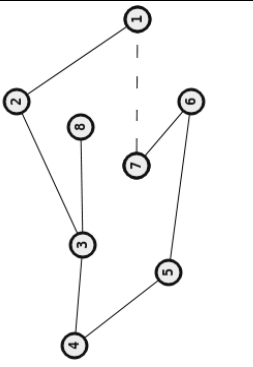
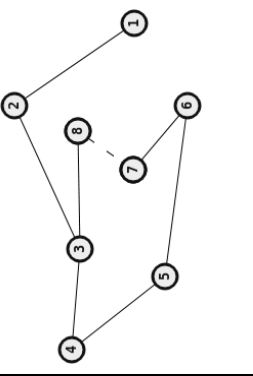
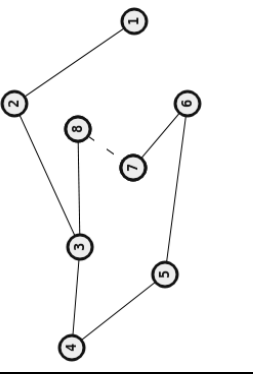
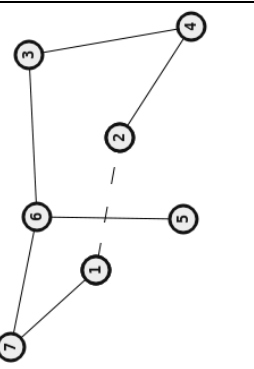
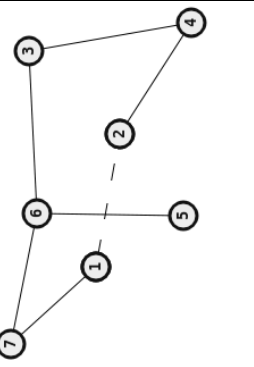
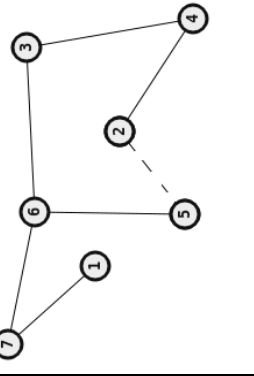
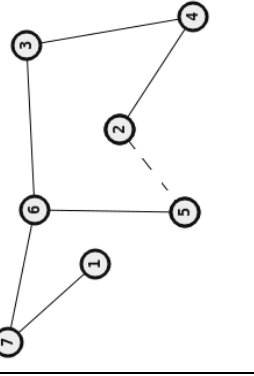
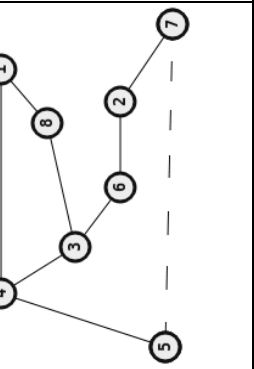
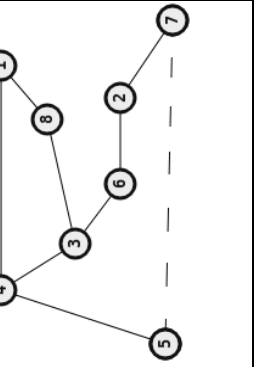
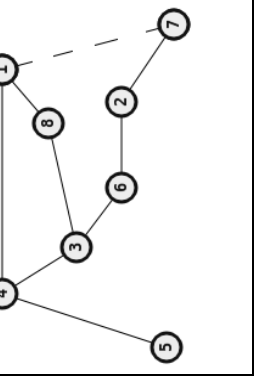
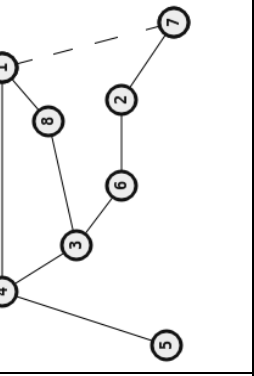
ПРИЛОЖЕНИЕ А

Тестирование обученной нечеткой нейронной сети в разработанном программном обеспечении



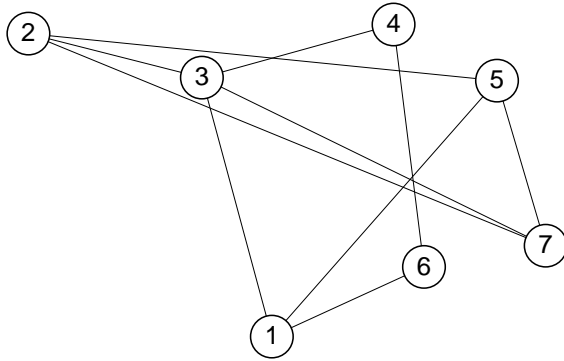
ПРИЛОЖЕНИЕ Б

Сравнительный анализ рекомендаций

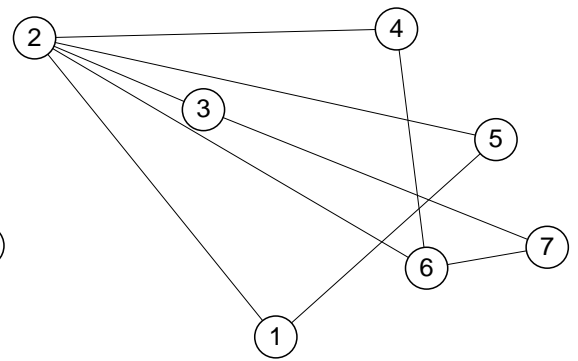
Новая топология 1 (эксперты)	Новая топология 1 (ПО)	Новая топология 2 (эксперты)	Новая топология 2 (ПО)
			
			
			
			

ПРИЛОЖЕНИЕ В

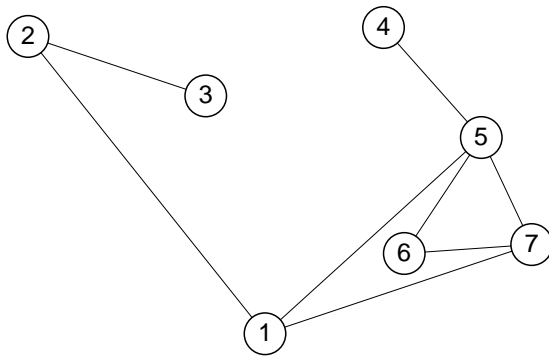
Примеры сгенерированных топологий информационных структур



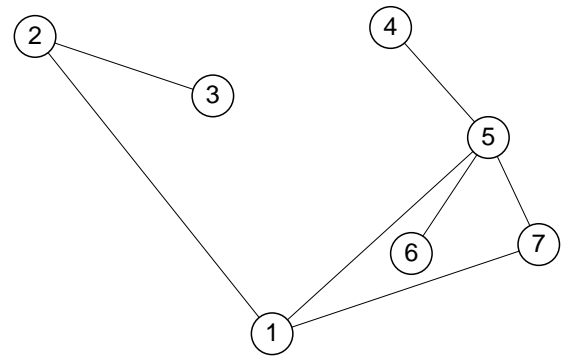
1-я топология



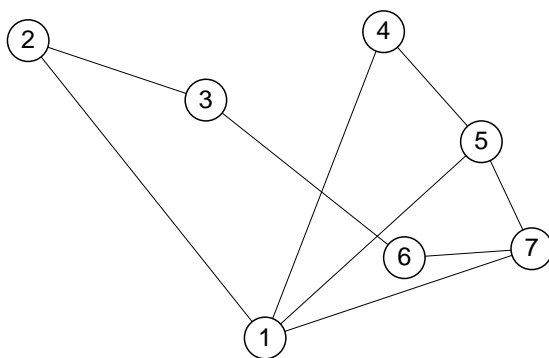
30-я топология



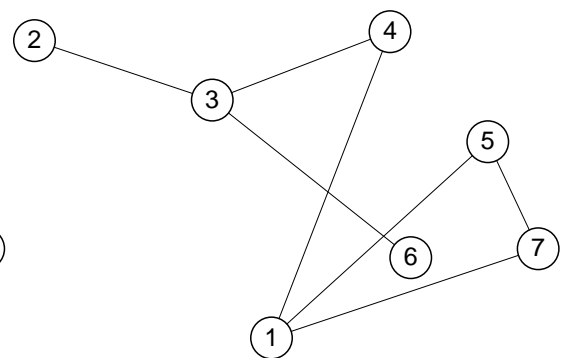
60-я топология



75-я топология



95-я топология



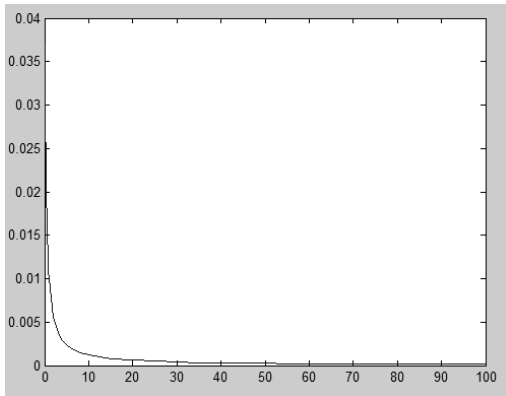
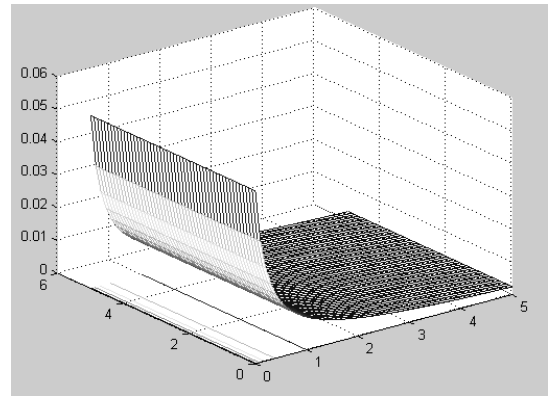
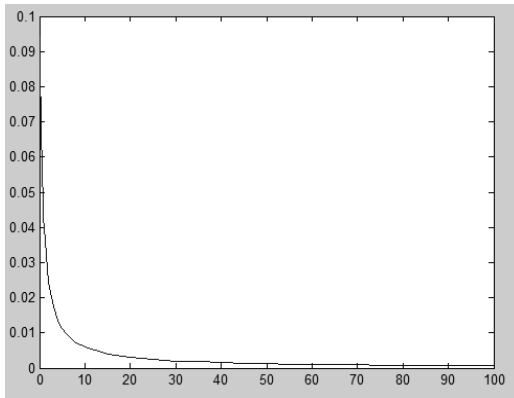
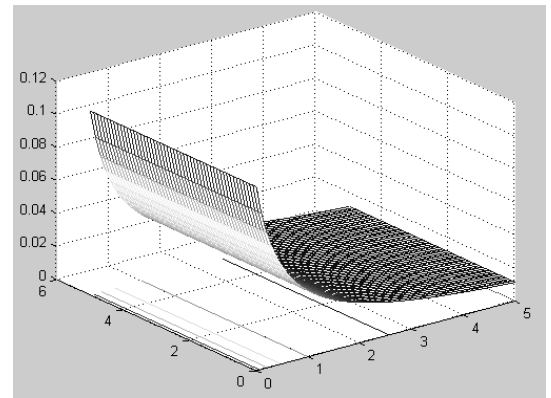
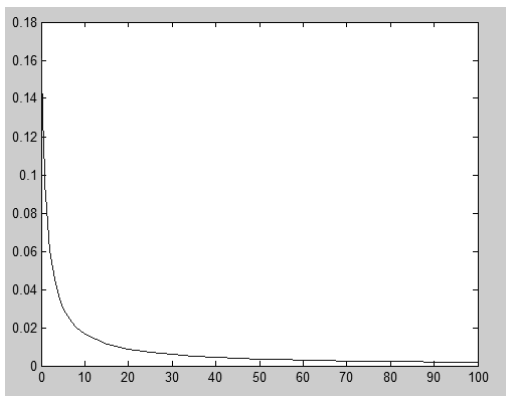
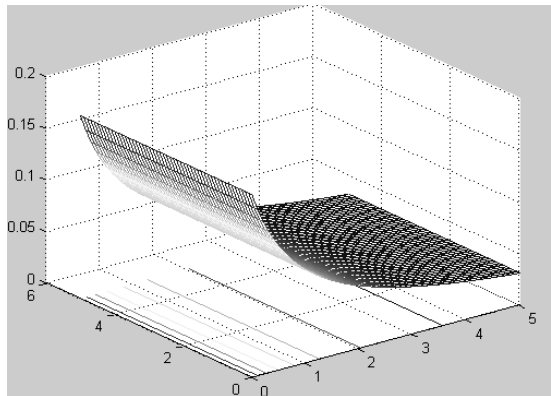
99-я топология

ПРИЛОЖЕНИЕ Г

Таблица истинности для различных видов импликаций

Вид импликации	Определение	$\mu_A(x)$	$\mu_A(y)$	$\mu_A(x) \rightarrow \mu_A(y)$
Стандартная четкая	$\mu_R(x, y) =$ $= \begin{cases} 1, \text{ если } \mu_A(x) \leq \mu_B(y), \\ 0, \text{ если } \mu_A(x) > \mu_B(y) \end{cases}$	1	1	1
		1	0	0
		0	1	1
		0	0	1
Классическая нечеткая (Клине-Даэнса)	$\mu_R(x, y) =$ $= \max\{1 - \mu_A(x), \mu_B(y)\},$ при $\mu_A(x) \geq \mu_B(y)$	1	1	1
		1	0	0
		0	1	1
		0	0	1
Заде	$\mu_R(x, y) = \max\{\min\{\mu_A(x), \mu_B(y)\}, 1 - \mu_A(x)\}$	1	1	1
		1	0	0
		0	1	1
		0	0	1
Лукашевича	$\mu_R(x, y) = \min\{1, 1 - \mu_A(x) + \mu_B(y)\}$	1	1	1
		1	0	0
		0	1	1
		0	0	1
Мамдани	$\mu_R(x, y) = \min\{\mu_A(x), \mu_B(y)\}$	1	1	1
		1	0	0
		0	1	0
		0	0	0
Ларсена	$\mu_R(x, y) = \mu_A(x) \cdot \mu_B(y)$	1	1	1
		1	0	0
		0	1	0
		0	0	0
Используемая в работе T-импликация	$\mu_R(x, y) = \mu_A(x) \cdot \mu_B(y) /$ $/(\gamma + (1 - \gamma)(\mu_A(x) + \mu_B(y) - \mu_A(x) \cdot \mu_B(y))), \gamma > 0$	1	1	1
		1	0	0
		0	1	0
		0	0	0

ПРИЛОЖЕНИЕ Д

Графики зависимости функции $\mu(a,b)$ от γ График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.1$ и $\mu(b)=0.1$ График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.1$ и $\mu(b)=0.1$ График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.2$ и $\mu(b)=0.2$ График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.2$ и $\mu(b)=0.2$ График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.3$ и $\mu(b)=0.3$ График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.3$ и $\mu(b)=0.3$

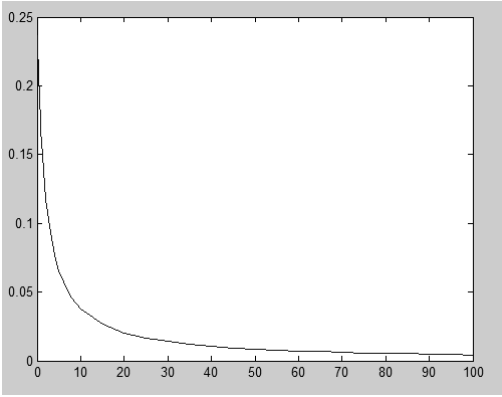


График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.4$ и $\mu(b)=0.4$

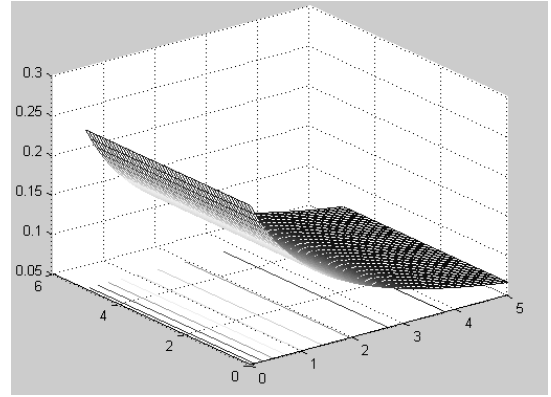


График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.4$ и $\mu(b)=0.4$

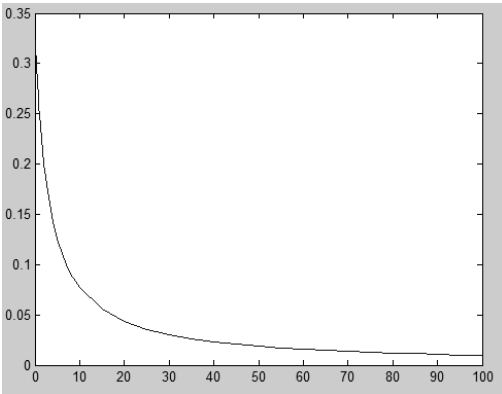


График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.5$ и $\mu(b)=0.5$

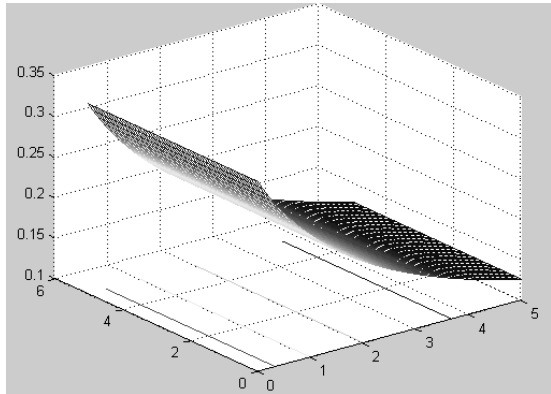


График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.5$ и $\mu(b)=0.5$

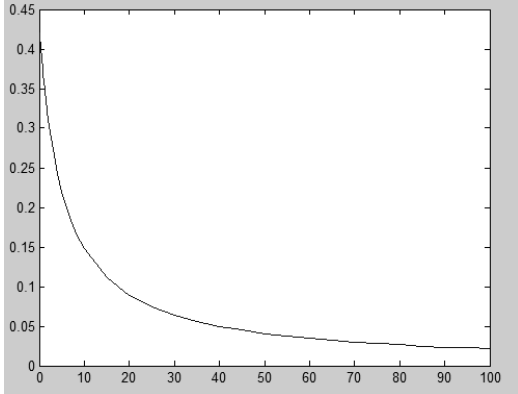


График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.6$ и $\mu(b)=0.6$

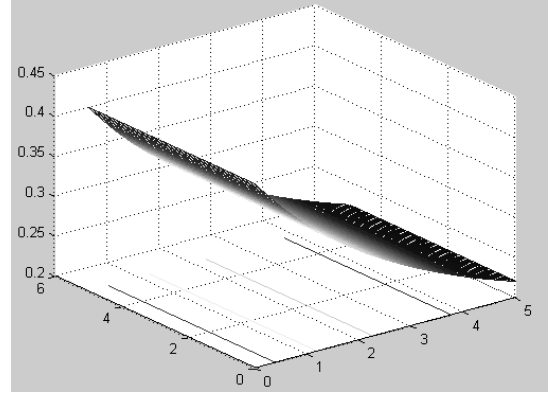


График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.6$ и $\mu(b)=0.6$

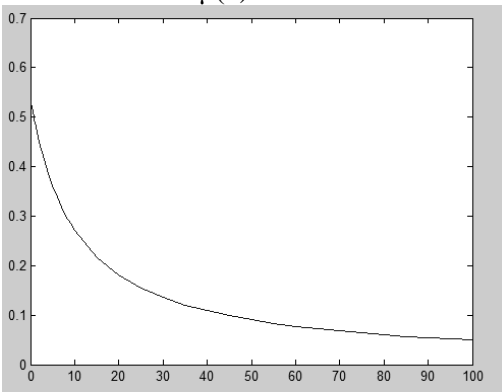


График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.7$ и $\mu(b)=0.7$

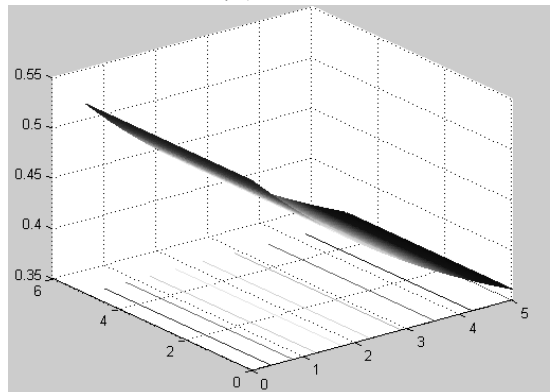


График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.7$ и $\mu(b)=0.7$

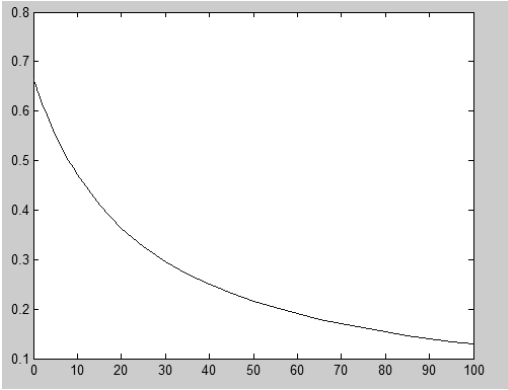


График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.8$ и $\mu(b)=0.8$

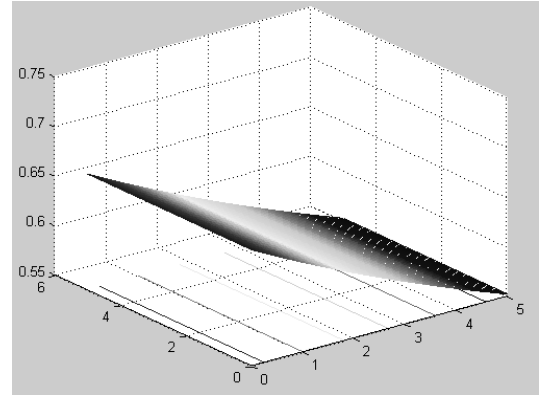


График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.8$ и $\mu(b)=0.8$

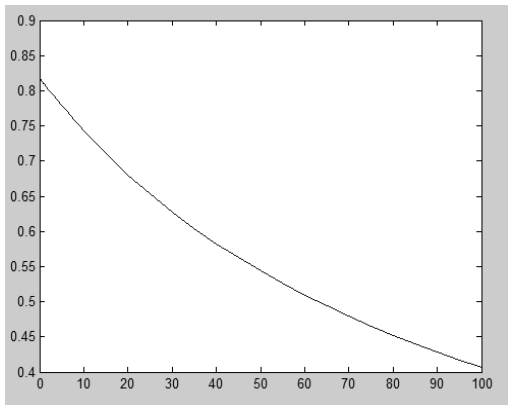


График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.9$ и $\mu(b)=0.9$

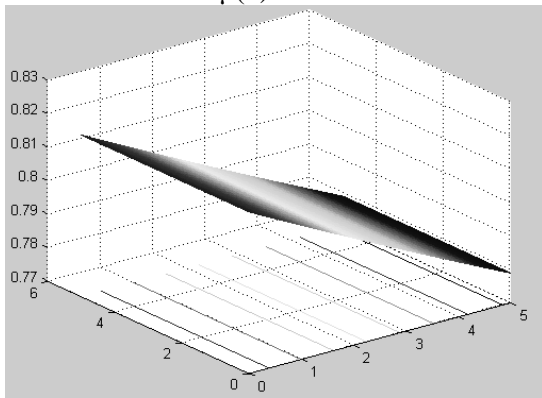


График зависимости $\mu(a,b)$ от γ при $\mu(a)=0.9$ и $\mu(b)=0.9$

ПРИЛОЖЕНИЕ Ж

Копии свидетельств о государственной регистрации программы
для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2011614063

Аудит сетевых систем с использованием полиномиальной
оценки живучести (AuditLanSystemsPolynomial)

Правообладатель(ли): *Долгов Артем Анатольевич (RU),
Хорохорин Михаил Александрович (RU)*

Автор(ы): *Долгов Артем Анатольевич,
Хорохорин Михаил Александрович (RU)*

Заявка № 2011612184

Дата поступления 4 апреля 2011 г.

Зарегистрировано в Реестре программ для ЭВМ
25 мая 2011 г.

Руководитель Федеральной службы по интеллектуальной
собственности, патентам и товарным знакам

Б.П. Симонов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

**СВИДЕТЕЛЬСТВО**

о государственной регистрации программы для ЭВМ

№ 2011614062**Аудит сетевых систем с использованием искусственных
нейронных сетей для оценки живучести (AuditLanSystemsANN)****Правообладатель(ли): Долгов Артем Анатольевич (RU),
Хорохорин Михаил Александрович (RU)****Автор(ы): Долгов Артем Анатольевич,
Хорохорин Михаил Александрович (RU)**

Заявка № 2011612183

Дата поступления 1 апреля 2011 г.

Зарегистрировано в Реестре программ для ЭВМ

25 мая 2011 г.

**Руководитель Федеральной службы по интеллектуальной
собственности, патентам и товарным знакам**

Б.П. Симонов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

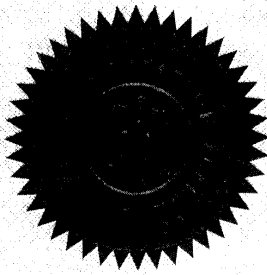
**СВИДЕТЕЛЬСТВО**

о государственной регистрации программы для ЭВМ

№ 2011610919**Алгоритм оценки живучести нечетких сетевых информационных систем с использованием полинома Татта (АОЖНСИСТ)**

Правообладатель(ли): *Государственное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет» (RU)*

Автор(ы): *Хорохорин Михаил Александрович, Громов Юрий Юрьевич, Долгов Артем Анатольевич (RU)*

Заявка № **2010617465**Дата поступления **29 ноября 2010 г.**Зарегистрировано в Реестре программ для ЭВМ
24 января 2011 г.

Руководитель Федеральной службы по интеллектуальной собственности, патентам и товарным знакам

Б.П. Симонов

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2011610918

Алгоритм оценки живучести сетевых информационных систем с использованием нейронных сетей (АОЖСИСНС)

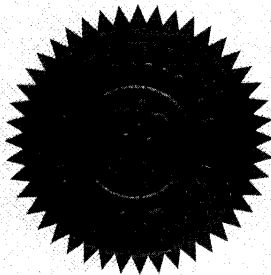
Правообладатель(ли): *Государственное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет» (RU)*

Автор(ы): *Долгов Артем Анатольевич, Громов Юрий Юрьевич, Хорохорин Михаил Александрович (RU)*

Заявка № 2010617464

Дата поступления 29 ноября 2010 г.

Зарегистрировано в Реестре программ для ЭВМ
24 января 2011 г.



Руководитель Федеральной службы по интеллектуальной собственности, патентам и товарным знакам

Б.П. Симонов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

**СВИДЕТЕЛЬСТВО**

о государственной регистрации программы для ЭВМ

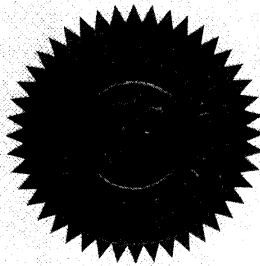
№ 2011610917**Алгоритм оценки живучести сетевых информационных систем с использованием полинома Татта (АОЖСИСТ)**

Правообладатель(ли): *Государственное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет» (RU)*

Автор(ы): *Долгов Артем Анатольевич, Громов Юрий Юрьевич, Хорохорин Михаил Александрович (RU)*

Заявка № 2010617463

Дата поступления 29 ноября 2010 г.

Зарегистрировано в Реестре программ для ЭВМ
24 января 2011 г.

Руководитель Федеральной службы по интеллектуальной собственности, патентам и товарным знакам

Б.П. Симонов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

**СВИДЕТЕЛЬСТВО**

о государственной регистрации программы для ЭВМ

№ 2011610916**Алгоритм оценки живучести нечетких сетевых информационных систем с использованием нейронных сетей (АОЖНСИСНС)**

Правообладатель(ли): *Государственное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет» (RU)*

Автор(ы): *Хорохорин Михаил Александрович, Громов Юрий Юрьевич, Долгов Артем Анатольевич (RU)*

Заявка № 2010617462

Дата поступления 29 ноября 2010 г.

Зарегистрировано в Реестре программ для ЭВМ
24 января 2011 г.

Руководитель Федеральной службы по интеллектуальной собственности, патентам и товарным знакам

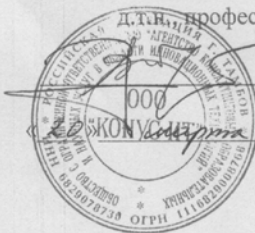
Б.П. Симонов

ПРИЛОЖЕНИЕ 3

Акты об использовании результатов исследования

«УТВЕРЖДАЮ»
 Генеральный директор ООО
 «Агентство консалтинговых,
 образовательных и научных услуг в
 области инновационных
 технологий»

Д.Т.Н. профессор
 Дидрих В.Е.
 2014 г.




АКТ

**об использовании результатов диссертационной работы
 ХОРОХОРИНА МИХАИЛА АЛЕКСАНДРОВИЧА
 на соискание ученой степени кандидата технических наук на тему:
 «Модели и алгоритмы получения оценки живучести систем с нечеткой
 информационной структурой, обеспечивающие сокращение времени расчета»**

Комиссия в составе председателя – члена совета директоров Ивановой Ольги Геннадьевны, кандидата технических наук, доцента, и члена инженера-программиста Паладьева Виктора Валерьевича составила настоящий Акт о том, что результаты диссертационной работы ХОРОХОРИНА МИХАИЛА АЛЕКСАНДРОВИЧА на соискание ученой степени кандидата технических наук, а именно предложенные алгоритм перехода от нечеткой продукционной модели к соответствующей ей нечеткой нейросетевой модели, который отличается применением Т-импликации, способного модифицировать алгоритм нечеткого вывода Мамдани и алгоритм обучения нечеткой нейронной продукционной сети, основанный на использовании предложенного квазидискретного Ньютоновского метода с немонотонной стабилизацией, в котором для решения задачи одномерной оптимизации используется модификация метода чисел Фибоначчи, использованы в НИР и ОКР, выполняемых по тематике разработки и исследования информационных структур различного предметного назначения.

Полученные программные реализации разработанных алгоритмов используются для исследования как уже функционирующих, так и разрабатываемых информационных структур, что позволяет оценивать их эффективность функционирования с целью её повышения.

Председатель комиссии:


 О.Г. Иванова

Член комиссии:


 В.В. Паладьев

«10» марта 2014 г.



«УТВЕРЖДАЮ»
Генеральный директор
ОАО «Медтехника»

Проскуряков Проскуряков В.С.

«28» марта 2014 г.

АКТ

**об использовании результатов диссертационной работы
ХОРОХОРИНА МИХАИЛА АЛЕКСАНДРОВИЧА
на соискание ученой степени кандидата технических наук на тему:
«Модели и алгоритмы получения оценки живучести систем с нечеткой
информационной структурой, обеспечивающие сокращение времени расчета»**

Комиссия в составе председателя – главного инженера Проскурякова Виктора Сергеевича и члена – заместителя главного инженера Ченчик Людмилы Николаевны составила настоящий Акт о том, что результаты диссертационной работы ХОРОХОРИНА МИХАИЛА АЛЕКСАНДРОВИЧА на соискание ученой степени кандидата технических наук, а именно программные реализации:

- алгоритма перехода от нечеткой продукционной модели к соответствующей ей нечеткой нейросетевой модели, который отличается применением Т-импликации, что позволило модифицировать алгоритм нечеткого вывода Мамдани;
- алгоритма обучения нечеткой нейронной продукционной сети, основанный на использовании предложенного квазидискретного Ньютоновского метода с немонотонной стабилизацией, в котором для решения задачи одномерной оптимизации используется модификация метода чисел Фибоначчи

использованы при выполнении работ по тематике синтеза информационных систем и структур, определения параметров и оптимизации работы в учреждениях здравоохранения на территории Тамбовской области, что позволило на 20% сократить временные затраты.

Председатель комиссии:

Проскуряков
Ченчик

В.С. Проскуряков

Член комиссии:

Л.Н. Ченчик

«28» марта 2014 г.

«УТВЕРЖДАЮ»
 Начальник войсковой части 5670
 « 23 » апреля 2014 г.




АКТ

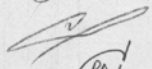
**об использовании результатов диссертационной работы
 ХОРОХОРИНА МИХАИЛА АЛЕКСАНДРОВИЧА
 на соискание ученой степени кандидата технических наук на тему:
 «Модели и алгоритмы получения оценки живучести систем с нечеткой
 информационной структурой, обеспечивающие сокращение времени расчета»**

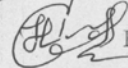
Комиссия в составе председателя – начальника учебного отдела, заместителя начальника центра по учебной работе Власова Ю.А., членов комиссии – начальника отдела измерений и объективного контроля Каданцева С.М., старшего помощника начальника учебного отдела Ильичева Н.С. составила настоящий Акт о том, что результаты диссертационной работы ХОРОХОРИНА МИХАИЛА АЛЕКСАНДРОВИЧА на соискание ученой степени кандидата технических наук – нечеткая продукционная модель оценки живучести систем с нечеткой информационной структурой, отличающиеся применением нечетких параметров, характеризующими как структуру, так и параметры информационных каналов, для свертки которых используются однопараметрические Т-нормы, параметры которых уточняются вследствие решения оптимизационной задачи, для решения которой используется разработанный алгоритм, основанный на методе чисел Фибоначчи, позволяющая получить оценку живучести систем с нечеткой информационной структурой в условиях негативных внешних воздействий; алгоритм перехода от нечеткой продукционной модели к соответствующей ей нечеткой нейросетевой модели и алгоритм обучения нечеткой нейронной сети, использующий разработанный квазидискретный Ньютоновский метод с немонотонной стабилизацией, позволяющие сократить количество вычисляемых операций и повысить эффективность расчета, используются в процессе организации подготовки и переподготовки курсантов и слушателей, а также при выполнении научно-исследовательских работ.

Председатель комиссии:

 Ю.А. Власов

Члены комиссии:

 С.М. Каданцев

 Н.С. Ильичев

« 23 » апреля 2014 г.

«УТВЕРЖДАЮ»
Первый проректор ФГБОУ ВПО
«Тамбовский государственный
технический университет»
д.п.н., профессор



[Signature] Молоткова Н.В.

август 2014 г.

АКТ

об использовании результатов диссертационной работы
ХОРОХОРИНА МИХАИЛА АЛЕКСАНДРОВИЧА
на соискание ученой степени кандидата технических наук на тему:
«Модели и алгоритмы получения оценки живучести систем с нечеткой
информационной структурой, обеспечивающие сокращение времени расчета»

Комиссия кафедры «Информационные системы и защита информации» в составе председателя – заведующей кафедрой Ивановой Ольги Геннадьевны, кандидата технических наук, доцента, и членов комиссии – доцента кафедры Минина Юрия Викторовича, кандидата технических наук, доцента; доцента кафедры Ивановского Михаила Андреевича, кандидата технических наук, доцента, составила настоящий Акт о том, что результаты диссертационной работы ХОРОХОРИНА МИХАИЛА АЛЕКСАНДРОВИЧА на соискание ученой степени кандидата технических наук используются в учебном процессе на кафедре «Информационные системы и защита информации» ФГБОУ ВПО «Тамбовский государственный технический университет» при непосредственном участии автора диссертационных исследований в разработке учебно-методических материалов лекций, лабораторных работ и обучающих программных комплексов по следующим дисциплинам кафедры: «Теория информационных процессов и систем», «Моделирование систем», «Моделирование сетевых процессов», «Проектирование информационных систем», «Моделирование информационных систем».

Результаты данного диссертационного исследования используются для обучения студентов специальности 230201 и направления 230400 «Информационные системы и технологии».

Председатель комиссии:

[Signature] О.Г. Иванова

Члены комиссии:

[Signature] Ю.В. Минин

[Signature] М.А. Ивановский

«21» *август* 2014 г.