

Министерство образования и науки Российской Федерации
ФГБОУ ВПО «Бурятский государственный университет»
Институт математики и информатики

На правах рукописи



Хандаров Фёдор Владимирович

**МЕТОД ОДНОВРЕМЕННОГО
СТРУКТУРНО-ПАРАМЕТРИЧЕСКОГО СИНТЕЗА
МНОГОСЛОЙНЫХ ПЕРСЕПТРОНОВ**

Специальность 05.13.18 – Математическое моделирование,
численные методы и комплексы программ

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
кандидат физико-математических наук,
доцент Т. Г. Дармаев

Улан-Удэ
2014

Оглавление

ВВЕДЕНИЕ.....	3
ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ И ОБЩАЯ СХЕМА СТРУКТУРНО-ПАРАМЕТРИЧЕСКОГО СИНТЕЗА.....	14
1.1 Постановка задачи структурно-параметрического синтеза.....	14
1.1.1 Постановка задачи обучения сети с фиксированной структурой	14
1.1.2 Выбор функции активации.....	17
1.1.3 Кодирование точек пространства поиска при различающихся структурах	20
1.1.4 Постановка задачи обучения сетей с различающимися структурами и общая схема СПС ...	22
1.2 Алгоритмическое наполнение общей схемы СПС	26
1.2.1 Методы модификации топологии.....	26
1.2.2 Методы параметрического улучшения	35
1.3 Выводы по главе.....	30
ГЛАВА 2. МЕТОД СТРУКТУРНО-ПАРАМЕТРИЧЕСКОГО СИНТЕЗА МНОГОСЛОЙНЫХ ПЕРСЕПТРОНОВ	31
2.1 Стратегия модификации топологии (синтез структуры) сети.....	31
2.2 Алгоритм нелокального параметрического улучшения.....	35
2.2.1 Сравнительный анализ методов ГСП.....	35
2.2.2 Гибридный метод ГСП на основе комбинации поиска с запретами и дифференциальной эволюции.....	49
2.3 Метод структурно-параметрического синтеза	60
2.4 Тестирование метода структурно-параметрического синтеза.....	68
2.4.1 Выбор тестовых задач.....	68
2.4.2 Описание задач.....	71
2.4.3 Результаты решения задач.....	74
2.5 Выводы по главе.....	82
ГЛАВА 3. ПРОГРАММНЫЙ КОМПЛЕКС И РЕШЕНИЕ ПРАКТИЧЕСКИХ ЗАДАЧ.....	84
3.1 Описание программного комплекса.....	84
3.2 Решение практических задач	97
3.2.1 Прогнозирование налоговых поступлений (по данным Республики Бурятия)	97
3.2.2 Прогнозирование результатов сдачи Единого государственного экзамена (ЕГЭ).....	104
3.3 Выводы по главе.....	110
ЗАКЛЮЧЕНИЕ	111
СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ	112
СПИСОК ЛИТЕРАТУРЫ.....	113
ПРИЛОЖЕНИЕ А	127
ПРИЛОЖЕНИЕ Б.....	129
ПРИЛОЖЕНИЕ В	131

ВВЕДЕНИЕ

Для построения математических моделей на основе эмпирической информации (по измеряемым данным наблюдений над объектом) распространенным подходом является использование *искусственных нейронных сетей (ИНС)* [19; 32]. ИНС представляют собой универсальный инструмент моделирования, они используются в качестве систем представления знаний, на практике играющих роль управляющего компонента либо модуля принятия решений [32].

Искусственные нейронные сети – передаточные модели, построенные по принципу организации и функционирования сетей нервных клеток живого организма. ИНС представляют собой упорядоченные множества обрабатывающих узлов, называемых *нейронами*, соединенных взвешенными связями, по которым передаются сигналы между узлами (подобная структура может быть описана ориентированным графом, вершины которого описывают нейроны, а ориентированные ребра – межнейронные связи).

Пусть некоторая рассматриваемая ИНС состоит из n нейронов, которые являются упрощенными моделями естественных нейронов, что выражается в виде сопоставления каждому i -тому нейрону сети некоторой передаточной (активационной) функции φ_i , принимающей аргумент вида

$$y_i = \varphi_i \left(\sum_{j=0}^n w_{ij} x_j \right), i = \overline{0, n}, \quad (1)$$

где y_i является выходным сигналом i -того нейрона, рассчитываемым на основе значений x_j ($j = \overline{0, n}$) сигналов, приходящих от остальных нейронов и w_{ij} – весов, приписанных межнейронным связям. Заметим, что индексация x в формуле (1) начинается с 0, хотя число нейронов – n ($x_0 = 1$ в этом случае играет роль смещения).

Все нейроны, таким образом, подразделяют на внутренние (скрытые), входные и выходные – в зависимости от источника и приемника сигналов. Входные нейроны получают свои сигналы из внешней среды, а выходные – отдают сигналы во внешнюю среду. Количество скрытых нейронов n и расположение связей между нейронами сети определяет топологию сети (структуру сети).

Процесс моделирования с использованием ИНС сводится к *выбору топологии* сети (структурной идентификации) и к дальнейшему подбору весовых коэффициентов (параметрической идентификации) при заданной топологии. В принятой терминологии задача подбора весовых коэффициентов известна как *задача обучения* сети.

На практике выделяют различные виды ИНС в зависимости от рассматриваемых в совокупности характеристик: структуры расположения нейронов [90; 91; 107; 108; 114], направления передачи сигналов (сети прямого распространения, рекуррентные сети и др. [63; 80; 109]), используемых активационных функций (радиально-базисные функции, сигмоидальные функции и др. [120; 133]), способа обучения (с учителем, без учителя, с подкреплением [24; 45]) и прочих характеристик [8].

На данный момент одним из наиболее исследованных классов ИНС являются *многослойные перцептроны (МП)* – сети с прямым распространением сигнала (от входов к выходам) со слоистой топологией нейронов и активационными функциями сигмоидального типа (гиперболический тангенс, сигмоидальная функция, функция Гаусса и др.), реализующие обучение с учителем. Математическое обоснование использования МП восходит к тринадцатой проблеме Гильберта [33] и содержится в теоремах Стоуна-Вейерштрасса [44], А. Н. Колмогорова и В. И. Арнольда [2; 26; 27]; работах R. Hecht-Nielsen [98], G. Cybenko [69], В. Крейнвича [111], А. Н. Горбаня [13; 89] и др. – см., например, обзор А. Pinkus [132], общим итогом которого является доказательство возможности аппроксимации с помощью МП любой непрерывной функции нескольких переменных при нелинейных функциях активации.

Современные исследования [8] показывают, что одной из наиболее структурно простых (по признаку числа нейронов) разновидностей МП является сеть *с полными перекрестными связями*, т. е. персептрон, обладающий связями между нейронами одного слоя: такая структура значительно проще структуры *с последовательными связями* при условии, что обе структуры реализуют одинаковую конфигурацию разделяющих поверхностей в пространстве признаков. А. И. Галушкин также отмечает, что в плане простоты можно выделять и другие многослойные персептроны *с перекрестными связями произвольной структуры* [8]. В нашем исследовании число нейронов сети является фактором, определяющим размерность оптимизационных задач, поэтому наиболее удачным можно считать использование такой разновидности МП с перекрестными связями произвольной структуры как сеть *с разомкнутыми перекрестными связями*, т. е. МП, в котором могут присутствовать связи между нейронами одного слоя, и они не образуют замкнутых контуров. Применение подобных структур даст значительное сокращение размерности решаемых задач.

Далее в работе под терминами «искусственная нейронная сеть», «сеть», «многослойный персептрон» в работе будут пониматься именно *многослойные персептроны с разомкнутыми перекрестными связями (МППС)*.

Процесс моделирования с использованием ИНС сводится к определению топологии сети и к дальнейшему подбору весов ее межнейронных связей, т. е. обучению сети, которое происходит до тех пор, пока отклонение между желаемым и производимым сетью откликами не достигнет требуемого уровня для обучающей выборки. Данная задача является многопараметрической задачей нелинейной оптимизации.

Для дальнейшего рассмотрения этапов моделирования (подбор топологии и обучение сети) введем далее некоторые необходимые обозначения. Считаем, что имеется выборка примеров (наблюдений), используемых для обучения сети, которая может быть представлена в виде

$$Teach = \left\{ \left(x^{(t)}, y^{(t)} \right) \right\}_{t=1, \overline{T}}, \quad (2)$$

где T – количество примеров. Данную выборку будем называть *обучающим множеством*.

При подаче того или иного примера на входы в сетях с фиксированной топологией (т. е. сетях с фиксированным количеством скрытых нейронов и расположением связей) формируется отклик на выходах – таким образом, сеть реализует аппроксимацию $y = f(x)$ некоторого отображения F вида

$$F : X \rightarrow Y, \quad (3)$$

где $X = \{x = (x_1, \dots, x_N)\}$ – пространство входных сигналов, а $Y = \{y = (y_1, \dots, y_M)\}$ – пространство соответствующих откликов. Получаемый отклик сети $\tilde{y} \in Y$ зависит от топологии сети (количества нейронов и расположения связей между ними) и значений весов связей между нейронами.

Обучение сети с учителем фактически представляет собой многопараметрическую задачу нелинейной оптимизации. Для ее постановки на множестве (2) вводится некоторая функция ошибки вида

$$e(w) = e(w, y, \tilde{y}), \quad (4)$$

выражающая различие между сформированным сетью y и желаемым y откликами, и обучение сети заключается в подборе матрицы смежности графа сети $w^* \in W$ такой, что

$$w^* = \arg \min_{w \in W} e(w, y, \tilde{y}). \quad (5)$$

Функция (4), как правило, выбирается таким образом, чтобы задача обучения сети на некотором обучающем множестве являлась задачей минимизации функции суммарной ошибки по всем примерам обучающего множества.

Для проверки качества обучения сети помимо обучающего множества из доступных примеров формируется также *тестовое множество*, содержащее примеры, не входящие в обучающее множество. Показателем итоговой *оценки качества* сети, обученной на обучающих примерах и аппроксимирующей нелинейное отображение (3), является значение суммарной ошибки на примерах тестового множества. Если тестовая ошибка устраивает пользователя так же, как

и ошибка обучения, то сеть признается *способной к обобщению*, т. е. способной производить качественный прогноз на незнакомых примерах.

Под *методами обучения* сети, т. е. нахождения (5), таким образом, могут пониматься любые оптимизационные методы, как общего назначения, так и специализированные.

Одним из основных этапов развития концепции ИНС стала разработка метода обратного распространения ошибки (ОРО), изначально использовавшегося в качестве метода обучения ИНС. Метод был независимо открыт рядом ученых, первые его упоминания можно найти в работах А. И. Галушкина [9], а также независимо описавшего данный метод в своей Ph.D.-диссертации П. Дж. Вербоса [149]. Дальнейшее развитие метод ОРО получил в 1986 году в работах С. И. Барцева и В. А. Охонина [4; 5], заложивших одну из известных научных школ в области ИНС, которую впоследствии возглавил А. Н. Горбань, а также одновременно и независимо в работах Д. Руммельхарта, Дж. Е. Хинтона и Р. Дж. Вильямса [137]. В настоящее время метод в основном используется уже не в качестве самостоятельного метода обучения, а как базовая составляющая более общих методов оптимизации, применяемых для обучения ИНС, позволяющая эффективно рассчитывать градиент функции ошибки. В частности, наиболее известными успешными попытками такого применения ОРО является использование метода Левенберга-Марквардта [95], метода сопряженных градиентов [125], алгоритма Broyden-Fletcher-Goldfarb-Shanno (далее – BFGS) [61] и других градиентных методов [64; 65].

На практике вопрос выбора того или иного *градиентного оптимизационного метода* обучения все еще остается открытым. В силу сложности ландшафта (овражность, наличие плато, изолированность локальных минимумов и многоэкстремальность) и высокой стоимости вычисления целевой функции ошибки обучения и ее градиента (для чего в общем случае требуется вычислить отклик сети на всем множестве обучающих примеров, которое может быть очень большим в реальных задачах, что и приводит к дороговизне вычисления целевой функции)

процедура обучения ИНС градиентными методами на практике оказывается весьма трудоемкой.

Между тем, в последнее время растет интерес к использованию при решении сложных задач, в том числе и обучения ИНС, эвристических методов *глобального случайного поиска (ГСП)*, что связано как с ростом возможностей вычислительной техники, так и с достигаемыми успехами при решении реальных задач с использованием этих методов [155]. Активно развиваются различные подходы глобального стохастического поиска (эволюционные/генетические алгоритмы, метод имитации отжига, дифференциальная эволюция, поиск с запретами и др.). Вопросами их применения к обучению ИНС занимаются такие отечественные и зарубежные ученые, как В. М. Курейчик [20; 28], В. В. Курейчик [20], В. Г. Редько [37], Ю.Р. Цой [52, 145], К. Stanley, R. Mikkulainen [141], Х. Yao [153] и мн. др. Однако, как отмечается авторитетными исследователями теории и практики глобального поиска А. А. Жиглявским и А. Г. Жилинским, «локальные методы оптимизации составляют важную часть глобальной методологии оптимизации» и «типичная глобальная стратегия оптимизации всегда является компромиссом между двумя конкурирующими целями: глобальностью и локальностью поиска», поэтому полный отказ от локальных методов представляется недостаточно обоснованным, особенно в таких сложных задачах, как обучение нейронных сетей.

Не менее важной задачей, чем подбор оптимального набора весов в фиксированной топологии является собственно определение данной топологии – структурный синтез сети. На этапе подбора топологии важно учитывать, что количество нейронов напрямую влияет на способность сети к обобщению [97]. Говорят, что *хорошей обобщающей способностью* обладает та сеть, ошибка которой на тестовой выборке достаточно мала (то есть незначительно отличается от ошибки на обучающей выборке). Понятие обобщающей способности тесно связано с понятиями *переобучения* и *недообучения*. Переобучение возникает в случае, когда модель вместо причинно-следственных связей, содержащихся в примерах, начинает описывать случайные характеристики обучающего множе-

ства, что происходит при использовании *избыточно сложных* сетей (избыток нейронов). При использовании же *недостаточно сложных* сетей (нехватка нейронов) возникает недообучение, когда алгоритм обучения не обеспечивает достаточно малой величины средней ошибки уже на обучающем множестве.

Изначально подбор структуры осуществлялся исследователем вручную (без применения средств автоматизации), исходя из его субъективного понимания предметной области. Впоследствии А. Н. Горбанем, Е. М. Миркесом, S. E. Fahlman, В. Hassibi, Y. LeCun и др. предложены различные конструктивные и деструктивные методы подбора топологии, применяемые после процедуры настройки весов. В последние годы в прямых методах настройку весов связей и подбор топологии производят в некотором смысле одновременно: пространство поиска охватывает топологически различные сети. В данном случае речь идет скорее о стратегиях переключения между различными топологиями. Особенно часто такой подход применяется в различного рода мультистартовых методах. Заметим, что при этом существующие методы все еще не обеспечивают получение топологий малых размеров, наиболее выгодных на практике. Естественным выходом из данной ситуации представляется разработка более гибких адаптивных механизмов, позволяющих динамически как усложнять, так и упрощать топологию непосредственно в процессе обучения.

Резюмируя вышесказанное, можно выделить следующие проблемы, возникающие при использовании на практике аппарата искусственных нейронных сетей вообще и МПРПС в частности:

- 1) сложность характера целевой функции и высокие затраты на расчет производных на больших обучающих выборках делают процедуру обучения ИНС градиентными методами при решении реальных задач весьма трудоемкой и не всегда удачной;
- 2) универсальной стратегии выбора оптимальной структуры ИНС не существует, а при неудачном подборе структуры в отдельных случаях процесс обучения может зайти в тупик; при этом известные деструктивные

методы и конструктивные методы не всегда обеспечивают получение оптимальной топологии.

Таким образом, можно утверждать, что перспективными направлениями исследований в области нейросетевого моделирования являются, во-первых, поиск способов вычислительно эффективного комбинирования градиентных алгоритмов и алгоритмов глобального поиска при обучении сетей (в частности, МПРПС); во-вторых решение проблемы подбора оптимальной топологии сетей путем спецификации структуры непосредственно в процессе обучения.

Актуальность работы следует из необходимости разработки новых методов обучения МПРПС, использующих комбинацию подходов глобального стохастического поиска и градиентных методов и позволяющих эффективно осуществлять спецификацию топологии сетей во время идентификации параметров, т. е. производить одновременный структурно-параметрический синтез сетей.

Целью работы является разработка метода структурно-параметрического синтеза МПРПС на основе подходов глобального случайного поиска и адаптации топологии в процессе обучения.

Для достижения поставленной цели были решены следующие **задачи**:

- 1) сформулировать постановку задачи структурно-параметрического синтеза МПРПС и предложить схему ее решения;
- 2) разработать на основе предложенной схемы вычислительно эффективные метод и алгоритмы структурно-параметрического синтеза МПРПС;
- 3) реализовать разработанные метод и алгоритмы в виде комплекса программ;
- 4) провести апробацию алгоритмов и программного обеспечения на представительных тестовых и прикладных задачах.

Методы исследования: анализ и обобщение результатов ранее проведенных исследований в области искусственных нейронных сетей; применение методов теории оптимизации, теории вероятностей, математической статистики, машинного обучения; исследование эффективности разработанных алгоритмов с помощью вычислительных экспериментов.

Работа **соответствует паспорту специальности 05.13.18 «Математическое моделирование, численные методы и комплексы программ»** по пунктам 1 (разработан новый метод структурно-параметрического синтеза нейросетевых моделей), 3 (разработан новый алгоритм глобального поиска), 4 (разработан комплекс программ нейросетевого моделирования), 5 (исследованы проблемы прогнозирования из областей экономики и образования – получены модели прогнозирования налоговых поступлений и результатов ЕГЭ).

Научную новизну результатов работы составляют:

- 1) разработана новая мультистартовая схема одновременного подбора топологии и настройки весов связей МПРПС, в которой для каждой сети реализуется собственная поисковая стратегия; на основе данной схемы разработан новый метод структурно-параметрического синтеза МПРПС;
- 2) разработан новый алгоритм глобального поиска, используемый в качестве способа нелокального параметрического улучшения МПРПС;
- 3) предложена новая стратегия модификации топологии МПРПС для мультистартовых методов.

Теоретическая и практическая значимость результатов работы. Разработанные метод и алгоритмы определяют перспективное направление развития в области разработки эффективных методов нейросетевого моделирования.

Разработанные метод и алгоритмы и их программная реализация использовались для выполнения работ в рамках грантов РГНФ и Бурятского государственного университета. Комплекс используется в работе Института математики и информатики Бурятского государственного университета, Управления довузовской подготовки Бурятского государственного университета, Отдела прогнозирования рынка труда Министерства экономики Республики Бурятия, а также в исследованиях Отдела региональных экономических исследований БНЦ СО РАН.

Степень достоверности и апробация результатов. Достоверность результатов диссертации обусловлена использованием апробированных научных методов и средств, разработанных авторитетными специалистами: опорой на известные результаты, полученные в теоремах Стоуна-Вейерштрасса [44], А. Н. Колмо-

горова и В. И. Арнольда [2; 26; 27] работах А. И. Галушкина [8-9], А. Н. Горбана [13; 89], А. А. Жиглявского и А. Г. Жилинскаса [21-22; 155], В. Крейнвича [111], Р. Шнабеля [15], G. Cybenko [69], R. Necht-Nielsen [98], Y. LeCun и др., сопоставлением с результатами расчетов других авторов, проведением вычислительных экспериментов.

Результаты работы обсуждались на Всероссийских семинарах «Нейроинформатика, ее приложения и анализ данных» (Красноярск, 2008, 2009); в школьно-семинаре молодых ученых «Математическое моделирование и информационные технологии: управление, искусственный интеллект и технологии программирования» (Улан-Удэ – оз. Байкал, 2006); на Международных конференциях «Инфокоммуникационные и вычислительные технологии и системы» (Улан-Удэ – оз. Байкал, 2009, 2010), «Математика, её приложения и математическое образование» (Улан-Удэ – оз. Байкал, 2009); на семинарах Института математики и информатики БГУ, Отдела региональных экономических проблем БНЦ СО РАН, Института динамики систем и теории управления СО РАН.

Работа по теме диссертации была отмечена II премией на I региональном конкурсе среди молодых ученых Республики Бурятия «Конкурентоспособность региона в условиях экологических и демографических ограничений» (ОР-ЭИ БНЦ СО РАН, Улан-Удэ, 2009).

Публикации и личный вклад автора. По теме исследования опубликовано 8 работ, 3 из которых опубликованы в журналах из перечня ВАК РФ.

Результаты Главы 1 опубликованы в работах [47-48]; Главы 2 – в [35; 46-48]; главы 3 – в [34-35; 46; 48; 50-51]. Все результаты, выносимые на защиту, получены автором лично. В работах [34-35; 46-51] постановка задачи, подготовка данных, разработка и апробация алгоритмов выполнены автором лично. В работах [46-47] сравнительный анализ разработанных алгоритмов и существующих методов произведен автором совместно с научным руководителем, Дармаевым Т.Г.; в работах [34; 49-51] экономическая трактовка полученных результатов выполнена автором совместно с Дондоковым З. Б.-Д.; в работе [35] подготовка данных и экономическая трактовка полученных результатов выполнена автором совмест-

но с Дондоковым З. Б.-Д. и Каницкой А. Л.; в работах [35;48] постановка задачи, разработка алгоритмов выполнены автором лично, программная реализация выполнена в неделимом соавторстве со Скворцовым М. В.

Структура работы. Диссертация состоит из введения, трех глав, заключения, трех приложений и списка литературы. Общий объем работы составляет 132 страницы. Список литературы содержит 155 наименований, из них 101 – на иностранных языках.

В Приложении А приводятся примеры описаний нейросетевых моделей для тестовых задач, решаемых в Главе 2. В Приложении Б – опросник профессиональной готовности, использованный в практической задаче в Главе 3. В Приложении В – описание лучшей сети для задачи, решаемой в Главе 3.

ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ И ОБЩАЯ СХЕМА СТРУКТУРНО-ПАРАМЕТРИЧЕСКОГО СИНТЕЗА

1.1 Постановка задачи структурно-параметрического синтеза

1.1.1 Постановка задачи обучения сети с фиксированной структурой

Для введения необходимых обозначений приведем как задачу оптимизации задачу обучения сети с фиксированной топологией.

Число входов сети n_i и число выходов n_o считаем фиксированными, потому что явное управление этими параметрами (т. е. изменение значений n_i, n_o) невозможно, поскольку предполагает исключение входных или выходных параметров из рассмотрения, что в случае решения практических задач не находится в нашей компетенции. Число всех нейронов сети, включая входные и выходные, будем далее обозначать $n = n(n_h) = n_i + n_h + n_o$, где n_h – число скрытых нейронов сети.

Пусть имеется обучающее множество (2) примеров из эмпирической выборки данных, где пара $(x^{(t)}, y^{(t)})$ составляет t -й пример наблюдения ($t \in [1, T]$).

Матрица смежности графа соответствующего МПРПС будет иметь вид

$$w \in W = \left\{ (w_{ij}), w_{ij} \in [-w_a, w_a], i, j = \overline{1, n} \right\}. \quad (6)$$

Будем считать без потери общности, что $w_{ij} \in [-w_a, w_a]$.

Обозначим как $y_1, \dots, y_{n_i+n_h}$ – выходы входных и скрытых нейронов, тогда

отклики сети будут рассчитываться как $y_j = \varphi \left(\sum_{i=1}^{n_i+n_h} w_{ij} y_i \right)$, при

$$j = \overline{n_i + n_h + 1, n_i + n_h + n_o}.$$

Граф соответствующей сети изображен на Рисунке 1.

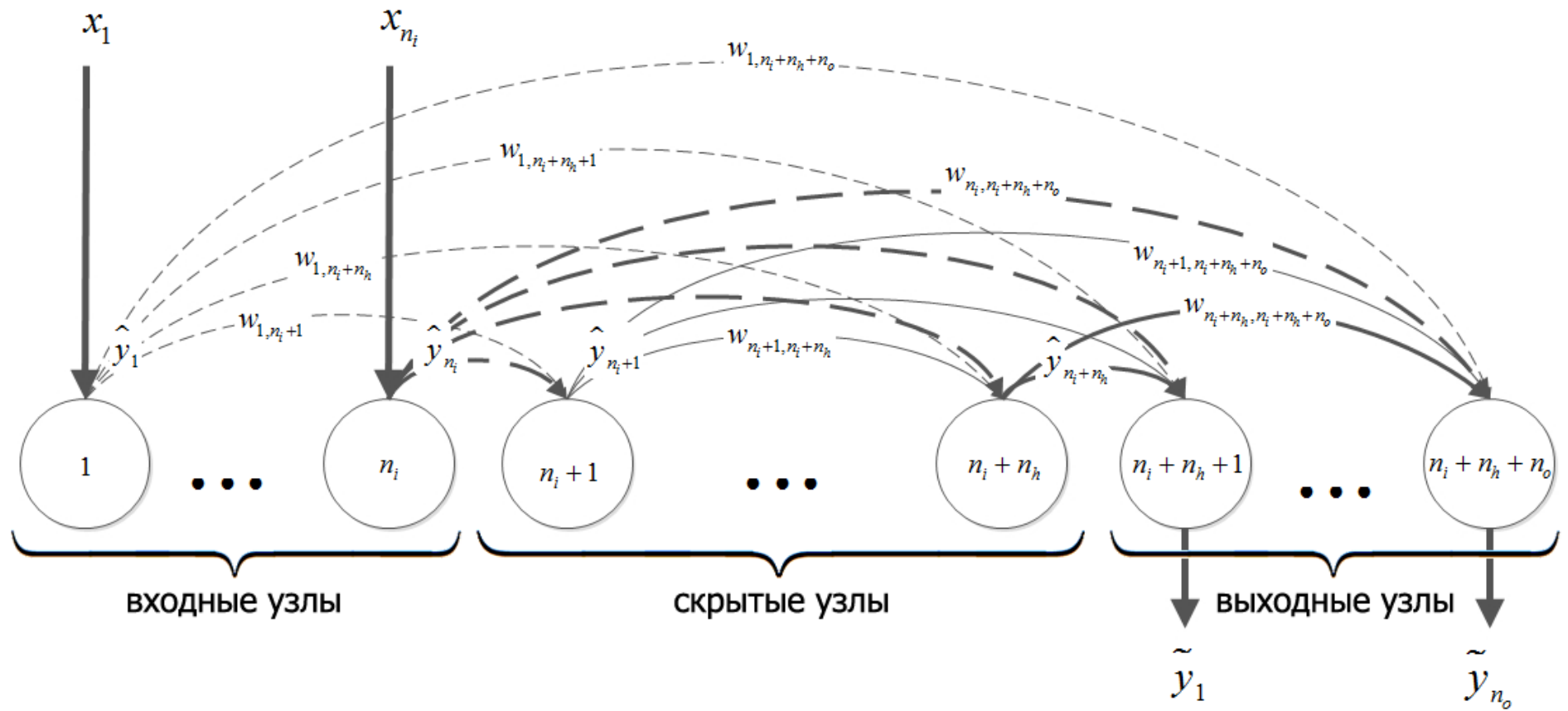


Рисунок 1 – Граф многослойного персептрона с разомкнутыми перекрестными связями

Задача обучения, т. е. настройки весов связей МПРПС как задача оптимизации заключается в подборе элементов матрицы $w^* = \arg \min_{w \in W} e(\text{Teach}, w)$. При этом функция ошибки $e(w)$ должна быть ограничена снизу и непрерывна в некоторой окрестности $U_\varepsilon(w)$, $\varepsilon > 0$. Пространство поиска W , т. е. множество матриц (6) при $w_{ij} \in [-w_a, w_a]$, является компактным, что гарантирует нам $w^* \in W$ – наличие минимума в области поиска.

В качестве (4) рассматривают различные функции, описывающие зависимость ошибки на тестовом множестве от аргумента (набора весов связей сети с фиксированной топологией). Наиболее распространено использование средне-квадратичной функции ошибки – целевой функции (далее – ЦФ) задачи оптимизации:

$$e(w) = \frac{1}{n_o} \frac{1}{T} \sum_{t=1}^T \sum_{k=1}^{n_o} \left(\tilde{y}_k^{(t)} - y_k^{(t)} \right)^2. \quad (7)$$

На практике применяются также и иные формы ЦФ, отражающие особенности решаемой задачи [143]. В настоящей работе, не теряя общности, мы будем использовать ошибку в виде (7).

Считая неизменным обучающее множество (2), получаем следующую задачу оптимизации при фиксированной топологии сети:

$$\begin{aligned} w \in W, \text{ где } W &= \left\{ (w_{ij})_{i,j=1,\overline{n}} \mid w_{ij} \in [-w_a, w_a] \right\}, \\ \tilde{y}^{(t)} &= \left(\tilde{y}_1^{(t)}, \dots, \tilde{y}_{n_o}^{(t)} \right), \text{ где } y_j = \varphi \left(\sum_{i=1}^{n_i+n_h} w_{ij} y_i \right) \text{ при } j = \overline{1, n_o}, \\ y_k &= \begin{cases} \varphi \left(\sum_{i=1}^{k-1} w_{ij} y_i \right) & \text{при } k \in [n_i + 1, n_i + n_h], \\ x_k & \text{при } k \in [1, n_i], \end{cases} \\ e(w) &= \frac{1}{n_o} \frac{1}{T} \sum_{t=1}^T \sum_{k=1}^{n_o} \left(\tilde{y}_k^{(t)} - y_k^{(t)} \right)^2 \rightarrow \min. \end{aligned} \quad (8)$$

В постановке (8) количество скрытых нейронов и расположение связей закреплено, поэтому начальное приближение определяется только весами имеющихся связей, которые в данном случае предлагается устанавливать случайными величинами, равномерно распределенными на интервале $[-w_a, w_a]$.

Для разработки метода структурно-параметрического синтеза нам требуется **новая** постановка задачи, достаточно широкая для того, чтобы используемые методы оптимизации можно было применять единообразно при поиске среди МПРПС с различным числом нейронов.

Для этого далее уточним вид используемой функции активации и представление сети в качестве точки пространства поиска.

1.1.2 Выбор функции активации

На практике наиболее общим случаем является использование в качестве функции активации φ некоторой сигмоидальной (гладкой нелинейной монотонной S-образной) передаточной функции, область значений которой, как правило, $[0, 1]$ или $[-1, 1]$. Сигмоидальная активационная функция должна производить усиление слабых сигналов и медленно насыщаться при сильных входных сигналах.

Следует заметить, что существуют исследования, посвященные использованию в МП более экзотических, в том числе и несигмоидальных, функций активации [73; 112], однако, как правило, это приводит к повышению вычислительных затрат без существенного повышения эффективности. Поэтому далее в работе мы не будем останавливаться на этом моменте, а станем рассматривать в качестве функции активации лишь сигмоидальные.

На Рисунке 2 изображена структура нейрона сигмоидальной сети, включающего сумматор и передаточный (активационный) механизм, реализуемый применением некоторой сигмоидальной функции активации φ .

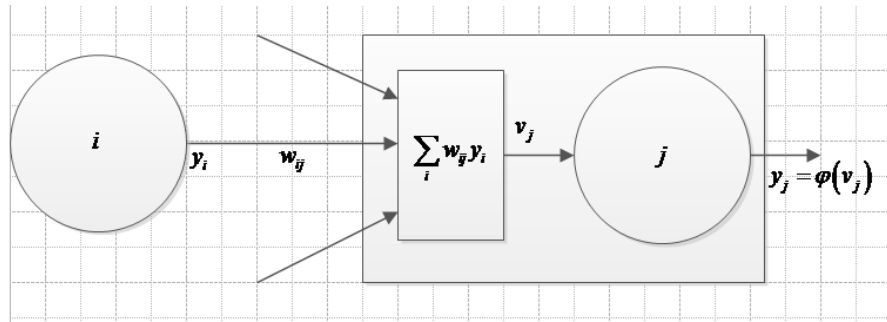


Рисунок 2 – Структура нейрона сигмоидальной сети

Использование гладкой функции активации позволяет применять для обучения сетей градиентные методы оптимизации – в этом случае возможен расчет градиента функции ошибки с помощью процедуры обратного распространения ошибки.

При этом несимметричные сигмоидальные активационные функции являются эмпирически известным источником систематической ошибки, ограничивая выходы нейронов значениями из интервала $[0,1]$ [38; 45].

В то же время целесообразность использования симметричных функций активации обоснована в работе Y. LeCun [118]: при помощи исследований собственных чисел матрицы Гессе показано, что использование таких функций (в отличие от несимметричных), с более высокой вероятностью позволяет добиться нулевого среднего для выхода любого нейрона, что, в свою очередь, приводит к уменьшению временных затрат на обучение.

Таким образом, в качестве функции активации требуется выбирать сигмоидальную симметричную функцию активации, причем, желательно с возможностью легкого вычисления производной. На практике наш выбор ограничен двумя вариантами: логистической функцией $y = \varphi(x) = 1 / (1 + e^{-\beta x})$ и функцией гиперболического тангенса x , что объясняется возможностью простого выражения производной через саму функцию: $y' = \varphi'(x) = \beta y(1 - y)$ в случае логистической функции и $y' = \varphi'(x) = \frac{\beta}{\alpha} (\alpha^2 - y^2)$ в случае гиперболического тангенса.

При этом очевидно, что фактически гиперболический тангенс можно считать аппроксимацией логистической функции, смещенной по оси ординат и обладающей свойством симметрии.

Для параметров α и β Y. LeCun подобраны значения $\alpha = 1,7159$, $\beta = 2/3$, исходя из требования $\varphi(1) \approx 1$, $\varphi(-1) \approx -1$, обоснованием которого является тот факт, что общий коэффициент усиления сжимающего преобразования составляет ≈ 1 , а интерпретация состояния сети упрощается, и, более того, абсолютное значение второй производной функции φ достигает максимума при ± 1 , что улучшает сходимость к концу эпохи обучения [115; 116]. График функции активации при использовании указанных коэффициентов α и β изображен на Рисунке 3.

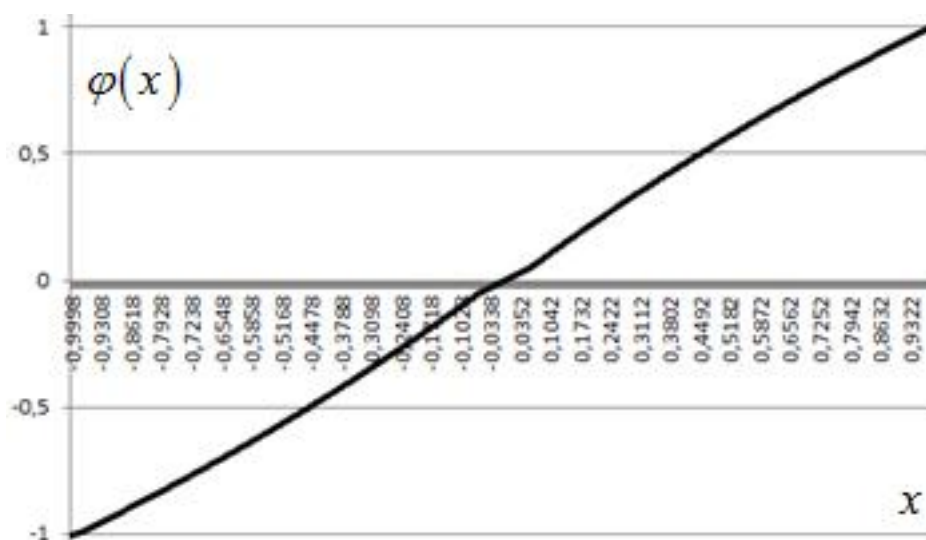


Рисунок 3 – График гиперболического тангенса со значениями коэффициентов, предложенными Y. LeCun

Заметим также, что несмещенность графика по оси абсцисс обеспечивает $\varphi(0) = 0$, что будет полезно также и при дальнейшей организации представления сетей (см. Раздел 1.1.3).

Учитывая все вышеизложенное, в дальнейшем в работе в качестве функции активации будем использовать гиперболический тангенс.

1.1.3 Кодирование точек пространства поиска при различающихся структурах

При постановке задачи в виде задачи оптимизации функции вида (4) используется матрица смежности графа, что обеспечивает константный доступ при прямом проходе (вычислении ЦФ) или обратном проходе (расчете градиента) по сети, однако приводит к следующим нежелательным последствиям:

- 1) затраты на хранение лишней информации об отсутствующих связях;
- 2) потеря удобства при работе с методами оптимизации;
- 3) рост вычислений при модификации топологии.

Принимая во внимание пункты 1 и 2, считаем целесообразным представлять пространство поиска как множество векторов, реализуемых при компьютерных расчетах в виде линейных массивов или связанных списков [1]. Рассмотрим подробнее связь подобной линейной структуры данных с матрицей смежности графа сети.

Элемент w_{ij} матрицы смежности $w = (w_{ij})_{i,j=1,n}$, находящийся на пересечении i -той строки и j -того столбца, обозначает вес связи от i -того нейрона к j -тому. Соответствующий список весов связей будет иметь размерность $L = n^2$, где $n = n_o + n_i + n_h$. Будем в точности соблюдать следующую нумерацию нейронов: первые n_i нейронов – входные, следующие n_o – выходные, оставшиеся n_h – скрытые нейроны. Тогда матрица смежности примет блочный вид (см. ниже), и мы сможем развернуть ее в линейную структуру меньшей размерности, поскольку при такой нумерации нейронов сетей и с учетом того, что используемые нами сети представляют собой направленные ациклические графы, в соответствующей матрице смежности не будет связей от нейрона с меньшим номером к нейрону с ббльшим номером. При этом в линейной структуре первые $n_i \cdot n_o$ элементов будут соответствовать связям от входных к выходным нейронам, следующие $n_i \cdot n_h$ элементов – связям от входных к скрытым нейронам, следую-

щие $n_h \cdot n_o$ – связям от скрытых к выходным нейронам и оставшиеся $\frac{n_h(n_h-1)}{2}$ элементов – связям между скрытыми нейронами. Количество элементов используемой линейной структуры составит

$$L = n_i \cdot n_h + n_h \cdot n_o + n_i \cdot n_o + \frac{(n_h-1)n_h}{2}. \quad (9)$$

Блочная структура матрицы смежности и соответствующего списка весов связей будут иметь вид

$$\begin{pmatrix} 0 & A & B \\ 0 & 0 & 0 \\ 0 & C & D \end{pmatrix}, (A \ B \ C \ D), \quad (10)$$

где в блоке A расположены элементы, соответствующие весам связей от входных к выходным нейронам, в блоке B – от входных к скрытым, в блоке C – от скрытых к выходным, в блоке D – между скрытыми нейронами.

Получение номера k позиции в списке весов связей от i -того нейрона к j -тому происходит по следующим формулам:

$$k = \begin{cases} p, & \text{при } i < n_i, n_i \leq j < u, \\ v + i \cdot n_h + j - u, & \text{при } i < n_i, u \leq j, \\ t + n_o \cdot (i - n_o) + j - n_i, & \text{при } n_i \leq i, j < u, \\ \frac{(2 \cdot n_h + u + s)(i - u)}{2} + v + t + n_o \cdot n_h + j + s, & \text{при } u \leq i, u \leq j, \end{cases} \quad (11)$$

где $p = n_o \cdot i + j - n_i$, $u = n_o + n_i$, $v = n_o \cdot n_i$, $t = n_i \cdot n_h$, $s = i + 1$.

Теперь при одноразовой индексации для каждого нового значения n_h , мы получаем константный доступ к элементам используемой линейной структуры при вычислении ЦФ и расчете градиента, как и при использовании матрицы смежности.

В предыдущем разделе настоящей работы в качестве функции активации был выбран гиперболический тангенс $\tanh(x)$, для которого выполняется условие $\tanh(0) = 0$. Таким образом, добавление и удаление некоторой связи равносильно

приравниванию к 0 соответствующего элемента списка весов связей $w_i = 0$. Тогда мы можем представлять сети с одинаковым количеством нейронов списками одинаковой длины L , соответствующей максимальному количеству связей, возможному при одинаковом количестве скрытых нейронов.

При конкретизации вышеописанного представления сетей в языке программирования необходимо учесть, что индексированность структуры для достижения константной скорости доступа к элементам, требуемой при вычислении значения ЦФ и применении методов оптимизации, представляется для нас более приоритетной, нежели эффективность в операциях вставки и удаления нейронов, требуемая для модификации топологии сетей. Поэтому на практике мы будем использовать стандартную структуру данных – динамический расширяемый массив (при реализации – `std::vector` из библиотеки STL языка C++ [40]).

Теперь мы можем перейти к постановке задачи обучения для сетей с нефиксированной топологией.

1.1.4 Постановка задачи обучения сетей

с различающимися структурами и общая схема СПС

Для получения задачи оптимизации сетей различных топологий мы можем переписать задачу (8) в следующем виде (обучающее множество (2) по-прежнему считаем неизменным):

$$\begin{aligned}
 w \in W, \text{ где } W &= \left\{ (w_i)_{i=1, L(n_h)} \mid w_i \in [-w_a, w_a] \right\}, \\
 \tilde{y}^{(t)} &= \left(\tilde{y}_1^{(t)}, \dots, \tilde{y}_{n_o}^{(t)} \right), \text{ где } y_j = \varphi \left(\sum_{i=1}^{n_i+n_h} w_{l(i,j,n_h)} y_i \right) \text{ при } j = \overline{1, n_o}, \\
 y_k &= \begin{cases} \varphi \left(\sum_{i=1}^{k-1} w_{l(i,j,n_h)} y_i \right) & \text{при } k \in [n_i + 1, n_i + n_h], \\ x_k & \text{при } k \in [1, n_i], \end{cases} \\
 e(w) &= \frac{1}{n_o} \frac{1}{T} \sum_{t=1}^T \sum_{k=1}^{n_o} \left(\tilde{y}_k^{(t)} - y_k^{(t)} \right)^2 \rightarrow \min.
 \end{aligned} \tag{12}$$

По сравнению с задачей (8) в постановку были добавлены зависимость $L(n_h)$ длины списка w от числа скрытых нейронов, рассчитываемая согласно (9), и $l(i, j, n_h)$ – позиция веса связи от i -го нейрона к j -тому в списке w , рассчитываемая согласно (11), чего удалось добиться благодаря выбору активационной функции и принятому способу кодирования МПРПС. В данной постановке (12) является задачей структурно-параметрического синтеза МПРПС для решения которой, в сущности, можно применять любые оптимизационные методы, однако в настоящей работе мы попытаемся предложить более проблемно-ориентированную схему. Как показано во Введении, вследствие сложности поверхности, формируемой ЦФ и большой размерности реальных задач, градиентные методы, применяемые в чистом виде, могут испытывать затруднения при решении задачи нелинейной оптимизации, поэтому мы попытаемся предложить более сложную схему решения (12), использующую помимо градиентных методов, также и методы глобального поиска.

Заострим внимание на том факте, что пространство поиска состоит из подпространств, каждое из которых соответствует той или иной фиксированной топологии сети. Целью моделирования при решении задач является получение таких сетей, которые достигают требуемой точности и обладают при этом минимально возможной структурой, иными словами, для получения допустимого решения требуется найти допустимое решение в любом из подпространств, при этом при равенстве по значению целевой функции то решение будет лучше, которое будет находиться в подпространстве меньшей сложности.

Требуемую точность решения может обеспечить применение некоторого градиентного метода, но лишь при условии старта из «удобной» точки, чего, конечно, нельзя гарантировать в каждом конкретном случае. Кроме того, желателен механизм, позволяющий находить такие «удобные» точки именно в подпространствах, соответствующих сетям с менее сложной топологией. Простейшим способом повысить вероятность попадания в удобную начальную точку и, следовательно, получить хорошее решение, является мультистарт, когда гради-

ентный метод запускается многократно из различных точек. Однако в общем случае не каждая точка популяции будет пригодна для осуществления градиентного спуска: мы могли бы воспользоваться этим и предпринимать отличные от градиентного спуска операции для таких точек на каждой итерации схемы с целью повышения вероятности нахождения решения допустимой точности и минимальной топологической сложности. Поэтому в настоящей работе будем использовать *популяционный* подход (в литературе также используется обозначение *методы поколений* [155]). Каждая итерация подобных методов заключается в том, что из имеющегося множества точек пространства поиска (текущего *поколения* или *популяции*) тем или иным способом порождается другое множество точек (следующее *поколение* или *популяция*). В отличие от простого мултистарта каждая точка следующего поколения зависит, в общем случае, от всех точек текущего поколения, а не от единственной родительской точки.

Таким образом, суть популяционной схемы решения поставленной задачи заключается в том, что мы выбираем некоторое конечное подмножество точек пространства поиска и производим *оценку перспективности* этих точек (квалификацию точек). Согласно полученной оценке, разбиваем все точки на две группы. В первой группе перспективных точек продолжаем градиентный спуск, а в группе точек, в которых определенное улучшение не достигнуто, будем производить либо *нелокальное параметрическое улучшение* (смену окрестности локального минимума) при сохранении топологии либо *модификацию топологии* сети. Упомянутую оценку перспективности будем производить, совершая **небольшое** число итераций некоторого градиентного метода в направлении тех локальных минимумов, в окрестностях которых оказались эти точки. Иллюстрация процесса описанной обработки точек приводится на Рисунке 4.

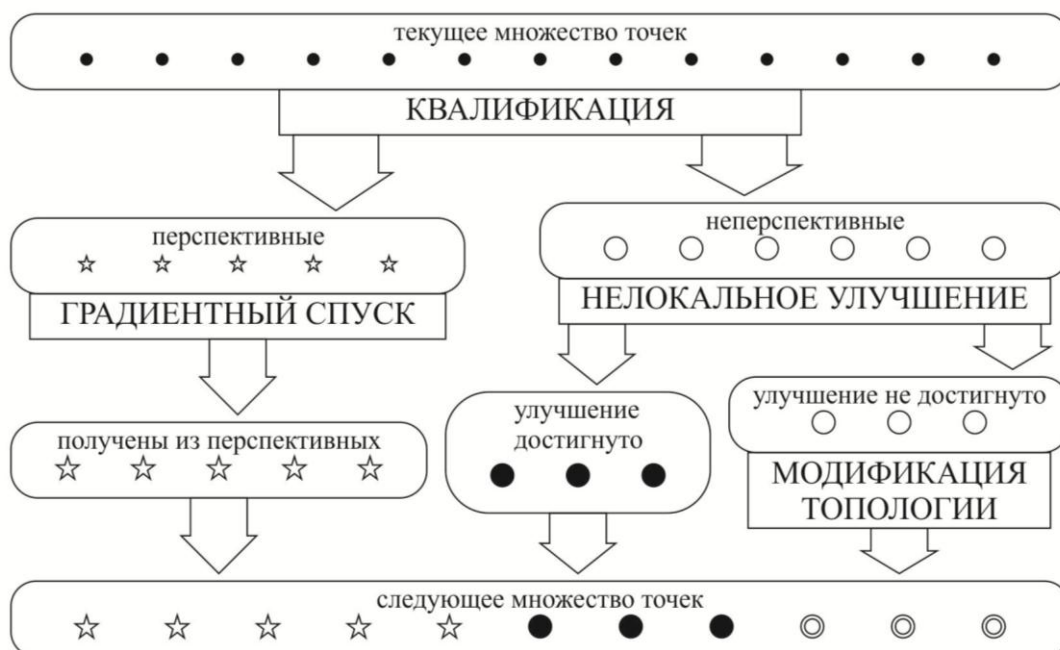


Рисунок 4 – Обработка точек при структурно-параметрическом синтезе

В результате применения данной итерационной поисковой схемы итоговое множество точек в общем случае будет отличаться – как параметрически, так и структурно – от первоначального. Критерием остановки могут являться достижение требуемой точности, превышение заданного числа итераций, превышение заданного числа вызовов ЦФ. Для алгоритмического наполнения данной схемы нам потребуется далее разработать алгоритм нелокального параметрического улучшения а также стратегию модификации топологии сети.

В качестве градиентного метода без потери общности будем применять метод BFGS [67; 79; 87; 139], который, как показывают экспериментальные исследования, более выгоден по признаку числа вызовов ЦФ, чем метод сопряженных градиентов [57; 70; 78 и др.]. Также в нашей работе мы ориентировались на известные схемы выбора метода решения задачи безусловной оптимизации с измененным числом переменных (см., например, [10]).

В работе была использована известная реализация BFGS за авторством Роберта Шнабеля (положительно определенная формула секущих в факторизованной форме) [16].

Заметим, наконец, что подробное исследование градиентных методов не входит в задачи настоящей работы – в случае применения разработанного ал-

горитмического и программного обеспечения на практике вместо BFGS с легкостью может быть использован любой другой градиентный метод. Основным требованием к предпочтению этого метода среди других при одинаковой достигаемой точности будет являться меньшее число вычислений целевой функции и ее градиента.

1.2 Алгоритмическое наполнение общей схемы СПС

1.2.1 Методы модификации топологии

Имеющиеся подходы к модификации топологии можно классифицировать различными способами. По принципу действия подходы делят на конструктивные (принцип усложнения) и деструктивные (принцип упрощения), по очередности применения – на динамические и производимые по окончании настройки.

Наиболее мощными **деструктивными** методами являются методы, основанные на *анализе чувствительности* [43]. Так, например, *Optimal Brain Damage (OBD)* [117] и его обобщение *Optimal Brain Surgeon (OBS)* [96] используют матрицу Гессе. Метод первого порядка, названный *контрастированием нейронной сети*, и также основанный на анализе чувствительности, был предложен А. Н. Горбанем [14]. При контрастировании производится анализ первых производных в ходе обучения градиентными методами, что не требует отдельного дифференцирования. В разное время также были предложены методы нулевого порядка: *прореживание* [32; 45] и *скелетонизация* [127; 154] нейронных сетей. Выделяется группа деструктивных методов, основанных на теории регуляризации Тихонова [41; 42], и направленных на минимизацию эмпирического риска [6]. Деструктивные методы применяются по завершении процесса настройки весов связей сети.

Если в деструктивных методах редуцируемая сеть должна быть в общем случае избыточной, то **конструктивные** методы, напротив, начинают работу с сетями минимальной топологии, постепенно увеличивая количество элементов до достижения требуемой производительности сети. Одним из первых успешных конструктивных методов является *CasCor* – каскадная корреляция Фальмана-Лебье [55; 77], в которой на каждом этапе обучения к сети добавляется новый скрытый нейрон. Существуют различные модификации метода каскадной корреляции – одной из наиболее успешных является *CasPer* [144], отличие которого от *CasCor* заключается в следующих моментах: 1) при добавлении производится выбор лучшего из нескольких встраиваемых нейронов; 2) после включения нейрона в сеть его веса не фиксируются, а продолжают изменяться. Непосредственно обучение может производиться с помощью любого подходящего алгоритма оптимизации (в оригинальной работе Фальмана был применен алгоритм *quickprop* [76]). Методы каскадной корреляции настраивают структуру (топологию) во время настройки параметров (весов связей).

Общим недостатком перечисленных методов является их вычислительная сложность при отсутствии гарантий оптимальности получаемой топологии, что в случае многоагентного подхода неприемлемо по причине увеличения количества вычислений пропорционально числу агентов.

При рассмотрении пространства поиска как совокупности подпространств операции модификации топологии становятся фактически операциями перехода между подпространствами и являются частью мета-алгоритма, управляющего перебором стартовых точек (принадлежащих в общем случае различным подпространствам), из которых производится запуск методов градиентного спуска.

Простейшими операциями модификации топологии могут быть удаление или добавление связи и удаление или добавление нейрона. Заметим, что набор возможных модификаций данным списком не ограничивается: очевидно, возможны и более сложные манипуляции с топологией (добавление или удаление целых подсетей, определяемых по некоторым правилам), однако в данной работе мы ограничимся простейшими операциями.

В многоагентных алгоритмах достаточно общим и наиболее употребительным является вероятностный подход к подбору структуры, когда в зависимости от значений различных параметров или наблюдаемой динамики изменяются вероятности применения операций модификации топологии [59; 93; 104; 140, 144; 152].

В качестве означенных параметров может выступать, к примеру, динамика изменения ЦФ, оцениваемая тем или иным способом структурная сложность или другие индивидуальные оценки модифицируемой сети и/или популяции в целом [11; 12; 93; 141; 153].

Основным преимуществом подобного вероятностного подхода является гибкое перемещение по всему пространству поиска – простая смена подпространств. Недостатками являются большое число параметров и необходимость их тонкой настройки для достижения требуемого эффекта, а также отсутствие гарантий универсальности при решении задач с различными поверхностями, формируемыми ЦФ.

В силу того, что каждая из операций модификации с некоторой вероятностью (пусть подчас невысокой) может быть применена на каждом этапе – можно говорить о реализации *параллельной* стратегии модификации топологии. Заметим, что параллельная стратегия обеспечивает высокую скорость сходимости для решения задач небольшой размерности, когда интенсивность исследования разных подпространств поиска, соответствующих различному количеству скрытых нейронов, предпочтительнее глубокого исследования фиксированной топологии. Однако при решении более сложных задач параллельная стратегия при поверхностной настройке параметров приводит, как правило, к излишнему усложнению топологии получаемых сетей. Последствием усложнения может быть как переобученность и последующая потеря обобщающей способности (обусловленная увеличением количества локальных минимумов пропорционально росту числа нейронов и, следовательно, легкостью их обнаружения), так и потеря точности (в силу недостаточно плотного исследования подпространств поиска).

В любом случае, глубокое исследование стратегии выбора модификации топологии в многоагентных методах представляет собой отдельную большую исследовательскую задачу.

В настоящей работе предлагается подход, более ориентированный на требования к получаемым решениям, а именно нацеленный на соблюдение приоритета упрощения перед усложнением. Схема такого подхода представлена на Рисунке 5.

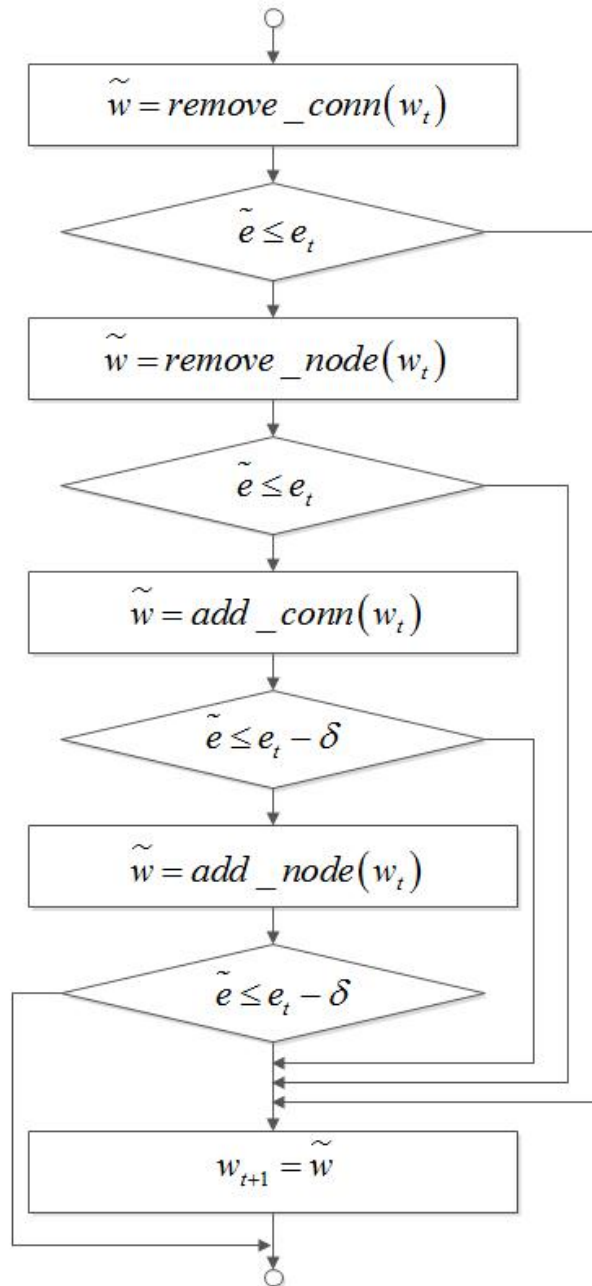


Рисунок 5 – Стратегия последовательного выбора операции модификации топологии

Операции *remove_conn*, *add_conn*, *remove_node*, *add_node* обозначают соответственно удаление/добавление связи, удаление/добавление нейрона. Суть этих

операций, а также детализация условий в схеме, представленной на Рисунке 5, подробно будут раскрыты в Разделе 2.1 при описании алгоритма модификации топологии.

1.3 Выводы по главе

В Главе 1 приведена известная постановка задачи обучения МПРПС с фиксированной топологией, эксплуатирующая представление сети в виде матрицы смежности. Описан более экономный способ представления МПРПС в виде линейных структур, отличающийся также легкостью осуществления операций модификации топологии.

Предложена общая популяционная схема структурно-параметрического синтеза МПРПС, основанная на разделении точек на перспективные и неперспективные для осуществления градиентного спуска. Для неперспективных точек далее должны быть определены алгоритм нелокального параметрического улучшения (смена окрестности локального минимума) при сохранении топологии, а также стратегия модификации топологии сети.

Выдвинуты требования к алгоритмическому наполнению: принято решение о применении для нелокального параметрического улучшения подхода глобального стохастического поиска, а для модификации топологии – последовательной стратегии выбора операции модификации с приоритетом операций удаления связи/нейрона перед операциями добавления связи/нейрона.

ГЛАВА 2. МЕТОД СТРУКТУРНО-ПАРАМЕТРИЧЕСКОГО СИНТЕЗА МНОГОСЛОЙНЫХ ПЕРСЕПТРОНОВ

2.1 Стратегия модификации топологии (синтез структуры) сети

В Разделе 1.2.1 было принято решение о применении последовательной стратегии модификации топологии с приоритетом операций упрощения и предложена ее схема (см. Рисунок 5). Опишем далее схему более подробно.

Пусть $w^{(k)}(t) = (w_1^{(k)}(t), \dots, w_L^{(k)}(t))$ – некоторая k -тая точка пространства поиска, т.е. МПРПС, для которой принято решение о модификации ее топологии, $L = L(n_h)$ – длина ее списка связей, тогда $\forall r \in [1, n_h]$: I_r – множество индексов связей, входящих в нейрон r , O_r – множество индексов связей, исходящих из нейрона r , $\zeta \in [0, 1]$ – случайная величина, равномерно распределенная на интервале $[0, 1]$, $\mu \in [-\varepsilon, \varepsilon]$ – случайная величина, нормально распределенная в ε -окрестности нуля, $e_t^{(k)} = e(w^{(k)}(t))$ – значение ЦФ, $\delta > 0$ – параметр качества приращения. Увеличение δ отражает потребность пользователя в небольшой топологии: если определить штраф большим, то можно добиться того, что топология вообще перестанет увеличиваться, и метод будет вынужден искать решение топологически меньшее или равное начальному. В работе использовалось значение $\delta = \sqrt{\varepsilon}$.

Алгоритм получения $w^{(k)}(t+1)$ путем модификации топологии решения $w^{(k)}(t)$ выглядит следующим образом (опустим для удобства индекс k):

1. Генерируем равномерно распределенное число $l \in \{[1, L]: w_l(t) \neq 0\}$.
2. Полагаем $w(t+1) = w(t)$, $w_l(t+1) = 0$.

3. Если $e_t - e_{t+1} \geq 0$, то улучшение достигнуто – переходим к шагу 16, в противном случае улучшения достигнуть не удалось – переходим к шагу 4.
4. Генерируем равномерно распределенное число $l \in \{[1, L] : w_l(t) = 0\}$.
5. Полагаем $w(t+1) = w(t)$, $w_l(t+1) = \zeta$.
6. Если $e_t - e_{t+1} > \delta$, то улучшение достигнуто – переходим к шагу 16, в противном случае улучшения достигнуть не удалось – перейдем к шагу 7.
7. Генерируем равномерно распределенное число $r \in [1, n_h]$ – номер позиции удаляемого нейрона.
8. Полагаем $n_h(t+1) = n_h(t) - 1$, $w(t+1)$ получаем из $w(t)$ путем удаления $n_i(t) + n_o(t) + n_h(t)$ элементов $\{w_l, l \in \{I_r \cup O_r\}\}$,
 $w(t) = (w_1(t), \dots, w_{L(t+1)}(t))$.
9. Если $e_t - e_{t+1} \geq 0$, то улучшение достигнуто – переходим к шагу 16, в противном случае улучшения достигнуть не удалось – переходим к шагу 10.
10. Генерируем равномерно распределенное число $r \in [0, n_h]$ – номер позиции добавляемого нейрона (0 в данном случае означает добавление перед первым нейроном, n_h – после последнего нейрона).
11. Полагаем $n_h(t+1) = n_h(t) + 1$. $w(t+1)$ получаем из $w(t)$ путем добавления $n_i(t) + n_o(t) + n_h(t)$ элементов на позиции, соответствующие входящим и исходящим связям нового нейрона, т. е. $L(t+1) = L(t) + n_i(t) + n_o(t) + n_h(t)$, $w(t) = (w_1(t), \dots, w_{L(t+1)}(t))$.
Значения добавленных элементов устанавливаем в окрестности нуля.
12. Равновероятно выбираем элементы $l_1 \in I_r$ и $l_2 \in O_r$.
13. Полагаем $w_{l_1}(t+1) = \zeta$, $w_{l_2}(t+1) = \zeta$.

14. Если $e_t - e_{t+1} > \delta$, то улучшение достигнуто – переходим к шагу 16, в противном случае улучшения достигнуть не удалось – переходим к шагу 15.
15. Улучшения достигнуть не удалось. Возвращаем значение $w(t+1) = w(t)$. Выход.
16. Улучшение достигнуто. Возвращаем новое значение $w(t+1)$. Выход.

Под улучшением решения в описанной стратегии понимается как снижение значения ЦФ, так и упрощение топологии МПРПС. При этом при усложнении топологии сети, мы требуем, чтобы полученное решение было явно лучше текущего $e_t - e_{t+1} > \delta$, а в случае упрощения – по крайней мере, не хуже $e_t - e_{t+1} \geq 0$.

При решении сложных практических задач хорошо себя показала следующая модификация изложенной стратегии. При успехе изменения топологии мы заменяем не родительскую точку $w^{(k)}(t+1) = w^{(k)}(t)$, а наихудшую точку популяции $w^{(n)}(t+1) = w^{(k)}(t)$. Родительские точки при этом остаются без изменений. Данная модификация ведет к интенсификации «исследования» каждого уровня топологии, вместе с тем снижая диверсификацию (в смысле охвата различных подпространств поиска, соответствующих сетям с различным числом скрытых нейронов в популяции).

Еще одной возможностью ускорения работы алгоритма, является детализация выбора удаляемых связей и нейронов. Здесь возможно применение различных механизмов осуществления подобного выбора, наиболее мощными из которых представляются процедуры контрастирования [14], OBS/OBD [96; 117] и др. Однако вместе с сокращением размерности получаемых топологий они несут и рост вычислительных затрат. В настоящей работе мы будем использовать более простой способ, который опишем ниже.

Число удаляемых и добавляемых связей всякий раз будем определять случайным образом – как равномерно распределенную случайную величину $\eta_{[0,Q]}$ из небольшого интервала $[0,Q]$. Стохастический характер выбора величины $\eta_{[0,Q]}$

позволяет добиться большей гибкости и универсальности, независимости от конкретики решаемой задачи.

Выбор величины Q обуславливается максимальным возможным числом связей $L_{\max}(n_h)$ в данной топологии и имеющимся числом связей L .

Основные способы выбора данной величины сводятся к двум следующим:

- 1) Q фиксировано на протяжении всего процесса синтеза;
- 2) Q выбирается адаптивно – пересчитывается каждый раз, исходя из каких-то средних параметров всей популяции, например, обратно пропорционально среднему приращению ЦФ $e(w)$: при $\overline{\Delta e} \rightarrow 0$, $Q \rightarrow L_{\max}(n_h)$.

Заметим, что при адаптивном выборе параметра Q увеличивается необходимость промежуточных расчетов, а именно – расчетов среднего значения ЦФ, требующих дополнительного расчета значений ЦФ для каждой сети, что, в общем случае, нежелательно. Поэтому в нашей работе мы будем использовать фиксированное значение Q . При этом остается открытым вопрос выбора этого фиксированного значения при различных параметрах решаемой задачи. В общем случае, мы должны отталкиваться от размерности самой задачи, определяемой числом нейронов на входном и выходном слоях или, с тем же смыслом, максимальным количеством связей, т. е. $Q = Q(n_o, n_i) = Q(L_{\max}(n_h))$. В нашей работе будем использовать следующее простое эвристическое правило: $Q = \varepsilon \frac{L_{\max}(n_h)}{n_h + 1}$, где $\varepsilon \in [0, 1]$. При этом значение Q будет отражать зависимость от размерности пространства поиска. В работе значение ε было принято за 0,1.

Добавление и удаление нейронов целесообразнее производить по одному, поскольку в противном случае мы рискуем получить неконтролируемый рост количества нейронов (объясняющийся тем, что сети с большим количеством нейронов в общем случае должны демонстрировать более высокую приспособляемость).

Добавление и удаление нейронов целесообразнее производить по одному, поскольку в противном случае мы рискуем получить неконтролируемый рост количества нейронов (объясняющийся тем, что сети с большим количеством нейронов в общем случае должны демонстрировать более высокую приспособляемость).

2.2 Алгоритм нелокального параметрического улучшения

1.2.2 Методы параметрического улучшения

В целом, в схеме, приведенной в Разделе 1.1.4, может быть применен любой глобальный метод безусловной оптимизации, и вопрос заключается в выборе наиболее эффективного из них. В настоящей работе предлагается использование методов глобального стохастического поиска (ГСП). Данные методы являются эвристическими, и меткая их характеристика дана, например, в монографии А. А. Жиглявского и А. Г. Жилинскаса: «эвристические методы: некогда презираемые, а нынче весьма уважаемые» (“heuristic methods: once scorned, now highly respectable”) [155]. На подобные изменения в настроениях специалистов могли повлиять следующие факторы [15]:

- 1) *практические успехи* – опыт решения реальных задач уже не позволяет отказывать эвристикам в признании, поскольку они подчас выигрывают у рациональных методов, особенно при решении задач высокой вычислительной сложности;
- 2) появление строгих математических доказательств о *невозможности существования гарантированных методов* поиска глобальных экстремумов любых непрерывных функций за разумное число операций.

Основное преимущество ГСП перед другими подходами заключается в робастности – малой чувствительности к «нехорошему» поведению ЦФ и наличию случайных ошибок в данных, полученных эмпирическим путем (что особенно важно в случае решения практических задач машинного обучения). Очевидно также, что алгоритмы ГСП менее чувствительны к структуре и размерности пространства поиска [22]. Таким образом, перечисленные особенности подхода ГСП свидетельствуют об оправданности его выбора для решения задачи настройки весов связей.

Основным минусом использования процедур ГСП является собственно эвристический характер данного подхода, не обеспечивающий в общем случае гарантии оптимального решения задачи. Наличие формальных доказательств теоретической сходимости вместе с тем практически бесполезно, поскольку такая сходимость, как правило, чрезвычайно медленна. Однако для решения достаточно большого числа практических задач вопрос получения оптимального решения не ставится. Часто для достижения цели достаточно найти некоторое приближение – ε -оптимальное решение – попасть в окрестность глобального или близкого к нему по значению локального минимума; кроме того, во многих задачах требуется лишь улучшить известное существующее решение за приемлемое время.

При решении практических задач обучения любых ИНС наиболее вычислительно затратным действием является, как правило, вычисление ЦФ и ее градиента (что связано с объемом имеющейся обучающей выборки). В связи с этим в качестве параметров сравнения различных сетей далее будут приняты как точность найденного решения, так и число вызовов ЦФ.

В данной работе методы ГСП рассматриваются исходя из деления на одноточечные и многоточечные (популяционные). Первые ставят своей задачей построение последовательности решений, сходящейся к глобальному минимуму, каждое из этих решений получается из предыдущего, а способ получения и определяет сам метод. Многоточечные методы на каждой итерации оперируют некоторым подмножеством допустимых решений, преследуя цели широкого охвата пространства поиска.

Далее при описании методов будем придерживаться единой схемы (см. Рисунок 6), выделяя этапы инициализации (*init*), проверки критерия останова (*stopping criteria*), получения пробных решений (*trial*) и перехода к следующему решению или множеству решений (*gen*). Будем также считать, что на каждом из этапов при выборе действия алгоритм опирается соответственно на наборы параметров set^{init} , set_t^{SC} , set_t^{trial} , set_t^{gen} (*parameters*), которые могут пересчитываться в конце каждой итерации.

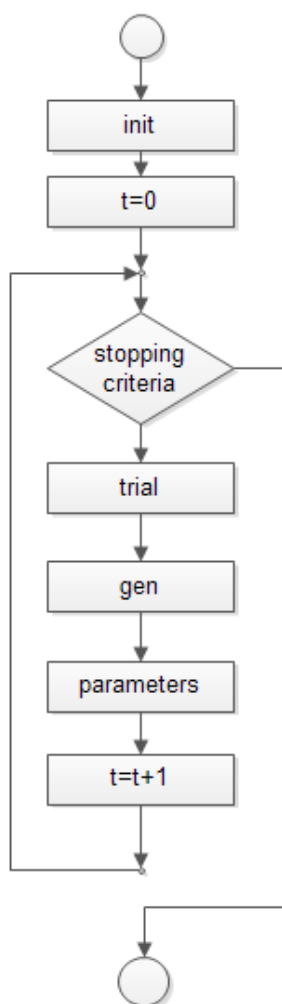


Рисунок 6 – Общая схема глобального стохастического поиска

Одноточечные методы ГСП. Цель одноточечных методов – построение последовательности $\{w_t\} \rightarrow w_{\min}$ при $t \rightarrow \infty$. Заметим, что при этом в общем случае не требуется выполнения условия $w_t \leq w_{t+1}$, но требуется $\{w_k\} \rightarrow w_{\min}$ при $t \rightarrow \infty$, где $\{w_k\}$ некоторая подпоследовательность $\{w_t\}$.

Для одноточечных методов схема, представленная на Рисунке 6, выглядит следующим образом:

1. Задаем или рассчитываем на основе особенностей решаемой задачи наборы параметров set^{init} , set_0^{SC} , set_0^{trial} , set_0^{gen} .
2. Генерируем по некоторому правилу начальную точку $w_0 = init(set^{init})$, $t = 0$.
3. Производим оценку $y_0 = f(w_0)$.

4. Если выполняется критерий останова $SC(w_t, set_t^{SC})$ – переходим к шагу 11, иначе – переходим к шагу 5.
5. Генерируем по некоторому правилу множество пробных точек $W = trial(w_t, y_t, set_t^{trial})$, которое может состоять и из одной точки.
6. Производим оценку пробных точек $Y = \left\{ f(w^{(k)}) \right\}_{k=1, p}$.
7. Получаем по некоторому правилу новую точку $w_{t+1} = gen(w_t, y_t, W, Y, set_t^{gen})$.
8. Пересчитываем параметры set_t^{SC} , set_t^{trial} , set_t^{gen} .
9. $t = t + 1$
10. Возвращаемся на шаг 3.
11. Выход.

Хорошо зарекомендовавшими себя односточечными методами на сегодняшний день являются, например, метод имитации отжига (simulated annealing) [74; 92; 106] и поиск с запретами (tabu search) [82; 84-86].

Конкретизируем далее приведенную схему на примере этих методов.

Имитация отжига. Идея метода имитации отжига принадлежит Н. Метрополису [74] и основывается на моделировании физического процесса кристаллизации вещества при отжиге металлов при понижающейся температуре. Атомы предварительно раскаленного металла при охлаждении стремятся образовать кристаллическую структуру с минимальной энергией, при этом с уменьшающейся вместе с температурой вероятностью они могут перейти и в состояние с большей энергией. Процесс завершается, когда температура падает до заранее заданного значения. Возможность перехода в худшее состояние зачастую позволяет отыскать состояние с энергией меньшей, чем начальная. Для решения оптимизационных задач метод был применен С. Киркпатриком [106], впоследствии развит Л. Ингбером [105].

Этапы инициализации и получения пробной точки представляют собой моделирование равномерного распределения по пространству поиска D : $w_0 = \text{uniform}(D)$ и $w_t = \text{uniform}(D)$.

Этап генерации пробного решения выглядит следующим образом. В соответствии с распределением Гиббса рассчитывается вероятность перехода к полученной пробной точке:

$$\text{Pr} = \text{Pr}(w_t, w, y_t, y) = \begin{cases} 0, & \text{если } y \leq y_t \\ \frac{y - y_t}{T_t}, & \text{если } y > y_t \end{cases},$$

после чего осуществляется переход с рассчитанной вероятностью

$$w_{t+1} = \begin{cases} w_t & \text{с вероятностью } \text{Pr} \\ w_t & \text{с вероятностью } 1 - \text{Pr} \end{cases}.$$

Пространство поиска D не изменяется на протяжении работы алгоритма, выступая параметром $\text{set}_t^{\text{init}}$ и $\text{set}_t^{\text{trial}}$. Единственным параметром $\text{set}_t^{\text{gen}}$ является T_t – т. н. «температура», которая изменяется следующим образом: $T_{t+1} = \alpha_t \cdot T_t$, $\alpha_t \in (0,1)$. Индекс t при коэффициенте α , означает, что при одном и том же значении температуры может быть проведена более чем одна внутренняя итерация алгоритма. Температура также является параметром set_t^{SC} : останов алгоритма происходит при падении температуры ниже заданного значения: $T_t \leq T_{\text{start}}$.

При соблюдении равномерности распределения на *trial*-этапе алгоритм достигает глобального минимума при $t \rightarrow \infty$. Известны варианты алгоритма, использующие другие распределения многомерных точек w_t , а также различные распределения для их отдельных координат.

Идея **поиска с запретами** принадлежит Ф. Гловеру [83] и основывается на ведении списка запретов tabu_t , в непрерывных задачах представляющем собой набор подмножеств поискового пространства, содержащих просмотренные ранее точки:

$$\begin{cases} \text{tabu}_0 = \{\emptyset\} \\ \text{tabu}_{t+1} = \text{tabu}_t \cup B(w_t) \end{cases}$$

Заметим, что изначально метод предназначался для решения задач дискретной оптимизации. В непрерывном случае метод становится концептуально схож с так называемыми методами покрытий [21-22, 155].

В качестве подмножеств $B(w_t)$ могут быть использованы, например, шары $B_h(w_t)$ радиуса $h > 0$. При попадании поиска в локальный минимум, запреты «вытаскивают» поиск из его окрестности.

Начальная точка без потери общности может быть получена также моделированием равномерного распределения $w_0 = \text{uniform}(D)$. Более сложным по сравнению с предыдущим методом является этап выбора пробной точки, на котором в некоторой окрестности точки w_t : $\forall j \in [1, p]: z_j = \text{uniform}(B_h(w_t))$ генерируется множество точек $\{z_j\}, j = \overline{1, p}$, называемых *соседями*. Из полученных *соседей* отбирается множество кандидатов $\{\tilde{z}_j\}, j = \overline{1, p}$ – точек, которые не попадают в текущий список запретов $\{\tilde{z}_j\} = \{z_j \mid z_j \notin \text{tabu}_t\}$. Итоговой пробной точкой является $w_t = \arg \min_{j=1, p} \{f(z_j)\}$.

На этапе перехода к следующему решению совершается безусловный переход $w_{t+1} = w_t$. Очевидно, что возможны случаи перехода к точке, ухудшающей текущее состояние, поэтому на протяжении всей работы метода требуется хранить информацию о рекорде w^* – лучшей по значению ЦФ точки, посещенной во время работы метода.

Параметрами набора $\text{set}_t^{\text{trial}}$, помимо списка запретов tabu_t , являются p – число генерируемых соседей, $h > 0$ – радиус генерации и w^* – рекордная точка и значение в ней ЦФ $y^* = f(w^*)$.

Возможна ситуация, когда не удается сформировать множество соседей (в силу покрытия запретами на t -й итерации допустимой для формирования соседей области пространства поиска). В этом случае, как правило, производится рестарт метода из новой начальной точки (с сохранением данных о рекордных точке w^* и значении в ней y^*) [68; 140].

Популяционные методы ГСП. *Популяционные методы (многоточечные методы, методы поколений)* на каждой итерации оперируют множеством точек, называемым *поколением* или *популяцией*:

$$P_t = \left\{ w_k^{(t)} \right\}_{k=1, n_p} = \left\{ \left(w_{k,1}^{(t)}, w_{k,2}^{(t)}, \dots, w_{k,n}^{(t)} \right), w_{k,j}^{(t)} \in D_j \right\}_{k=1, n_p}, \text{ где } n_p - \text{размер популяции}$$

(фиксированный или изменяющийся по определенным правилам на протяжении всей работы алгоритма), $D_1 \times \dots \times D_n = D \subset \mathbb{R}^n$ – пространство поиска.

Схема популяционных методов является достаточно общей и отличается от одноточечной схемы лишь тем, что оперирует не с одной точкой, а с некоторым множеством.

1. Задаем или рассчитываем на основе особенностей решаемой задачи наборы параметров set^{init} , set_0^{SC} , set_0^{trial} , set_0^{gen} .
2. Генерируем по некоторому правилу начальную популяцию $W_0 = \left\{ w_0^{(k)} \right\} = init \left(set^{init} \right)$, $k = \overline{1, p}$, $t = 0$.
3. Производим оценку всех точек начальной популяции $Y_0 = \left\{ f \left(w_0^{(k)} \right) \right\}_{k=1, p}$.
4. Если выполняется критерий останова $SC \left(W_t, set_t^{SC} \right)$ – переходим к шагу 11, иначе – переходим к шагу 5.
5. Генерируем по некоторому правилу множество пробных точек $W = trial \left(W_t, Y_t, set_t^{trial} \right)$.
6. Производим оценку $Y = \left\{ f \left(w^{(k)} \right) \right\}_{k=1, p}$.

7. Получаем по некоторому правилу множество новых точек

$$W_{t+1} = gen(W_t, Y_t, W, Y, set_t^{gen}).$$
8. Пересчитываем параметры set_t^{SC} , set_t^{trial} , set_t^{gen} .
9. $t = t + 1$
10. Возвращаемся на шаг 3.
11. Выход.

При таком представлении одноточечных и популяционных методов можно говорить о том, что первые являются подклассом вторых.

Конкретизируем приведенную схему на примере одного из наиболее популярных в последнее время классов популяционных методов – эволюционных алгоритмах, получивших свое название от принципа биологической эволюции: этапы получения пробных решений в таких методах моделируют процесс естественного отбора в *популяциях*.

Множество точек t -й итерации называют «родительским» по отношению к множеству точек $t+1$ -й итерации, которое называют «дочерним» или множеством «потомков», точки *популяции* называют *особями* популяции [20; 39], акцентируя внимание на том, что моделируется эволюционный процесс. Операции получения пробных точек называют эволюционными операторами, реализующими отбор, воспроизводство и мутацию точек-особей:

$$\begin{aligned} parents_t &= selection(W_t); \\ children_t &= crossover(parents_t); \\ W_t &= mutation(children_t). \end{aligned}$$

Популяция следующей итерации получается на основе предыдущей популяции и на основе множества пробных решений $W_{t+1} = gen(W_t, W_t)$.

Реализация эволюционных операторов *selection*, *crossover*, *mutation*, а также процедуры получения нового решения *gen* и определяют конкретный эволюционный алгоритм. Очевидно, что эволюционные алгоритмы являются разновидностью ГСП с упором на адаптации параметров и обмене информацией между решениями в процессе исследования пространства поиска [47].

Большая свобода в выборе эволюционных операторов влечет за собой увеличение числа настраиваемых параметров и позволяет разработать очень хороший эволюционный алгоритм для решения некоторой конкретной задачи. Однако в случае применения ИНС сделать это гораздо труднее, поскольку поведение ЦФ зависит от обучающей выборки и может значительно измениться даже при решении одной и той же задачи. Существует достаточно много разновидностей и конкретных реализаций эволюционных алгоритмов, в той или иной степени отличающихся друг от друга. Одной из наиболее успешных и широко применяемых является т. н. **генетический алгоритм**, который исторически отличается от остальных разновидностей эволюционных алгоритмов большей ролью, придаваемой оператору скрещивания, производящего рекомбинацию текущих решений популяции. В данной работе для проведения сравнительного анализа будем использовать следующую его модификацию (см. Таблицу 1):

Таблица 1 – Параметры генетического алгоритма

Эволюционный оператор	Конкретизация эволюционного оператора
<i>selection</i>	турнирная селекция, с размерностью турнира 3
<i>crossover</i>	двухточечный кроссовер
<i>mutation</i>	$w_t^{(k)} = \text{uniform}\left(B_\varepsilon\left(\text{children}_t^{(k)}\right)\right)$, где $\varepsilon = \text{uniform}(0,1)$, а $B_\varepsilon(x)$ – шар радиуса ε
<i>gen</i>	$W_{t+1} = W_t$ с одной <i>элитной</i> особью: одна худшая по приспособляемости особь W_t заменяется одной лучшей по приспособляемости особью W_t

Еще одним значимым и очень эффективным методом, использующим идеи эволюционных алгоритмов, является метод **дифференциальной эволюции**, предложенный Р. Сторном и К. Прайсом [142]. Его основное отличие от других методов заключается в способе генерации пробных решений:

При любом $k \in [1, p]$ случайным и равновероятным образом выбираются три различных точки $a, b, c = \text{uniform}\left\{W_t / w_t^{(k)}\right\}$, и генерируется так называемый му-

тантный вектор $t = a + \alpha(b - c)$. Операции умножения на скаляр и сложения здесь применяются к точкам как к векторам n -мерного пространства.

Некоторые координаты полученного мутантного вектора замещаются координатами вектора предыдущей итерации $w_t^{(k)}$:

$$w_{t,j}^{(k)} = \begin{cases} t_j & \text{с вероятностью } \beta, \\ w_{t,j}^{(k)} & \text{с вероятностью } 1 - \beta. \end{cases}$$

Вектор следующего поколения определяется следующим образом:

$$\forall k \in [1, p]: w_{t+1}^{(k)} = \begin{cases} w_t^{(k)} & \text{если } y_t^{(k)} < y_t^{(k)}, \\ w_t^{(k)} & \text{если } y_t^{(k)} < y_t^{(k)}. \end{cases}$$

Метод отличается небольшим числом настраиваемых параметров, а точнее всего тремя: p – размер популяции, α – коэффициент взвешивания, β - вероятность замены координаты при скрещивании.

На практике рекомендуется брать $\alpha = \text{uniform}[-1.0, 1.0]$ (новое значение для каждого вектора или каждой его координаты), а $\beta = 0.9$.

Как показывают исследования, увеличение p на практике не дает существенного повышения скорости [71]. Из этих соображений значение параметра p принималось равным десяти.

Одной из проблем, возникающих при применении метода на практике, может стать выход поиска за границы при решении задачи с ограничениями. Впрочем, проблема легко разрешается, в простейшем случае – приравниванием вышедшей за пределы координаты ближайшему граничному значению.

2.2.1 Сравнительный анализ методов ГСП

Ландшафт поверхности функции ошибки зависит от имеющейся обучающей выборки. Даже для одной и той же задачи он может значительно различаться при различных наборах обучающих примеров. Разрабатываемый алгоритм при этом должен эффективно работать на всевозможных классах поверхности функции ошибки, как «плохих», так и «хороших». Также следует учитывать высокую стоимость вычисления ЦФ и ее градиента: на практике это означает, что алгоритм должен, по крайней мере, производить как можно меньше вычислений ЦФ. В данном разделе произведем оценку свойств, описанных в предыдущей главе алгоритмов с тем, чтобы предложить эффективный гибридный алгоритм параметрического улучшения без изменения топологии: улучшение в данном случае производится за счет изменения параметров – весов связей – при фиксированной топологии.

Эффективность методов будем оценивать, во-первых, по числу вызовов ЦФ; во-вторых, по точности найденного решения. Подчеркнем, что быстроту работы алгоритмов целесообразно оценивать именно по числу вызовов ЦФ, а не по затраченному времени в секундах, поскольку при применении методов к обучению нейронной сети на реальных задачах с большими выборками основную часть времени занимает именно вычисление ЦФ.

В силу стохастичности методов, ведущей к высокой зависимости от выбора начального приближения, в качестве результирующих следует рассматривать усредненные данные нескольких запусков (в настоящей работе усреднялись результаты 100 запусков из одних и тех же для всех сравниваемых методов стартовых точек, выбранных моделированием равномерного распределения по всему пространству поиска).

В качестве критерия остановки работы всех алгоритмов будем использовать один и тот же показатель: превышение заданного количества вычислений ЦФ.

Для обеспечения применимости результатов на практике размерность задач будем брать равной 100, что примерно соответствует полносвязному МПРПС при 9 входных, 2 выходных и 6 скрытых нейронах. Такой масштаб уже является

показательным и позволяет получить общее представление о возможностях исследуемых методов.

Тестирование описанных методов будем производить на наборе функций, графики которых (при $N=2$) представлены на Рисунке 7 и описание приведено в Таблице 2.

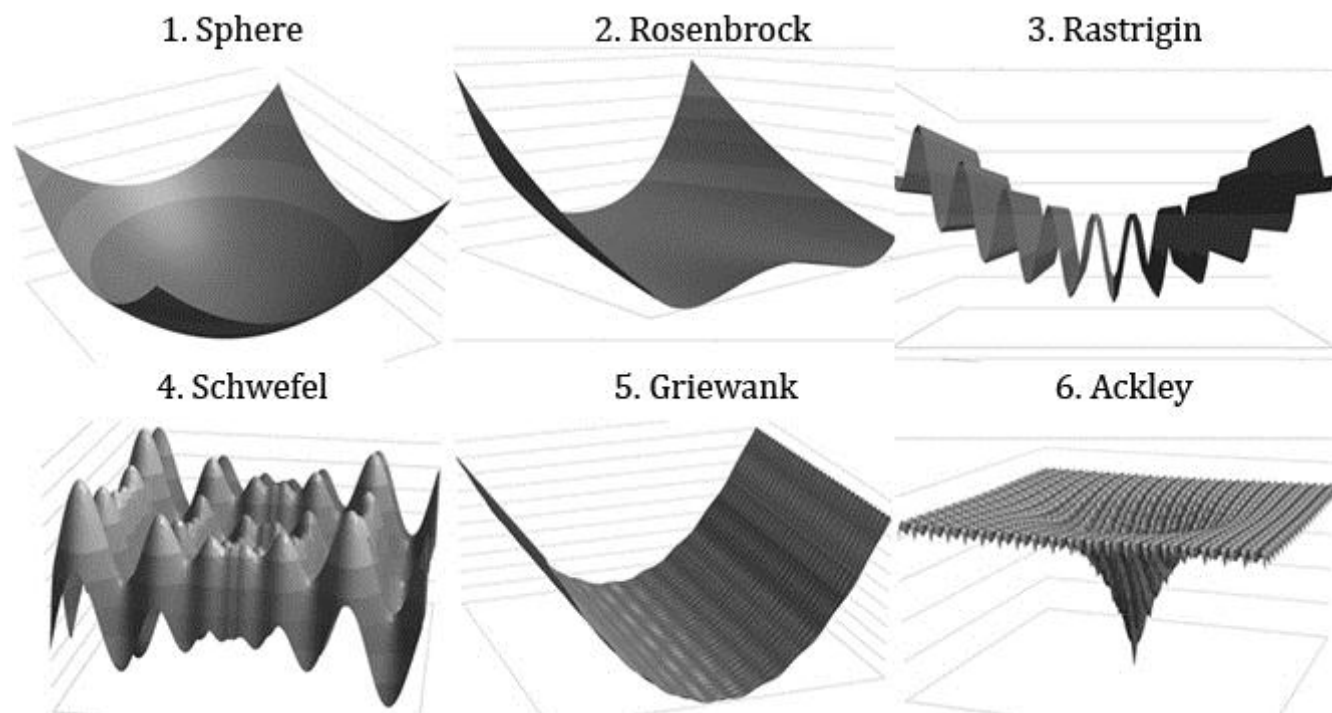


Рисунок 7 – Графики тестовых функций (при $N=2$)

Таблица 2 – Описание тестовых функций

№	Функция	Область определения	Минимум
1	Sphere (Многомерная сфера) $f(w) = \sum_{i=1}^n w_i^2$	-5,12; 5,12	$w_i^* = 0.0$ $f^* = 0$
2	Rosenbrock (Седло Розенброка) $f(w) = \sum_{i=1}^{n-1} \left(100 \cdot (w_{i+1} - w_i^2)^2 + (w_i - 1)^2 \right)$	-2,048; 2,048	$w_i^* = 1.0$ $f^* = 0$
3	Rastrigin (Функция Растригина) $f(w) = 10 \cdot n + \sum_{i=1}^n (w_i^2 - 10 \cdot \cos(2\pi w_i))$	-5,12; 5,12	$w_i^* = 0.0$ $f^* = 10$
4	Schwefel (Функция Швевеля) $f(w) = 418,9829 \cdot n + \sum_{i=1}^n (-w_i \cdot \sin(\sqrt{ w_i }))$	-500; 500	$w_i^* = 420,9687$ $f^* = 0.0$
5	Griewank (Функция Гриванка) $f(w) = 1 + \sum_{i=1}^n \left(\frac{w_i^2}{4000} \right) - \prod_{i=1}^n \cos\left(\frac{w_i}{\sqrt{i}}\right)$	-600; 600	$w_i^* = 0.0$ $f^* = 0$
6	Ackley (Функция Экли) $f(w) = 20 + e - 20 \cdot e^{-0,2 \cdot \sqrt{\frac{1}{n} \sum_{i=1}^n w_i^2}} - e^{\frac{1}{n} \cdot \sum_{i=1}^n \cos(2\pi w_i)}$	-30; 30	$w_i^* = 0.0$ $f^* = 0$

Функция многомерной сферы является достаточно простой для всех методов оптимизации: из любой стартовой точки направление убывания приводит нас в единственный имеющийся минимум. Rosenbrock отличается наличием медленно убывающего плато. Функции Rastrigin, Schwefel, Griewank и Ackley являются многоэкстремальными и потому достаточно сложными для любого алгоритма.

Заметим, что поверхность, формируемая функцией ошибки обучения многослойного персептрона, не так проста, как поверхность Sphere, и, скорее, похожа на функции Ackley или Schwefel. Рассматриваемый тестовый набор призван дать оценку рассматриваемых методов ГСП в грубом приближении.

Далее оценим возможности методов имитации отжига (SA), поиска с запретами (TS), генетического алгоритма (GA) и дифференциальной эволюции

(DE). На Рисунке 8 приведены графики лучшего решения, выбранного по показателю количества вычислений ЦФ.

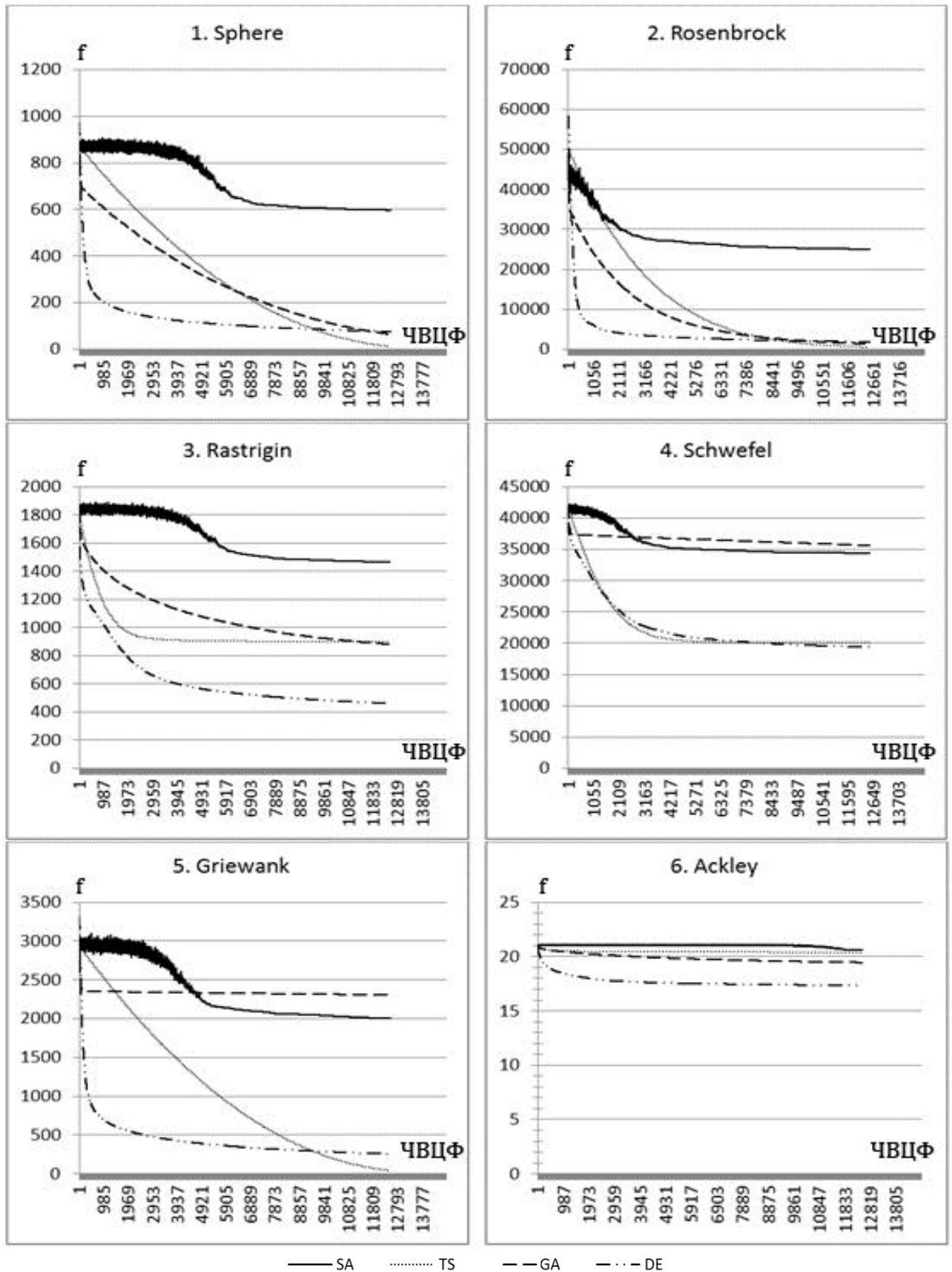


Рисунок 8 – Графики сравнения различных методов ГСП

Анализ приведенных графиков позволяет сделать следующие выводы:

- 1) метод имитации отжига показывает наихудшие результаты, что, скорее всего, связано с большой размерностью тестовых задач;
- 2) поиск с запретами показывает уверенную сходимость к глобальному минимуму на функциях Sphere, Rosenbrock, Griewank, что демонстрирует его возможности при обработке несложных (немногоэкстремальных) ландшафтов;
- 3) поиск с запретами работает значительно хуже на функциях Schwefel, Rastrigin и Ackley, попадая в ловушку локальных минимумов и не успевая выбраться из нее за приемлемое время;
- 4) метод дифференциальной эволюции показывает лучшие результаты на многоэкстремальных функциях Rastrigin и Ackley, однако проигрывает поиску с запретами в остальных случаях, за исключением функции Schwefel, где оба метода показывают примерно одинаковые результаты;
- 5) генетический алгоритм (в приведенной реализации) обнаруживает поведение, схожее с поведением метода дифференциальной эволюции, но показывает худшие в сравнении с ним результаты на всех задачах, кроме Sphere;
- 6) на сложных многоэкстремальных функциях Schwefel, Rastrigin и Ackley глобальный минимум не достигнут ни одним из методов.
- 7) Методы SA, GA и DE можно охарактеризовать как «исследующие пространство», причем, у метода DE подобное «исследование» получается лучше, чем у конкурентов. Метод TS более последовательно покрывает пространства поиска и, очевидно, сильно зависит от выбора начальной точки, однако при верном выборе поиск достаточно быстро спускается к близлежащему локальному минимуму.

2.2.2 Гибридный метод ГСП на основе комбинации поиска с запретами и дифференциальной эволюции

Как показано в предыдущем разделе, поиск с запретами достаточно хорошо находит минимум любой функции **при удачном выборе** начальной точки, когда организуется выбор соседей вблизи текущей точки. При этом можно говорить

о некоторой «осмысленности» поиска, придающей ему определенную скорость сходимости в направлении ближайшего локального минимума. Основной проблемой поиска с запретами является снижение скорости при выталкивании поиска из ловушки локального минимума.

В данном разделе мы опишем улучшение поиска с запретами – адаптацию радиуса, увеличивающую скорость сходимости при движении в окрестности локального минимума. Кроме того, рассмотрим возможность комбинации поиска с запретами с методом дифференциальной эволюции.

Способ генерации соседей на практике, как правило, более сложен, нежели простое равномерное распределение точки w_t : $\forall j \in [1, p]: z_j = \text{uniform}(B_h(w_t))$.

С целью более однородного исследования пространства вокруг точки w_t , шар $B_h(w_t)$ тем или иным образом разбивается на подобласти

$B_h(w_t) = \bigcup_{j=1}^p C_{h_{j-1}, h_j}(w_t)$ для генерации соседей по одному в каждой из этих подобластей $\forall j \in [1, p]: z_j = \text{uniform}(C_{h_{j-1}, h_j}(w_t))$. Тогда, в свою очередь, возникает задача разбиения, а точнее, выбора радиусов h_j . Предложены различные способы

подобного разбиения, например, $h_{p-j+1} = \frac{h_p(p-j+1)}{p}$ – линейное разбиение

или $h_j = \left(h_{j-1}^n - \frac{h_{j-1}^n}{p} \right)^{\frac{1}{p}}$ – геометрическое разбиение. При выборе способа следует

соблюдать требования, как вычислительной простоты организации процесса, так и качества самого разбиения.

В нашей работе будем придерживаться описываемого ниже способа. Пусть количество подобластей фиксировано и равно p . Пусть также задан (или каким-либо образом вычислен) некоторый базовый радиус генерации $h_{\text{trial}} > 0$.

Для любого $j \in [1, p]: h_j = 2 \cdot h_{j-1}$, при этом $h_0 = \frac{h_{trial}}{2^{\lfloor p \rfloor}}$, тогда внутренние по-

добласти будут исследоваться тщательнее внешних, в силу меньшего объема внутренних областей.

Далее нам потребуется определить величину *радиуса генерации* $h_{trial} > 0$ в основном цикле алгоритма. Уместно сделать ее адаптивно изменяющейся, также как и радиус шара (*табу-радиус*), заносимого в список запретов. Заметим, что постоянное значение радиуса (радиусов) при генерации и внесении в список запретов не является подходом, способствующим универсальности метода.

Радиус h_{trial} для единообразия будем далее обозначать, как $r_t^{(trial)}$ – и он должен быть таким, чтобы значение $h_0 = \frac{r_t^{(trial)}}{2^{\lfloor p \rfloor}}$ не падало ниже некоторого

фиксированного h_{min} , определяемого из требуемой точности искомого решения

задачи оптимизации. Также значение $h_p = 2^{\frac{\lfloor p \rfloor}{2}-1} \cdot r_t^{(trial)}$ не должно превышать не-

которое фиксированное h_{max} , определяемое из ограничений задачи оптимизации.

Число генерируемых соседей p выбирается небольшим, как правило, из интервала $[4, 20]$.

Табу-радиус будем определять как половину евклидова расстояния от точки предыдущей итерации до новой сгенерированной точки: $r_t^{(tabu)} = \frac{\rho(w_t, w_{t+1})}{2}$. Ра-

диус генерации $r_t^{(trial)}$ также должен зависеть от результатов предыдущей итерации. Будем определять его как равномерно распределенное число из интервала

$$\left[\frac{\rho(w_t, w_{t+1})}{2}, 2 \cdot \rho(w_t, w_{t+1}) \right].$$

Сравним результаты, полученные при использовании одинакового значения радиусов для занесения просмотренных точек в список запретов и генерации со-

седей и при использовании различных значений, рассчитываемых, как описано выше (см. Рисунок 9).

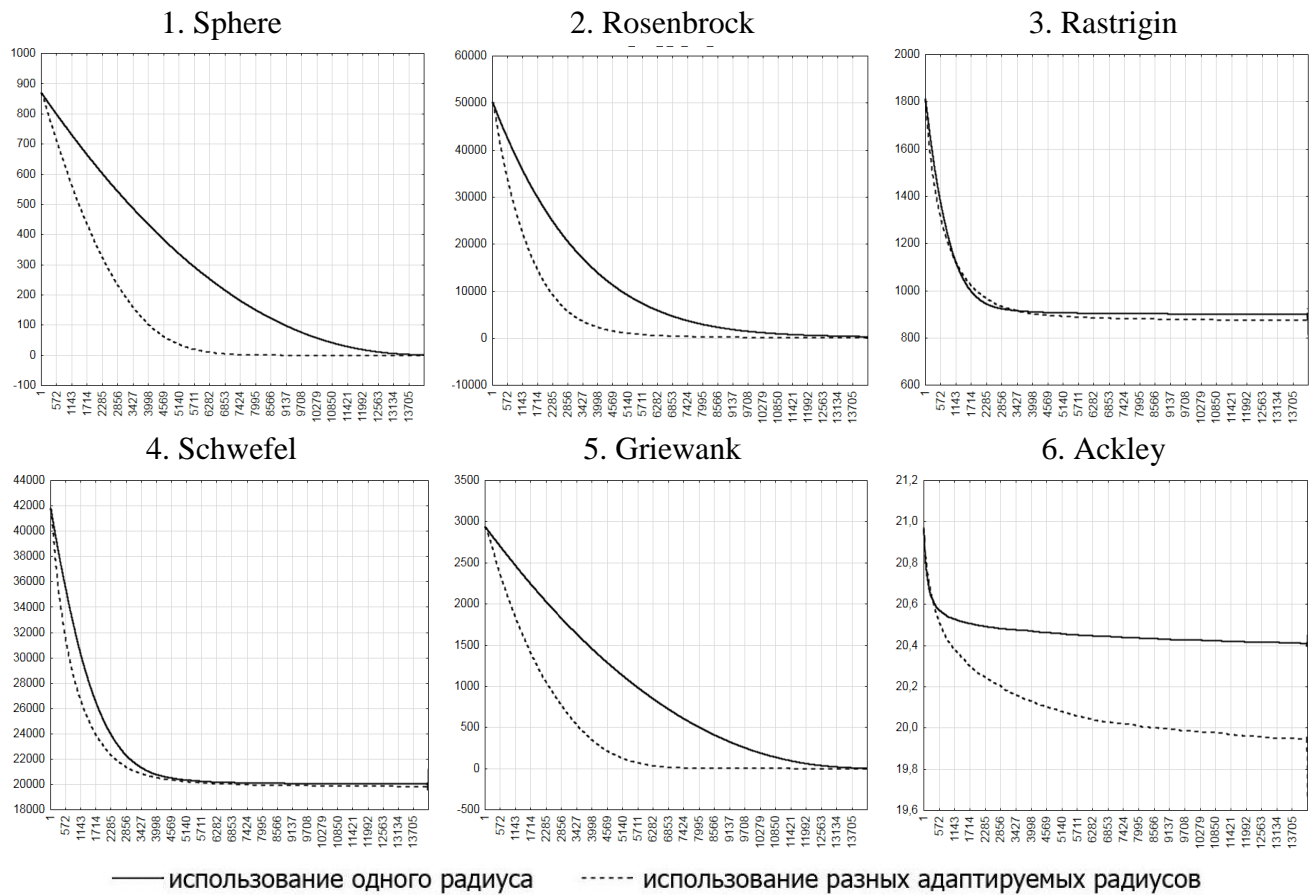


Рисунок 9 – Графики сравнения различных способов обращения с радиусами (по оси ординат – лучшее найденное значение ЦФ, по оси абсцисс – число вызовов ЦФ)

Как видно из графиков, поиск с различными значениями радиуса для списка запретов и генерации соседей опережает по скорости сходимости поиск с одним радиусом.

Работа поиска с запретами подразделяется на этапы спуска к локальным минимумам и подъема из окрестностей уже посещенных локальных минимумов. Если на этапе спуска метод действует сравнительно эффективно, достаточно быстро спускаясь в направлении поиска, то на этапах подъема эффективность поиска заметно снижается. Таким образом, желательным улучшением представляется более быстрый переход в окрестность другого локального минимума, минуя длительный процесс подъема.

В настоящей работе попробуем для организации подобного перехода использовать механизм дифференциальной эволюции. Будем различать шаги гло-

бального поиска, сделанные по правилам поиска с запретами (*ts*-шаг) и по правилам дифференциальной эволюции (*de*-шаг). Будем далее обозначать точки $w^{(de)}$ и $w^{(ts)}$, полученные в результате совершения *ts*-шага и *de*-шага, как *ts*-точки и *de*-точки соответственно; $y^{(de)}$ и $y^{(ts)}$ – значения ЦФ в этих точках.

Заметим, что совершение *de*-шага имеет смысл производить лишь в случаях, когда последовательность *ts*-шагов привела поиск на дно окрестности локального минимума, показателем чего является либо невозможность формирования *ts*-точки $w^{(ts)}$, либо ухудшение значения $y^{(ts)}$ по сравнению с текущей точкой $y_t < y^{(ts)}$. Общая схема получения пробной точки w_t показана на Рисунке 10.

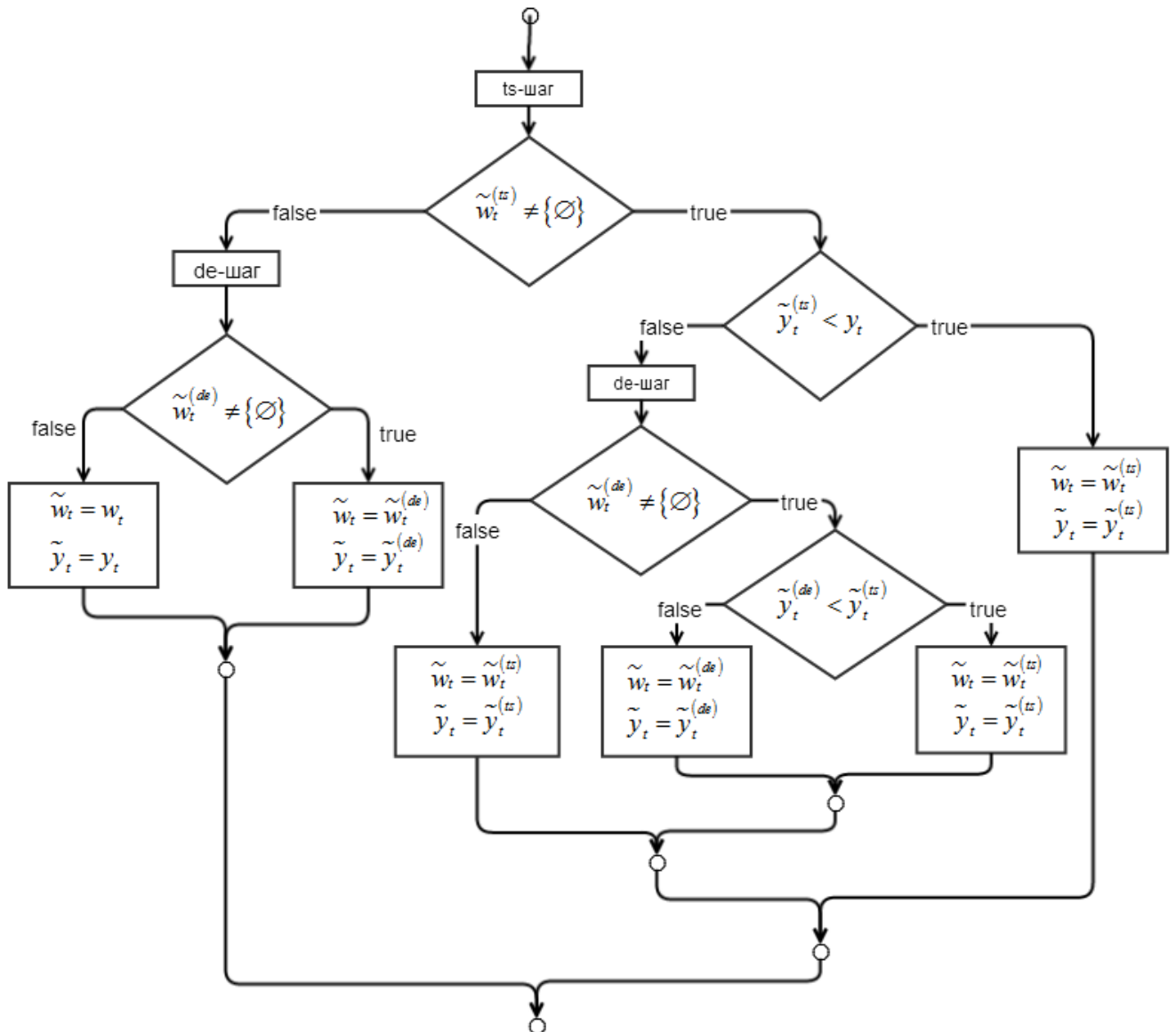


Рисунок 10 – Схема получения пробной точки в разрабатываемом методе

Очевидно, структура списка запретов в таком случае будет иметь сегментированный вид, где переход к каждому новому сегменту будет означать совершение de-шага, а сам сегмент – последовательность точек, полученных произведенными подряд ts-шагами. Таким образом, список запретов будет иметь вид:

$$tabu_t = \bigcup_{j=0}^S tabu_{t,j}, \text{ где } S \text{ – число сегментов списка запретов.}$$

В качестве точек популяции для совершения de-шага будем брать точки, принадлежащие различным сегментам списка запретов – по одной от каждого сегмента. Случайным и равновероятным образом будем выбирать три различных номера сегментов $v_0, v_1, v_2 \in [0, S-1]$, в каждом из которых также равновероятно выберем по одной точке: $p_j = \text{choise}\{tabu_{t,v_j}\}$ ($j \in [0, 2]$). Далее из координат выбранных точек сгенерируем мутантный вектор: $t = p_0 + \alpha(p_1 - p_2)$. Некоторые координаты полученного мутантного вектора заменим координатами точки, выбираемой из последнего сегмента списка запретов $x = \text{choise}_{last}\{tabu_{t,S}\}$.

Описанный способ совершения de-шага возможен при $S \geq 4$ (поскольку нам необходимо не менее 4-х точек и, соответственно, 4-х сегментов), а при $S < 4$ будем выбирать точку начала нового сегмента случайным образом равновероятно по всему пространству поиска за исключением уже запрещенных к этому моменту областей.

Заметим, что возможна потеря хороших решений в результате резких «скачков» на de-шаге. Попробуем исправить ситуацию, предложив способ, сохраняющий координаты рекордного решения. Для этого воспользуемся необходимостью определения действия в случае выхода за границы параллелепипедных ограничений, которые часто возникают в задаче обучения ИНС в качестве ограничений на изменение весов связей.

Будем считать, что на веса связей ИНС и, следовательно, на координаты точек в оптимизационной задаче наложены параллелепипедные ограничения вида

$$\forall t : b_l \leq w_{t,j}, w_{t,j} \leq b_r.$$

В случае применения de-шага возможна ситуация, когда какие-либо координаты полученной точки выходят за указанные границы $w_{t,j} < b_l$ или $w_{t,j} > b_r$. Воспользуемся ситуацией и будем изменять эти координаты на случайные значения, полученные моделированием нормального распределения в окрестности соответствующих координат рекордной точки: $w_{t,j} = normal(w_j^*, \tau)$. Если же j -тая координата и после этого «выскакивает» за границы, то приравняем:

$$w_{t,j} = \begin{cases} b_l, & \text{если } w_{t,j} < b_l \\ b_r, & \text{если } w_{t,j} > b_r \end{cases}.$$

Параметр $\tau > 0$ в данном случае следует выбирать достаточно небольшим, чтобы оставаться в окрестности рассматриваемой точки. В работе использовалось значение $\tau = \frac{b_r - b_l}{10^3}$.

Далее требуется определить $choise_{last}$ и $choise$ – процедуры выбора точек из сегментов. Очевидно, следует различать следующие варианты:

- 0) выбирается последняя точка сегмента (предполагаемое дно исследованной окрестности локального минимума);
- 1) выбирается первая точка сегмента (одна из наиболее удаленных от предполагаемого локального минимума точек в исследованной окрестности);
- 2) из сегмента равновероятно выбирается случайная точка.

Также нас интересует использование параметра α (коэффициента взвешивания) при генерации мутантного вектора – мы располагаем двумя вариантами:

- 0) новое значение α для каждого нового мутантного вектора (одинаковое для всех координат);
- 1) новое значение α для каждой координаты нового мутантного вектора.

Далее произведем вычислительный эксперимент, с помощью которого определим оптимальный набор трех параметров. Для каждой ЦФ многократно выберем стартовую точку случайным образом, равновероятно по всему пространству поиска и произведем запуск метода на каждой комбинации (наборе) пара-

метров, затем по каждой функции и набору параметров подсчитаем простейшую статистику: минимальное, максимальное, среднее значения и стандартное отклонение. Полученные результаты сведены в Таблице 3.

Таблица 3 – Результаты выбора процедур $choise_{last}$, $choise$ и значения α

$choise_{last}$, $choise$, α	Знач.	Sphere	Rosenbrock	Rastrigin	Schwefel	Griewank	Ackley	Err
0-0-0	мин.	1,69	185,88	332,92	10605,3	6,96	15,55	
	макс.	8,01	642,07	618,94	15001,5	29,69	19,95	
	ср.	4,05	354,78	467,37	12954,8	16,75	18,21	0,79
	откл.	1,67	95,06	56,33	1159,97	5,18	1,06	
0-0-1	мин.	0,47	144,22	306,25	10212,8	2,61	15,24	
	макс.	2,51	449,21	568,17	15622,2	10,43	19,92	
	ср.	1,2	247,2	447,85	12788,7	5,51	18,25	0,55
	откл.	0,5	63,8	57,83	1122,96	1,89	1,15	
0-1-0	мин.	1,42	251,8	327,11	9907,08	7,84	16,4	
	макс.	8,46	599,18	552,53	15681,4	40,55	19,68	
	ср.	4,56	390,88	458,45	12914,6	18,36	17,9	0,42
	откл.	1,56	77,62	56,72	1292,32	6,09	0,86	
0-1-1	мин.	0,62	149,56	364,92	10557,1	2,9	14,45	
	макс.	4,32	500,96	590,85	15867,8	12,19	19,66	
	ср.	1,6	285,18	467,99	12907	6,2	17,8	0,26
	откл.	0,72	80,73	53,64	1122,94	2,26	0,99	
0-2-0	мин.	1,36	213,77	371,54	9079,47	7,71	16,13	
	макс.	8,4	594,67	585,69	15296,6	29,48	19,76	
	ср.	4,41	375,59	471,96	12627,6	17,19	18,06	0,61
	откл.	1,71	77,26	49,15	1253,47	4,7	0,93	
0-2-1	мин.	0,57	167,97	324,72	9988,1	3,11	15,66	
	макс.	3,13	499,25	626,86	14860,4	12,63	19,91	
	ср.	1,34	255,69	450,69	12436,4	6,33	17,89	0,12
	откл.	0,55	63,98	57,38	1311,8	2,1	0,95	
1-0-0	мин.	33,85	1006,76	415,97	13482,5	110,87	16,83	
	макс.	87,5	2603,02	707,2	18905,6	292,6	19,95	
	ср.	59,51	1776,66	563,33	16131,7	199,62	18,74	4,53
	откл.	12,6	360,6	65,09	1197,04	42,41	0,83	
1-0-1	мин.	12,71	931,47	461,07	12848,5	69,77	16,85	
	макс.	80,96	2313,31	670,86	19860,2	228,33	19,95	
	ср.	48,28	1466,05	551,16	15626,8	164,34	18,36	3,55
	откл.	14,14	281,04	50,3	1383,58	36,46	0,72	
1-1-0	мин.	47,19	1524,56	455,17	13672,6	180,52	16,93	
	макс.	145,65	3684,67	724,86	19927,4	515,2	19,89	
	ср.	89,6	2492,44	593,75	17240,7	316,28	18,45	5,69
	откл.	18,03	511,98	63,62	1264,69	70,97	0,83	

Окончание таблицы 3

$choise_{last},$ $choise, \alpha$	Знач.	Sphere	Rosenbrock	Rastrigin	Schwefel	Griewank	Ackley	Err
1-1-1	мин.	56,09	1486,76	475,49	13311,5	160,72	17,01	
	макс.	137,14	4054,91	704,21	20541	540,61	19,9	
	ср.	87,32	2410,02	593,44	16667,9	296,29	18,16	5,13
	откл.	19,68	594,19	61,05	1555,17	69,32	0,7	
1-2-0	мин.	39,38	1099,81	490,88	13354,8	117,57	16,8	
	макс.	86,58	3184,83	780,19	19379,9	280,81	19,95	
	ср.	64,35	1885,19	588,83	16329,9	212,06	18,62	4,76
	откл.	12,72	458,8	60,04	1300,62	42,27	0,81	
1-2-1	мин.	32,08	1000,27	436,86	13493,8	128,73	16,31	
	макс.	83,77	2345,64	668,33	18886,9	268,68	19,95	
	ср.	55,01	1665,89	563,57	15982,8	202,74	18,27	3,91
	откл.	11,77	347,59	50,22	1203,85	36,88	0,75	
2-0-0	мин.	4,41	325,1	364,85	10235,6	17,71	16,4	
	макс.	16,6	851,97	606,48	16142,2	73,14	19,93	
	ср.	10,48	563,23	471,75	13393,6	38,68	18,31	1,26
	откл.	2,62	120,52	59,12	1263,94	12,69	0,81	
2-0-1	мин.	1,65	206,43	353,45	9401,26	9,66	15,88	
	макс.	8,5	571,28	643,11	16427,1	36,82	19,95	
	ср.	4,3	416,5	470,64	13154,8	17,02	18,32	1,01
	откл.	1,47	78,66	65,49	1305,71	5,51	1,03	
2-1-0	мин.	6,55	424,48	403,5	10292,4	12,35	16,6	
	макс.	20,1	771,37	600,41	15615,7	80,69	19,94	
	ср.	11,63	586	488,82	13679,3	45,91	18,19	1,35
	откл.	3,16	91,67	50,39	1090,68	13,67	0,88	
2-1-1	мин.	1,9	215,48	327,11	11251,8	7,81	16,19	
	макс.	8,99	645,94	543,85	17018,5	38,34	19,76	
	ср.	5,07	422,57	453,58	13352,2	20,46	17,95	0,56
	откл.	1,67	100,77	47,12	1251,27	7,17	1,01	
2-2-0	мин.	5,13	326,77	349,6	11063,8	14,32	16,71	
	макс.	17,61	956,69	546,3	17210,3	62,79	19,95	
	ср.	9,95	565,05	467,74	13423,6	36,2	18,39	1,31
	откл.	2,86	110,39	46,42	1325,39	10,97	0,84	
2-2-1	мин.	1,95	233	365,28	10017,8	9,51	16,35	
	макс.	10,79	652,09	568,5	15861,8	45,09	19,95	
	ср.	4,97	451,66	460,16	13051,2	19,56	18,19	0,81
	откл.	1,7	98,27	47,79	1161,72	6,86	0,9	

По Таблице 3 определим лучший набор параметров по абсолютному отклонению Err по формуле для j -той строки $Err_j = \sum_{k=1}^6 (x_{jk} - \min_k) / (\max_k - \min_k)$, где \min_k и \max_k – минимальное и максимальное значения по k -тому столбцу, x_{jk} – значение в j -той строке.

Произведенные эксперименты показывают, что лучшие показатели среднего значения получаются при выборе в процедуре *choise_{last}* последней точки сегмента. Наилучшие результаты получены при коде 0-2-1, т. е. когда в процедуре *choise_{last}* выбирается последняя точка сегмента, в процедуре *choise* – случайная точка сегмента, а α выбирается для каждой координаты нового мутантного вектора.

Теперь, когда мы определили различные параметры разработанного алгоритма, произведем его сравнение с базовыми методами. Для этого запустим алгоритм многократно из тех же стартовых точек, на которых производилось сравнение методов в начале данной главы. Результаты представлены на Рисунке 12. Новый разработанный метод обозначен, как TS_DE.

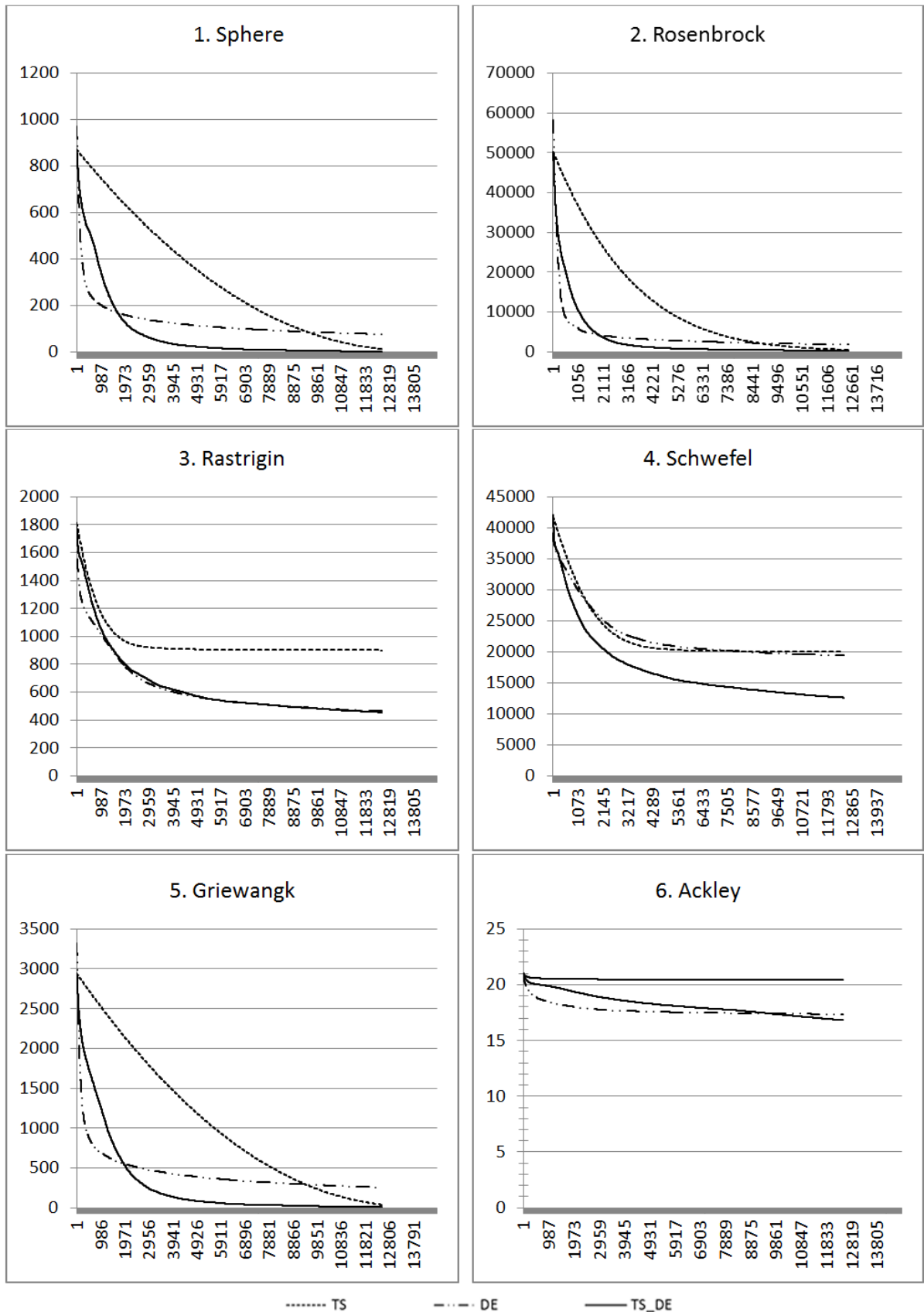


Рисунок 12 – Графики сравнения разработанного гибридного метода с родительскими подходами дифференциальной эволюции и поиска с запретами (по оси ординат – лучшее найденное значение ЦФ, по оси абсцисс – число вызовов ЦФ)

Полученный таким образом гибридный алгоритм сочетает в себе «исследовательские» способности метода дифференциальной эволюции и скорость сходимости поиска с запретами при движении в сторону локального минимума [46]. Как видно из графиков, во всех задачах показано улучшение по сравнению с оригинальными, негибридизированными, методами поиска с запретами и дифференциальной эволюции.

2.3 Метод структурно-параметрического синтеза

Одна из основных проблем обучения ИНС – застревание поиска в локальных минимумах. В алгоритме нелокального параметрического улучшения для борьбы с данной проблемой мы применяем подходы ГСП. В случае, когда точка оказалась в окрестности некоторого локального минимума, будем считать, что известные градиентные методы также позволяют улучшить решение в окрестности. В методе структурно-параметрического синтеза дополнительно будем пытаться сократить вычисления. Основная идея метода заключается в следующем: в процессе решения задачи все точки делятся на две отдельные группы, для каждой из которых моделируется распределение, зависящее от проверки возможности локального улучшения: перспективные точки улучшаются локально, для неперспективных (неулучшаемых) вызывается алгоритм нелокального параметрического улучшения с целью попадания в область притяжения локального минимума с меньшим значением, нежели текущее. И, наконец, если алгоритм нелокального параметрического улучшения терпит неудачу, производится модификация топологии. Этап определения перспективности станем далее называть *квалификационным* этапом. Также отдельно будем рассматривать

лучшую и худшую по значению ЦФ точки популяции. В Таблице 4 опишем действия для выделенных групп точек.

Таблица 4 – Описание действий с выделенными на квалификационном этапе группами точек

Группа точек	Описание группы и предлагаемые действия
Лучшая точка	Лучшая по значению ЦФ точка (найдена окрестность лучшего на текущий момент локального минимума) – пытаемся улучшить результат, оставаясь в зоне притяжения найденного локального минимума.
Перспективные точки	Точки, для которых показано наилучшее приращение на разведочном этапе локального спуска (найлены зоны притяжения локальных минимумов, сами минимумы еще не достигнуты) – пытаемся улучшить результат, оставаясь в зоне притяжения найденного локального минимума.
Неперспективные (застойные) точки	Точки, для которых приращение на разведочном этапе локального спуска отсутствует (в точках достигнут локальный минимум) – дальнейшее локальное улучшение бессмысленно – пытаемся применить алгоритм параметрического улучшения, а в случае его неудачи – алгоритм структурного синтеза.
Худшая точка	Худшая по значению ЦФ точка – точка подлежит замене – через данную точку в популяцию вводятся сети с модифицированной топологией. Если точка не была заменена, то пытаемся ее улучшить, оставаясь в зоне притяжения найденного локального минимума

Тогда в общем виде алгоритм может быть описан следующим образом:

1. Моделируем распределение вероятностей P_1 на n_1 -кратном произведении $W \times \dots \times W$, где $n_1 \geq 1$ данное целое. Устанавливаем номер итерации $t = 1$. Получаем n_t точек $\left\{ w_t^{(k)} \right\}_{k=1, \dots, n_t}$ из W .
2. Квалификационный этап. Производим некоторое небольшое фиксированное количество итераций локального спуска из каждой точки P_t , получив $\left\{ w_t^{(k)} \right\}_{k=1, \dots, n_t}$. Вычисляем значения ЦФ f в полученных точках

$\left\{ f_t^{(k)} = f\left(w_t^{(k)}\right) \right\}_{k=1, \dots, n_t}$. Выделяем *перспективные* и *неперспективные*

точки для дальнейшего локального улучшения, отдельно выделяем точки с наилучшим и наихудшим текущими значениями ЦФ. Получаем в итоге n_t точек

$$\left\{ w_t^{(k)} \right\}_{k=1, \dots, n_t} = \left\{ w_t^{(best)} \right\} \cup \left\{ w_t^{(worst)} \right\} \cup \left\{ w_t^{(k)} \right\}_{k \in Perspective} \cup \left\{ w_t^{(k)} \right\}_{k \notin Perspective}.$$

3. Производим некоторое фиксированное количество итераций локального спуска из лучшей точки $w_t^{(best)}$, получив $w_{t+1}^{(best)}$.
4. Для перспективных точек $\left\{ w_t^{(k)} \right\}_{k \in Perspective}$ производим некоторое количество итераций локального спуска пропорционально размеру улучшения на этапе квалификации, получив $\left\{ w_{t+1}^{(k)} \right\}_{k \in Perspective}$.
5. Полагаем f_t^{worst} в качестве $f_t^{worst} = f\left(w_t^{(worst)}\right)$.
6. Для каждой из неперспективных точек полагаем $\left\{ w_t^{(k)} \right\}_{k \notin Perspective}$:
 - 6.1. Пытаемся найти улучшение $f^{*(k)}$ посредством применения алгоритма параметрического улучшения.
 - 6.2. Если $f^{*(k)} < f\left(w_t^{(k)}\right)$, то $w_{t+1}^{(k)} = w^{*(k)}$ – переходим к шагу 7, в противном случае переходим к шагу 6.3.
 - 6.3. Получаем точку $w^{(k)}$ посредством применения алгоритма структурного синтеза $w_t^{(k)}$.
 - 6.4. Если $f\left(w^{(k)}\right) < f_t^{worst}$, то заменяем худшую сеть полученной модификацией $w_t^{worst} = w^{(k)}$ и полагаем $f_t^{worst} = f\left(w^{(k)}\right)$.

7. Если, $t = T_{\max}$, то переходим к шагу 8, в противном случае полагаем $t = t + 1$ и переходим к шагу 2.
8. Выход.

Рассматриваемое пространство поиска $W = \left\{ W^{l(0)}, \dots, W^{l(n_h^{\max})} \right\}$, имеет слож-

ную структуру, то есть поиск ведется в подпространствах $W^{l(i)} \subset W$ разной размерности, соответствующих сетям с $i \geq 0$ скрытых нейронов. Модификация топологии сети равносильна операции перехода из текущего подпространства в подпространство другой размерности. Поскольку в общем случае сеть с более сложной топологией способна доставить меньшее значение ЦФ, модификации, особенно усложняющие, следует проводить с осторожностью, иначе мы можем получить неконтролируемое усложнение сетевой топологии. Поэтому в нашей алгоритмической схеме мы на каждой итерации фактически допускаем появление только одной сети с модифицированной топологией, которая при этом заменяет худшую по значению ЦФ сеть популяции, если та, к тому же, показывает отсутствие приращения. Иллюстрация такого подхода приведена на Рисунке 13.

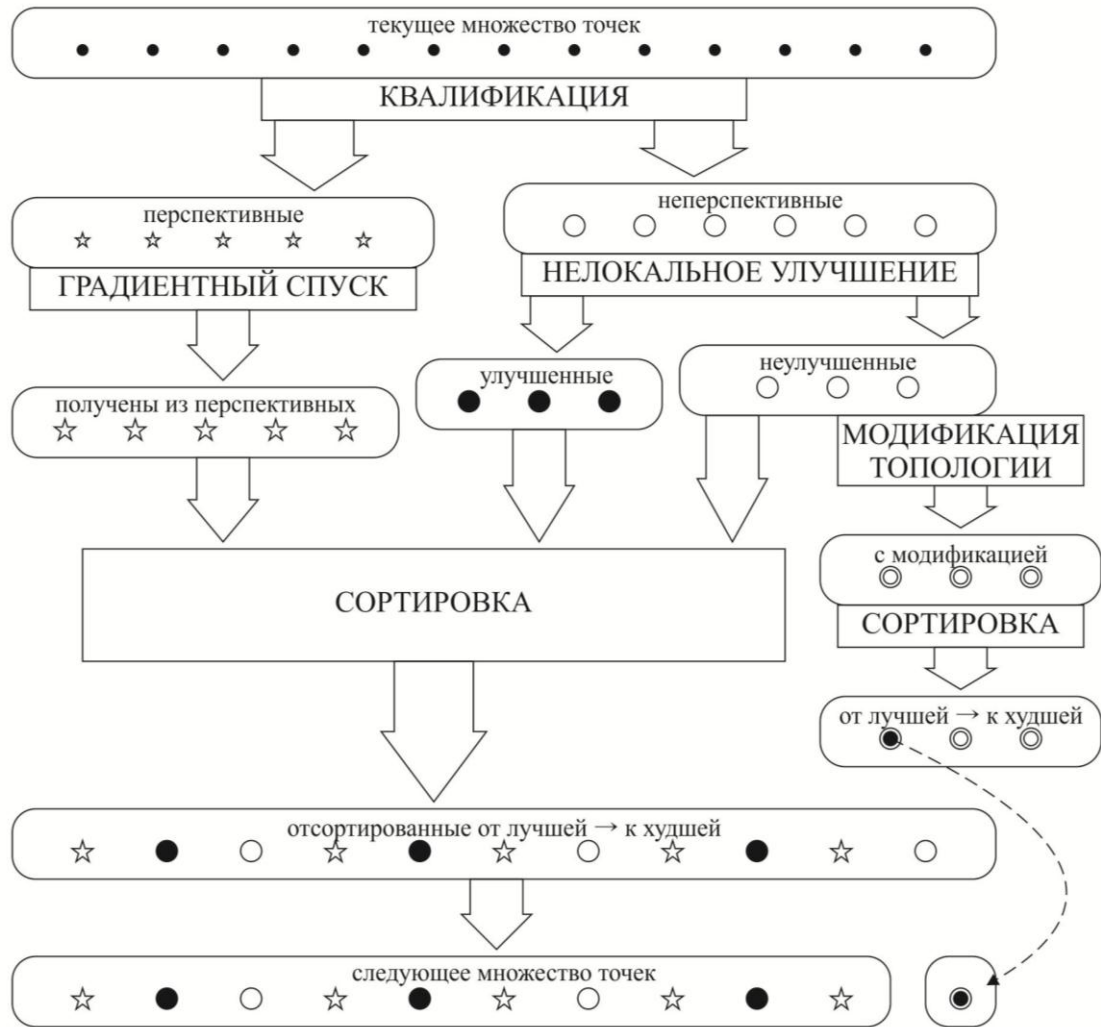


Рисунок 13 – Схема СПС с модификацией топологии

Размер популяции, не теряя общности, будем считать фиксированным на всех этапах.

Количество итераций квалификационного этапа, как показывает практика, достаточно брать равным 10, а количество итераций $i_t^{(k)}$ основного этапа будем высчитывать для каждой точки следующим образом. Пусть

$\Delta f_t^{(k)} = f(w_t^{(k)}) - f(w_t^{(k)})$ – приращение, полученное на этапе предварительного

локального спуска в k -той точке, $\bar{\Delta f}_t$ – среднее значение по всем приращениям

$\{\Delta f_t^{(k)}\}_{k=1, \dots, n}$. Тогда положим

$$i_t^{(k)} = \left\lfloor 100 \cdot \frac{\Delta f_t^{(k)}}{\bar{\Delta f}_t} \right\rfloor. \quad (13)$$

Оценка (13) позволяет выделять точки наиболее перспективные для осуществления локального спуска при наличии таковых. В случае если все точки имеют примерно равную «перспективность», благодаря (13) они все получат около 100 итераций локального спуска.

В результате проведения описанной процедуры худшая точка популяции будет заменена на лучшую по значению ЦФ точку, полученную посредством модификации топологии одной из тех неперспективных точек, для которых не дал улучшения алгоритм нелокального параметрического улучшения. Таким образом, каждая неперспективная точка популяции либо может быть улучшена (путем применения алгоритма нелокального параметрического улучшения), либо может остаться без изменений. Во втором случае путем применения операции модификации топологии к этой неперспективной сети будет получена новая точка, относящаяся к другому подпространству поиска. Лучшая из всех таких новых точек, полученных в результате применения операции модификации топологии к неперспективным сетям, заменит худшую точку популяции (если оценка первой будет лучше оценки последней). Неизменившаяся неперспективная точка на одной из следующих итераций станет худшей точкой популяции и в свою очередь будет заменена.

Дополнительно ускорить выход на необходимое число нейронов позволяет старт популяции из точек, принадлежащих подпространствам разной размерности, то есть различным топологиям сетей. На практике достаточно задавать число скрытых нейронов каждой сети случайной величиной, равномерно распределенной на интервале $[0, 3]$. Заметим, что сети с нулевым числом скрытых нейронов, будут заведомо хуже по значению ЦФ, чем сети с каким-либо отличным от нуля числом скрытых нейронов, однако их наличие в популяции на начальном этапе ведет к более интенсивному исследованию пространства поиска. Происходит это по той причине, что сеть без скрытых нейронов на некоторое первоначальное время становится лучшей сетью популяции, устанавливая нижнюю границу оценки ЦФ, поскольку интенсивнее перебирает доступные локальные минимумы (в силу меньшей размерности поискового пространства). Достигнутый рекордный

минимум ЦФ заставляет сети с бóльшим количеством нейронов более интенсивно осуществлять поиск с помощью глобальных процедур, не тратя время на локальный спуск. Таким образом, при наличии простых сетей поиск интенсифицируется, а итоговой лучшей сетью популяции становится, тем не менее, изначально более сложная сеть.

Критерием останова всего алгоритма может являться выполнение одного из следующих условий:

- 1) превышение максимального количества итераций;
- 2) достижение заданного значения ЦФ;
- 3) падение приращения ниже заданного значения.

На практике мы достаточно часто будем сталкиваться со случаями стагнации популяции, когда полученные с помощью локальных процедур хорошие решения с очень малой вероятностью могут быть улучшены с помощью процедур ГСП. Для решения такой проблемы определим процедуру *рестарта алгоритма*, позволяющую осуществлять перезапуск поиска при его стагнации. При этом лучшее из найденных на момент рестарта решений имеет смысл сохранить, т. е. перезапускать все сети популяции кроме лучшей. Возможно также перезапускать сети с какой-либо вероятностью или перезапускать только сети с нулевым приращением на квалификационном этапе и т. п. Критерий перезапуска – падение среднего приращения по результатам итерации или по результатам только частичной настройки до заданного пользователем значения. Очевидно, данный подход можно использовать в случае, если критерием останова всего алгоритма не является падение приращения ниже заданного. В комбинации с условием достижения заданного значения ЦФ рестарт может существенно повысить эффективность. Данный вариант следует выбирать в случае, когда имеется априорная информация о желаемом значении ЦФ.

В настоящей работе хорошие результаты показывает рестарт при падении приращения до 10^{-6} в среднем на квалификационном этапе (перезапускаются не все точки, а только показавшие нулевое приращение) и выбор в качестве кри-

терия останова всего алгоритма сложного условия: достижение заданного значения ЦФ или превышение заданного количества итераций.

Дополнительное ускорение сходимости может быть достигнуто, если после каждой итерации производить дополнительный локальный спуск из точек с наилучшим и наихудшим значениями ЦФ.

Разработанный метод представляет собой процедуру глобального поиска в пространстве МПРПС с различными топологиями, использующую методы глобального стохастического поиска, локального градиентного спуска и оснащенную механизмом последовательной модификации топологий сетей; далее в работе будем сокращенно обозначать его как метод структурно-параметрического синтеза (метод СПС, или просто – СПС).

2.4 Тестирование метода структурно-параметрического синтеза

2.4.1 Выбор тестовых задач

Исследование производительности алгоритмов настройки ИНС представляет собой нетривиальную задачу в смысле выявления универсальности разрабатываемых алгоритмов. Значительная часть исследований проводится в излишне проблемно-ориентированном ключе, когда ИНС выступает лишь в роли инструмента, а на первый план выходит успешность частного решения некоторой конкретной задачи. Другими словами, разрабатываемые методы настройки ИНС часто подгоняются под решение одной-двух тестовых задач и становятся бесполезными в широком применении. Потенциал полученных таким образом алгоритмов не раскрывается в полной мере, в результате чего исследования, как правило, не представляют интереса для пользователей.

В настоящей работе особое значение придается универсальности разработанного алгоритмического и программного обеспечения и практической его применимости, исходя из чего и будут подбираться задачи для тестирования.

Отдельные аспекты применения алгоритмов настройки ИНС могут быть исследованы как теоретически (получение верхних и нижних оценок параметров, оценка сходимости по вероятности и почти всюду и т. п.), так и статистически (с помощью эмпирических оценок на тестовых функциях). Поскольку в данной работе применяются методы стохастической природы, мы станем придерживаться эмпирического подхода к их тестированию, тем более что к настоящему моменту сформировался некоторый набор хороших стандартизованных тестов, позволяющих производить сравнительный анализ производительности алгоритмов.

Для тестирования алгоритмов используются задачи двух типов: искусственные модельные задачи и задачи на основе реальных данных. В качестве наиболее популярных модельных задач можно выделить задачу о функции четности (N-Parity Problem) [150; 151] и задачу о балансировании шестов (Pole-Balancing Prob-

lem) [66]. Моделирование функции четности и, в особенности, ее частный случай XOR представляют собой наиболее популярные модельные задачи классификации, которые не могут быть решены с помощью линейного разделения пространства входных признаков. Функция четности от n переменных представляет собой булеву функцию $f : \{0,1\}^n \rightarrow \{0,1\}$, где $f(x) = 1$ тогда и только тогда, когда число единиц вектора $x \in \{0,1\}^n$ четно. Иначе говоря, f определена как $f(x) = x_1 \oplus x_2 \oplus \dots \oplus x_n$. При использовании ИНС это требует наличия скрытых нейронов.

Задача о балансировании шестов представляет собой задачу из области теории управления, в которой требуется смоделировать контроллер для сложной и неустойчивой нелинейной системы. Роль такой системы в задаче играет движущаяся тележка, на которой расположено некоторое количество шестов (перевернутых маятников), и целью является удержание шестов от падения в течение максимально возможного времени посредством управления тележкой. В разновидностях задачи требуется балансировать одним или двумя шестами, тележка может двигаться в одном или двух измерениях, данные об угловых скоростях могут быть доступны или не доступны для управляющего тележкой контроллера.

Помимо двух упомянутых выше задач распространены задача об искусственном муравье [110], задача о двух спиралях [111], и др. Однако несмотря на имеющиеся достоинства (высокую сложность, проработанность постановок), искусственные модельные задачи имеют два существенных недостатка:

- 1) заведомая определенность возможности получения точного решения в силу наличия строгих априорных закономерностей;
- 2) отсутствие аналогий в реальных предметных областях.

Таким образом, очевидно, что применение при тестировании алгоритма только искусственных модельных задач не гарантирует подтверждения способностей разработанного алгоритма к решению задач реального мира. Поэтому в нашей работе обратимся к реальным задачам.

Крупнейшим в мире репозиторием реальных задач является UCI Machine Learning Repository [62], основанный в 1987 году Patrick Murphy и David Aha (Калифорнийский университет в Ирвине, Калифорния, США). На сегодняшний день репозиторий UCI включает в себя 238 наборов реальных данных из различных областей. Для этих наборов сформулированы задачи классификации, кластеризации и регрессии.

Репозиторий UCI многие годы находится в числе 100 наиболее цитируемых источников в области информатики, что является несомненным показателем его авторитетности (к настоящему времени на него ссылается более тысячи работ в ведущих журналах).

Однако заметим, что непосредственное использование наборов данных из репозитория для обучения ИНС неэффективно, поскольку первоначальное назначение репозитория – тестирование алгоритмов машинного обучения вообще, а не только алгоритмов обучения ИНС, поэтому имеющиеся в нем наборы данных допускают возможность различной их трактовки (могут быть использованы различным образом). При работе же с ИНС данная особенность лишает полученные результаты объективности.

При этом на основе UCI Machine Learning Repository создано несколько эталонных бенчмарков (специально отобранных и предобработанных наборов тестовых задач), предназначенных для тестирования именно ИНС, поэтому обратимся к одному из них. Наиболее авторитетным среди таких бенчмарков является PROBEN1 [134]. PROBEN1 включает в себя комплект наборов данных из репозитория UCI, постановки задач (из областей распознавания образов и аппроксимации функций) для этих наборов, подробное руководство по проведению тестирования, а также авторские решения всех представленных задач. Поскольку автор не пользовался никакими автоматизированными способами подбора топологии, далее применительно к сетям, описанным в сборнике, будем говорить о ручном способе подбора.

Итак, в настоящей работе для тестирования разрабатываемых алгоритмов настройки ИНС в качестве основного средства мы будем использовать сборник PROBEN1.

2.4.2 Описание задач

Сборник задач PROBEN1 включает в себя 11 задач классификации и 4 задачи на аппроксимацию функций. Подавляющее большинство задач реального мира (которые могут быть решены с применением ИНС) может быть сведено к подобным постановкам. Мы продемонстрируем возможности разработанного метода СПС на примере решения трех задач классификации и одной задачи на аппроксимацию. В сборнике PROBEN1 все задачи представлены в трех вариантах, отличающихся составом обучающего, валидационного и тестирующего множеств. В нашей работе мы, не теряя общности, будем использовать первые варианты разбивки для каждой задачи.

Диагностика типа грудной опухоли (Cancer). Набор данных по раку груди создан доктором В. Уолбергом из клиники университета Висконсина. Требуется решить задачу классификации злокачественных и доброкачественных опухолей на основе характеристик, полученных с помощью гистологических исследований [152]. Набор данных содержит 699 примеров. Из них 458 (65,5%) примеров описывают доброкачественные опухоли и 241 (34,5%) – злокачественные.

Входные параметры включают в себя 9 непрерывных на $[0,1]$ переменных:

- 1) толщина слоя;
- 2) степень однородности размеров клеток;
- 3) степень однородности формы клеток;
- 4) степень маргинальности спайки (слипание клеток);
- 5) размер эпителиальных одиночных клеток;
- 6) количество свободных ядер;
- 7) хроматин;
- 8) нормальные ядра;
- 9) митозы.

Выходной параметр представляет собой пару булевых переменных; при получении результата (0,1) опухоль признается доброкачественной, при получении результата (1,0) – злокачественной.

Идентификация типа стекла (Glass). Имеются данные по образцам (крупицам и осколкам) стекла [75; 81]. Требуется решить задачу классификации типов стеклянных осколков (широко распространённую в судебно-следственной практике) по его химическому составу и показателю преломления. Входные параметры включают 9 непрерывных на интервале $[0,1]$ переменных:

- 1) показатель преломления;
- 2) весовая доля оксида (оксидов) натрия, %;
- 3) весовая доля оксида (оксидов) марганца, %;
- 4) весовая доля оксида (оксидов) алюминия, %;
- 5) весовая доля оксида (оксидов) кремния, %;
- 6) весовая доля оксида (оксидов) калия, %;
- 7) весовая доля оксида (оксидов) кальция, %;
- 8) весовая доля оксида (оксидов) бария, %;
- 9) весовая доля оксида (оксидов) железа, %.

Булев вектор выходных переменных описывает один из 6 возможных типов стекол: состояние данного вектора $b_i = \begin{cases} 0, & i = k \\ 1, & i \neq k \end{cases}$ – соответствует k -тому типу стекла. Всего в наборе содержится 214 примеров (см Таблицу 5).

Таблица 5 – Состав набора данных Glass

Тип стекла	Количество примеров, шт.	Количество примеров, %
Оконное термополированное стекло	70	32,7103
Оконное нетермополированное стекло	76	35,5140
Термопол. стекло окна трансп. средства	17	7,9439
Стекло тары	13	6,0748
Стекло посуды	9	4,2056
Стекло фар автомобиля	29	13,5514

Диагностика заболеваний диабетом индейского племени пима (Diabetes). База данных содержит результаты медицинских наблюдений за женщинами индейского племени Пима в возрасте старше 21 года. Среди индейцев племени пима, живущих на территории США зафиксирован самый высокий в мире процент заболеваемости сахарным диабетом 2 типа. Данная проблема лишь частично обусловлена генетически, поскольку заболеваемость этим типом диабета практически отсутствует среди индейцев-пима, живущих в Мексике. Спусковым механизмом для данного заболевания стало разрушение традиционной экономики племени и переход к новому питанию. Используя данные наблюдений американского Института диабета и заболеваний почечно-пищеварительного тракта [146], требуется диагностировать диабет по косвенным характеристикам. Всего имеется 768 примеров, 500 (65,1%) примеров описывают отсутствие диабета и 268 (34,9%) – наличие диабета. Входные параметры включают 8 непрерывных на интервале $[0,1]$ переменных:

- 1) число беременностей;
- 2) концентрация плазма-глюкозы в стоматическом тесте;
- 3) диастолическое давление крови (mm Hg);
- 4) толщина кожи на складке трехглавой мышцы (mm);
- 5) содержание сыворотки инсулина (микро-U/ml);
- 6) индекс веса тела (вес в килограммах/(рост в метрах)²);
- 7) наличие диабета в роду;
- 8) возраст (в годах).

Выходной параметр представляет собой пару булевых переменных; при получении результата (0,1) диагностируется наличие диабета, при получении результата (1,0) – отсутствие диабета.

Прогнозирование сердечных заболеваний (HeartAC). Данные по риску сердечных заболеваний были собраны доктором Р. Детрано, в Кливленд Клиник [103]. В задаче требуется выяснить, не уменьшена ли в диаметре по крайней мере одна из четырех основных сердечных камер более чем на 50%,

и определить вероятность развития заболевания сердечно-сосудистой системы. Данные представлены в 303 наблюдениях. В 164 (54%) случаях не диагностировано уменьшение диаметра ни одной из четырех камер, в 139 (46%) случаях диаметр хотя бы одной камеры уменьшен.

В 35 непрерывных на интервале $[0,1]$ входных переменных описываются разнообразные показатели: пол, возраст, отношение к курению, жалобы на боли, кровяное давление и проч.; одна непрерывная на интервале $[0,1]$ выходная переменная описывает вероятность развития заболевания сердечно-сосудистой системы.

2.4.3 Результаты решения задач

Для каждой из описанных задач будем оценивать следующие показатели качества получаемых решений:

- 1) ошибка обучения;
- 2) точность получаемого решения (ошибка на тестовом множестве);
- 3) число скрытых нейронов;
- 4) число связей;
- 5) число эпох, затраченных на обучение.

Поскольку результаты работы нашего алгоритма во многом зависят от случая (в силу стохастической природы его составляющих), целесообразным будет производить многократные запуски алгоритма для каждой задачи и вычислять минимальное, максимальное, среднее значения и стандартное отклонение всех интересующих нас параметров итоговых решений по всем запускам. Количество запусков для каждой задачи установим равным 100, согласно рекомендациям сборника PROBEN1. Полученные результаты будем сравнивать с эталонными результатами, описанными в сборнике PROBEN1 (см. Таблицу 6). При этом сети, используемые в PROBEN1, представляют собой полносвязные многослойные персептроны, поэтому число скрытых нейронов и число связей в них является фиксированным и приводится без значения стандартного отклонения.

Для ошибок на обучающем и тестовом множествах в PROBEN1 фиксировались среднее и стандартное отклонение по 60 запускам (минимальное и максимальное значения не фиксировались). Настройка весов связей сетей в PROBEN1 происходила с использованием алгоритма RPROP [136].

Таблица 6 – Результаты вычислительных экспериментов на задачах PROBEN1

Параметр	Значения параметра	Cancer		Glass		Diabetes		HeartAC	
		СПС	PROB	СПС	PROB	СПС	PROB	СПС	PROB
Ошибка на тест. множ-ве	мин.	1,8975	–	9,1659	–	16,6084	–	2,4051	–
	макс.	4,5092	–	9,6524	–	18,4625	–	4,7781	–
	средн.	2,5133	1,6000	9,4581	9,7500	17,5192	17,3000	3,5833	2,8200
	ст. откл.	1,1253	0,4100	0,1979	0,4100	0,5790	1,9100	0,8788	0,2200
Ошибка на обуч. множ-ве	мин.	3,3432	–	7,6660	–	14,7117	–	4,2834	–
	макс.	3,7641	–	9,2251	–	16,2692	–	4,6163	–
	средн.	3,5172	2,8700	8,4396	7,6800	15,7503	14,7400	4,2821	3,8600
	ст. откл.	0,2113	0,2700	0,6945	0,7900	0,4196	2,0300	0,2849	0,3200
Число скрытых нейронов	мин.	2	–	2	–	1	–	2	–
	макс.	5	–	4	–	2	–	4	–
	средн.	2,7	6	2,4	24	1,6	32	3	2
	ст. откл.	1,059	–	0,8944	–	0,5164	–	0,8944	–
Число связей	мин.	28	–	65	–	18	–	71	–
	макс.	33	–	101	–	32	–	140	–
	средн.	30,6	100	73,2	572	25	370	106,2	110
	ст. откл.	2,0736	–	15,6269	–	4,2439	–	25,5648	–
Число эпох обучения	мин.	5	–	2	–	8	–	1	–
	макс.	60	–	11	–	65	–	10	–
	средн.	27,2	–	6,6	–	19	–	4,2	–
	ст. откл.	21,9249	–	4,3359	–	17,4741	–	3,2342	–

На основании результатов проведенных экспериментов можно сделать следующие выводы:

- 1) Среднее значение ошибки на тестовом множестве, достигаемое при использовании метода СПС, незначительно уступает значению ошибки, получаемому при ручном подборе, при решении всех задач кроме задачи Glass – задача Glass при этом обладает наименьшей размерностью;

- 2) Минимальное значение ошибки на тестовом множестве, достигаемое при использовании метода СПС, улучшает значение ошибки, получаемое при ручном подборе, при решении всех задач кроме задачи Cancer;
- 3) Сети, получаемые при использовании метода СПС, отличаются простой топологией.

Таким образом, при использовании СПС мы получаем сети с очень простой топологией (с двумя-тремя скрытыми нейронами), незначительно уступающие по значению ошибки на тестовом множестве сетям с топологией, подобранной автором сборника PROBEN1. При этом для случая лучшей по всем запускам сети, как правило, можно наблюдать некоторое (небольшое) снижение величины ошибки по сравнению со значением ошибки, получаемым при обучении сетей с вручную подобранной топологией.

Заметим, что временные затраты на обучение при использовании метода СПС могут быть выше, чем затраты на обучение единственной сети с фиксированной топологией. Однако при этом следует учесть следующие факторы:

- 1) На подбор топологии и дальнейшие поиски хорошей стартовой точки затрачивается время исследователя, а не машинное время;
- 2) При неудачном ручном подборе топологии и/или начальных значений весов связей вычислительная сложность может значительно вырасти.

Примеры полученных по каждой задаче сетей и значения весов их связей представлены в Приложении А.

Приведем историю поиска лучших решений для каждой задачи (см. Таблицу 7). Для каждой операции с сетью описаны текущее значение ЦФ (ЦФ), приращение ЦФ в результате данной операции (Приращ.), текущее число скрытых нейронов (Нейр.) и вид операции (Операция).

Таблица 7 – История поиска лучших решений

а) задача Cancer

№	ЦФ	Приращ.	Нейр.	Операция	№	ЦФ	Приращ.	Нейр.	Операция
0	209,3375	0,0000	2	Иниц.	5	16,6619	-1,3182	2	Квалиф.
1	48,3193	-161,0180	2	Квалиф.	6	4,4109	-12,2511	2	Персп. спуск
2	45,6794	-2,6399	2	Персп. спуск	7	4,1059	-0,3049	2	Лучш. спуск
3	45,6751	-0,0043	2	Квалиф.	8	4,0851	0,0000	2	Итог. спуск
4	17,9801	-27,6950	2	ГСП					

б) задача Glass

№	ЦФ	Приращ.	Нейр.	Операция	№	ЦФ	Приращ.	Нейр.	Операция
0	115,8118	0,0000	4	Иниц.	12	107,5027	-0,0035	4	Персп. спуск
1	109,4165	-6,3954	4	Квалиф.	13	107,4933	-0,0094	4	Квалиф.
2	107,9053	-1,5112	4	Персп. спуск	14	107,4877	-0,0057	4	Персп. спуск
3	107,7830	-0,1223	4	Квалиф.	15	107,4763	-0,0114	4	Квалиф.
4	107,7503	-0,0327	4	Персп. спуск	16	107,4649	-0,0114	4	Персп. спуск
5	107,6046	-0,1457	4	Квалиф.	17	107,3500	-0,1148	4	Квалиф.
6	107,5467	-0,0579	4	Персп. спуск	18	107,2110	-0,1390	4	Персп. спуск
7	107,5372	-0,0095	4	Квалиф.	19	107,2103	-0,0007	4	Квалиф.
8	107,5350	-0,0022	4	Персп. спуск	20	37,7449	-11,9346	4	Квалиф.
9	107,5237	-0,0113	4	Квалиф.	21	10,2123	-27,5326	4	Персп. спуск
10	107,5176	-0,0062	4	Персп. спуск	22	9,8059	0,0000	4	Итог. спуск
11	107,5062	-0,0114	4	Квалиф.					

в) задача Diabetes

№	ЦФ	Приращ.	Нейр.	Операция	№	ЦФ	Приращ.	Нейр.	Операция
0	49,9924	0,0000	2	Иниц.	11	23,8235	-0,0793	2	Персп. спуск
1	49,9885	-0,0040	2	Квалиф.	12	23,8203	-0,0032	2	Квалиф.
2	49,8824	-0,1060	2	Худш. спуск	13	23,7686	-0,0517	2	Персп. спуск
3	49,8824	0,0000	2	Квалиф.	14	23,7254	-0,0432	2	Квалиф.
4	27,6657	-22,2167	2	ГСП	15	23,5960	-0,1294	2	Персп. спуск
5	24,9584	-2,7073	2	Квалиф.	16	23,5730	-0,0230	2	Квалиф.
6	23,9298	-1,0285	2	Персп. спуск	17	19,1052	-4,4678	2	Персп. спуск
7	23,9282	-0,0016	2	Квалиф.	18	17,4989	-1,6063	2	Квалиф.
8	23,9265	-0,0017	2	Квалиф.	19	14,9779	-2,5210	2	Персп. спуск
9	23,9263	-0,0002	2	Персп. спуск	20	14,9153	-0,0626	2	Лучш. спуск
10	23,9029	-0,0234	2	Квалиф.	21	14,9153	0,0000	2	Итог. спуск

Окончание таблицы 7

г) задача HeartAC

№	ЦФ	Приращ.	Нейр.	Операция	№	ЦФ	Приращ.	Нейр.	Операция
0	49,4550	0,0000	2	Иниц.	8	4,9350	-0,3611	2	Квалиф.
1	28,4237	-21,0313	2	ГСП	9	4,5423	-0,3927	2	Персп. спуск
2	16,1986	-12,2252	2	Квалиф.	10	4,5418	-0,0006	2	Квалиф.
3	7,9689	-8,2297	2	Персп. спуск	11	4,5371	-0,0047	2	Квалиф.
4	7,4506	-0,5183	2	Квалиф.	12	4,5187	-0,0184	2	Персп. спуск
5	6,3382	-1,1124	2	Персп. спуск	13	4,4746	-0,0441	2	Лучш. спуск
6	6,2199	-0,1183	2	Квалиф.	14	4,4746	0,0000	2	Итог. спуск
7	5,2961	-0,9239	2	Персп. спуск					

В рамках разработанного метода СПС производим следующие операции с сетью (в скобках приведены сокращения, используемые для обозначения операций в таблице):

- 1) Инициализация сети («Иниц.») – выполняется для каждой точки в начале алгоритма (задание начальных значений весовых коэффициентов);
- 2) Квалификационный локальный спуск («Квалиф.») – выполняется для всех точек в начале каждой итерации алгоритма (число шагов небольшое и фиксировано до начала алгоритма);
- 3) Локальный спуск («Персп. спуск») – применяется к перспективным точкам (число шагов зависит от значения приращения, полученного на этапе последней квалификации);
- 4) Локальный спуск («Лучш. спуск») – применяется к лучшей в конце очередной итерации точке популяции (число шагов задается до начала алгоритма);
- 5) Локальный спуск («Худш. спуск») – применяется к худшей после очередной квалификации точке популяции (число шагов задается до начала алгоритма);
- 6) Алгоритм нелокального параметрического улучшения («ГСП») – применяется к неперспективным точкам (параметры фиксированы и задаются до начала алгоритма);

- 7) Модификация топологии («Модиф. топологии») – применяется к неперспективным точкам, для которых не удалось добиться улучшения путем применения алгоритма нелокального параметрического улучшения (параметры фиксированы и задаются до начала алгоритма);
- 8) Рестарт («Рестарт») – применяется к точкам с нулевым приращением на этапе квалификации (при падении среднего по популяции приращения ниже заданного до начала алгоритма значения)
- 9) Локальный спуск («Итог. спуск») – применяется по окончании работы алгоритма (при выполнении критерия останова) к лучшей сети.

Из истории поиска лучших решений видно, что процедура локального спуска из точки за некоторое небольшое количество итераций находит локальный минимум. Также видно, что именно применение процедуры ГСП выбивает точки из текущего локального минимума и приводит их в область меньшего локального минимума, что и доказывает эффективность применения данного метода.

Однако, кроме того, анализ решений задач показывает, что для лучших сетей, полученных в ходе вышеописанных решений, не возникло необходимости применения процедур модификации топологии или рестарта. Для оценки целесообразности операции модификации топологии была проведена серия экспериментов, в которой две из вышеприведенных задач решались без использования модификации топологии (см. Таблицу 8). При отказе от модификаций топологии происходит очевидное ухудшение всех показателей. Таким образом, подтверждается эффективность применения данной операции: модификации топологии могут не участвовать в развитии лучшей итоговой сети, однако они оказывают непосредственное влияние на общую динамику развития всей популяции.

Таблица 8 – Результаты вычислительных экспериментов без использования модификации топологии (задачи Cancer и Glass)

Параметр	Значения параметра	Cancer		Glass	
		С модиф. топологии	Без модиф. топологии	С модиф. топологии	Без модиф. топологии
Ошибка на тестовом множестве	мин.	1,8975	3,2376	9,1659	9,4359
	макс.	4,5092	7,2638	9,6524	14,0641
	средн.	2,5133	5,1202	9,4581	11,5646
	ст. откл.	1,1253	1,4429	0,1979	1,9204
Ошибка на обучающем множестве	мин.	3,3432	3,3200	7,6660	10,2323
	макс.	3,7641	5,1146	9,2251	12,3645
	средн.	3,5172	4,1470	8,4396	11,3923
	ст. откл.	0,2113	1,1931	0,6945	0,8267
Число скрытых нейронов	мин.	2	2	2	2
	макс.	5	5	4	5
	средн.	2,7	3,3	2,4	3,6
	ст. откл.	1,059	1,3375	0,8944	1,2156
Число связей	мин.	28	24	65	62
	макс.	33	39	101	110
	средн.	30,6	32,1	73,2	80,4
	ст. откл.	2,0736	5,5267	15,6269	52,9395
Число эпох обучения	мин.	5	4	2	6
	макс.	60	92	11	74
	средн.	27,2	46,2	6,6	43
	ст. откл.	21,9249	51,6555	4,3359	39,1121

Среди преимуществ разработанного алгоритма необходимо отметить явную экономию вычислительных затрат, обусловленную понижением уровня топологической сложности сетей. Напомним, что скорость одной итерации алгоритма обучения (параметризации) одной нейросетевой модели находится в прямой зависимости от топологии, т. е. размерности решаемой задачи многомерной минимизации. Масштаб такой экономии в разработанном методе тем заметнее, что скорость вычислений остается высокой, даже несмотря на использование многоточечной схемы, влекущей за собой многократное увеличение количества расчетов.

Для измерения скорости проведем эксперименты на задачах Glass и Diabetes. Выбор задач обусловлен преимуществом разработанного метода СПС в получении уровня топологической сложности (см. Таблицу 8). Для задачи HeartAC превосходство ручного подбора в скорости параметрической оптимизации очевидно.

Будем сравнивать по скорости сходимости следующие операции:

- 1) Подбор весов сетей фиксированной топологии, описанной в PROBEN1;
- 2) Подбор весов сетей фиксированной топологии, предварительно найденной с помощью СПС (время на подбор топологии не учитывалось);
- 3) Подбор и весов, и топологии с помощью СПС.

В первых двух случаях начальное приближение будем выбирать случайно, моделированием равномерного распределения значений весов связей на допустимом интервале, и для подбора весов станем использовать разработанный алгоритм параметрического улучшения.

Результаты вычислительных экспериментов представлены на Рисунке 14.

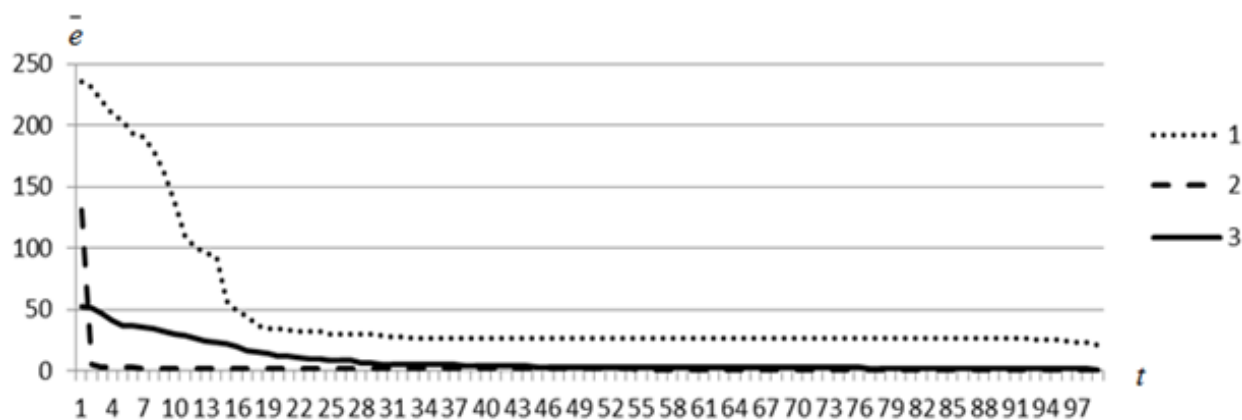


Рисунок 14 – Результаты вычислительных экспериментов по измерению скорости сходимости операций

Из Рисунка 14 видно, что снижение среднего значения функции ошибки \bar{e} в эксперименте 2 происходит значительно быстрее, чем в эксперименте 1 и эксперименте 3. Лучший старт для эксперимента 3 обусловлен применением популяционного подхода и небольшой топологией, благодаря чему в среднем старт чаще происходил из более хорошего локального минимума. Плохой старт и медленная настройка для эксперимента 1 объясняются громоздкой топологией

и, следовательно, более случайным характером подбора хорошего начального решения и трудоемкостью дальнейшего решения задачи оптимизации большей размерности. Начальное значение и крутое падение графика функции ошибки в эксперименте 2 обусловлено соответственно стартом из единственной точки и дальнейшим использованием метода СПС.

Из проведенных экспериментов можно сделать окончательный вывод о предназначении разработанного метода: основное преимущество СПС заключается в решении проблем подбора структуры и выбора метода обучения, при этом в некоторых случаях дополнительно может быть достигнуто значительное ускорение в вычислениях, связанное с топологической простотой синтезируемых моделей.

2.5 Выводы по главе

Для стратегии последовательного выбора операции модификации топологии с приоритетом операций упрощения предложено улучшение, уточняющее описанную в Главе 1 популяционную схему и нацеленное на интенсификацию исследования каждого уровня топологии: при успехе модификации топологии заменяется не родительская точка, а наихудшая точка популяции.

Произведено исследование различных подходов глобального стохастического поиска, не использующих градиент. Определены подходы, показывающие хорошие результаты при спуске в окрестности к минимуму (поиск с запретами) и обладающие глобальными поисковыми свойствами (дифференциальная эволюция). На их основе разработан новый алгоритм глобального поиска. В результате вычислительных экспериментов показано, что новый алгоритм превосходит оба

родительских подхода. В настоящей работе разработанный алгоритм используется в качестве алгоритма нелокального параметрического улучшения МПРПС.

Предложен новый метод структурно-параметрического синтеза, использующий разработанные алгоритм нелокального параметрического улучшения и стратегию модификации топологии, конкретизирующий общую схему структурно-параметрического синтеза, описанную в Главе 1.

Для тестирования разработанного метода использовался известный бенчмарк PROBEN1, созданный на основе данных UCI Machine Learning Repository. Показано превосходство разработанного метода перед ручным подбором топологии и чисто градиентными методами.

Показано, что, несмотря на использование популяционной схемы синтеза, скорость работы разработанного программного обеспечения является высокой. Данный факт связан с небольшой топологической сложностью сетей, полученных с помощью метода СПС.

Основное преимущество разработанного метода заключается в решении проблемы подбора структуры и выбора метода обучения искусственной нейронной сети при решении практических задач, при этом в большинстве случаев дополнительно может быть достигнуто значительное ускорение в вычислениях, связанное с топологической простотой синтезируемых моделей.

ГЛАВА 3. ПРОГРАММНЫЙ КОМПЛЕКС И РЕШЕНИЕ ПРАКТИЧЕСКИХ ЗАДАЧ

3.1 Описание программного комплекса

Общие сведения. Программный комплекс нейросетевого моделирования, разработанный на основе метода структурно-параметрического синтеза, предназначен для решения прикладных задач, допускающих постановку в форме задач классификации и регрессии на основе эмпирических данных.

Функциональное назначение. Программный комплекс состоит из взаимодействующих модулей различного назначения, использующих ряд разработанных библиотек классов, нацеленных на выполнение единой задачи построения нейросетевой модели в виде МПРПС по имеющимся данным наблюдений за поведением объекта или процесса. Программный комплекс может работать в режиме «Эксперимент», предназначенном для сбора и анализа информации о структурно-параметрическом синтезе персептронных моделей, либо для создания таких моделей. Также в комплексе предусмотрен режим «Эксплуатация», целью которого является расчет прогнозных значений по описанию модели, созданной на основе какой-либо выборки данных.

Логическая структура. Описание логической структуры представлено на Рисунке 15. Основные модули комплекса описаны в Таблице 9.

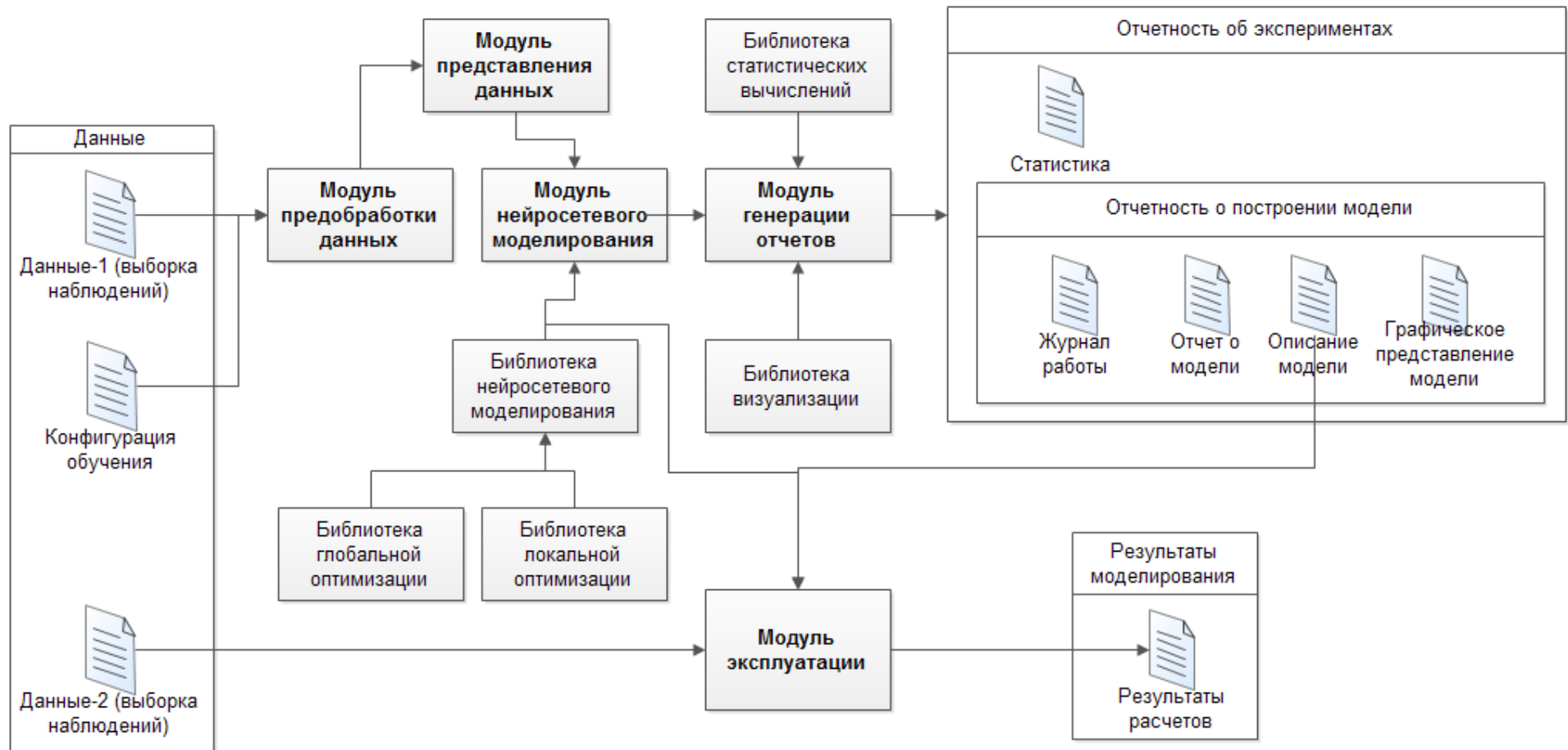


Рисунок 15 – Логическая структура разработанного программного комплекса

Таблица 9 – Описание модулей программного комплекса

Модуль и его файловый состав	Основные методы и структуры данных
Модуль представления данных (Problem.h, Problem.cpp)	Класс Problem считывает и хранит данные о решаемой задаче, включая обучающую и тестовую выборки.
Модуль предобработки данных (PreprocessingLib.h)	Функции библиотеки применяются к объекту класса Problem, производят предварительную обработку данных: шкалирование, устранение дублирующихся и неинформативных наблюдений, обнаружение пропусков данных.
Модуль нейросетевого моделирования (SPS, SPS.cpp)	Модуль содержит реализацию разработанного метода структурно-параметрического синтеза.
Библиотека нейросетевого моделирования (Ann.h, Ann.cpp)	Класс Ann содержит описание нейросетевой модели, включает методы инициализации (полностью, случайным образом, случайным образом со считыванием из конфигурационного файла некоторых параметров и с полным считыванием сохраненной ранее сети из файла), методы расчета ошибки, пакетного и попримерного градиента, обучающие методы простого обратного распространения ошибки и стохастического градиента. Последние два метода возвращают объект класса TrainReport – для самостоятельного применения.
Библиотека глобальной оптимизации (Population.h, Population.cpp, sa.h, ts.h, ga.h, de.h, tsde.h)	Основная структура данных класса Population – это <code>std::vector<Ann> pop</code> (контейнер, содержащий объекты класса Ann). Классы sa, ts, ga, de, tsde содержат соответственно реализацию методов поиска с запретами, имитации отжига, генетических алгоритмов, дифференциальной эволюции, разработанного метода нелокального параметрического улучшения.

Окончание таблицы 9

Модуль и его файловый состав	Основные методы и структуры данных
Библиотека локальной оптимизации (LocalOptLib.h)	Функции библиотеки применяются к объекту класса Ann и включают алгоритмы градиентной оптимизации, а также обслуживающие процедуры (кроме простейших алгебраических операций). Основные функции могут возвращать как объект класса TrainReport – применяться самостоятельно, так и объект класса void (для использования в составе глобальных методов).
Модуль генерации отчетов (TrainReport.h, TrainReport.cpp, TrainReportPopulation.h, TrainReportPopulation.cpp)	Содержит описание процесса обучения для единственной сети и для метода СПС, методы обработки и вывода результатов с различной детализацией.
Библиотека статистических вычислений (StatsUtils.h)	Функции реализуют расчеты простейших статистик (среднее, минимум, максимум, процентиля) и применяются к базовым типам C++ STL. Используются методами классов TrainReport и TrainReportPopulation.
Библиотека визуализации (GrapUtils.cpp)	Функции модуля генерируют описание сетей на языке DOT для дальнейшей визуализации с использованием сторонних библиотек (например, GraphViz ¹)
Модуль эксплуатации (Model.cpp)	Приложение считывает описание нейросетевой модели и пакет примеров, выводит информацию о результатах применения модели к примерам.
Общие утилиты (Utils.h)	Различные утилиты, шаблоны вывода, конвертации различных типов данных STL, а также некоторые компараторы.

¹ Пакет утилит для визуализации графов, описанных на языке DOT, и различного дополнительного программного обеспечения для визуализации структурированных данных – <http://www.graphviz.org/>

Технические средства. Разработка программного комплекса производилась полностью на языке C++ (стандарт C++11). Модули программного комплекса скомпилированы под MinGW 4.7.1 и могут запускаться под 32/64-битные операционные системы MS Windows. Имеющийся исходный код может быть перекомпилирован под любую операционную систему, для которой портирован компилятор MinGW.

В качестве генератора псевдослучайных чисел в проекте используется Вихрь Мерсенна [123], разработанный в 1997 году японскими учёными М. Мацумото и Т. Нисимура и обеспечивающий быструю генерацию высококачественных псевдослучайных чисел.

Вычислительные затраты при реализации поисковых алгоритмов прежде всего можно сократить за счет снижения стоимости вычисления ЦФ. При решении задачи оптимизации ИНС существенного уменьшения затрат удалось добиться за счет снижения стоимости вычисления экспоненты, что связано с возможностью представления через нее гиперболического тангенса:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1}. \quad (14)$$

Очевидно, для вычисления значения (14) достаточно однократного вычисления экспоненты от аргумента $2x$. Заметим, что при работе с ИНС для нас было бы достаточно использования какой-либо аппроксимации экспоненты, позволяющей ускорить вычисления.

Реализация экспоненты входит в стандартную поставку любого языка программирования, однако, как правило, в них она достаточно медленна, что объясняется преследованием целей совместимости с устаревшими архитектурами ЭВМ и сохранения высокой точности вычислений. В отличие от распространенного подхода, при котором используются таблицы подстановки с линейной интерполяцией экспоненты, в нашей работе мы опираемся на подход Н. Шраудольфа [138], основанный на манипуляции с компонентами стандартного представления в ЭВМ числа с плавающей точкой (используются векторные команды Streaming SIMD Extensions).

Streaming SIMD Extensions (SSE) – это векторные команды с плавающей запятой, выполняемые процессором в специальном блоке. Система команд SSE представляет собой развитие системы команд MMX (MultiMedia eXtensions – мультимедийные расширения). MMX предлагает работу с целочисленными векторами с количеством элементов от 1 до 8, при этом используются 64-разрядные регистры MMX, физически размещаемые в регистрах сопроцессора с плавающей запятой.

В системе команд SSE (а также SSE2, SSE3 и SSE4) используются 128-битные специальные регистры XMM и отдельные операционные устройства. Допускается обработка представлений как с плавающей, так и с фиксированной запятой.

Для работы с SSE необходимо использовать либо команды ассемблера, либо т. н. «интринсики» (intrinsic) – специальные мини-функции на языке высокого уровня (в данной работе – C++), позволяющие напрямую работать с регистрами SSE и выполнять SSE-операции. За одну операцию SSE параллельно выполняются либо четыре операции одинарной точности с плавающей запятой (float), либо две операции двойной точности (double).

Ранее SSE-команды также применялись при реализации передаточных функций посредством использования устаревшей к настоящему моменту команды возведения числа 2 в дробную степень [59; 131].

В Таблице 10 приведены результаты следующего вычислительного эксперимента: производилось многократное (по 100 запусков) вычисление трех реализаций экспоненты для каждого из 10^7 случайных чисел, равномерно распределенных на интервале $[-w_a, w_a]$, где $w_a = 22.5$.

Таблица 10 – Сравнение производительности реализаций вычисления экспоненты²

Значения параметра	std::exp(), мс	table_exp(), мс	fast_exp(), мс
Минимальное	928,00	872,00	552,00
Максимальное	1108,00	1046,00	671,00
Среднее	1007,81	959,75	604,21
Станд. отклонение	33,52	39,33	25,26

Из Таблицы 10 видно, что при использовании `fast_exp()` быстродействие улучшается почти в два раза. Также заметим, что значение $\tanh(w)$ оказывается в окрестности единицы с некоторой гарантированной точностью $\varepsilon > 0$ при достаточно небольших значениях аргумента w (Таблица 11).

Таблица 11 – Значения аргумента w , достаточные для достижения требуемой точности ε

Требуемая точность ε	Значение аргумента w
10^{-2}	2,6466
10^{-3}	3,8002
10^{-4}	4,9517
10^{-5}	6,1030
10^{-6}	7,2543

Отсюда следует, что можно сократить интервал вычисления активационной функции соответственно размерности пространства откликов сети. Например, при единственном отклике сети и требуемой точности $\varepsilon = 10^{-6}$ можно реализовать вычисление активационной функции так, как показано на Рисунке 16. Значение 7,2543 соответствует последней строке Таблицы 11.

² Для следующих характеристик рабочей станции и программной реализации: процессор Intel Core 2 Duo E4500, всякий раз было задействовано одно ядро с тактовой частотой 2,20 ГГц и реальной частотой шины 200 МГц, операционная система – MS Windows XP Pro SP3, компилятор MinGW 4.7.2

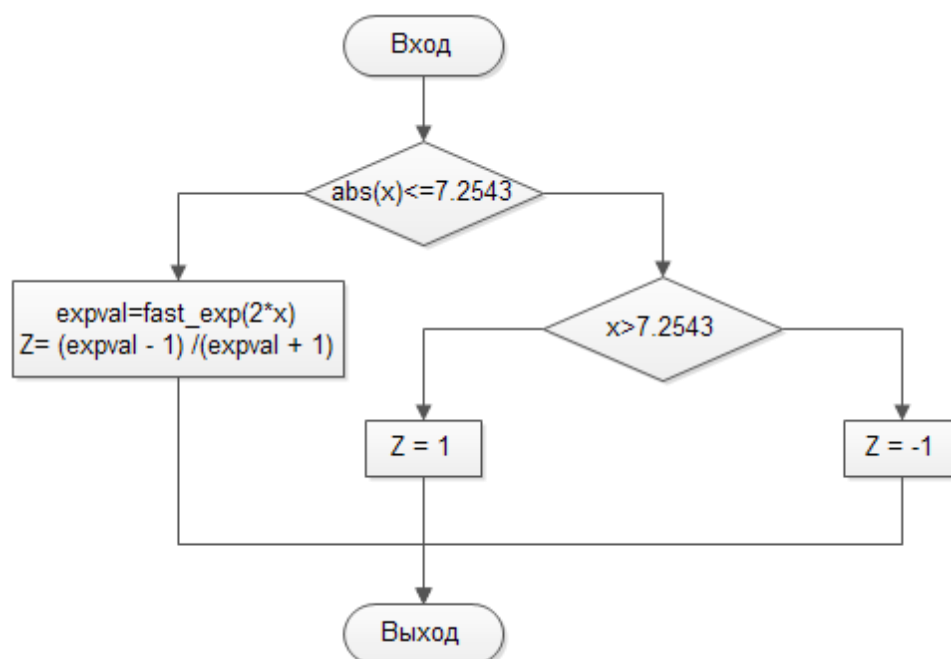


Рисунок 16. Вычисление активационной функции $Z = \varphi(x)$

В качестве вспомогательного технического средства в программном комплексе используется библиотека визуализации графов GraphViz (для визуализации структуры нейросетевых моделей).

Входные данные. Ввод данных осуществляется с помощью файла конфигурации и файла данных. Формат конфигурационного файла описан в Таблице 12.

Таблица 12 – Формат конфигурационного файла

Тип и имя параметра	Описание	Рекомендуемое значение
string fname	Название задачи	Определяется пользователем
double wa	Граничное значение изменения весов	22,5
int p_size	Размер популяции	10
int start_p_size	Начальный размер популяции	10
double start_error	Величина ошибки, до достижения которой будут производиться основные итерации	Определяется пользователем
int expn_num	Количество экспериментов	100
int over_num	Число вызовов процедуры локального спуска для итоговой подстройки лучшей сети	1000

Окончание таблицы 12

int ts_num	Число проб при ts-шаге	10
int de_num	Число проб при de-шаге	10
int tabu_size	Размер списка запретов	1000
int max_f_call	Максимальное число вызовов ЦФ	15000
int de_try_max	Максимальное число попыток генерации «соседа»	10
double re-start_value	Порог, при падении значения среднего приращения популяции до которого запускается механизм рестарта популяции	-0,0001
double re-start_value	Интервал вариации начального числа скрытых нейронов для генерации начального приближения	[0,0..2,0]

При запуске программного комплекса проверяется наличие файла с именем fname в формате dt, описывающего выборку данных для обучения.

Структура файла имеет следующий вид. В первых строках идет описание количества переменных в наблюдениях:

```

bool_in=<Число булевых входных переменных>
real_in=<Число непрерывных входных переменных>
bool_out=<Число булевых выходных переменных>
real_out=<Число непрерывных выходных переменных>
training_examples=<Число обучающих примеров>
validation_examples=<Число валидационных примеров>
test_examples=<Число тестовых переменных>

```

Далее в <N> строках идет описание N примеров, первые Tr из которых – обучающие, вторые V – валидационные, последние Ts – тестовые.

Заметим, что в разработанном методе СПС валидационное множество не используется, поэтому далее мы о нем говорить не будем. Используются примеры только обучающего множества для обучения и тестового для тестирования.

Выходные данные. Данные, полученные в результате применения программного комплекса, выводятся в виде нескольких файлов. **Файл журнала** процесса обучения имеет текстовый формат и содержит информацию

об инициализации, основных итерациях и завершении процесса построения модели.

На Рисунке 17 показана инициализация процесса построения модели. В первой строке выводится время начала работы и название решаемой задачи, далее идет информация о начальном размещении 10 точек, и информация о значении ЦФ.

```
20130514_0336_cancer1
pop_size=10
p start: 48.6224/4
0: 4;50;48.6224
(...)
9: 3:39;106.981
```

Рисунок 17 – Инициализация процесса построения

Старт основных итераций начинается строкой «MAIN ITERATIONS STARTED» (Рисунок 18), далее для каждой следующей итерации указывается ее номер и лучшее значение ЦФ на момент ее начала.

```
MAIN ITERATIONS STARTED
lucky_iter=1; start=48.6224
```

Рисунок 18 – Начало основных итераций

На этапе квалификации (Рисунок 19) для каждой точки показаны значение ЦФ до и после применения локального спуска, число скрытых нейронов и связей, достигнутое приращение. В конце указаны индексы лучшей и худшей по значению ЦФ точек, а также значение ЦФ лучшей точки.

```
===QUALIFICATION START:
0: 48.6224 => 31.731 (4/50) -16/8914;
1: 49.8085 => 29.1833 (4/54) -20.6252
(...)
9: 106.981 => 31.0859 (3/39) -75.8952;
best_num=8
worst_num=4
====> best=31.731
```

Рисунок 19 – Этап квалификационного ранжирования точек

Этап обработки начинается и заканчивается пометками «MUTATIONS START» и «MUTATIONS END» (Рисунок 20). Для каждой точки указывается ее индекс и квалификация (“perspective” или “stagnant”). Худшая точка не улучшается, из лучшей производится 100 итераций спуска. Stagnant-точки пересеиваются (global), а в случае неудачи для них производится модификация топологии.

```

===MUTATIONS START:
sum_probs: -102.494; -11.3883; 0
person k=0
perspective
perspective improved 74 iters: 31.731 => 25.0781 == -6.65292
person k=1
perspective
perspective improved 90 iters: 29.1833 => 24.7986 == -4.38467
person k=2
stagnant 0.0715819
stagnant improved global: 49.9999 => 21.8178 == -28.1821
person k=3
stagnant 5.36271
stagnant improved global: 49.8688 => 17.8433 == -32.0255
person k=4
4 worst
(...)
===MUTATIONS END:
(...)

```

Рисунок 20 – Этап обработки точек

В данном примере на этапе обработки у неперспективных точек с индексами 8 и 9 вероятность нелокального параметрического улучшения оказалась небольшой и была вызвана операция модификации топологии (Рисунок 21). Для точки 8 удаление связи привело к ее улучшению. Для точки 9 неудачу потерпели операции удаления и добавления связей, и успешным оказалось удаление нейрона.

```

(...)
person k=8
stagnant 0.00014975
stagnant improvement failed
-1 conn: 49.8569 => 49.8569 == - 6.18237e-006
2/32 => 2/31
structure changed: 49.8569 => 49.8569 = - 6.18237e-006
person k=9
stagnant 0
stagnant improvement failed
- conn failed
+ conn failed
-1 node: 50 => 42.2591 == -7.74087
2/33 = 1/25
structure changed: 50 => 42.2591 == -7.7408
(...)

```

Рисунок 21 – Модификация топологии

В результате подобных операций каждая итерация завершается попыткой дополнительно улучшить худшую и лучшую точки (Рисунок 22). Дополнительно в журнал выводится общая статистика результатов итерации.

```

===TUNING START:
worst tuning: 50 => 50 ==0
best tuning: 17.8433 => 11.0061 == -6.83717
===TUNING END:

==ITER RESUKTS START:
48.6224 => 11.0061 == -37.6163
==ITER RESUKTS END:

```

Рисунок 22 – Завершение итерации

Основные итерации заканчиваются выводом сообщения «MAIN ITERATIONS STOPPED» (Рисунок 23), после чего сообщается номер эксперимента.

```

MAIN ITERATIONS STOPPED
expn=0
0: 3.46132 => 3.46124
1: 3.46124 => 3.46117
2. 3.46117 => 3.46109

```

Рисунок 23 – Завершение основных итераций

Далее начинается итоговое обучение лучшей сети. По окончании итогового обучения (Рисунок 24) выводится значение ЦФ до и после начала итогового обучения.

```
1497: 3.17432 => 3.17342
1498: 3.17342 => 3.17337
1499: 3.17342 => 3.17337
3.46132 => 3.17337 == 0
```

Рисунок 24 – Завершение работы

Помимо файла журнала обучения программный комплекс в процессе работы создает и другие выходные файлы; их описание приведено в Таблице 13.

Таблица 13 – Выходные файлы программного комплекса

Вид	Формат	Структура
Файл с полным описанием итоговой модели, пригодный для использования в модуле эксплуатации	Текстовый	В первой строке приведены параметры n_i , n_o , n_h – число входных, выходных и скрытых нейронов, записываемое через пробел. Далее описывается матрица смежности графа персептронной модели: в каждой из последующих строк указываются вещественные числа, описывающие элементы матрицы (веса связей).
Файл отчета об обучении модели	CSV	В начале располагается блок краткого описания “BRIEF” со следующими характеристиками: метка точки (служебная информация), число входных, выходных и скрытых нейронов, число связей, итоговое значение ЦФ, приращение ЦФ на последней итерации. Затем приводится история операций с точкой и соответствующие значения ЦФ, приращения ЦФ, числа скрытых нейронов. Далее в таком же виде идут описания лучших сетей в последующих экспериментах.
Файл со статистикой экспериментов	CSV	Перечисление максимального, минимального и среднего значений, а также стандартного отклонения следующих характеристик: ошибка на тестовом множестве, ошибка на обучающем множестве, число скрытых нейронов, число связей, число эпох обучения.

Вызов и загрузка. Перед началом работы с программным комплексом пользователю необходимо подготовить входные файлы с данными

<имя_задачи>.dt и конфигурационный файл econf.cfg, задающий набор параметров, в описанных форматах и расположить их в каталоге с программным обеспечением. Далее требуется запустить один из файлов – express.exe или experiment.exe. Выбор файла запуска зависит от целей пользователя: произвести вычислительный эксперимент по построению моделей или получить прогноз уже имеющейся модели для одного наблюдения или пакета наблюдений.

Разработанный программный комплекс пригоден для решения любых задач, допускающих построение модели в виде МПРПС по данным наблюдений, при этом качество модели зависит, очевидно, от качества предоставленной выборки эмпирических данных. Отметим, что предварительная обработка данных возлагается на пользователя, но это условие не снижает степени практической применимости разработанного программного комплекса.

К настоящему моменту подготовлена техническая документация для регистрации разработанного программного обеспечения в ФГБУ «Федеральный институт промышленной собственности».

3.2 Решение практических задач

3.2.1 Прогнозирование налоговых поступлений

(по данным Республики Бурятия)

В практической деятельности часто возникают задачи получения прогноза на основе анализа лишь динамики прогнозируемой величины. Аппарат для решения таких задач востребован, например, в сфере онлайн-решений, когда взаимосвязи влияющих факторов изменяются на глазах: к примеру, при автоматизации биржевой или букмекерской торговли. Потребность в подобном аппарате существует также в ситуации, когда учет влияющих факторов затруднен или невозмо-

жен, как это, например, происходит в области маркетинга и других, сложных для прогнозирования экономических системах.

Одной из задач, при решении которой востребовано применение такого аппарата, является прогнозирование налоговых поступлений, каковое является важной проблемой регионального экономического планирования [35; 49]. В настоящее время экономическая картина мира претерпела значительные изменения; одним из основных последствий таких изменений стало вовлечение региональных экономических систем в процессы глобализации. Сложившаяся ситуация закономерным образом порождает необходимость учета при анализе процессов региональных экономических систем гораздо большего числа факторов, чем в ранних моделях налогового прогнозирования, когда региональные экономические системы были более замкнутыми.

При этом на качество прогноза налоговых поступлений и возможность его практического использования в планировании налоговых доходов региона решающим образом влияет понимание существующих закономерностей и использование эффективной методологической основы прогнозирования. И в этом случае камнем преткновения становится непостоянство системы факторов, влияющих на поступление налогов, т. е. ее нестационарность (в силу отсутствия возможности выделения явного тренда, сезонности, цикличности) [30; 50; 51].

Таким образом, актуальной является задача разработки не только мощных систем многофакторного прогнозирования, но и инструментария построения краткосрочных прогнозов, опирающегося лишь на динамику непосредственно прогнозируемой величины и способного на быструю реакцию при изменениях системы влияющих факторов [34].

В данном контексте перспективным представляется использование аппарата нейросетевого моделирования наряду с инструментарием, традиционно используемым для прогнозирования временных рядов [35].

В нашей работе мы рассмотрим применение метода СПС для синтеза нейросетевой модели прогнозирования налоговых поступлений по Республике Бурятия. Временной интервал исторических данных, используемых для построе-

ния системы прогнозирования, составляет четыре года с разбивкой по месяцам. В Таблице 14 представлены исходные данные.

Таблица 14 – Налоговые доходы Р. Бурятия в бюджеты всех уровней в 2004–2007 гг., тыс. руб.

Месяц	2004	2005	2006	2007
Январь	876935	1708948	2076007	1231694
Февраль	701101	1728481	2459420	846623
Март	1259393	1849992	2477204	1690235
Апрель	2136617	1758829	1910765	1376684
Май	1053282	1499963	2921238	1879502
Июнь	1131094	1427282	2585433	1512825
Июль	195137	1394902	2132191	2344140
Август	917856	1310242	2333428	1443117
Сентябрь	857267	1546974	2359101	2351188
Октябрь	1275043	1461098	2701702	2281840
Ноябрь	866602	1423545	2730066	2654730
Декабрь	972774	1770500	2093905	2665092

На Рисунке 25 данные отображены в виде графика.



Рисунок 25 – Налоговые доходы Республики Бурятия в бюджеты всех уровней в 2004-2007 гг., тыс. руб.

Визуальный анализ графика показывает, что поведение прогнозируемой величины в рассматриваемый период достаточно плохо предсказуемо.

Для использования гиперболического тангенса в качестве активационной функции нам потребуется привести исходные данные к такому виду, чтобы значения поступлений принадлежали интервалу $[0,1]$. При этом учтем, что шкалирование следует производить таким образом, чтобы используемый аппарат мог выдать нам значения бóльшие чем максимальное и меньшие чем минимальное среди имеющихся наблюдений (на случай, если в прогнозируемый период будет наблюдаться выход за пределы области значений поступлений предыдущего периода). Максимальное наблюдаемое значение составляет 2921238, минимальное – 195137. Таким образом, наблюдаемый интервал изменений примет вид $[195137, 2921238]$, абсолютное значение разницы между минимальным и максимальным значениями составит $I=2726101$. Позволим прогнозируемому значению принимать значения в интервале $[195137-\alpha, 2921238+\alpha]$, где $\alpha = \frac{I}{2}$. Тогда шкалирование произведем по форму-

ле $x = \frac{x-a}{b-a}$, где $a = 195137 - \alpha = -1167914$, $b = 2921238 + \alpha = 4284289$.

После шкалирования данные примут вид, представленный в Таблице 15.

Таблица 15 – Налоговые доходы Республики Бурятия в бюджеты всех уровней в 2004-2007 гг., тыс. руб. (шкалированные величины)

Месяц	2004	2005	2006	2007
Январь	0,3751	0,5277	0,5950	0,4401
Февраль	0,3428	0,5312	0,6653	0,3695
Март	0,4452	0,5535	0,6686	0,5242
Апрель	0,6061	0,5368	0,5647	0,4667
Май	0,4074	0,4893	0,7500	0,5589
Июнь	0,4217	0,4760	0,6884	0,4917
Июль	0,2500	0,4701	0,6053	0,6442
Август	0,3826	0,4545	0,6422	0,4789
Сентябрь	0,3714	0,4979	0,6469	0,6454
Октябрь	0,4481	0,4822	0,7097	0,6327
Ноябрь	0,3732	0,4753	0,7149	0,7011
Декабрь	0,3926	0,5389	0,5983	0,7030

Для прогнозирования временного ряда воспользуемся методом «скользящего окна» (методом кадрирования). Данный метод используется для получения набора обучающих примеров из упорядоченной по времени последовательности наблюдений. Для использования метода необходимо задать два параметра: ширину t_1 входного окна (количество входов сети) и ширину t_2 выходного окна (количество выходов сети).

Пусть временной ряд $\{z_i\}_{i=1, \overline{N}}$ (где N – количество всех имеющихся наблюдений) является одномерным, тогда набор обучающихся примеров $\{(x^{(i)}, y^{(i)})\}_{i=1, \overline{K}}$ формируется на основе $\{z_i\}_{i=1, \overline{N}}$ по формулам:

$$\begin{cases} x^{(i)} = (z_i, z_{i+1}, \dots, z_{t_1+i-1}), \\ y^{(i)} = (z_{t_1+i}, z_{t_1+i+1}, \dots, z_{t_1+i+t_2-1}). \end{cases} \quad (15)$$

Метод кадрирования при прогнозировании одномерного временного ряда ни в коей мере не конкурирует с методами, опирающимися на экономический анализ влияющих факторов, однако он хорошо справляется со своим основным назначением – получением модели краткосрочного прогнозирования в условиях высокой степени неопределенности влияющих факторов (что, как было отмечено, является достаточно широко распространенной проблемой в реальных задачах экономического прогнозирования). Подчеркнем также, что нашей задачей не является построение модели по всей имеющейся выборке – для получения прогноза достаточно модели, наиболее точно отражающей зависимость любой точки временного ряда от фиксированного числа значений предыдущих наблюдений.

Ошибку при обучении сети будем рассчитывать так же, как и рассчитывали ранее – в виде (7).

В качестве критерия останова работы алгоритма установим падение среднего приращения ниже заданной величины $\varepsilon = 0.001$ по абсолютному значению. В силу небольшого количества обучающих примеров разбиения на обучающее и тестовое множества не будем производить с целью получения модели, наиболее

точно характеризующей всю картину наблюдений. Принятую ошибку будем рассчитывать также по всем имеющимся в нашем распоряжении наблюдениям.

Сравнение метода СПС станем производить с результатами работы Е. В. Чимитдоржиевой и В. В. Доржиевой [54], в которой используется модель и методология, называемая АРПСС – авторегрессия проинтегрированного скользящего среднего, в литературе встречается под названием ARIMA – AutoRegressive Integrated Moving Average, также такие модели называют моделями Бокса-Дженкинса. Заметим, что данная методология разработана специально для применения в исследованиях нестационарных процессов (т. е. несохраняющих во времени свои вероятностные характеристики), и представляет собой, по сути, следующее: с помощью интегрирования и взятия разности процесс приводится к стационарному виду и, далее, применяется базовая методология – ARMA (AutoRegressive Moving Average).

В указанной работе для оценки полученных моделей использовалась средняя относительная ошибка [54]:

$$e(w) = \frac{1}{n_o} \cdot \frac{1}{T} \cdot \sum_{t=1}^T \sum_{k=1}^{n_o} \frac{|\tilde{y}_k^{(t)} - y_k^{(t)}|}{y_k^{(t)}} \cdot 100\%. \quad (16)$$

Наилучшая средняя относительная ошибка, полученная с помощью применения АРПСС составила 14,9% [54]. Значения ошибки прогнозирования, рассчитанной в виде (16), для различных размеров входных окон (15) при использовании СПС приводятся в Таблице 16. Для каждого из размеров входного окна рассматривалась лучшая по десяти запускам сеть.

Таблица 16 – Средняя относительная ошибка при различных размерах окна

Размер окна	Средняя относительная ошибка, %	Среднеквадратичная ошибка
3	12,6375	0,7745
4	11,3446	0,7169
5	10,6931	0,5862
6	11,0180	0,5787
7	10,5125	0,5336
8	10,6042	0,7161
9	10,8216	0,7576

Как видно из Таблицы 16, наилучшее значение достигается при 7-месячном входном окне и составляет 10,5125%, что несколько улучшает результат, полученный при использовании метода АРПСС. Топология сети, на которой достигнута данная величина ошибки изображена на Рисунке 26.

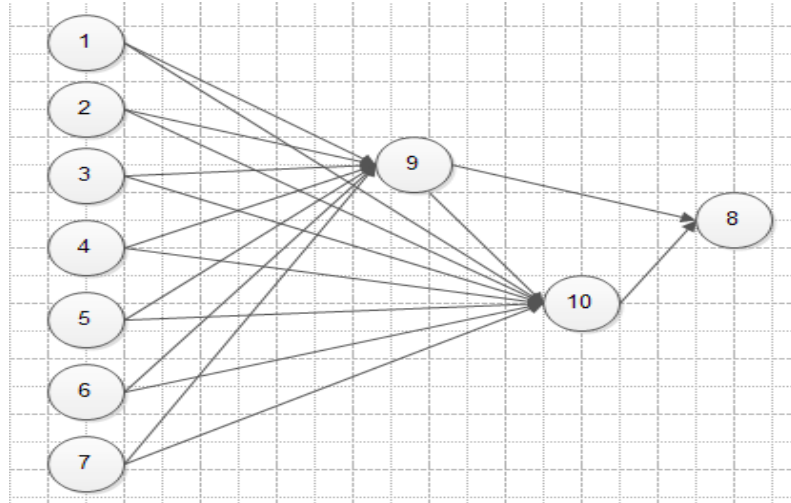


Рисунок 26 – Топология лучшей сети для задачи о прогнозировании налога

Матрица смежности этой сети приводится в Таблице 17.

Таблица 17 – Матрица смежности сети для задачи о прогнозировании налога

	0	1	2	3	4	5	6	7	8	9	
0	0	0	0	0	0	0	0	0	-0,1851	5,7448	0,2148
1	0	0	0	0	0	0	0	0	-0,1135	1,8494	9,9538
2	0	0	0	0	0	0	0	0	0,2509	-2,9348	-8,0268
3	0	0	0	0	0	0	0	0	0,2567	-7,7335	-5,0663
4	0	0	0	0	0	0	0	0	-0,2834	-4,9960	7,7421
5	0	0	0	0	0	0	0	0	0,5849	-6,8301	5,3707
6	0	0	0	0	0	0	0	0	0,7868	-8,8508	5,0352
7	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	-6,1526	0	-1,8215
9	0	0	0	0	0	0	0	0	-6,2239	0	0

На Рисунке 27 изображен график годового прогноза для входного окна, равного 7-ми месяцам.

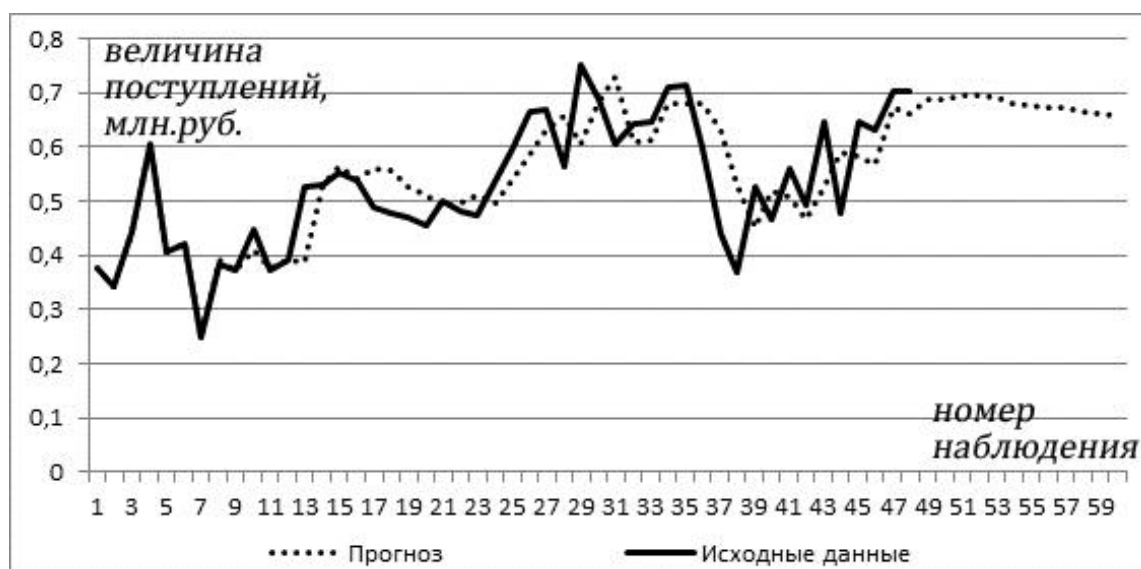


Рисунок 27 – Результат 12-месячного прогноза, построенного с помощью метода СПС с 7-ми месячным входным окном (в шкалированном виде)

Небольшая величина визуально заметного смещения прогноза вправо относительно исходных данных, говорит о том, что модель достаточно чутко реагирует на изменения, улавливая общий тренд. Можно сделать общий вывод об эффективности применения ИНС для построения прогноза в условиях изменения системы влияющих факторов путем построения модели, отражающей зависимость любой точки временного ряда от фиксированного числа значений предыдущих наблюдений. Качество получаемых моделей иллюстрируется расчетами, показывающими улучшение ошибки по сравнению с традиционными методами прогнозирования (АРПСС).

Для дальнейших исследований перспективным представляется изучение применения метода СПС для построения прогноза процесса, приведенного к стационарному виду.

3.2.2 Прогнозирование результатов сдачи

Единого государственного экзамена (ЕГЭ)

Профориентационное тестирование школьников широко используется на различных этапах системы образования Российской Федерации. Раннее выявление потенциала и выбор направления развития позволяет добиться лучших результатов в последующей образовательной и профессиональной деятельности.

Вместе с тем для средних общеобразовательных учреждений, различных частных и вузовских центров подготовки абитуриентов актуальным является создание инструментария прогнозирования как выбора предметов для сдачи Единого государственного экзамена (ЕГЭ), так и его результатов с точностью до баллов.

Единый государственный экзамен в настоящее время сдается по 14 предметам: Русский язык, Математика, Физика, Химия, Биология, История, География, Обществознание, Литература, Английский язык, Немецкий язык, Французский язык, Испанский язык.

Первые два предмета – обязательны для сдачи, выбор для сдачи ЕГЭ остальных двенадцати предметов остается за школьником.

Несмотря на то, что существует достаточно широко представленный инструментарий оценки профессиональных предпочтений, средства прогнозирования выбора предметов ЕГЭ и оценок в баллах, получаемых при сдаче экзаменов, основанные на показателях профессиональных предпочтений школьников, пока отсутствуют. Особенно ценным являлся бы инструмент, позволяющий построение прогноза по *неявным факторам*, а не напрямую по результатам ЕГЭ-подобных тестов. Подобное прогнозирование может опираться на различные исходные данные: информацию о личностных предпочтениях, успеваемости, достижениях, знаниях, навыках, умениях школьника.

В работе рассматривается задача разработки такой модели прогнозирования, основанной на исходном тексте Опросника профессиональной готовности (ОПГ) Л. Н. Кабардовой [31] (см. Приложение Б). Опросник включает в себя 50 утверждений, касающихся профориентационных предпочтений школьника. В оригинале опросник направлен на выявление профессиональных предпочтений с точностью до одной из пяти областей: «Человек – человек», «Человек – знаковая система», «Человек – природа», «Человек – художественный образ», «Человек – техника» и предполагает сложную систему подсчета баллов. В нашей работе мы значительно упростили процедуру опроса: каждое из утверждений было оценено школьником по шкале от 1 до 5.

Строка исходных данных, таким образом, представляет собой упорядоченный набор 50 целых чисел от 1 до 5, по которым требуется выдать прогноз результатов ЕГЭ по 14 предметам в баллах от 1 до 100, при этом будем считать, что если предмет не выбран, то по нему условно получено 0 баллов.

Исходные данные предоставлены Образовательным центром «Горизонт» и Управлением довузовской подготовки ФГБОУ ВПО «БГУ». Было опрошено 493 школьника 11-х классов в период 2011-2013 гг. Опрос производился в сентябре-октябре до выбора предметов для сдачи ЕГЭ. Как видно из Таблицы Б.1 (см. Приложение Б), утверждения ОПГ являются достаточно общими, легко воспринимаются испытуемыми, затруднений в восприятии и верной трактовке утверждений у школьников не возникает. При этом ряд анкет (75 шт.) был забракован, поскольку не удалось выяснить, какие предметы в итоге сдавал школьник, или были обнаружены пропущенные утверждения. Таким образом, окончательная выборка составила 418 наблюдений.

Простая предобработка данных заключалась в выполнении следующих шагов:

- 1) Входные и выходные данные отмасштабированы в $[0,1]$.
- 2) Выходные переменные по предметам «Испанский язык», «Английский язык», «Французский язык» и «Немецкий язык» объединены в одну переменную «Иностранный язык».

В результате применения разработанного программного комплекса была получена нейросетевая модель, обладающая следующими характеристиками: 50 входных, 11 выходных, 2 скрытых нейрона, 673 связи. Ошибка обучения составила 0,0336, ошибка тестирования составила 0,0465. Матрица смежности соответствующей сети описана в Приложении В.

Полученная нейросетевая модель способна по данным профориентационного анкетирования предсказать результат ЕГЭ в баллах. Обучение сети как оптимизационный процесс нацелено на минимизацию среднеквадратичной ошибки шкалированных в $[0,1]$ данных. При подаче на вход обученной нейросетевой мо-

дели того или иного набора из 50 вопросов получаем выходной вектор из 11 значений, соответствующих баллам по указанному выше набору предметов.

Результаты применения обученной модели на имеющейся выборке представлены в Таблице 18 в виде абсолютной ошибки в баллах.

Таблица 18 – Статистика предсказанных баллов

Предмет	мин.	макс.	среднее	ст. откл.
РУС	0,17	39,06	13,55830	8,94713
МАТ	0,07	78,88	18,68160	14,23720
ФИЗ	0,18	63,90	17,10030	14,22340
ХИМ	0,01	69,92	11,05870	13,99400
БИО	0,16	70,87	15,84810	14,56040
ИСТ	0	70,96	16,43910	14,14090
ГЕО	0,01	70,13	4,63986	8,24672
ОБЩ	0,04	65,07	24,53810	13,42390
ЛИТ	0,02	77,65	7,31928	10,95440
ИНЯ	0,01	77,91	11,83940	13,48930
ИНФ	0,01	94,85	7,10675	11,14790

Как видно из таблицы, максимум среднего значения ошибки составляет около 24,5, а стандартное отклонение ошибки не превышает 14,6.

Оценим далее полученный прогноз только для тех случаев, когда предмет был выбран и по нему действительно был сдан экзамен (см. столбец «Экзамен» в Таблице 19).

Таблица 19 – Результаты прогнозирования по отдельным предметам

Точность прогноза		до 5 баллов				до 10 баллов				до 25 баллов			
из числа:		всех		сдававших		всех		сдававших		всех		сдававших	
Предмет	Экзамен	чел	%	чел	%	чел	%	чел	%	чел	%	чел	%
РУС	418	77	18,42	77	18,42	169	40,43	169	40,43	366	87,56	366	87,56
МАТ	418	71	16,99	71	16,99	137	32,78	137	32,78	302	72,25	302	72,25
ФИЗ	88	330	78,95	0	0	331	79,19	1	1,14	343	82,06	13	14,77
ХИМ	45	373	89,23	0	0	373	89,23	0	0	377	90,19	4	8,89
БИО	80	339	81,10	1	1,25	340	81,34	2	2,50	347	83,01	9	11,25
ИСТ	74	344	82,30	0	0	345	82,54	1	1,35	356	85,17	12	16,22
ГЕО	12	406	97,13	0	0	406	97,13	0	0	406	97,13	0	0
ОБЩ	222	214	51,20	18	8,11	232	55,50	36	16,22	326	77,99	130	58,56
ЛИТ	18	400	95,69	0	0	400	95,69	0	0	400	95,69	0	0
ИНЯ	44	374	89,47	0	0	374	89,47	0	0	376	89,95	2	4,55
ИНФ	30	388	92,82	0	0	388	92,82	0	0	389	93,06	1	3,33

Не станем учитывать при расчете ошибки прогноз по не выбранным для сдачи экзамена предметам. При этом будем считать, что если прогнозное значение отрицательное, то предсказано получение 0 баллов, т.е. предмет для сдачи не будет выбран.

Полученные прогнозы свидетельствуют, что полученная модель верно определяет результат с точностью до 25 баллов по любому предмету для 72,25% школьников.

В прогнозе сдачи экзамена по необязательным предметам наибольший интерес представляет, очевидно, количество верно определенных результатов именно для школьников, сдававших предмет. Результаты в Таблице 19 показывают, что по этим предметам данный показатель гораздо ниже, чем число верно определенных результатов среди всех школьников. В этом случае наилучшие результаты показаны по предметам «Русский язык» – 87,56% школьников, «Математика» – 72,25% школьников. Данный факт свидетельствует о наличии взаимосвязи между профессиональными предпочтениями и результатами обязательных экзаменов, причем для предмета «Русский язык» эта взаимосвязь достаточно ярко выражена. Лучший прогноз результатов экзамена по необязательным предметам продемонстрирован моделью для предмета «Обществознание» – верно определено 58,56% результатов экзамена по данному предмету. Данный факт свидетельствует о том, что модель фактически не обучилась прогнозировать результат сдачи экзамена по необязательным предметам, но обучилась предсказывать, что предмет не будет выбран для сдачи. Такой результат вполне закономерен и обусловлен нацеленностью метода на минимизацию среднеквадратичной ошибки по всем предметам в среднем.

В целом же, качество прогноза, на наш взгляд, можно поставить в зависимость от следующих факторов:

- 1) количество школьников, сдающих экзамен по определенным предметам;
- 2) содержание исходных утверждений опросника;

- 3) достоверность фактических результатов (поскольку существует вероятность списывания, в этих случаях модель неизбежно будет показывать ошибочный результат).

Анализ полученных результатов показывает, что разработанный алгоритмический инструмент с задачей построения профориентационных моделей справляется достаточно хорошо (даже при работе с материалом, изначально не предполагавшим получения прогноза: данные наблюдений взяты из имеющихся, а не подготовлены специально). Для улучшения качества прогноза возможны следующие действия:

- 1) построение различных моделей для отдельного прогнозирования результатов экзаменов по обязательным предметам и предметам, которые сдаются по желанию школьника;
- 2) переработка содержания опросника с учетом современных реалий и большей направленности на предмет прогнозирования.

Использование подобного инструментария имеет достаточно широкие перспективы и может быть нацелено на решение определенных практических задач.

Во-первых, получение прогноза сдачи Единого государственного экзамена в баллах позволяет произвести оценку возможностей школьников и организовать более эффективную подготовку к экзамену путем работы над заранее спрогнозированными результатами. Во-вторых, прогноз может быть использован с целью правильного выбора – для определения предметов, прогнозируемый результат экзамена по которым наиболее успешен. В решении таких задач могут быть заинтересованы как школьники и их родители, так и средние общеобразовательные учебные заведения, образовательные центры, вузы.

Кроме того, становится возможным выявление подозрительных случаев: когда ученики показывают неожиданно высокие результаты.

3.3 Выводы по главе

В Главе 3 описан разработанный программный комплекс, изложена процедура его применения в практических задачах из области экономики и образования.

Разработанный программный комплекс пригоден для решения любых задач моделирования по данным наблюдений, допускающих использование нейросетевого подхода. Комплекс реализован на языке C++ в виде набора модулей и библиотек классов.

Построена нейросетевая модель прогнозирования временного ряда данных налоговых поступлений (на примере Республики Бурятия): получено улучшение по сравнению с моделью на основе методологии ARIMA.

Рассмотрена задача из области менеджмента образования – предсказание результатов Единого государственного экзамена и предложена методика прогнозирования по данным профориентационного анкетирования. Показаны перспективы разработанной методики, описаны способы улучшения результатов.

ЗАКЛЮЧЕНИЕ

В работе получены и выносятся на защиту следующие основные научные результаты и разработки:

1. Мультистартовая схема одновременного подбора топологии и настройки весов связей МПРПС, отличающаяся от известных способом оценки сетей на предмет перспективности применения к ним методов локального спуска, глобального поиска или модификации топологии; метод одновременного структурно-параметрического синтеза, разработанный на основе данной схемы.

2. Новый алгоритм глобального поиска на основе метода поиска с запретами, который отличается от известных ведением сегментированного списка запретов и совершением шага по методу дифференциальной эволюции, эксплуатирующего данную сегментированность.

3. Новая стратегия последовательного выбора операции модификации топологии МПРПС, предназначенная для использования в мультистартовых методах, которая отличается от известных детерминированным порядком выбора операций модификации топологии с приоритетом операций упрощения и нацеленностью на интенсификацию исследования каждого уровня топологии.

4. Программный комплекс для решения задач нейросетевого моделирования, реализующий предложенные метод и алгоритмы.

5. Нейросетевые модели прогнозирования временного ряда налоговых поступлений (на примере Республики Бурятия) и прогнозирования результатов Единого государственного экзамена по данным профориентационного анкетирования, построенные с применением разработанного программного комплекса.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ИНС	искусственная нейронная сеть
МП	многослойный персептрон
МППС	многослойный персептрон с разомкнутыми перекрестными связями
ОРО	метод обратного распространения ошибки
BFGS	алгоритм Broyden–Fletcher–Goldfarb–Shanno
OBD	метод Optimal Brain Damage
OBS	метод Optimal Brain Surgeon
ГСП	метод глобального стохастического поиска
SA	Simulated Annealing – метод имитации отжига
TS	Tabu Search – метод поиска с запретами
GA	Genetic Algorithm – генетический алгоритм
DE	Differential Evolution – метод дифференциальной эволюции
ЦФ	целевая функция
СПС	метод структурно-параметрического синтеза
АРСС	авторегрессия проинтегрированного скользящего среднего
ARIMA	AutoRegressive Integrated Moving Average
ARMA	AutoRegressive Moving Average
ЕГЭ	Единый государственный экзамен

СПИСОК ЛИТЕРАТУРЫ

1. Алгоритмы : построение и анализ = Introduction to Algorithms / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн ; под ред. И. В. Красикова. – 2-е изд. – М. : Вильямс, 2005. – 1296 с.
2. Арнольд, В. И. О функциях трех переменных / В. И. Арнольд // Доклады АН СССР. – 1957. – Т. 114, № 4. – С. 679–681.
3. Ахмеров, Р. Р. Методы оптимизации гладких функций / Р. Р. Ахмеров. – Новосибирск : Изд-во Новосибирского ун-та, 1993. – 100 с.
4. Барцев, С. И. Адаптивные сети обработки информации / С. И. Барцев, В. А. Охонин. – Красноярск : Институт физики СО АН СССР, 1986. – Препринт № 59Б. – 20 с.
5. Барцев, С. И. Принцип двойственности в организации адаптивных сетей обработки информации / С. И. Барцев, С. Е. Гилев, В. А. Охонин // Динамика химических и биологических систем. – Новосибирск : Наука, 1989. – С. 6-55.
6. Вапник, В. Н. Восстановление зависимостей по эмпирическим данным / В. Н. Вапник. – М. : Наука, 1979. – 448 с.
7. Васильев, Ф. П. Методы оптимизации. – М. : Факториал Пресс, 2002. – 824 с.
8. Галушкин, А. И. Нейронные сети: основы теории / А. И. Галушкин. – М. : Горячая линия – Телеком, 2012. – 496 с.
9. Галушкин, А. И. Синтез многослойных систем распознавания образов / А. И. Галушкин. – М. : Энергия, 1974. – 367 с.
10. Гилл, Ф. Практическая оптимизация / Ф. Гилл, У. Мюррей, М. Райт. – М. : Мир, 1985. – 510 с.
11. Глаз, А. Б. Оценка вероятности образования оптимальной структуры перцептрона при ее оптимизации методами случайного поиска / А. Б. Глаз, Л. А. Растринин // Задачи статистической оптимизации. – Рига : Зинатне, 1971. – С. 131-142.
12. Глаз, А. Б. Трехрядный статистический перцептрон со специальным устройством статистической адаптации структуры / А. Б. Глаз, Л. А. Растринин // Перцептрон – система распознавания образов. – Киев : Наукова думка, 1975. – С. 334-385.]

13. Горбань, А. Н. Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей / А. Н. Горбань // Сибирский журнал вычислительной математики. – 1998. – Т. 1, № 1. – С. 12-24.
14. Горбань, А. Н. Обучение нейронных сетей / А. Н. Горбань. – М. : СССР - США СП Параграф, 1990. – 160 с.
15. Горнов, А. Ю. Метод «криволинейного» поиска глобального экстремума в задаче оптимального управления / А. Ю. Горнов, Т. С. Зароднюк // Современные технологии. Системный анализ. Моделирование. – Иркутск : ИрГУПС, 2008. – № 3. – С. 19-27.
16. Дэннис, Дж. Численные методы безусловной оптимизации и решения нелинейных уравнений / Дж. Дэннис, Р. Шнабель ; пер. с англ. – М. : Мир, 1988. – 440 с.
17. Евстигнеев, Е. Н. Налоги и налогообложение / Е. Н. Евстигнеев. – СПб. : Питер, 2005. – 272 с.
18. Евтушенко, Ю. Г. Методы решения экстремальных задач и их применение в системах оптимизации / Ю. Г. Евтушенко. – М. : Наука, 1982. – 432 с.
19. Ежов, А. А. Нейрокомпьютинг и его применение в экономике и бизнесе / А. А. Ежов, С. А. Шумский. – М. : МИФИ, 1998. – 224 с.
20. Емельянов, В. В. Теория и практика эволюционного моделирования / В. В. Емельянов, В. В. Курейчик, В. М. Курейчик. – М. : ФИЗМАТЛИТ, 2003. – 432 с.
21. Жиглявский, А. А. Математическая теория глобального случайного поиска / А. А. Жиглявский. – Л. : Изд-во ЛГУ, 1985. – 296 с.
22. Жиглявский, А. А. Методы поиска глобального экстремума / А. А. Жиглявский, А. Г. Жилинскас. – М. : Наука, 1991. – 247 с.
23. Измаилов, А. Ф. Численные методы оптимизации : учеб. пособие / А. Ф. Измаилов, М. В. Солодов. – М. : ФИЗМАТЛИТ, 2005. – 304 с.
24. Каллан, Р. Основные концепции нейронных сетей / Р. Каллан. – М. : Вильямс, 2001. – 288 с.
25. Кельберт, М. Я. Вероятность и статистика в примерах и задачах. В 2 т. Т. 2 : Марковские цепи как отправная точка теории случайных процессов и их приложения / М. Я. Кельберт, Ю. М. Сухов. – М. : МЦНМО, 2010. – 560 с.

26. Колмогоров, А. Н. О представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного / А. Н. Колмогоров // Доклады АН СССР. – 1957. – Т. 114, № 5. – С. 953-956.
27. Колмогоров, А. Н. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных / А. Н. Колмогоров // Доклады АН СССР. – 1956. – Т. 108, № 2. – С. 179-182.
28. Курейчик, В. М. Эволюционная адаптация в обучении нейронных сетей / В. М. Курейчик, Б. К. Лебедев, В. И. Божич // Известия Южного федерального университета ; технические науки. – Таганрог : Изд-во ТТИ ЮФУ, 1999. – Т. 13, № 3. – С. 25-28.
29. Левитин, А. В. Алгоритмы : введение в разработку и анализ / А. В. Левитин. – М. : Вильямс, 2006. – 576 с.
30. Налоговая реформа в России : проблемы и решения. В 2 т. Т. 1 / С. Синельников-Мурылев, П. Кадочников, К. Непесов, Д. Некипелов, Е. Шкробела. – М. : ИЭПП, 2003. – 395 с.
31. Никишина, В. Б. Психодиагностика в системе социальной работы : учеб. пособие для студ. высш. учеб. заведений / В. Б. Никишина, Т. Д. Василенко. – М. : Изд-во ВЛАДОС-ПРЕСС, 2004. – 208 с.
32. Осовский, С. Нейронные сети для обработки информации / С. Осовский. – М. : Финансы и статистика, 2004. – 344 с.
33. Проблемы Гильберта / под ред. П. С. Александрова. – М. : Наука, 1969. – 240 с.
34. Прогнозирование поступлений налога на доходы физических лиц с использованием искусственных нейронных сетей / Ф. В. Хандаров [и др.] // Байкальские экономические чтения : материалы международной научно-практической конференции. – Улан-Удэ : Изд-во ВСГУТУ, 2011. – С. 195-198.
35. Прогнозирование поступлений налога на прибыль организаций с помощью искусственных нейронных сетей / Ф. В. Хандаров [и др.] // Нейроинформатика, её приложения и анализ данных : материалы XVI Всероссийского семинара (19-21 сентября 2008) / под ред. А. Н. Горбаня, Е. М. Миркеса. – Красноярск : ИВМ СО РАН, 2008. – С. 56-58.
36. Растрингин, Л. А. Адаптация сложных систем / Л. А. Растрингин. – Рига : Зинатне, 1981. – 375 с.

37. Редько, В. Г. Эволюция, нейронные сети, интеллект : модели и концепции эволюционной кибернетики / В. Г. Редько. – М. : УРСС, 2005. – 224 с. – (Синергетика : от прошлого к будущему).
38. Рудой, Г. И. Выбор функции активации при прогнозировании нейронными сетями / Г. И. Рудой // Машинное обучение и анализ данных. – 2011. – Т. 1. № 1. – С. 16-39.
39. Рутковский, Л. Методы и технологии искусственного интеллекта / Л. Рутковский. – М. : Горячая линия – Телеком, 2010. – 520 с.
40. Страуструп, Б. Программирование : принципы и практика использования C++ / Б. Страуструп. – М. : Вильямс, 2011. – 1248 с.
41. Тихонов, А. Н. О регуляризации некорректно поставленных задач / А. Н. Тихонов // Доклады АН СССР. – 1963. – Т. 153, № 1. – С. 42-52.
42. Тихонов, А. Н. О решении некорректно поставленных задач и методе регуляризации / А. Н. Тихонов // Доклады АН СССР. – 1963. – Т. 151, № 3. – С. 501-504.
43. Томович, Р. Общая теория чувствительности / Р. Томович, М. Вукобратович ; пер. с сербск. и с англ., под ред. Я. З. Цыпкина. – М. : Советское радио, 1972. – 240 с.
44. Фихтенгольц, Г. М. Курс дифференциального и интегрального исчисления. В 3 т. Т. 3 / Г. М. Фихтенгольц. – М. : ФИЗМАТЛИТ, 2008. – 728 с.
45. Хайкин, С. Нейронные сети : полный курс / С. Хайкин. – 2-е изд. – М : Вильямс, 2006. – 1104 с.
46. Хандаров, Ф. В. Гибридный метод глобальной оптимизации на основе поиска с запретами и дифференциальной эволюции / Ф. В. Хандаров, Т. Г. Дармаев // Вестник Бурятского государственного университета. Выпуск: Математика, информатика. – 9 (1). – 2014. – С.69-75.
47. Хандаров, Ф. В. Об одном эволюционном алгоритме настройки искусственных нейронных сетей / Т. Г. Дармаев, Ф. В. Хандаров // Вестник Бурятского государственного университета. Спецвыпуск В. – 2012. – С. 197-203.
48. Хандаров, Ф. В. Открытая кроссплатформенная библиотека для решения нейросетевых задач и инструментальная среда для быстрого создания нейронных сетей различных топологий / Ф. В. Хандаров, М. В. Скворцов // Нейроинформатика, её приложения и анализ данных : материалы XVI Всероссийского семинара (19-21 сентября 2008) / под ред. А. Н. Горбаня, Е. М. Миркеса. – Красноярск : ИВМ СО РАН, 2008. – С. 59-65.

49. Хандаров, Ф. В. Прогнозирование налога на доходы физических лиц с использованием искусственных нейронных сетей / Ф. В. Хандаров, З. Б.-Д. Дондоков // Вестник Бурятского государственного университета. Серия : Математика и информатика. – 2012. – Вып. 1 – С. 60-67.
50. Хандаров, Ф. В. Прогнозирование налоговых поступлений с использованием гибридных нейронных сетей / Ф. В. Хандаров, З. Б.-Д. Дондоков // Инфокоммуникативные и вычислительные технологии и системы : материалы III Международной конференции (6-11 сентября 2010). – Улан-Удэ : Изд-во БурГУ, 2010. – С. 272-275.
51. Хандаров, Ф. В. Прогнозирование поступлений налога на прибыль организаций на основе гибридных нейронных сетей / З. Б.-Д. Дондоков, Ф. В. Хандаров // Кулагинские чтения : XI Международная научно-практическая конференция. – Чита : ЗабГУ, 2011. – С. 151-153.
52. Цой, Ю. Р. Эволюционный подход к настройке и обучению искусственных нейронных сетей [Электронный ресурс] / Ю. Р. Цой, В. Г. Спицын // Нейроинформатика. – 2006. – Т. 1, № 1. – С. 34-61. – Режим доступа : <http://www.niisi.ru/iont/ni/Journal/N1/TsoySp.pdf>, свободный. – Загл. с экрана.
53. Цыпкин, Я. З. Информационная теория идентификации / Я. З. Цыпкин. – М. : Наука, 1995. – 336 с.
54. Чимитдоржиева, Е. Ц. Влияние мирового кризиса на бюджетную политику субъектов Российской Федерации / Е. Ц. Чимитдоржиева, В. В. Доржиева // Экономический журнал. – 2009. – № 2 (16). – С. 18-32.

Литература на иностранном языке:

55. A limited memory algorithm for bound constrained optimization / R. H. Byrd [et al.] // SIAM Journal on scientific computing. – 1995. – Vol. 16, no. 5. – P. 1190-1208.
56. Adams, A. Function evaluation and the cascade-correlation architecture / A. Adams, S. Waugh // Proceedings of the 1995 IEEE International conference on neural networks. – 1995. – Vol. 2. – P. 942-946.
57. Alekseev, A. K. Comparison of advanced large-scale minimization algorithms for the solution of inverse ill-posed problems / A. K. Alekseev, I. M. Navon, J. L. Steward // Optimization methods & software. – 2009. – Vol. 24, no. 1. – P. 63-87.

58. ALGLIB : open source [Electronic resource] / сост. С. Бочканов, В. Быстрицкий. – Numerical analysis library, 1999-2013. – Режим доступа : <http://alglib.sources.ru>, свободный. – Загл. с экрана.
59. Alippi, C. Simple approximation of sigmoidal functions : realistic design of digital neural networks capable of learning / C. Alippi, G. Storti-Gajani // Proceedings of the IEEE International symposium on circuits and systems (Singapore, 1991). – 1991. – Vol. 3. – P. 1505-1508.
60. Angeline, P. J. An evolutionary algorithm that constructs recurrent neural networks / P. J. Angeline, G. M. Saunders, J. B. Pollack // IEEE Transactions on neural networks. – 1994. – Vol. 5, no. 1. – P. 54-65.
61. Asirvadam, V. S. Memory efficient BFGS neural-network learning algorithms using MLP-network : a survey / V. S. Asirvadam, S. F. McLoone, G. W. Irwin // Control applications, 2004 : Proceedings of the 2004 IEEE International conference on control applications (Taipei, Taiwan, Sept. 2-4, 2004). – Vol. 1. – P. 586-591.
62. Asuncion, A. UCI Machine learning repository [Electronic resource] / A. Asuncion, D. J. Newman. – Irvine, CA : University of California, 2007. – Режим доступа : <http://www.ics.uci.edu/~mllearn/MLRepository.html>, свободный. – Загл. с экрана.
63. Back, A. D. A low-sensitivity recurrent neural network / A. D. Back, A. C. Tsoi // Neural computation. – 1998. – Vol. 10. – P. 165-188.
64. Battiti, R. First and second order methods for learning : Between steepest descent and Newton's method / R. Battiti // Neural computation. – 1992. – Vol. 4, no. 2. – P. 141-166.
65. Bishop, C. M. Neural networks for pattern recognition / C. M. Bishop. – Oxford : University press, 1995. – 482 p.
66. Brownlee, J. The pole balancing problem – a review of a benchmark control theory problem : technical report no. 7-01 / J. Brownlee. – Swinburne University of technology, 2005. – 12 p.
67. Broyden, C. G. The convergence of a class of double-rank minimization algorithms / C. G. Broyden // Journal of the institute of mathematics and its applications. – 1970. – Vol. 6. – P. 76-90.
68. Chelouah, R. Tabu search applied to global optimization / R. Chelouah, P. Siarry // European journal of operational research. – 2000. – Vol. 123, no. 2. – P. 256-270.

69. Cybenko, G. Approximation by superposition of a sigmoidal functions / G. Cybenko // Mathematics of control, signals, and systems. – 1989. – Vol. 2. – P. 303-314.
70. Dao, V. N. P. A performance comparison of different back propagation neural networks methods in computer network intrusion detection / V. N. P. Dao, V. R. Vemuri // Differential equations and dynamical systems. – 2002. – Vol. 10, no. 1-2. – P. 201-214.
71. Differential evolution (DE) for continuous function optimization (an algorithm by Kenneth Price and Rainer Storn) [Electronic resource]. – Режим доступа : <http://www1.icsi.berkeley.edu/~storn/code.html>, свободный. – Загл. с экрана.
72. Draft technical report on C++ library extensions / M. H. Austern (ed.). – ISO/IEC DTR 19768, 2005. – 186 p.
73. Duch, W. Survey of neural transfer functions / W. Duch, N. Jankowski // Neural computing surveys. – 1999. – Vol. 2. – P. 163-212.
74. Equation of state calculations by fast computing machines / N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller // The journal of chemical physics. – 1953. – Vol. 21 (6). – P. 1087-1092.
75. Evett, I. W. Rule induction in forensic science : technical report / I. W. Evett, E. J. Spiehler // KBS in Government. – Aldermaston, Berkshire : Central research establishment, Home office forensic science service, 1987. – P. 107-118.
76. Fahlman, S. E. Faster-learning variations on back-propagation : an empirical study // Proceedings of the 1988 Connectionist models summer school / D. Touretzky, G. Hinton, T. Sejnowski (eds.). – Pittsburg, PA, 1988. – P. 38-51.
77. Fahlman, S. E. The cascade-correlation learning architecture / S. E. Fahlman, C. Lebiere // Advances in neural information processing systems. – 1990. – Vol. 2. – P. 524-532.
78. Fallgren, M. On the robustness of conjugate-gradient methods and quasi-Newton methods : master thesis at the Royal institute of technology / M. Fallgren. – Skolan för teknikvetenskap, Kungliga Tekniska högskolan, 2006. – 41 p.
79. Fletcher, R. A new approach to variable metric algorithms / R. Fletcher // Computer journal. – 1970. – Vol. 13. – P. 317-322.
80. Frasconi, P. Local feedback multi-layered networks / P. Frasconi, M. Gori, G. Soda // Neural computation. – 1992. – Vol. 4. – P. 120-130.

81. German, B. Database glass identification [Electronic resource]. – Aldermaston, Berkshire : Central research establishment home office forensic science service – Режим доступа : <http://archive.ics.uci.edu/ml/datasets/Glass+Identification>, свободный. – Загл. с экрана.
82. Glover, F. A user's guide to tabu search / F. Glover, E. Taillard // *Annals of operations research*. – 1993. – Vol. 41, no. 1. – P. 1-28.
83. Glover, F. Future paths for integer programming and links to artificial intelligence / F. Glover // *Computers & operations research*. – 1986. – Vol. 13, no. 5. – P. 533-549.
84. Glover, F. *Tabu Search* / F. Glover, M. Laguna. – Springer New York, 2013. – 3359 p.
85. Glover, F. Tabu search : part I / F. Glover // *ORSA Journal on computing*. – 1989. – Vol. 1, no. 3. – P. 190-206.
86. Glover, F. Tabu search : part II / F. Glover // *ORSA Journal on computing*. – 1990. – Vol. 2, no. 1. – P. 4-32.
87. Goldfarb, D. A family of variable metric updates derived by variational means / D. Goldfarb // *Mathematics of computing*. – 1970. – Vol. 24. – P. 23-26.
88. Good, P. I. *Resampling methods : a practical guide to data analysis* / P. I. Good. – 3rd ed. – Boston, MA : Birkhauser Boston, 2006. – 218 p.
89. Gorban, A. N. Approximation of continuous functions of several variables by an arbitrary nonlinear continuous function of one variable, linear functions, and their superpositions / A. N. Gorban // *Applied mathematics letters*. – 1998. – Vol. 11, no. 3. – P. 45-49.
90. Gorban, A. N. Principal graphs and manifolds / A. N. Gorban, A. Y. Zinovyev // *Handbook of research on machine learning applications and trends : algorithms, methods and techniques* / E. S. Olivas [et al.] (eds.). – PA : Information Science Reference, 2009. – Chapter 2. – P. 28-59.
91. Gradient-based learning applied to document recognition / Y. LeCun [et al.] // *Proceedings of the IEEE*. – 1998. – Vol. 86, no. 11. – P. 59-67.
92. Granville, V. Simulated annealing: a proof of convergence / V. Granville, M. Krivanek, J.-P. Rasson // *IEEE Transactions on pattern analysis and machine intelligence*. – 1994. – Vol. 16, no. 6. – P. 652-656.
93. Gruau, F. Genetic synthesis of modular neural networks / F. Gruau // *Proceedings of the 5th international conference on genetic algorithms*. – San Francisco, CA : Morgan Kaufmann, 1993. – P. 318-325.

94. Gruau, F. Neural network synthesis using cellular encoding and the genetic algorithm : unpublished PhD thesis / F. Gruau. – L'universite Claude Bernard, Lyon, 1994.
95. Hagan, M. T. Training feedforward networks with the Marquardt algorithm / M. T. Hagan, M. B. Menhaj // IEEE Transactions on neural networks. – 1994. – Vol. 5, no. 6. – P. 989-993.
96. Hassibi, B. Optimal brain surgeon and general network pruning / B. Hassibi, D. G. Stork, G. J. Wolff // IEEE International conference on neural networks (San Francisco, CA, 28 Mar-01 Apr, 1993). – 1993. – Vol. 1. – P. 293-299.
97. Hastie, T. The elements of statistical learning : data mining, inference, and prediction / T. Hastie, R. Tibshirani, J. Friedman. – 2nd ed. – Springer, 2009. – 533 p.
98. Hecht-Nielsen, R. Kolmogorov's mapping neural network existing theorem / R. Hecht-Nielsen // Proceedings of IEEE First annual international conference on neural networks. – San Diego, CA, 1987. – Vol. 3. – P. 11-14.
99. Hestenes, M. R. Methods of conjugate gradients for solving linear systems / M. R. Hestenes, E. Stiefel // Journal of research of the national bureau of standards. – 1952. – Vol. 49, no. 6. – P. 409-436.
100. Hinton, G. E. Deterministic Boltzmann machine learning performs steepest descent in weight-space / G. E. Hinton // Neural computation. – 1989. – Vol. 1, no. 1. – P. 143-150.
101. Horling, B. A survey of multi-agent organizational paradigms / B. Horling, V. Lesser // The knowledge engineering review. – 2005. – Vol. 19, no. 4. – P. 281-316.
102. Ingber, L. Simulated annealing: practice versus theory / L. Ingber // Mathematical and computer modelling. – 1993. – Vol. 18, no. 11. – P. 29-57.
103. International application of a new probability algorithm for the diagnosis of coronary artery disease / R. Detrano [et al.] // American journal of cardiology. – 1989. – Vol. 64. – P. 304-310.
104. ISO/IEC 14882:2011 Programming Language C++.
105. James, D. A comparative analysis of simplification and complexification in the evolution of neural network topologies / D. James, P. Tucker // Proceedings of genetic and evolutionary computation conference (GECCO – 2004). – New York, NY : Springer-Verlag, 2004.
106. Kirkpatrick, S. Optimization by simulated annealing / S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi // Science. – 1983. – Vol. 220 (4598). – P. 671-680.

107. Kohonen, T. Self-organized formation of topologically correct feature maps / T. Kohonen // *Biological cybernetics*. – 1982. – Vol. 43. – P. 59-69.
108. Kohonen, T. Self-organizing maps of symbol strings / T. Kohonen, P. Somervuo // *Neurocomputing*. – 1998. – Vol. 21. – P. 19-30.
109. Kosko, B. Bidirectional associative memories / B. Kosko // *IEEE Transactions on systems, man and cybernetics*. – 1988. – Vol. 18. – P. 49-60.
110. Koza, J. R. Genetic programming : on the programming of computers by natural selection / J. R. Koza. – Cambridge, MA : MIT Press, 1992. – 819 p.
111. Kreinovich, V. Y. Arbitrary nonlinearity is sufficient to represent all functions by neural networks : a theorem / V. Y. Kreinovich // *Neural networks*. – 1991. – Vol. 4. – P. 381-383.
112. Kuroe, Y. On activation functions for complex-valued neural networks : existence of energy functions / Y. Kuroe, M. Yoshida, T. Mori // *Artificial neural networks and neural information processing : proceedings of the joint international conference ICANN/ICONIP-2003 (Istanbul, Turkey, June 26-29, 2003)*. – Springer Berlin Heidelberg, 2003. – P. 985-992.
113. Lang, K. J. Learning to tell two spirals apart / K. J. Lang, M. J. Witbrock // *Proceedings of the 1988 Connectionist models summer school / D. Touretzky, G. Hinton, T. Sejnowski (eds.)*. – Pittsburg, PA, 1988. – P. 52-61.
114. LeCun, Y. A theoretical framework for back-propagation / Y. LeCun // *Proceedings of the 1988 Connectionist models summer school / D. Touretzky, G. Hinton, T. Sejnowski (eds.)*. – Pittsburg, PA, 1988. – P. 21-28.
115. LeCun, Y. Generalization and network design strategies : technical report CRG-TR-89-4 / Y. LeCun. – Department of computer science, University of Toronto, Canada, 1989. – 19 p.
116. LeCun, Y. Efficient learning and second-order methods : a tutorial at NIPS / Y. LeCun. – Denver, 1993. – 71 p.
117. LeCun, Y. Optimal brain damage / Y. LeCun, J. S. Denker, S. A. Solla // *Advances in neural information processing systems*. – 1990. – Vol. 2 – P. 598-605.
118. LeCun, Y. Second order properties of error surfaces : learning time and generalization / Y. LeCun, I. Kanter, S. A. Solla // *Advances in neural information processing systems*. – 1991. – Vol. 3. – P. 918-924.

119. Levenberg, K. A method for the solution of certain non-linear problems in least squares / K. Levenberg // Quarterly journal of applied mathematics – 1944. – Vol. 2. – P. 164-168.
120. Light, W. A. Ridge functions, sigmoidal functions and neural networks / W. A. Light // Approximation theory VII / E. W. Cheney, C. K. Chui, L. L. Schumacher (eds.). – Boston : Academic Press, 1992. – P. 163-206.
121. Livieris, E. A survey on algorithms for training artificial neural networks : technical report no. 08-01 / E. Livieris, P. Pintelas. – Patras : University of Patras, 2008. – 29 p.
122. Marquardt, D. An algorithm for least-squares estimation of nonlinear parameters / D. Marquardt // SIAM Journal on applied mathematics. – 1963. – Vol. 11, no. 2. – P. 431-441.
123. Matsumoto, M. Mersenne twister : a 623-dimensionally equidistributed uniform pseudorandom number generator / M. Matsumoto, T. Nishimura // ACM Transactions on modeling and computer simulations. – 1998. – Vol. 8, no. 1. – P. 3-30.
124. MinGW : minimalist GNU for Windows [Electronic resource]. – MinGW.org, 2013. – Режим доступа : www.mingw.org, свободный. – Загл. с экрана.
125. Moller, M. F. A scaled conjugate gradient algorithm for fast supervised learning / M. F. Moller // Neural networks. – 1993. – Vol. 6, no. 4. – P. 525-533.
126. Morales, J. L. Remark on "Algorithm 778 : L-BFGS-B : Fortran subroutines for large-scale bound constrained optimization" / J. L. Morales, J. Nocedal // ACM Transactions on mathematical software. – 2011. – Vol. 38, no. 1. – P. 1-4.
127. Mozer, M. Skeletonization : a technique for trimming the fat from a network via relevance assessment / M. Mozer, P. Smolensky // Advances in neural information processing systems. – 1989 – Vol. 1 – P. 107-115.
128. Nakamura, M. Guaranteed intervals for Kolmogorov's theorem (and their possible relation to neural networks) / M. Nakamura, R. Mines, V. Kreinovich // Interval computations. – 1993. – Vol. 3. – P. 183-199.
129. Nocedal, J. Numerical Optimization / J. Nocedal, S. Wright. – 2nd ed. – Springer, 2006. – 664 p.
130. Nocedal, J. Updating quasi-newton matrices with limited storage / J. Nocedal // Mathematics of Computation. – 1980. – Vol. 35, no. 151. – P. 773-782.

131. Pesulima, E. E. Digital implementation issues of stochastic neural networks / E. E. Pesulima, A. S. Pandya, R. Shankar // Proceedings of the international joint conference on neural networks (Washington, 1990). – 1990. – Vol. 2. – P. 187-190.
132. Pinkus, A. Approximation theory of the MLP model in neural networks / A. Pinkus // Acta numerica. – 1999. – Vol. 8. – P. 143-195.
133. Powell, M. J. D. Radial basis functions for multivariable interpolation : a review / M. J. D. Powell // Proceedings of the IMA Conference on algorithms for the approximation of functions and data. – Shrivvenham : Royal military college of science, 1985. – P. 143-167.
134. Prechelt, L. PROBEN1 – a set of benchmarks and benchmarking rules for neural network training algorithms : technical report no 21/94 / L. Prechelt. – Universitat Karlsruhe, 1994. – 38 p.
135. Reed, R. Pruning algorithms – a survey / R. Reed // IEEE Transactions on neural networks. – 1993. – Vol. 4, no. 5. – P. 740-747.
136. Riedmiller, M. A direct adaptive method for faster backpropagation learning : the RPROP algorithm / M. Riedmiller, H. Braun // Proceedings of the IEEE International conference on neural networks (April, 1993). – San Francisco, CA, 1993. – P. 586-591.
137. Rumelhart, D. E. Learning internal representations by error propagation / D. E. Rumelhart, G. E. Hinton, R. J. Williams // Parallel distributed processing / D. E. Rumelhart, J. L. McClelland, PDP Research Group. – Cambridge, MA : MIT Press, 1986. – Vol. 1 – Chapter 8. – P. 318-362.
138. Schraudolph, N. N. A fast, compact approximation of the exponential function : technical report / N. N. Schraudolph. – Lugano : IDSIA, 1998. – 11 p.
139. Shanno, D. F. Conditioning of quasi-newton methods for function minimization / D. F. Shanno // Mathematics of computation. – 1970. – Vol. 24, no. 111. – P. 647-656.
140. Siarry, P. Fitting of tabu search to optimize functions of continuous variables / P. Siarry, G. Berthiau // International journal for numerical methods in engineering. – 1997. – Vol. 40, no. 13. – P. 2449-2457.
141. Stanley, K.O. Evolving neural networks through augmenting topologies / K. O. Stanley, R. Miikkulainen // Evolutionary computation. – 2002. – Vol. 10, no. 2. – P. 99-127.

142. Storn, R. Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces : technical report TR-95-012 / R. Storn, K. V. Price. – Berkeley, ICSI, 1995. – 15 p.
143. Telfer, A. B. Energy functions for minimizing misclassification error with minimum-complexity networks / A. B. Telfer, H. Szu // Neural networks. – Vol. 7, no. 5. – 1994. – P. 809-817.
144. Treadgold, N. K. A cascade network algorithm employing progressive RPROP / N. K. Treadgold, T. D. Gedeon // Proceedings international work-conference on artificial and natural neural networks (Canary islands, 1997). – 1997. – P. 733-742.
145. Tsoy, Y. R. Using genetic algorithm with adaptive mutation mechanism for neural networks design and training / Y. R. Tsoy, V. G. Spitsyn // Optical memory and neural networks. – 2004. – Vol. 13, no. 4. – P. 225-232.
146. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus / J. W. Smith [et al.] // Proceedings of the symposium on computer applications and medical care. – 1988. – P. 261-265.
147. Volter, K. M. Introduction to variance estimation / K. M. Volter. – 2nd ed. – New York : Springer-Verlag, 2007. – 447 p.
148. Weigend, A. S. Generalization by weight-elimination with application to forecasting / A. S. Weigend, D. E. Rumelhart, B. A. Huberman // Advances in neural information processing systems. – 1991. – Vol. 3 – P. 875-882.
149. Werbos, P. J. Beyond regression : New tools for prediction and analysis in the behavioral sciences : Ph.D. thesis / P. J. Werbos. – Harvard University. – MA, 1974.
150. Wilamowski, B. M. Parity-n problems as a vehicle to compare efficiency of neural network architectures / B. M. Wilamowski, H. Yu, K. T. Chung // Industrial electronics handbook. Vol. 5 : Intelligent systems. – 2nd ed. – CRC Press, 2011. – Chapter 10. – P. 10-1-10-8.
151. Wilamowski, B. M. Solving parity-n problems with feedforward neural network / B. M. Wilamowski, D. Hunter, A. Malinowski // Proceedings of the international joint conference on neural networks (Portland, Oregon, July 20-23, 2003). – 2003. – Vol. 4. – P. 2546-2551.

152. Wolberg, W. H. Multisurface method of pattern separation for medical diagnosis applied to breast cytology / W. H. Wolberg, O. L. Mangasarian // Proceedings of the national academy of sciences of the USA. – 1990. – Vol. 87, no. 23. – P. 9193-9196.
153. Yao, X. Evolving artificial neural networks / X. Yao // Proceedings of the IEEE. – 1999. – Vol. 87, no. 9. – P. 1423-1447.
154. Yoshimura, A. A new measure for the estimation of the effectiveness of hidden units / A. Yoshimura, T. Nagano // Proceedings of the annual conference JNNS. – 1992. – P. 82-83.
155. Zhigljavsky, A. Stochastic global optimization / A. Zhigljavsky, A. Zilinskas. – Springer Science-Business Media, 2008. – 262 p.

ПРИЛОЖЕНИЕ А

В приложении приводятся примеры описаний сетей, полученных при решении практических задач. Описания приводятся в виде матриц смежности. Многоточиями заменены группы столбцов и строк, содержащие нулевые значения.

Таблица А. 1 – Матрица смежности сети для решения задачи Cancer

	...	9	10	11	12
0	...	-0,2433	0,2492	0	0
1	...	0,1257	0,9253	1,0663	0
2	...	0	1,0863	1,2350	-2,4699
3	...	0	0,2807	0	-3,6224
4	...	0	-0,1127	-0,1889	-1,6141
5	...	-0,0395	0,8949	0,7074	0
6	...	0	0,5351	0,3949	-3,5872
7	...	0	0,3845	0,4634	-1,8346
8	...	-0,0818	0,9262	0,4627	0
9	...	0	0	0	0
10	...	0	0	0	0
11	...	-4,0285	-1,0040	0	0
12	...	-4,1518	0,2262	0	0

Таблица А. 2 – Матрица смежности сети для решения задачи Glass

	...	9	10	11	12	13	14	15	16
0	...	0,283	0	0,352	-0,960	-1,280	-0,500	1,206	2,098
1	...	0	0	0,091	0	0,469	0,344	0	0
2	...	0,653	0,056	0,068	-0,210	0	-0,160	-1,250	-0,380
3	...	-0,700	0,563	0	1,239	0,237	-0,150	0	-0,760
4	...	0,216	0,281	-0,130	-0,210	0	0,140	0,020	0,305
5	...	1,159	0,111	-0,450	2,869	0,325	-0,190	0,031	-1,840
6	...	-0,730	3,103	0,314	-0,490	0	0,652	4,195	0,420
7	...	1,329	0,356	-0,640	1,497	0,190	2,638	1,262	-1,690
8	...	-0,160	0,590	-0,150	0	-0,030	-0,070	0	0
...
15	...	0	-1,750	0	-0,400	0	0	0	0
16	...	0,658	0	-0,400	1,513	0,631	-0,250	0	0

Таблица А. 3 – Матрица смежности сети для решения задачи Diabetes

	...	8	9	10	11
0	...	0	0,3604	1,5387	2,8310
1	...	1,0448	-1,2060	1,7132	0
2	...	0	0,2420	0	-0,9172
3	...	0,1139	-0,3526	-0,3861	1,0122
4	...	0,2978	-0,6998	-1,7803	-1,5992
5	...	1,4581	-2,2454	-0,8103	-2,6668
6	...	0,6747	-0,3174	-0,0211	0,4343
7	...	-0,1777	-0,0692	2,0825	0
...
10	...	0,6337	-1,1412	0	-2,2130
11	...	1,5802	-3,5564	0	0

Таблица А. 4 – Матрица смежности сети для решения задачи HeartAC

	...	35	36	37
0	...	0,0751	-0,5944	-0,2811
1	...	0,1236	-0,2100	-0,7987
2	...	-0,1328	-0,4442	2,0406
3	...	0	-0,1298	-1,0193
4	...	0	0	0
5	...	0,1922	0	1,1430
6	...	0	1,3307	-0,6395
7	...	0,1630	-1,4798	0
8	...	1,4873	-2,9154	2,3886
9	...	0	1,9718	-2,3979
10	...	0,9390	0	1,9480
11	...	-1,5525	1,7332	-0,9592
12	...	-0,0215	0,2178	-1,0799
13	...	1,3877	0	0
14	...	-1,3974	-3,3009	-0,7022
15	...	-0,4871	-1,1574	0
16	...	-1,2678	0	0
17	...	-0,5194	2,9579	0
18	...	0	-0,3412	0
19	...	2,0434	0,3446	0
20	...	-0,4429	-1,8416	0
21	...	0,1188	0,0835	-0,6638
22	...	0	-1,8799	-0,7676
23	...	0,4387	-3,1186	1,8519
24	...	2,2127	0	0
25	...	-0,0229	-0,2160	0
26	...	0,0885	0,5164	0,2333
27	...	0	0	1,5317
28	...	-3,0099	0	-0,6225
29	...	0,2824	-2,2247	0
30	...	0	0	-1,7778
31	...	0	0,0998	0,9392
32	...	0	-1,0299	0
33	...	0	-0,5400	0,1896
34	...	0	-0,2153	0
35	...	0	0	0
36	...	-1,1915	0	0
37	...	0	0	0

ПРИЛОЖЕНИЕ Б

Таблица Б. 1 – Утверждения опросника профессиональной готовности

1	Делать выписки, вырезки из различных текстов и группировать их по определенному признаку.
2	Выполнять практические задания на лабораторных работах по физике (собирать и составлять схемы, устранять в них неисправности, разбираться в принципе действия прибора).
3	Длительное время (более одного года) самостоятельно выполнять все работы, обеспечивающие рост и развитие растений (поливать, удобрять, пересаживать).
4	Сочинять стихи, рассказы, заметки, писать сочинения, признаваемые многими интересными, достойными внимания.
5	Сдерживать себя, не «выплескивать» на окружающих свое раздражение, гнев, обиду, плохое настроение.
6	Выделять из текста основные мысли и составлять на их основе краткий конспект, план, новый текст.
7	Разбираться в физических процессах и закономерностях, решать задачи по физике.
8	Вести регулярно наблюдение за развивающимся растением и записывать данные наблюдений в специальный дневник.
9	Мастерить красивые изделия из дерева, ткани, металла, сухих растений, ниток.
10	Терпеливо, без раздражения объяснять непонятное кому-либо, даже если приходится повторять это несколько раз.
11	Легко находить ошибки в письменных работах по русскому языку, литературе.
12	Разбираться в химических процессах, свойствах химических элементов, решать задачи по химии.
13	Разбираться в особенностях развития и во внешних отличительных признаках многочисленных видов растений.
14	Создавать законченные произведения живописи, графики, скульптуры.
15	Много и часто общаться с разными людьми, не уставая от этого.
16	На уроках иностранного языка отвечать на вопросы и задавать их, пересказывать тексты и составлять рассказы на заданную тему.
17	Отлаживать какие-либо механизмы (велосипед, мотоцикл), ремонтировать электротехнические приборы (пылесос, утюг, светильник).
18	Свое свободное время тратить преимущественно на уход и наблюдение за каким-нибудь животным.
19	Сочинять музыку, песни, имеющие успех у сверстников и взрослых.
20	Внимательно, терпеливо, не перебивая, выслушивать людей.
21	При выполнении заданий по иностранному языку без особых трудностей работать с текстами.
22	Налаживать и чинить электронную аппаратуру (приемник, магнитофон, телевизор, аппаратуру на дискотеках).
23	Регулярно, без напоминания взрослых, ухаживать за животными: кормить, чистить (животное и клетку), лечить, обучать.
24	Публично, при зрителях, разыгрывать роли, изображать кого-либо, декламировать стихи, прозу.
25	Увлекать делом, игрой, рассказом детей младшего возраста.
26	Выполнять задания по математике, химии, в которых требуется составлять логическую цепочку действий, используя при этом различные формулы, законы, теоремы.
27	Ремонтировать замки, краны, мебель, игрушки.

Продолжение таблицы Б. 1.

28	Разбираться в породах и видах животных (лошадей, птиц, рыб, насекомых), знать их характерные внешние признаки и повадки.
29	Всегда четко видеть, что сделано писателем, драматургом, художником, режиссером, актером талантливо, а что — нет, и уметь обосновать это устно или письменно.
30	Организовывать людей на какие-либо дела, мероприятия.
31	Выполнять задания по математике, требующие хорошего знания формул, законов и умения правильно применять их при решении.
32	Выполнять действия, требующие хорошей координации движений и ловкости рук: работать на станке, электрической швейной машине, монтировать изделия из мелких деталей.
33	Сразу замечать изменения в поведении животного или внешнем виде растения.
34	Играть на музыкальных инструментах, публично исполнять песни, танцевальные номера.
35	Выполнять работу, требующую обязательных контактов с множеством разных людей.
36	Выполнять количественные расчеты, подсчеты данных (по формулам и без них), выводить на основе этого различные закономерности, выявлять следствия.
37	Из типовых деталей, предназначенных для сборки определенных моделей или изделий, конструировать новые, придуманные самостоятельно.
38	Специально заниматься углубленным изучением биологии, ботаники, зоологии: читать научную литературу, слушать лекции, научные доклады.
39	Создавать на бумаге и воплощать новые, интересные модели (одежды, причесок, украшений), интерьеры помещений, конструкции изделий.
40	Влиять на людей: убеждать их, предотвращать конфликты, улаживать ссоры, разрешать споры.
41	Работать с условно-знаковой информацией: составлять и рисовать схемы, карты, чертежи.
42	Выполнять задания, в которых требуется мысленно представить расположение предметов или фигур в пространстве.
43	Длительное время заниматься исследовательскими работами в биологических кружках, на биостанциях, в зоокружках и питомниках.
44	Быстрее и чаще других замечать в обычном необычное, удивительное, прекрасное.
45	Сопереживать людям (даже не очень близким), понимать их проблемы, оказывать посильную помощь.
46	Аккуратно и безошибочно выполнять «бумажную», работу: писать, выписывать, подсчитывать.
47	Выбирать наиболее рациональный (простой, короткий) способ решения задачи — технической, логической, математической.
48	Занимаясь растениями или животными, выполнять тяжелую физическую работу, переносить неблагоприятные погодные условия, грязь, специфический запах.
49	Настойчиво, терпеливо добиваться совершенства в создаваемом или исполняемом произведении (в любой сфере творчества).
50	Говорить, сообщать что-либо, излагать свои мысли вслух.

ПРИЛОЖЕНИЕ В

Таблица В. 1 – Матрица смежности сети для задачи прогнозирования результатов ЕГЭ

	51	52	53	54	55	56	57	58	59	60	61	62	63	
1	...	0,1405	-0,5974	0,0204	-0,0550	-0,0320	-0,0008	0,0027	-0,0989	0,0307	0,0810	-0,0179	-9,9963	3,6338
2	...	-0,0695	0,3874	0,0156	-0,0047	-0,0263	-0,0143	0,0080	0,0098	-0,0902	-0,0325	0,0012	9,9829	-9,9998
3	...	0,1179	-0,5857	0,0388	0,0291	-0,0280	0,1089	-0,0255	-0,0797	0,0271	-0,0498	-0,0046	-9,9884	9,9990
4	...	-0,1323	0,4261	0,0145	0,0330	0,0491	0,0051	-0,0010	0,1311	-0,0797	0,0074	-0,0008	9,9901	-9,9920
5	...	-0,0749	0,6231	0,0004	0,0814	0,1263	-0,0048	-0,0371	0,0974	-0,0357	0,0168	-0,0010	9,9927	-9,9982
6	...	0,0341	0,5752	-0,0996	0,0237	0,0662	0,0363	-0,0075	0,0762	-0,0310	0,0483	0,0310	10,0000	-9,9918
7	...	-0,0203	0,1571	-0,0231	-0,0665	-0,0416	-0,0019	0,0147	0,1009	0,0119	-0,0128	-0,0146	6,9704	-2,6650
8	...	0,0772	-0,1400	0,0345	-0,0064	-0,0018	-0,0095	0,0189	-0,0714	0,0229	-0,0282	-0,0137	-5,7338	-7,0208
9	...	-0,1038	0,5679	0,0609	-0,0234	0,0222	0,0336	0,0213	-0,0496	-0,0144	0,0069	-0,0490	9,9930	-9,9967
10	...	-0,1111	0,3182	0,0550	-0,0248	0,0042	-0,0242	0,0388	0,0887	-0,0459	0,0195	-0,0817	9,9977	-9,9926
11	...	-0,0278	0,4522	-0,0256	0,0872	0,1554	-0,0414	0,0015	0,0163	0,0510	-0,0031	-0,0105	9,9876	-9,9919
12	...	0,1463	0,0725	-0,0428	0,0054	0,0549	-0,0596	0,0271	-0,0232	-0,0144	-0,0212	0,0223	-8,2222	5,5734
13	...	0,0679	-0,2748	-0,0109	0,0384	0,0716	0,0748	-0,0399	-0,0711	-0,0032	-0,0154	0,0352	-9,9850	-2,9783
14	...	0,1477	-0,0889	0,0165	-0,0426	0,0109	0,0351	0,0343	0,0451	-0,0492	0,0022	-0,0044	-8,9388	9,6789
15	...	0,0764	0,3777	0,0159	0,0014	0,0557	0,0242	-0,0032	0,0270	0,0024	-0,0961	0,0144	9,8228	-9,9926
16	...	0,0514	-0,4525	-0,0391	-0,0637	0,0022	-0,0503	0,0038	-0,0381	-0,0005	0,0071	-0,0238	-9,9874	9,9972
17	...	-0,0654	0,4020	0,0696	-0,0089	0,0626	-0,0145	-0,0027	-0,0330	0,0280	0,0468	-0,0030	9,9828	-4,5962
18	...	0,0315	0,4165	-0,0218	0,0600	0,0748	-0,0330	0,0315	-0,0152	0,0457	0,0274	0,0075	9,9835	-10,0000
19	...	0,0301	-0,2593	0,0132	0,0078	-0,0205	0,0186	-0,0225	-0,0111	-0,0491	-0,0353	0,0044	-9,4065	9,8368
20	...	0,1983	-0,7873	0,0516	0,0336	-0,0559	0,0351	-0,0237	-0,0675	0,0649	-0,0268	0,0522	-9,9869	8,0025
21	...	0,1347	-0,3192	-0,0105	0,0149	0,0532	-0,0361	-0,0296	-0,0085	0,0169	0,0992	-0,0054	-9,9786	2,0477
22	...	-0,1074	0,5382	0,0421	0,0505	0,0419	-0,0793	0,0468	-0,0063	0,0173	0,0861	0,0088	9,9839	-9,9969
23	...	-0,0020	-0,3334	-0,0230	0,0180	-0,0079	0,0434	0,0173	0,0975	-0,0151	-0,0477	0,0204	-5,4281	8,6577
24	...	0,0077	-0,1313	-0,0089	0,0519	0,0631	-0,0061	0,0028	-0,0610	-0,0345	-0,0814	0,0185	-9,9683	9,9997
25	...	0,1199	-0,5501	-0,1022	-0,0258	-0,1429	-0,0815	-0,0075	-0,0042	0,0626	0,0770	0,0246	-9,9985	9,9934
26	...	0,1861	-0,3737	0,0344	-0,0058	-0,0064	0,0516	0,0127	0,1224	0,0397	-0,0086	-0,0139	-9,9925	9,9999
27	...	-0,0681	0,6391	0,1264	0,0366	0,0336	-0,0060	-0,0206	0,0704	-0,0513	-0,0117	-0,0213	9,9947	-7,3627
28	...	0,0698	-0,5372	-0,0152	-0,0270	-0,1354	0,0055	0,0077	-0,1183	0,0557	-0,0399	-0,0089	-9,9789	9,9995
29	...	0,0946	-0,2386	0,0445	0,0174	-0,0585	-0,0416	-0,0299	0,0800	0,0058	0,0239	0,0409	-6,5282	8,1470
30	...	0,0785	0,3671	0,0074	-0,0309	-0,0174	0,0188	0,0290	0,0776	0,0097	-0,0259	0,0246	9,9792	-9,9879

Продолжение таблицы В. 1

	51	52	53	54	55	56	57	58	59	60	61	62	63	
31	...	0,1403	-0,3955	0,0188	-0,0131	-0,1261	0,0288	-0,0006	0,0204	-0,0119	-0,0064	-0,0236	-9,9910	9,9874
32	...	0,1289	-0,4544	-0,0569	0,0223	-0,0431	-0,0444	-0,0110	-0,0697	0,0181	0,0246	0,0377	-9,9779	8,3743
33	...	0,0297	-0,3213	-0,0666	-0,0037	-0,0453	0,0968	-0,0151	0,0067	0,0498	-0,0473	0,0014	-9,9933	4,7459
34	...	-0,0928	0,5981	0,0865	-0,0250	-0,0497	-0,0355	0,0150	0,0488	-0,0065	0,0031	0,0069	9,9976	-8,1624
35	...	0,0064	0,5804	-0,0007	-0,0079	0,0093	-0,0251	0,0495	-0,0327	0,0440	0,0358	0,0350	9,9974	-9,9954
36	...	0,2146	-0,3590	0,1106	0,0218	0,0577	0,0135	-0,0120	0,1159	0,0149	-0,0511	0,0316	-9,9899	9,9942
37	...	-0,1464	0,3709	-0,0090	-0,0136	-0,0442	0,0372	-0,0471	0,1367	-0,0376	-0,0321	-0,0180	9,9727	-9,9937
38	...	-0,0524	0,4911	0,0374	0,0018	0,0076	-0,0538	-0,0207	0,0361	0,0094	-0,0381	-0,0026	9,9826	-9,9950
39	...	0,2249	-0,4220	-0,0226	-0,0689	-0,0660	0,0398	-0,0069	0,0714	-0,0109	0,0688	-0,0265	-9,9843	9,9959
40	...	0,0846	-0,4294	0,0436	-0,0425	0,0359	0,0002	0,0002	-0,1132	0,0255	-0,0036	0,0171	-9,9941	5,5942
41	...	0,0418	0,4068	0,0991	-0,0477	-0,0869	0,0876	0,0107	0,1118	0,0581	0,0646	0,0330	8,7547	-9,9989
42	...	-0,2037	0,4340	-0,0581	-0,0088	-0,0135	0,0802	0,0197	0,1217	-0,0617	-0,0587	-0,0269	9,7789	-3,1469
43	...	0,0378	0,4097	-0,1170	0,0626	0,0881	-0,0372	-0,0132	0,0142	-0,0213	0,1151	0,0029	9,9918	-9,9897
44	...	0,1557	-0,4281	-0,0814	-0,0175	-0,0418	0,0114	-0,0124	0,0049	-0,0085	-0,0357	-0,0023	-9,9923	9,9957
45	...	-0,0671	-0,1981	0,1127	-0,0467	0,0120	-0,0839	0,0199	-0,1102	-0,0033	-0,0241	-0,0067	-9,9740	9,9840
46	...	-0,1382	0,5203	-0,1183	0,0353	0,0222	-0,0027	-0,0043	0,0531	0,0014	0,0814	-0,0213	9,9934	-9,9972
47	...	-0,0908	0,4833	-0,0188	-0,0301	0,0488	-0,0659	0,0244	-0,1215	-0,0051	0,0251	-0,0006	9,9869	-5,1341
48	...	-0,0569	-0,2253	-0,0251	0,0440	-0,0839	0,1267	0,0231	-0,0678	0,0071	-0,0761	0,0009	-9,9923	5,4197
49	...	0,0223	0,3314	0,0711	0,0020	0,0702	-0,0028	-0,0220	0,0822	-0,0334	0,0329	-0,0101	9,9766	-9,9883
50	...	0,0481	-0,1709	-0,0657	0,0007	0,0376	0,0440	-0,0082	-0,0665	0,0402	-0,0069	0,0091	-9,2318	9,8020
...
62	...	0,0718	-0,4402	-0,0298	0,0020	-0,0418	0,0597	-0,0011	-0,0501	0,0219	-0,0179	0,0153	0,0000	-5,3010
63	...	-0,0980	0,4473	0,0053	-0,0092	0,0261	-0,0166	0,0056	0,0689	-0,0134	0,0093	0,0004	0,0000	0,0000