ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Федеральное государственное
образовательное бюджетное учреждение
высшего профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
им. проф. М. А. БОНЧ-БРУЕВИЧА»

М. П. Белов, Ф. В. Филиппов

ИНТЕЛЛЕКТУАЛИЗАЦИЯ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ

ЧАСТЬ 1

УЧЕБНОЕ ПОСОБИЕ

СПб ГУТ)))

САНКТ-ПЕТЕРБУРГ
2014

УДК 004.032(075.8)
ББК 32.973-018я73
Б43

Рецензенты:
доктор технических наук, профессор кафедры системного анализа и управления Национального минерально-сырьевого университета «Горный» *Д. А. Первухин*,

доктор технических наук, профессор кафедры автоматизированного электропривода и электротехники Санкт-Петербургского государственного университета растительных полимеров *В. Д. Кулик*

Утверждено редакционно-издательским советом СПбГУТ в качестве учебного пособия

Белов, М. П.
Б43 Интеллектуализация инфокоммуникационных систем : учебное пособие. Часть 1 / М. П. Белов, Ф. В. Филиппов ; СПбГУТ. – СПб., 2014. – 84 с.

Приводятся основы теории интеллектуализации инфокоммуникационных систем, рассмотрено применение нейронных сетей, генетических алгоритмов, Интернет-технологии, смысловая разметка контента, набор данных для поисковых машин, универсальный способ представления знаний, автоматизация интерпритации контента, язык SPARQL.

Предназначено для студентов, обучающихся по направлению подготовки 09.03.02, а также для аспирантов и специалистов в области телекоммуникаций. Возможно использование при выполнении курсовых и дипломных работ.

УДК 004.032(075.8)
ББК 32.973-018я73

© Белов М. П., Филиппов Ф. В. 2014

© Федеральное государственное образовательное бюджетное учреждение высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», 2014

СОДЕРЖАНИЕ

Введение 4

1. СИНТЕЗ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ 8

1.1. Задачи синтеза современных инфокоммуникационных систем 8

1.2. Методы структурного синтеза РП ИКС П 10

1.2.1. *Метод морфологического синтеза структуры ИКС П* 14

1.2.2. *Структурный синтез по альтернативным деревьям* 15

1.2.3. *Синтез проектных решений с учетом дополнительных условий* 16

1.2.4. *Структурный синтез с ограничениями на основе N-дольных графов* 17

1.2.5. *Генетико-морфологический метод* 18

1.3. Применение генетических алгоритмов для структурного синтеза ИКС П 22

1.4. Параметрическая оптимизация структур ИКС П 28

1.5. Оптимизация работы блочно модульной системы обработки данных с использованием генетических алгоритмов 31

1.6. Применение генетических алгоритмов для оптимизации процесса передачи данных в инфокоммуникационных системах 36

2. ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ В РП ИКС П 48

2.1. Область применения нейронных сетей 48

2.2. Топология нейронных сетей 52

2.3. Модели нейронных сетей 53

2.4. Алгоритм обратного распространения 64

2.5. Нейронные сети Хопфилда и Хэмминга 69

2.6. Требования к алгоритму обучения 74

2.7. Выбор эталонов для обучения нейросети 75

2.8. Повышение эффективности обучения НС обратного распространения 77

Список литературы 82

ВВЕДЕНИЕ

В промышленности интенсивно создаются и развиваются информационно-управляющие системы управления распределенными технологическими процессами на предприятиях. Такие системы (инфокоммуникационные системы – ИКС) интегрируют телекоммуникационные, промышленные сети, компьютерные и информационные ресурсы в единую инфокоммуникационную среду, в которой процессы обработки, хранения и распространения информации представляются единым процессом информационного обслуживания потребителей [22]. Потоки информации о производственных процессах на объекты управления – электронный документооборот между различными структурными подразделениями и информация, поступающая с промышленных агрегатов и технологических комплексов – обеспечивают ИКС, безотказное функционирование которых возможно только при условии упреждения сбоев, возникающих как по причине коллизий трафика, так и из-за тренда параметров аппаратуры. В связи с чрезвычайной сложностью развивающихся ИКС важное значение имеет оптимизация процесса передачи данных в распределенном технологическом процессе.

В настоящее время растет интеграция внедряемых на предприятиях информационных технологий в единую инфраструктуру с целью повышения эффективности производственных и бизнес-процессов предприятий различных отраслей. В результате происходит усложнение промышленной инфокоммуникационной системы предприятия, превращая ее в распределенную (гетерогенную) в пространстве сложную инфокоммуникационную систему.

Основными сетевыми элементами ИКС являются периферийные устройства (телефонные аппараты, IP-, GSM- и SIP-телефоны, WEB-серверы, агрегаты и механизмы, базы данных (БД), CAD/CAM-системы, службы бизнес и видео и т. д.) и устройства коммутации, маршрутизации и сигнализации.

Система управления (СУ) ИКС представляется увязанными в единую сеть центрами управления (БД, автоматизированными рабочими местами, сетевыми картами промышленных механизмов и агрегатов, коллекторами и т. п.), биллинг-системой, городской сетью передачи данных и/или межрегиональной компанией (МРК) связи, предназначенной для переноса служебной информации в пределах СУ (предполагается, что СУ МРК в общем случае включает в свой состав СУ филиалов компании).

В состав СУ ИКС входят:

– информационно-математическая база, включающая методы, модели и алгоритмы, и предназначенная для выработки управляющих воздействий;

– технические службы, являющиеся элементами технологической цепи управления, на которые возлагаются функции подготовки, передачи и хранения информации.

Инструментальные средства СУ ИКС ориентируются на решение трех классов задач – долгосрочных, среднесрочных и текущих задач оператив-

ного технологического управления на ИКС. В свою очередь, каждый класс распадается на частные управленческие задачи, различающиеся целевыми требованиями, критерием оптимальности, ограничениями и составом вектора управляемых переменных [22].

Каналы переноса пользовательского и служебного трафика могут быть общими. К служебному трафику относятся биллинговый управленческий трафик, создаваемый запросами внешних клиентов центра управления сетью, трафик между центром управления сетью и оборудованием сбора данных.

Служебный трафик может передаваться как случайным образом, так и по расписанию. Поэтому актуально повышение эффективности оперативного управления распределенной промышленной ИКС предприятия (РП ИКС П), а это в свою очередь возможно только при разработке соответствующего алгоритма управления.

Перспективными методами оценки состояния и оптимизации процессов передачи информации в РП ИКС П считаются интенсивно развиваемые технологии генетических алгоритмов (ГА) и нейронных сетей (НС). По сравнению с методами ГА статистические методы НС дают меньшую точность в решении задач оценки состояния и оптимизации процессов передачи информации в РП ИКС П и не обладают адаптивными свойствами [29]. Растущее применение ГА и НС обусловлено их достоинствами: способностью обучения на новых данных и неограниченной сферой применения.

Основные тенденции развития ИКС в начале третьего тысячелетия можно выразить двумя словами: биологизация и гибридизация. Под *биологизацией* понимается построение и исследование моделей поведения сложных объектов и способов управления ими на основе имитации механизмов, реализованных природой в живых существах. Такой подход обусловлен тем фактом, что многие так называемые классические методы обработки информации воспринимаются в настоящее время как простейшие реализации универсальных способов функционирования биологических объектов. С другой стороны, стремительное увеличение вычислительных мощностей и развитие математического аппарата позволили подступиться к решению таких задач, размерность которых еще 5–10 лет назад была непреодолимым барьером для исследователя.

Гибридизация состоит в совместном применении различных методов и/или моделей для обработки информации об одном и том же объекте. Парадигма такого подхода основана на согласии с тем, что любая сколь угодно сложная искусственная модель реального объекта всегда будет примитивнее и проще оригинала, и только многоаспектное его изучение с последующей интеграцией получаемых результатов позволит обрести необходимые знания или приблизиться к оптимальному решению.

В конце XX в. Япония провозгласила программу Real World Computing (вычисления в реальном мире). Речь идет прежде всего о том, чтобы дать вычислительным и управляющим техническим системам возможность

самостоятельно, без помощи человека воспринимать воздействия внешнего мира и действовать в нем. Авторы программы огромную роль – до 40 % ее содержания – отводят исследованию естественных и созданию искусственных нейронных систем (ИНС), в дальнейшем нейронных сетей (НС).

Современные РП ИКС П имеют сложную структуру построения. При проектировании таких систем возможно применение большого количества разнообразного инфокоммуникационного оборудования. Поэтому на стадии проектирования необходимо синтезировать оптимальную структуру построения таких объектов управления, а также за счет соответствующих параметров инфокоммуникационного оборудования обеспечить требуемые динамические характеристики РП ИКС П. В этом случае можно говорить о структурной неопределенности.

Эффективное управление конкретными ИКС невозможно без оптимального структурно-параметрического синтеза, анализа и исследования систем и процессов управления.

Инструментальные программные средства, осуществляющие структурно-параметрический синтез и оптимизацию, являются неотъемлемой частью систем автоматизированного проектирования (САПР) новых поколений, точно так же, как почти во всех современных САПР имеется модуль параметрического синтеза и оптимизации. Также структурно-параметрический синтез поддерживается в перспективных CASE-системах (Computer-Aided System Engineering) и в CALS-технологиях (Computer Aided Logistic System).

Современные методы структурно-параметрического синтеза позволяют уменьшить сроки проектирования, повысить эффективность проектирования.

Но не всегда традиционные методы структурного синтеза могут решить поставленную задачу, поэтому в настоящее время большой интерес представляет применение для этих целей современных инструментальных средств, таких как нейронные сети и генетические алгоритмы.

Все чаще управление процессом работы (передача, обработка, хранение и т. п.) с информацией в РП ИКС П осуществляется не только по стандартным законам управления, но и по модели информационного процесса и объекта управления.

Рост потребностей в комплексной автоматизации делопроизводства предприятий, организаций и фирм обуславливает внедрение новых современных информационно-управляющих технологий, что в свою очередь приводит к возрастающей в разы обработке различной информации. Особенностью обработки информационных данных является то, что она происходит в реальном времени. Это предъявляет определенные требования не только к техническим средствам, но и к алгоритмам обработки данных. В этом случае необходимо разрабатывать алгоритмы обработки данных с использованием современных подходов.

Разработка и сопровождение сложных интеллектуальных инфокоммуникационных систем без использования интернет-технологий не возможна.

Интернет – крупнейший информационный репозитарий, содержимое которого непрерывно растет и представлено на самых разнообразных языках. Становится все труднее находить смысл во всем этом содержимом в силу того, что поиск основан на содержании страниц, а не на семантическом или смысловом значении этого содержания [4]. Семантические веб-технологии – это эффективный способ структурирования данных в Интернете. Как только будет создан семантический Интернет, он даст возможность разметки всего содержания Интернета, описания каждого элемента информации и обеспечения семантического значения этих элементов. Такую структуру можно отождествить с распределенной базой данных, которая связана в глобальном масштабе с содержанием документов в Интернете. Причем эта связь осуществляется способом, понятным компьютерам.

Широкое применение семантических технологий позволит использовать Интернет, как гигантскую уникальную информационную базу знаний во всех областях человеческой деятельности. Развитие информационных технологий по пути структурирования данных Интернета и совершенствования систем управления полученными структурами является важнейшей задачей повышения «интеллекта» поисковых машин. Вследствие этого данные, важные для пользователей, станут доступными и для машин, которые будут мгновенно, в удобной форме готовить их для пользователей.

1. СИНТЕЗ ИНФОКОММУНИ КАЦИОННЫХ СИСТЕМ

1.1. Задачи синтеза современных инфокоммуникационных систем

Целью синтеза является построение РП ИКС П, удовлетворяющей сформулированным в техническом задании требованиям. Поэтому при проектировании РП ИКС П ее синтез можно условно разбить на два этапа:

1) выбор структур РП ИКС П, т. е. элементов и топологии причинно-следственных связей между ними, структур операторов элементов, в частности, алгоритмов управляющих устройств и значений их параметров;

2) параметрический синтез. На этапе структурного синтеза кроме синтеза структуры РП ИКС П и топологии причинно-следственных связей между ее подсистемами решаются также задачи синтеза: 1) управляющих воздействий; 2) компенсаторов возмущений; 3) регуляторов.

В зависимости от того, какую структуру имеет ИКС П и какие технологические задачи она решает, следует различать подзадачи синтеза: 1) автономных систем; 2) отдельных каналов связи; 3) расширенных систем.

Методы синтеза ИКС П можно разделить на три группы: аналитические; графические; численные.

Аналитические методы синтеза линейных систем обычно сводятся к действиям над полиномами, дробно-рациональными функциями, матрицами (числовыми, полиномиальными, дробно-рациональными). Наиболее часто используются алгебраические процедуры раскрытия определителей, решения систем уравнений; различных типов, определения собственных значений матриц и корней полиномов.

Недостаток аналитических методов заключается в том, что они требуют полной определенности постановок задач, недостижимой во многих практических ситуациях [1]. Поэтому аналитические методы должны являться частью более общих процедур синтеза, строящихся как последовательное раскрытие неопределенности.

Операторные методы позволяют решать задачи синтеза автономных систем или отдельных каналов передач ИКС П.

К графическим, или графоаналитическим, методам синтеза относятся частотные методы синтеза ИКС П. Частотные методы, использующие возможности компьютерной графики, позволили создать новые методы синтеза многомерных систем [1].

Использование графических образов при расчете систем управления ИКС П на этапах построения и структурного представления моделей, получения решения и отображения результатов является важным условием их «физичности», тесной связи процедур исследования с параметрами объектов и требованиями к поведению систем.

Численные методы синтеза связаны с многократным вычислением характеристик и показателей качества полностью определенных вариантов систем. Показатели качества сопоставляются со сформулированными требованиями. Анализ повторяется многократно до исчерпания исходного множества решений.

Синтез как многовариантный анализ так или иначе связан с перебором вариантов. Полный перебор применяется в случае не очень большой мощности множества возможных структур построения ИКС П. Прямой (простой) перебор реализуется с помощью программных средств гибкого редактирования моделей, быстрого построения характеристик и вычисления показателей качества.

В случае большого числа исходных вариантов необходимо применять комбинаторные методы, позволяющие сократить перебор за счет использования априорной информации и текущей информации, получаемой в результате анализа предыдущих вариантов.

Поисковые методы синтеза позволяют найти значения параметров, доставляющие минимум показателям качества, или построить поверхности равного значения показателей качества в пространстве параметров без полного перебора. При этом используются весьма разнообразные методы оптимизации или нелинейного программирования [30].

Достоинством численных методов синтеза путем полного перебора и многовариантного анализа является универсальность и надежность. Их недостаток – большая трудоемкость при значительной исходной неопределенности ситуации.

В современных технологиях синтеза, использующих компьютерные программы, интегрируются все перечисленные методы – аналитические, графические, численные, а также применяется теория нейронных сетей и генетических алгоритмов.

От компьютерных программ требуется максимальная графичность процедур ввода и редактирования моделей, образность формирования требований и отображения результатов.

Интерактивный характер синтеза должен поддерживаться за счет быстроты выполнения вычислительных процедур, что обеспечивается наиболее полным использованием аналитических методов и символьных алгоритмов, применения эффективных численных алгоритмов решения систем уравнений, поиска корней полиномов и собственных значений матриц, условных экстремумов функций и т. д.

Возможно применение топологического синтеза в случае, когда возмущение нельзя измерять непосредственно, что исключает создание системы компенсации, функционирующей по принципу разомкнутого управления [1].

1.2. Методы структурного синтеза РП ИКС П

Для определения оптимальной структуры РП ИКС П исходными данными обычно являются следующие.

1. Выполняемые системой функции, которые могут быть формализованы в виде множества решаемых задач $E=\{E_i\}$. Каждая из задач E_i, $i=\overline{1, l}$ может состоять из этапов q_i, $q_i=\overline{1, Q_i}$ и иметь вариантов δ_i, $\delta_i=\overline{1, \Delta_i}$ решения в РП ИКС П.

2. Связи между задачами и их этапами, которые могут задаваться в виде графа $G_E=\{E_{qi}, E'_{qi}\}$, где $\{E_{qi}, E'_{qi}\}\in E$. Дуги графа $\{E_{qi}, E'_{qi}\}$ характеризуют отношения следования, существующие между решаемыми задачами и их этапами, и соответствуют направлениям информационных потоков.

3. Множество возможных узлов ИКС П $M=\{M_j\}$ и связей между ними, которые задаются в виде графа $G_M=\{M_j, M'_j\}$, где $j, j'=\overline{1, J}$. Вершины графа G_M отображают узлы, а дуги – связи между ними.

В некоторых случаях может быть задан конечный набор вариантов возможных узлов ИКС П и связей между ними, т. е. G_M^{γ}, $\gamma=\overline{1, \gamma}$, где G_M^{γ} – γ-й возможный вариант.

4. Виды и характеристики технических средств, применение которых возможно в ИКС П, пусть $A=\{a_l\}$ – множество возможных технических средств и $l=\overline{1, L}$ – тип технического средства.

5. Внешние для системы источники и потребители информации по всем этапам задач.

Тогда задача оптимальной структуры ИКС П состоит в нахождении:

– узлов системы M;

– связей между ними G_M;

– возлагаемых на технические средства задач E и вариантов их решения (δ_i, $i=\overline{1, I}$) в распределении их по уровням и узлам системы и в выборе комплекса технических средств $\overline{A}$, при которых максимизируется эффект решения задач в ИКС П, т. е.

$$\max \sum_{q_i\in E, \delta_i\in\Delta} \omega_{q_i}^{\delta_i} \sum x_{q_i, j, a_i}^{\delta_i},$$

$$\overrightarrow{E}\in E, \overrightarrow{M}\in M, \overrightarrow{G}_E\in G_E, \overrightarrow{G}_M\in G_M, \overrightarrow{A}\in A,$$

где $\omega_{q_i}^{\delta_i}$ – эффект от внедрения q_i-го этапа i-й задачи при использовании δ_i-го варианта его решения; переменная $x_{q_i, j, a_i}^{\delta_i}$ принимает значение 1, если

q_i -й этап i-й задачи при использовании δ_i-го варианта его реализации решается в j-м узле a_l-м техническим средством l-го типа, и значение 0 – в противном случае. Здесь предполагается, что каждый этап задачи решается в одном узле.

При этом очевидно, что $\sum x^{\delta_i}_{q_i, j, a_i} = 1$.

Эффективность системы базируется на обоснованном соотношении выгод (затрат) и сроков их получения.

Оптимальная структура ИКС П определяется при ограничении на ресурсы, загрузку технических средств, каналов передачи информации и своевременность решения задач, т. е.

$$\sum_{q_i \in E,\ a_l \in A, \delta_i \in \Delta} R^{\delta_i}_{q_i, j, a_l, k} x^{\delta_i}_{q_i, j, a_l} \le R_k,$$

где $k = \overline{1, K}$ – тип ресурса; R_k – величина используемого ресурса.

Таким образом, данное выражение может быть определено, как функционал качества, предназначенный для синтеза эффективной структуры ИКС П.

Суммарное количество вариантов анализа построения ИКС П можно рассчитать по формуле

$$\sum_{q_i \in E,\ a_l \in A, \delta_i \in \Delta} \lambda^{\delta_i}_{q_i, j} t^{\delta_i}_{q_i, j, a_l, k} x^{\delta_i}_{q_i, j, a_l} \le \rho_{j, a_l},$$

где $\lambda^{\delta_i}_{q_i, j}$ – интенсивность (частота) решения q_i-го этапа i-й задачи при δ_i-м варианте решения; ρ_{j, a_l} – загрузка a_l-го технического средства l-го типа в j-м узле; $t^{\delta_i}_{q_i, j, a_l, k}$ – время выполнения q_i-го этапа i-й задачи в j-м узле a_l-м техническим средством при δ_i-м варианте решения.

Временные ограничения для различных задач ИКС П могут иметь сложный вид и требуют анализа работы различных узлов. Например, для оперативных задач необходимо, чтобы вероятность превышения времени решения задачи допустимой величины $t_{\text{доп}\, q_i}$ была не более заданной ε_{qi}:

$$P\left\{ t^{\delta_i}_{q_i, j, a_l} + \tau^{\delta_i}_{q_i, j, a_l} > t_{\text{доп}\, q_i} \right\} \le \varepsilon_{qi},$$

где $\tau^{\delta_i}_{q_i, j, a_l}$ – время ожидания в j-м узле.

Рассмотренная задача является задачей нелинейного математического программирования.

Рационально построенная ИКС П обычно имеет иерархическую структуру, причем узлы системы одного уровня можно разбить на группы, в которых они идентичны. Это позволяет рассматривать работу лишь одного «типового» узла для каждой из групп одного уровня.

Критерий качества функционирования для таких систем обычно аддитивен и имеет вид

$$W = \sum_{r_m=1}^{R_m} \sum_{m=1}^{M} \sum_{j=1}^{M} W_{j_r,r},$$

где $W_{j_r,r}$ – критерий для j-го узла m-го уровня r-й группы; $r_m = \overline{1, R_m}$ – число групп подсистем m-го уровня системы: $m = 1$; M – число уровней системы.

В случае, когда задано множество решаемых задач, множество узлов системы и связей между ними ($\overline{E}, \overline{M}, \overline{G_m}$ соответственно), а также распределение задач по узлам системы, возникает задача выбора комплекса технических средств, обеспечивающего функционирование ИКС П по заданным критериям качества.

Для анализа динамики функционирования многоступенчатых систем и оценки таких характеристик, как задержка в управлении, влияние ненадежности технических устройств и алгоритмов функционирования систем, их удобно формализовать в виде сетей массового обслуживания соответствующей структуры и использовать аналитические методы для определения оценок и методов статистических испытаний в более сложных случаях.

Среди всех подходов к решению задачи структурного синтеза наибольшее распространение в системах автоматизированного проектирования получили различные методы, принадлежащие к классу комбинаторно-логических. В основе этого подхода лежит хорошо организованный перебор в массиве решений.

Рассмотрим основные допущения:

– проектируемый объект, будь то ИКС П или процесс, имеет структуру;

– проектируемый объект принадлежит к некоторому классу объектов (множество аналогов и прототипов), имеющих одинаковое функциональное назначение;

– множество аналогов и прототипов обладает достаточной мощностью для того чтобы поиск новых сочетаний в этом комбинаторном пространстве был результативен;

– составные части объектов класса обладают «хорошими комбинаторными способностями».

Это значит, что принципы действия ИКС П не различаются настолько, чтобы запретить объединение разных компонентов в составе новой ИКС П.

Структуру определенных классов ИКС П, имеющих одинаковое функциональное назначение, принято называть обобщенной. Элементы этого

класса являются аналогами и прототипами реализуемого проекта. Обобщенная структура представляет собой «комбинаторное пространство», в котором находятся различные сочетания элементов, образующие структуры разрабатываемого технического объекта. В качестве средств описания обобщенных структур используются табличные, алгебраические, логические и сетевые модели. Классификация методов комбинаторно-логического синтеза представлена на рис. 1.1.

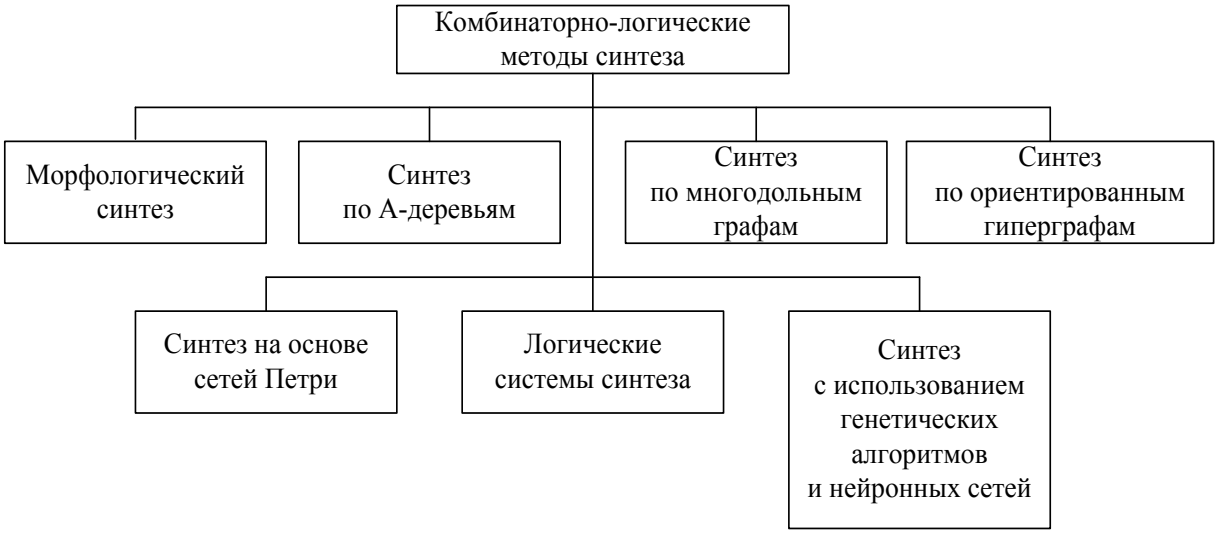

Рис. 1.1. Классификация методов комбинаторно-логического синтеза

Наибольшее распространение среди этих моделей получили морфологические таблицы и А-деревья (И–ИЛИ-деревья – в исследованиях по искусственному интеллекту и в программировании).

Морфологические таблицы применяются на ранних стадиях проектирования и конструирования: при решении изобретательских задач, для обликового проектирования некоторых ИКС П и т. п. Таблицы позволяют получать составы (структуры без связей), структуры с линейными связями и регулярные структуры, причем все решения состоят из равного количества элементов.

Под обобщенной структурой понимается структура определенного класса ИКС П, имеющих одинаковое функциональное назначение. Обобщенная структура составляется на основе знаний данной предметной области, обобщения инженерного опыта и традиций, анализа аналогов и прототипов. Она содержит информацию об альтернативных вариантах устройства объектов данного класса.

Существует несколько модификаций метода синтеза на основе обобщенной структуры. Методы синтеза отличаются друг от друга способом описания обобщенной структуры, методами поиска решений, средствами описания запрещений на сочетание различных альтернатив и т. д.

Разновидности метода синтеза на основе обобщенной структуры, которые получили широкое распространение в практике проектирования, рассмотрены в [6].

1.2.1. *Метод морфологического синтеза структуры ИКС П*

Метод морфологического синтеза структуры ИКС П применяется на ранних стадиях проектирования и конструирования; он позволяет найти и систематизировать все возможные способы построения системы, имеющие данное функциональное назначение. Средством, описывающим обобщенную структуру класса, служит морфологическая таблица. Это простой объект с хорошо отработанными правилами заполнения и поиска решений. Алгоритмы работы с морфологическими таблицами подробно рассмотрены в [21].

Любую морфологическую таблицу можно задать в виде матрицы инцидентности. Матрица инцидентности представляет собой прямоугольную (0,1)-матрицу $A = \| a_{ij} \|$ с размерами $L \times E$, где L – число морфологических классов; E – число элементов системы (реализация). Элемент матрицы a_{ij}, стоящий на пересечении i–j-й строки и j-го столбца, равен 1, если реализация j входит в морфологический класс с номером I, 0 – в противном случае.

Для того чтобы по матрице инцидентности A синтезировать вариант ИКС П, необходимо выбрать совокупность из L (по числу морфологических классов) элементов матрицы, для которой выполняются условия:

– все элементы матрицы инцидентности, принадлежащие совокупности, равны единице;

– никакие два элемента не лежат на одной линии матрицы (линией матрицы называется ее строка либо столбец);

– выбранные элементы покрывают все строки матрицы инцидентности.

Число элементов в совокупности равняется числу строк матрицы инцидентности. Если все элементы совокупности упорядочить по возрастанию номеров строк, которые покрываются этими элементами, и вместо каждого элемента записать номер столбца, который ими покрывается, то получится **L**-вектор.

Этот вектор представляет собой трансверсаль семейства ($a_{(1)}, a_{(2)}$, …, $a_{(L)}$), где все $a_{(i)}$ представляют собой морфологические классы.

На ранних этапах проектирования для описания обобщенных структур часто используется математический аппарат многодольных графов (N-дольных). Это граф вида $G = \left(X_1, X_2, ..., X_n, R\right)$, где $X = UX_i$ – множество вершин, а R – множество ребер. Подмножество X_i принято называть долями графа, ребра графа могут соединять только вершины разных долей. Вершины, принадлежащие одной доле, всегда являются независимым подмножеством.

При описании обобщенных структур систем и процессов элементы N-дольных графов получают следующее толкование:

– доли представляют основные технические подфункции класса;

– вершины каждой доли соответствуют техническим реализациям.

Такая интерпретация делает этот аппарат очень близким морфологическим таблицам по способу решения задачи структурного синтеза и по возможностям описания структурных связей между объектами.

1.2.2. *Структурный синтез по альтернативным деревьям*

Альтернативным деревом (А-деревом) называется вектор (**G**, **S**, γ), в котором: **G** = G(**X**, **D**) – корневое ориентированное дерево со множеством вершин $\mathbf{X}=\{x_i\}$ и множеством дуг $\mathbf{D}=\{d_j\}$; $\mathbf{S}=\{S_k\}$ – множество названий (номеров) связок; отображение γ: $S \to 2^{\mathbf{D}}$ ставит в соответствие каждому имени связки s, принадлежащей **S**, подмножество дуг γ(s) из множества **D**, причем все дуги одной связки имеют общую начальную вершину.

Объекты этого типа нашли широкое применение в программировании и в исследованиях по искусственному интеллекту. В этих отраслях знаний их принято называть И–ИЛИ-деревьями.

Рассмотрим правила соответствия между элементами деревьев и обобщенными структурами:

– корень дерева соответствует основной технической функции класса объектов;

– висячие вершины (листья) представляют технические реализации;

– связки дерева описывают способы разбиения технических функций на подфункции.

Альтернативное дерево составляется для конкретного класса систем, изделий, процессов и описывает обобщенную структуру этого класса. Вершинам дерева могут соответствовать функции, системы, подсистемы, элементы, которые реализуют эти функции, а также признаки функций и систем. Если у вершины альтернативного дерева есть потомки, это означает, что объект, который вершина описывает, имеет структуру. Например, функция сводится к набору подфункций, система – к совокупности подсистем и элементов, признак является агрегированным и выражается через композицию более простых признаков. Концевые вершины (листья) альтернативного дерева соответствуют элементарным, бесструктурным в данном классе объектам: функциям, элементам, признакам.

Для того чтобы получить по А-дереву решение задачи синтеза, нужно выполнить следующую последовательность шагов.

– выбрать одну из связок, исходящих из корневой вершины;

– для каждого потомка корневой вершины выбрать по одной исходящей связке;

– продолжать этот процесс до тех пор, пока каждая из полученных вершин-потомков не станет концевой вершиной в альтернативном дереве.

Подграф, который получается в результате такого процесса, называется графом решения (деревом решения, решающим графом), а концевые вершины, принадлежащие решающему графу, представляют собой решение задачи синтеза.

Можно дать точное рекурсивное определение графа решения. Обозначим через x корневую вершину альтернативного дерева G, через $\mathbf{X}_n$ – множество всех концевых вершин, через G_r – граф решений, связывающий x с множеством $\mathbf{X}_n$.

1. Если x – элемент $\mathbf{X}_n$, то G_r состоит из единственной вершины x.

2. Если x не принадлежит $\mathbf{X}_n$, и если из x исходит связка **K**, направленная на вершины $\{x_1, x_2, ..., x_k\}$, причем существует граф решения, соединяющий вершины из $\mathbf{X}_n$ с каждой вершиной x_i, i = l, k, то G_r состоит из вершины x, связки **K**, вершин $\{x_1, x_2, ..., x_k\}$ и графов решения, идущих в $\mathbf{X}_n$ из каждой вершины множества $\{x_1, x_2, ..., x_k\}$.

3. Если не выполнены условия 1 и 2, то графа решения, который бы связывал x с $\mathbf{X}_n$, не существует.

Данное определение корректно, так как в А-деревьях нет циклов, поэтому нет вершин, имеющих таких потомков, которые были бы их предками. Это гарантирует конечность и результативность описанной рекурсивной процедуры для А-деревьев.

1.2.3. *Синтез проектных решений с учетом дополнительных условий*

Все рассмотренные модели структурного синтеза ИКС П являются композиционными. Это значит, что они описывают только разрешенные сочетания элементов, полученные по правилам композиции данной обобщенной структуры. Среди всех решений задачи синтеза, которые определяются по рассмотренным моделям, могут содержаться нежизнеспособные, нереализуемые варианты.

Число различных параметров, характеристик, аспектов, режимов и т. п., которые описывают свойства реальных ИКС П, чрезвычайно велико. Только ограниченное их количество может быть учтено и описано при моделировании.

Возможность и необходимость объединения элементов в решении зависит не только от функциональной структуры класса ИКС П, которая успешно описывается формальными средствами дискретной математики. Во многих проектных ситуациях на выбор структур накладываются дополнительные условия и ограничения, не представленные на языке обобщенной структуры. Эти условия могут отражать как объективные связи различной физической природы (например, требования технологии, эргономики, технического дизайна и проч.), так и субъективные предпочтения экспертов.

Задача разработки моделей структурного синтеза, учитывающих дополнительную информацию о правилах объединения элементов в составе технической системы или процесса, является важной и актуальной.

Рассмотрим возможные способы формализации таких условий. Все дополнительные условия, наложенные на элементы обобщенных структур, можно разделить на две группы: бинарные и многоместные. Первые действуют на парах элементов, вторые могут регламентировать выбор подмножеств объектов любой мощности.

Анализ различных классов ИС показывает, что многоместные дополнительные условия композиции встречаются достаточно редко. Большая часть таких условий, имеющих прикладное значение, может быть сведена к сочетанию бинарных отношений и правил. Далее будем рассматривать только такие условия.

Если абстрагироваться от физической природы причин, накладывающих дополнительные ограничения на выбор альтернатив, то на уровне принятия решений все объективные характеристики и субъективные соображения могут рассматриваться как условия выбора (предусловия) и последствия выбора (постусловия) структурных элементов.

Для этих целей можно применить язык теории графов и аппарат математической логики.

1.2.4. *Структурный синтез с ограничениями на основе N-дольных графов*

N-дольным называется граф вида **G** = (**X**, **R**), где множество вершин **X** разбито на совокупность непересекающихся подмножеств $X_1 \cup X_2 \cup ... \cup X_N$, а из существования ребра $r = (z, y)$ следует, что инцидентные ребру вершины принадлежат разным подмножествам, т. е. $z \in \mathbf{Z}$; $y \in \mathbf{Z}$; $\mathbf{Z}, \mathbf{Y}, \mathbf{X}, \mathbf{Y} \subseteq \mathbf{Z}$ и $\mathbf{Z} \neq \mathbf{Y}$. Подмножества вида $\mathbf{X}_i$ принято называть долями графа. Из определения следует, что все доли являются независимыми множествами вершин.

Средствами *N*-дольных графов можно описывать обобщенные структуры с не очень сложной системой разбиения функции на подфункции.

Техническим подфункциям соответствуют доли многодольного графа. Технические реализации подфункций представляются в виде вершин. Две вершины связываются ребром тогда и только тогда, когда не существует запретов на сочетание соответствующих реализаций в составе одного технического объекта или процесса.

Аппарат многодольных графов не только позволяет описывать обобщенные структуры простыми функциональными структурами – с его помощью можно описывать решения задачи структурного синтеза, учитывающего бинарные запреты на сочетания. Любой полный *N*-вершинный подграф многодольного графа является решением поставленной задачи.

Требование разложимости графа на несколько долей не ограничивает общность постановки задачи. Анализ показывает, что любой граф можно разложить на совокупность долей, вершины которых не связаны друг с другом. Такое представление возможно даже для самых плотных типов графов – полных графов с произвольным числом вершин. Это значит, что

для поиска структурных решений по многодольным графам не требуется изобретать никаких специализированных алгоритмов. Для этих целей можно воспользоваться стандартными схемами, которые предложены для графов общего вида. В работах по теории графов показывается, что задача поиска полного подграфа с фиксированным числом вершин может быть решена несколькими разными способами: как напрямую, так и при помощи сведения к графовым конфигурациям другого вида. Так, хорошо изучен метод, когда исходная задача сводится к поиску независимого множества в дополнительном графе, который, в свою очередь, формулируется как задача покрытия транспонированной матрицы инцидентности.

Принадлежность задачи к классу *NP*-трудных задач высокой размерности означает, что в настоящее время не известен эффективный способ решения, имеющий полиномиальную сложность. Вопрос о принципиальной возможности существования быстрого алгоритма остается открытым.

Отсутствие эффективного решения для задачи в общей постановке не означает, что такое решение не может быть получено для ее частных случаев или ограничений. Высокие критерии точности и строгости, которые являются обязательными для математической постановки задачи, часто удается смягчить без ущерба для реальной проектной ситуации. Например, в оптимизационной по постановке задаче можно ограничиться поиском субоптимального решения.

Очевидным недостатком рассмотренного аппарата является его громоздкость. Обобщенные структуры, описывающие реальные классы объектов, могут содержать несколько десятков долей и сотни вершин в каждой из них. Любые традиционные способы кодирования таких громоздких графов (матрицы смежности, инцидентности, списки смежности и проч.) будут, очевидно, неэффективными. Задача компактного представления многодольных графов высокой размерности имеет простое и эффективное решение. Достаточно вместо самого графа хранить данные о его дополнении.

Более формально, дополнением графа $\mathbf{G} = G(\mathbf{X}, \mathbf{R})$ называется граф $\mathbf{L}(G) = L(\mathbf{X}, \mathbf{D})$, причем ребро $r = (x, y)$ существует в $\mathbf{G}$ тогда и только тогда, когда аналогичные вершины не связаны ребром $d = (x, y)$ в графе $\mathbf{L}$. Очевидно, что объединение любого графа и его дополнения дает в итоге полный граф на том же множестве вершин, который содержит $n(n - 1)/2$ ребер.

Графы, описывающие реальные классы технических объектов, обладают очень высокой размерностью. Для сокращения объема описания целесообразно хранить данные о дополнении многодольного графа, которое в общем случае имеет намного меньше ребер, чем исходный граф.

1.2.5. *Генетико-морфологический метод*

Решаемая на этапе технического предложения задача выбора компоновки ИКС П, является основной задачей проектирования, оказывающей влияние на технико-экономические показатели ИКС П (объем и скорость передаваемой информации, количество возможных ошибок, надежность и т. д.).

Сложностью данной задачи является выбор оптимального варианта построения ИКС П, поскольку для ее решения требуются большие затраты времени и труда проектировщика.

При традиционном проектировании практически невозможно с достаточной полнотой и точностью рассчитать, оценить и сопоставить всевозможные варианты структур построения ИКС П, соответствующих конкретным условиям производства, поскольку оно, как правило, осуществляется эмпирически, основываясь при этом в основном на опыте и интуиции наиболее квалифицированных проектировщиков. Однако применение интуитивного и/или опытного подхода не гарантирует определение оптимальной компоновки для решения поставленной задачи. Это происходит из-за сложности формализации задачи выбора структур построения ИКС П, охватывающей достаточно большое множество возможных структур. Решить этот вопрос можно путем применения современных информационных технологий автоматизации, в том числе искусственных нейронных сетей, позволяющей освободить проектировщика от необходимости выполнения большого количества рутинных работ.

На каждом уровне подсистем возможны различные схемные и конструктивные решения, создающие отдельные морфологические модели с характерными морфологическими признаками, для каждой из которых имеются альтернативы их реализации.

Для структурного отбора структуры построения ИКС П можно применить: 1) радиальную базисную НС (например, реализованную с помощью программного продукта Matlab (рис. 1.2)); 2) алгоритмы, построенные на базе генетических алгоритмов.

Сеть позволяет сделать наиболее оптимальную оценку и выборку компоновок ИС из совокупного множества компоновок по структурным признакам, сформулированным проектировщиком для решения данной задачи с необходимым для этого качеством.

Радиальные базисные сети PNN (Probabilistic Neural Networks) относятся к вероятностным сетям.

Они наиболее эффективно позволяют решать задачи обобщения и классификации векторов при большом количестве обучающих данных, поскольку имеют большее количество нейронов по сравнению со стандартными сетями с прямой передачей сигналов и методом обратного распространения ошибки.

Главное отличие и преимущество нейросетей перед классическими средствами прогнозирования и классификации заключается в их способности к обучению. В общем случае, обучение – такой выбор параметров сети, при котором сеть лучше всего справляется с поставленной проблемой. Нейронная сеть определяет ответ с помощью примеров, сгруппированных в обучающие множества.

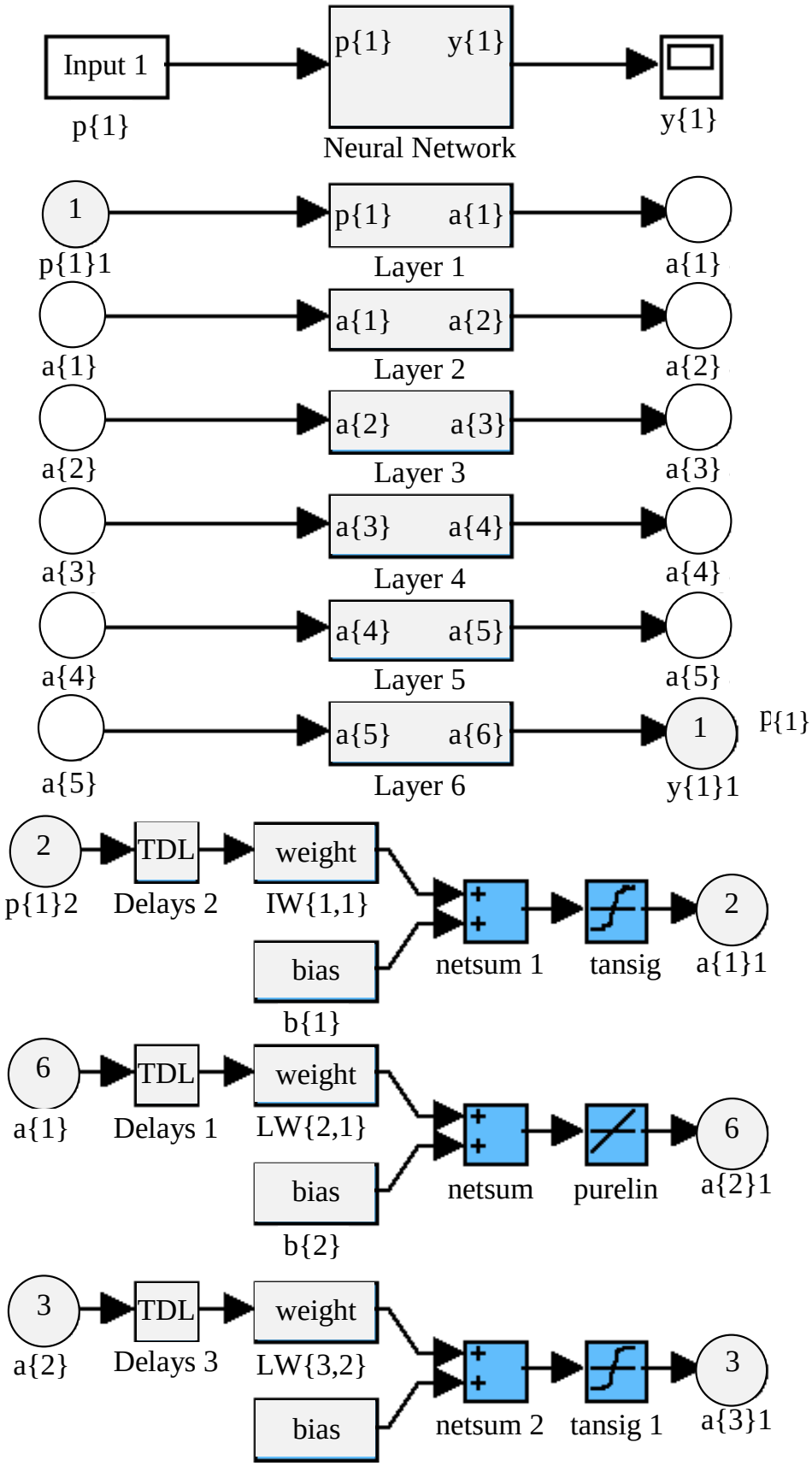

Рис. 1.2. Фрагмент модели элементов нейронной сети
для структурного синтеза структуры построения ИКС П,
построенный с помощью оператора gensim(netn)

Обучающее множество нейронной сети представим в виде матрицы связанности T:

$$T = \begin{bmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,j-1} & k_{1,j} \\ k_{2,1} & \ddots & & & k_{2,j} \\ \vdots & & \ddots & & \vdots \\ k_{i-1,1} & & & \ddots & k_{i-1,j} \\ k_{i,1} & k_{i,2} & \cdots & k_{i,j-1} & k_{i,j} \\ c_1 & c_2 & \cdots & c_{j-1} & c_j \end{bmatrix},$$

которая, формируется из i-го количества векторов входа (входные данные обучающего алгоритма):

$$p_i = \begin{bmatrix} k_{i,1} & k_{i,2} & \ldots & k_{i,j-1} & k_{i,j} \end{bmatrix},$$

и вектора цели T_C (выходные данные обучающего алгоритма):

$$T_c = \begin{bmatrix} c_1 & c_2 & \ldots & c_{i-1} & c_i \end{bmatrix}.$$

В итоге программа выдаст одну или несколько оптимальных компоновок.

Почти все соображения, положенные в основу отбора структур построения ИКС П, так или иначе связаны со структурой последних. С другой стороны, обобщенные структурные формулы, подобные координатному коду, записанному ранее, являются аппаратом достаточно универсальным и гибким, способным отражать различные оттенки структуры компоновок.

Таким образом, применение НС дает возможность пользователю автоматизировать и ускорить процесс отбора оптимальных структур ИКС П, определив совокупное множество вариантов структур для конкретных условий работы, осуществив структурный отбор подмножеств, удовлетворяющих формализованным структурным признакам отбора, и подготовить данные для окончательного выбора компоновок путем сравнения их технико-экономических показателей. Пользователь имеет возможность самостоятельно выбирать структурные признаки, подходящие его условиям, и самостоятельно формализовывать их. Для работы с данным алгоритмом пользователю требуется знание только основной функции формообразования для правильного выбора структурных условий. При верном подборе пользователем структурных признаков и их формализации, программа выдает результат, который полностью совпадает с результатом, полученным вручную или с использованием программ с четкой логикой.

Возможно усложнение программы и алгоритма обобщением структурных правил для всех основных типов РП ИКС П, что позволит существенно сократить время подготовки программы к работе.

1.3. Применение генетических алгоритмов для структурного синтеза ИКС П

Основные принципы генетических алгоритмов (ГА) были сформулированы Голландом (Holland) в 1975 г. и хорошо описаны во многих работах [2, 11, 12], поэтому в данной монографии теорию построения ГА подробно рассматривать не будем, остановимся на рассмотрении вопросов, связанных с применением для структурно-параметрического синтеза. В отличие от эволюции, происходящей в природе, ГА только моделируют те процессы в популяциях, которые являются существенными для развития.

В общем виде генетический алгоритм может быть представлен в виде схемы, приведенной на рис. 1.3.

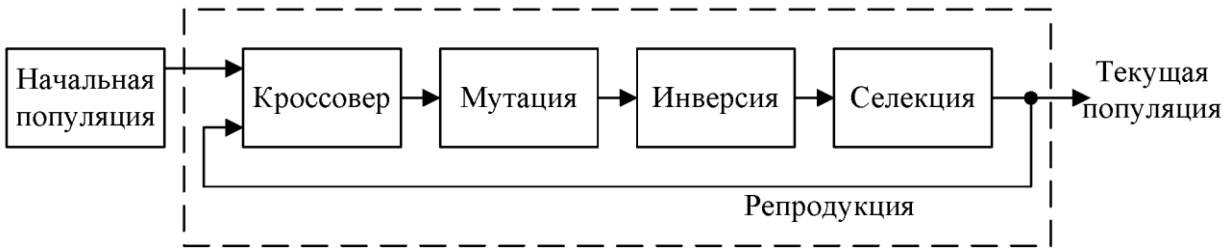

Рис. 1.3. Схема генетического алгоритма

Генетические алгоритмы – это адаптивные методы поиска, которые в последнее время часто используются для решения задач оптимизации (поиска оптимального решения). Они основаны на генетических процессах биологических организмов: биологические популяции развиваются в течение нескольких поколений, подчиняясь законам естественного отбора. Подражая этому процессу, генетические алгоритмы способны «развивать» решения реальных задач, если те соответствующим образом закодированы. Например, ГА могут использоваться, чтобы проектировать структуры ИКС П, для поиска оптимального отношения скорость передачи/объем информации, или определять структуры ИКС П, сочетающие хорошие динамические характеристики с минимальными экономическими затратами.

ГА работают с совокупностью особей, каждая из которых представляет возможное решение поставленной задачи структурно-параметрического синтеза.

Поясним некоторые термины, которые встречаются в ГА.

Хромосома – это вектор (последовательность) из нулей и единиц. Каждая позиция (бит) называется геном. *Кроссовер* – это операция, при которой две хромосомы обмениваются своими частями. *Мутация* – это случайное изменение одной или нескольких позиций в хромосоме. *Приспособленность индивидуума* – это значение целевой функции на этом индивидууме. *Выживание наиболее приспособленных* – это популяция следующего поколения, которая формируется в соответствии с целевой функцией. Чем приспособленнее индивидуум, тем больше вероятность его участия в кроссовере, т. е. размножении.

На первом шаге алгоритма формируется исходная популяция особей, где каждая особь представляет собой решение задачи, иначе говоря, является кандидатом на решение. Формирование исходной популяции, как правило, происходит с использованием какого-нибудь случайного закона, в ряде случаев исходная популяция может быть также результатом работы другого алгоритма.

Необходимо отметить, что особи популяции могут состоять как из одной, так и из нескольких хромосом. Каждая хромосома особи, в свою очередь, состоит из генов, причем их количество в хромосоме определяется количеством варьируемых параметров решаемой задачи (аргументов целевой функции).

Для применения генетических операторов значения этих параметров должны быть представлены в виде двоичной последовательности, т. е. двоичной строки, состоящей из нескольких бит. Количество бит при кодировании гена (хромосомы) зависит от требуемой точности решаемой задачи. Оптимальным считается такой выбор, когда выполняется соотношение

$$n \geq \log_2 \frac{x_{\max} - x_{\min}}{\Delta},$$

где n – количество бит, используемых для кодирования значения аргумента; $x_{\max}$ и $x_{\min}$ – максимальное и минимальное значения аргумента x целевой функции; Δ – погрешность решения задачи.

Количество особей M в популяции определяется, как правило, эмпирическим путем, желательно из интервала $n \leq M \leq 2 \cdot n$.

Кодом, применяемым при реализации ГА, обычно является код Грея. Значения кодов Грея рассмотрены в табл. 1.1.

Таблица 1.1

Кодирование			Кодирование по коду Грея		
десятичное	двоичное	шестнадцатеричное	десятичное	двоичное	шестнадцатеричное
0	000	0h	0	0000	0h
1	0001	1h	1	0001	1h
2	0010	2h	3	0011	3h
3	0011	3h	2	0010	2h
4	0100	4h	6	0110	6h
5	0101	5h	7	0111	7h
6	0110	6h	5	0101	5h
7	0111	7h	4	0100	4h
8	1000	8h	12	1100	Ch
9	1001	9h	13	1101	Dh
10	1010	Ah	15	1111	Fh
11	1011	Bh	14	1110	Eh
12	1100	Ch	10	1010	Ah
13	1101	Dh	11	1011	Bh
14	1110	Eh	9	1001	9h
15	1111	Fh	8	1000	8h

На втором шаге работы генетического алгоритма происходит отбор, или селекция, наиболее приспособленных особей, имеющих более предпочтительные значения функции пригодности по сравнению с остальными особями. После чего к отобранным особям применяются операторы скрещивания и мутации.

В классическом генетическом алгоритме отбор наиболее приспособленных особей осуществляется случайным образом с помощью различных методов генерации дискретных случайных величин, имеющих различные законы распределения. Среди данных методов следует упомянуть пропорциональный отбор, отбор с вытеснением, равновероятный отбор, и т. д. Каждый из перечисленных методов имеет как свои достоинства, так и недостатки, например, общим недостатком указанных методов является то, что при их использовании в некотором поколении наилучшие особи популяции могут быть потеряны. Одним из способов преодоления этого недостатка является использование сочетания алгоритмов отбора.

На третьем шаге реализуется операция скрещивания особей. Смысл данной процедуры сводится к нахождению новых комбинаций генетического кода хромосом путем обмена случайным образом участков генетического кода у двух особей, прошедших отбор. Это способствует «получению» дополнительного числа новых особей, среди которых могут оказаться как более, так и менее приспособленные особи. Это явление объясняется тем, что точка скрещивания выбирается случайным образом.

На четвертом шаге к особям популяции, полученным в результате скрещивания, применяется операция мутации. С помощью данной операции можно получать принципиально новые генотипы и фенотипы особи, что приводит к еще большему разнообразию особей в рассматриваемой популяции. Суть этого оператора заключается в следующем: в популяции случайным образом выбирается особь и так же случайно выбирается позиция гена, в которой значение изменяется на противоположное ($0 \to 1$ или $1 \to 0$). Внесение таких случайных изменений позволяет существенно расширить пространство поиска приемлемых решений задачи целочисленной оптимизации.

Работа ГА носит итерационный характер, т. е. все указанные шаги применяются многократно и ведут к постепенному изменению исходной популяции в направлении улучшения значения функции пригодности.

В качестве критериев окончания работы ГА принято рассматривать следующие условия: 1) сформировано заданное число поколений; 2) популяция достигла заданного уровня качества (например, 75 % особей имеют одинаковую генетическую структуру или одинаковое значение функции пригодности); 3) достигнут некоторый уровень сходимости, при котором улучшение популяции не происходит.

Имеются много способов реализации идеи биологической эволюции в рамках ГА, которые возможно применить для синтеза, исследования и управления. Традиционным считается ГА, представленный на рис. 1.4.

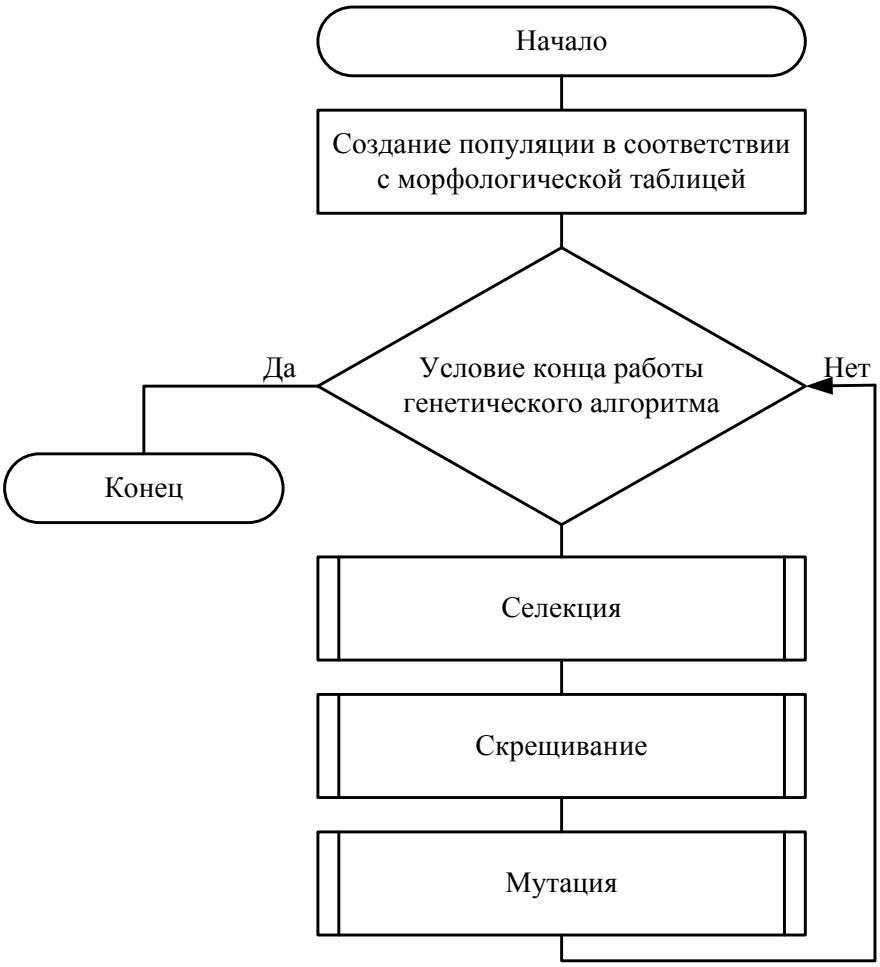

Рис. 1.4. Схема работы ГА

Модель отбора определяет, каким образом следует строить популяцию следующего поколения. Как правило, вероятность участия индивидуума в скрещивании берется пропорционально его приспособленности. Часто используется так называемая *стратегия элитизма*, при которой несколько лучших индивидуумов переходят в следующее поколение без изменений, не участвуя в кроссовере и отборе. В любом случае каждое следующее поколение будет в среднем лучше предыдущего. Когда приспособленность индивидуумов перестает заметно увеличиваться, процесс останавливают и в качестве решения задачи оптимизации берут наилучшего из найденных индивидуумов.

Приведем полное описание работы ГА.

1. Создание структуры решения искомой задачи в виде массива $a[i]$, $i = 1, \ldots, n$, где n – максимальное число компонентов структуры.

2. Создание показателя эффективности структуры, заполненной конкретными значениями.

3. Задание некоторого массива различных структур S_k, $k = 1, ..., N$, размерностью N, большей, чем число компонентов n в структуре. Данный массив можно сгенерировать случайно, задав нули и единицы в каждой структуре.

4. Расчет показателей эффективности J_k для каждой структуры S_k.

5. Естественный отбор структур по некоторому правилу выбора наилучших среди заданного массива структур. Пример: можно по правилу вида $J_0 = M(J_k)$ – найти среднее значение J_k, если $J_k < J_0$, то структура остается, иначе умирает. Если разница между J_0 предыдущим и новым меньше какого-то малого числа, то конец расчета.

6. Замена выбывших структур на новые, рожденные от наиболее приспособленных структур с помощью генетических операторов:

а) мутация – замена в структуре одного из значений случайно выбранного компонента. Пример: из (1, 1, 0, 1, 0, 0, 1, 0) получится (1, 1, 0, 1, 1, 0, 1, 0);

б) инверсия – перестановка в структуре некоторой ее части наоборот. Пример: из (1, 1, 0, 1, 0, 0, 1, 0) получится (1, 1, 0, 0, 1, 0, 1, 0);

в) кроссинговер – создание структуры, основанной на двух структурах – заменой одной части первой структуры на ту же область во второй. Пример: из (*A, B, C, D, E*) и (*a, b, c, d, e*) получится (*A, B, c, d, E*).

7. Переход к этапу 4.

Влияние параметров генетического алгоритма на эффективность поиска оптимальной структуры ИС. Стандартная схема ГА предполагает ограничение численности потомков путем использования так называемой вероятности кроссовера. Такая модель придает величине, соответствующей численности потомков, недетерминированный характер. Есть метод предлагающий отойти от вероятности кроссовера и использовать фиксированное число хромосомных пар на каждом поколении, при этом каждая хромосомная пара «дает» двух потомков. Такой подход хорош тем, что делает процесс поиска более управляемым и предсказуемым в смысле вычислительных затрат.

В качестве генетических операторов получения новых генотипов «потомков», используя генетическую информацию хромосомных наборов родителей, применяют два типа кроссоверов: одно- и двухточечный. Вычислительные эксперименты [11] показали, что даже для простых функций нельзя говорить о преимуществе того или иного оператора. Более того, было показано, что использование механизма случайного выбора одно- или двухточечного кроссовера для каждой конкретной брачной пары подчас оказывается более эффективным, чем детерминированный подход к выбору кроссоверов, поскольку достаточно трудно определить, который из двух операторов более подходит для каждого конкретного ландшафта приспособленности. Использование случайного выбора позволяет сгладить различия этих двух подходов и улучшить показатели среднего ожидаемого результата. Проведенный вычислительный эксперимент показал, что случайный выбор оказался эффективнее.

Аналогичный подход применяется и при реализации процесса мутации новых особей, однако в этом случае преимущество перед детерминированным подходом не так очевидно в силу традиционно малой вероятности мутации. В основном вероятность мутации составляет 0,001–0,01.

Выбор родительской пары. При выборе родительской пары можно использовать несколько подходов.

1. Случайный выбор родительской пары, когда обе особи, которые составят родительскую пару, случайным образом выбираются из всей популяции, причем любая особь может стать членом нескольких пар. Несмотря на простоту, такой подход универсален для решения различных классов задач. Однако он достаточно критичен к численности популяции, поскольку эффективность алгоритма, реализующего такой подход, снижается с ростом численности популяции.

2. Селективный метод состоит в том, что «родителями» могут стать только те особи, значение приспособленности которых не меньше среднего значения приспособленности по популяции, при равной вероятности таких кандидатов составить брачную пару. Такой подход обеспечивает более быструю сходимость алгоритма. Однако из-за быстрой сходимости селективный выбор родительской пары не подходит тогда, когда ставится задача определения нескольких экстремумов, поскольку для таких задач алгоритм, как правило, быстро сходится к одному из решений. Кроме того, для некоторого класса задач со сложным ландшафтом приспособленности быстрая сходимость может превратиться в преждевременную сходимость к квазиоптимальному решению. Этот недостаток может быть отчасти компенсирован использованием подходящего механизма отбора (о чем будет сказано далее), который бы «тормозил» слишком быструю сходимость алгоритма.

3. Два способа формирования родительской пары – *инбридинг* и *аутбридинг*. Оба эти метода построены на формировании пары на основе близкого и дальнего «родства» соответственно. Под «родством» здесь понимается расстояние между членами популяции, в смысле геометрического расстояния особей в пространстве параметров. В связи с этим различают генотипный и фенотипный (или географический) инбридинг и аутбридинг. Под инбридингом понимается такой метод, когда первый член пары выбирается случайно, а вторым с большей вероятностью будет максимально близкая к нему особь. Аутбридинг же, наоборот, формирует брачные пары из максимально далеких особей.

Использование генетических инбридинга и аутбридинга оказалось более эффективным по сравнению с географическим для всех тестовых функций при различных параметрах алгоритма. Наиболее полезно применение обоих представленных методов для многоэкстремальных задач. Однако два этих способа по-разному влияют на поведение генетического алгоритма. Так, инбридинг можно охарактеризовать свойством концентрации поиска в локальных узлах, что фактически приводит к разбиению популяции на отдельные локальные группы вокруг подозрительных на экстремум участков ландшафта, аутбридинг, напротив, направлен как раз на предупреждение сходимости алгоритма к уже найденным решениям, заставляя алгоритм просматривать новые, неисследованные области.

Механизм отбора. Наиболее эффективными являются два механизма отбора: элитный и отбор вытеснением.

Элитный отбор основан на построении новой популяции только из лучших особей репродукционной группы, объединяющей в себе родителей, их потомков и мутантов. В основном это объясняют потенциальной опасностью преждевременной сходимости, отдавая предпочтение пропорциональному отбору [11]. Быстрая сходимость, обеспечиваемая элитным отбором, может быть, когда это необходимо, с успехом компенсирована подходящим методом выбора родительских пар, например аутбридингом. Комбинация «аутбридинг – элитный отбор» является одной из наиболее эффективных.

Отбор вытеснением. Будет ли особь из репродукционной группы заноситься в популяцию нового поколения, определяется не только величиной ее приспособленности, но и тем, есть ли уже в формируемой популяции следующего поколения особь с аналогичным хромосомным набором. Из всех особей с одинаковыми генотипами предпочтение сначала отдается тем, чья приспособленность выше. При этом достигаются две цели: 1) не теряются лучшие найденные решения, обладающие различными хромосомными наборами; 2) в популяции постоянно поддерживается достаточное генетическое разнообразие. Вытеснение в данном случае формирует новую популяцию скорее из далеко расположенных особей, вместо особей, группирующихся около текущего найденного решения. Этот метод особенно хорошо себя показал при решении многоэкстремальных задач, при этом помимо определения глобальных экстремумов появляется возможность выделить и те локальные максимумы, значения которых близки к глобальным.

Турнирный отбор реализует n турниров, чтобы выбрать n особей. Каждый турнир построен на выборке k элементов из популяции и выборе лучшей особи среди них. Наиболее распространен турнирный отбор с $k = 2$.

Двухточечный кроссовер и *равномерный кроссовер*. В двухточечном кроссинговере выбираются две точки разрыва, и родительские хромосомы обмениваются участком генетического кода, который находится между двумя этими точками. В равномерном кроссинговере каждый бит первого родителя наследуется первым потомком с заданной вероятностью; в противном случае этот бит передается второму потомку, и наоборот.

Размер популяции является весьма важным элементом ГА. Если популяция слишком мала, генетического материала может не хватить для решения задачи. Размер популяции также влияет на коэффициент применения операторов скрещивания и мутации.

1.4. Параметрическая оптимизация структур ИКС П

Процедура постановки задачи оптимизации параметров ИКС П носит неформальный характер и включает следующие этапы: выбор критериев оптимальности, формирование целевой функции, выбор управляемых параметров, назначение ограничений, нормирование управляемых и выходных параметров.

Под оптимизацией понимается процесс поиска наилучшего варианта проектируемой РП ИКС П в условиях множества альтернатив. При проектировании ИКС П необходимо найти их структуру и параметры, обеспечивающие наилучшее сочетание показателей качества и эффективности. При этом возникает проблема формализации понятия «наилучший». Для выбора наилучшего варианта среди определенного множества необходимо сформулировать некоторое правило предпочтения. Основой такого правила может быть однозначная численная характеристика ИКС П, представляющая собой скалярную функцию. Эта характеристика содержательно отображает цель поиска, в связи с чем ее называют целевой функцией. Она позволяет количественно выразить качество объекта и поэтому называется также функцией качества. Таким образом, в основе построения правила предпочтения лежит целевая функция.

В качестве критериев оптимальности, принимаются выходные параметры ИКС П. При этом часто выбирают не все выходные параметры ИКС П, а только те, которые оказывают наибольшее влияние на достижение конечной цели проектирования всей ИКС. Остальные выходные параметры используются при формировании функций ограничений.

На основе выбранных критериев осуществляется формирование целевой функции.

Всесторонняя оценка эффективности и качества объекта возможна при использовании множества критериев. Задача оптимизации в этом случае становится многокритериальной.

Векторный характер критериев оптимальности создает проблему формирования целевой функции. Сложность ее обусловлена не только большим количеством используемых критериев, но и их характером. Обычно улучшение одного из критериев приводит к ухудшению других критериев. Такие критерии называются конфликтными. Если определить оптимальные параметры объекта по каждому из критериев в отдельности, то они окажутся различными. Поэтому при наличии векторного критерия возможно лишь некоторое компромиссное решение, которое в наибольшей мере отвечает задачам проектирования.

В зависимости от типа или класса РП ИКС П критерии могут быть детерминированные или статистические.

Если оптимизация параметров ИКС П осуществляется на основе детерминированной математической модели, то используемые критерии оптимальности (выходные параметры ИКС П) будут также детерминированными. Если же для оптимизации параметров используется вероятностная математическая модель ИКС П, которая учитывает случайные воздействия внешней среды и (или) случайный разброс параметров, то значения критериев представляют собой статистические оценки выходных параметров объекта, т. е. критерии оказываются статистическими. Значения критериев

на каждом шаге поиска при этом представляют собой случайные числа, так как они обусловлены не только изменением управляемых параметров, но и случайными факторами. Поиск решения в этом случае обеспечивается вероятностными итеративными алгоритмами. Эти алгоритмы можно разделить на три группы: использующие детерминированный поиск; использующие случайный поиск; комбинированного типа, использующие детерминированный и случайный поиск.

Методика оптимизации параметров ИКС П. Задача параметрической оптимизации ИКС П заключается в поиске параметров, при которых целевая функция достигает экстремального значения.

В достаточно общем виде математическую задачу оптимизации можно сформулировать следующим образом: минимизировать (максимизировать) целевую функцию с учетом ограничений на управляемые переменные.

При записи математических задач оптимизации в общем виде обычно используется следующая символика:

$$Q(\alpha) \xrightarrow[\alpha \in U]{} \min(\max),$$

где $Q(\alpha)$ – целевая функция; U – допустимое множество, заданное ограничениями на управляемые переменные.

Часто для большей эффективности работы алгоритмов оптимизации в целевую функцию (критерий качества) вводят штрафную или барьерную функцию. Штрафная функция равна 0, если $\alpha \in U$, и быстро возрастает при удалении точки $\alpha \in U$ от допустимого множества.

Разработку алгоритмов управления ИКС П выполняют, как правило, с учетом двух важнейших оценок качества: быстродействия и связанной с ним производительности.

Методы формального синтеза алгоритмов управления ИКС П и сложности, возникающие при этом, достаточно хорошо освещены в литературе. Синтез оптимальных алгоритмов управления локальными и взаимосвязанными системами может производиться в соответствии с оптимизирующими функционалами вида [11].

Особенностями задачи оптимизации параметров ИКС П являются следующие: 1) исходные модели ИКС П представляются в виде сложных структурных схем со значительным числом блоков, операторы которых принадлежат разным классам; 2) оптимизация по различным параметрам (скорость передачи информации, объем передаваемой информации, количество ошибок и др.) предполагает гибкое формирование различных функционалов качества; 3) оптимизируемые переходные процессы в ряде случаев имеют специфический, нетиповой вид, автоматическая оценка которых общепринятыми показателями качества представляет собой некоторую проблему при оптимизации по прямым показателям качества.

В случае, когда при проектировании РП ИКС П производится декомпозиция (горизонтальная или вертикальная), необходимо говорить о ло-

кальной оптимизации каждого элемента ИКС П. Сущность локальной оптимизации заключается в том, что на каждом уровне декомпозиции применяются свои критерии оптимальности и осуществляется оптимизация лишь некоторой части параметров РП ИКС П, относящихся к внутренним параметрам проектируемого элемента ИКС П. В результате оказывается, что при проектировании любого элемента ИКС П используемые критерии получены как результат оптимизации параметров более крупного элемента ИКС П. По существу декомпозиция РП ИКС П приводит к декомпозиции критериев. Следовательно, критерии на всех уровнях декомпозиции РП ИКС П взаимосвязаны и подчинены конечной цели – достижению высоких показателей эффективности и качества функционирования РП ИКС П.

Проектирование элементов ИКС П, выделяемых при декомпозиции РП ИКС П, основано на моделировании некоторого характерного режима их функционирования. Проектные задачи различаются также видами зависимостей критериев от оптимизируемых параметров, количеством используемых критериев и способом формирования целевой функции.

Многообразие исследовательских и проектных задач привело к разработке множества методов оптимизации, обладающих различными свойствами и возможностями поиска экстремума целевой функции с учетом особенностей математических моделей ИКС П. Проектировщик должен хорошо знать особенности решаемой задачи и свойства существующих методов оптимизации, предвидеть характер изменения целевой функции, что позволит осуществить обоснованный выбор метода и повысить вероятность решения задачи с минимальными затратами.

1.5. Оптимизация работы блочно модульной системы обработки данных с использованием генетических алгоритмов

Процедуры обслуживания заявок в системах обработки данных реального времени (СОД РВ) зависят от времени поступлении заявки на обработку, состава и взаимосвязей необходимых для ее обслуживания задач и от текущего состояния информационной базы, определяющего альтернативные возможности обработки данных.

Использование совокупности взаимосвязанных матричных и графовых моделей для анализа информационных потоков и технологии обработки данных в СОД РВ [3], [16], [17] в настоящее время из-за возросшего потока данных в инфокоммуникационных системах не позволяет оперативно их обрабатывать и управлять ими.

Поэтому предлагается строить алгоритм обработки информационных потоков данных в СОД РВ с использованием в начале традиционного подхода (использование матриц и интегрированных информационных графов (ИИГ)), а на заключительном этапе – генетических алгоритмов (ГА).

Использование традиционных подходов [3], [16], [17] позволит сформировать полную и качественную информационно-управляющую структуру данных инфокоммуникационной системы. Применение же генетических алгоритмов позволит более оптимально построить процесс обработки информационных потоков данных в СОД РВ и управление передачей данных между абонентами инфокоммуникационной системы.

Обобщенной формой представления взаимосвязей информационных элементов, процедур и информационных элементов при решении задач являются технологические матрицы сложности и достижимости, которые затем преобразуются в ИИГ обработки данных.

Построение и структуризация ИИГ для решения задач обработки данных реального времени осуществляется следующим образом.

Пусть задано множество задач СОД РВ $\{\eta_n\}$, $n=\overline{1,N}$. Технологии решения каждой задачи соответствует направленный ИИГ $G^n=(D^n,R^n)$, где $D^n=\{d_l^n\}$; $l=\overline{1,L^n}$; L^n – множество вершин ИИГ, отражающих информационные элементы задачи η_n; $R_n=\{p_r\}$ – множество отношений между информационными элементами $d_l^n \in D^n$. Каждому ИИГ G^n соответствует квадратная бинарная матрица смежности $C_n=\left\|C_u^n\right\|$ размера $L^n \times L^n$. Элемент $\left\|C_u^n\right\|$ матрицы C_n равен 1, если элементы d_l^n и $d_{l'}^n$ графа G^n связаны отношениями p_r, и равен 0 в противном случае.

Структурированный ИИГ взаимосвязей информационных элементов задачи, преобразованный к виду, не содержащему циклов обработки, называется скелетным ИИГ задачи G_c^n. Он состоит из ряда уровней или непересекающихся подмножеств вершин, каждая из которых является выходным результатом обработки предыдущего уровня или подмножества информационных элементов. С использованием ИИГ G_c^n определяется множество процедур обработки данных, необходимых для решения задач $\{\eta_n\}$, $n=\overline{1,N}$. Для каждой упорядоченной пары элементов $(d_l^n,d_{l'}^n)\in D^n$ определим подмножества

$$D_1^n=\{d_l^n \mid \exists d_l^n((d_l^n,d_{l'}^n)\in D^n)\},$$
$$D_2^n=\{d_l^n \mid \exists d_l^n((d_l^n,d_{l'}^n)\in D^n)\},$$
$$D_1^n \cup D_2^n = D^n.$$

Затем определим на множестве D^n декартово произведение $D^n=D_1^n \times D_2^n$. Пара элементов $\left(d_l^n,d_{l'}^n\right)$ связано с процедурой p_r, если она

принадлежит отношению R_n. Совокупность процедур задачи образует множество $\{p_r\} = R^n$. Полное множество процедур анализируемого множества $\{\eta_n\}$ задач определяется путем объединения $R = \bigcup_{n}^{N} R^n = \{p_r\}$.

Для определения в задаче входных, промежуточных и выходных данных, последовательности их получения и контуров обратной связи, а также анализа взаимосвязей в системе введено понятие матрицы достижимости.

Под матрицей достижимости M_n понимается квадратная бинарная матрица, проиндексированная одинаковым образом по обеим осям множеством информационных элементов D_n. Элемент d_l достижим из элемента $d_{l'}(d_l R d_{l'})$, если на графе $G^n = (D^n, R^n)$ можно указать направленный путь от вершины d_l к вершине $d_{l'}$ (либо $l = l'$),

$$m_{ll'} = \begin{cases} 1, \textit{ если для получения элемента } d_{l'} \textit{ используется элемент } d_l, \\ 0, \textit{ в противном случае.} \end{cases}$$

Матрица M_n определяется на основе матрицы C_n. При этом они связаны булевым уравнением

$$M_n = C_m^{m+1} = C_m^m \neq C_m^{m-1}.$$

Анализ структур обработки данных для каждой η_n задачи СОД и определение необходимой последовательности получения информационных элементов упрощается, если элементы построенной матрицы достижимости упорядочить по уровням (этапам) их обработки. Получение матрицы M_n методом свертки циклов позволяет уменьшить ее размерность, облегчить анализ и синтез структуры решения как отдельных задач η_n системы, так и функционирования всей СОД РВ.

Взаимосвязь между процедурами обработки данных при обслуживании каждой заявки СОД РВ, наборами входных и промежуточных данных удобно представлять с помощью таблицы инциденции [3] обработки множеств запросов $Y_n = \left\| y_{rl}^n \right\|$, которая представляет собой матрицу вида

$$y_{rl}^n = \begin{cases} +1, \textit{ если } d_l \textit{ является входным элементом процедуры } p_r \\ \quad \textit{при решении задачи } \eta_m, \\ 0, \textit{ если при решении задачи } \eta_m \textit{ задачи элемент } d_l \textit{ не используется} \\ \quad \textit{процедурой } p_r, \\ -1, \textit{ если } d_l \textit{ является выходным элементом процедуры } p_r \\ \quad \textit{при решении задачи } \eta_m. \end{cases}$$

В матрице Y_n каждая строка отображает процедуру обработки, а каждый столбец – использование всеми процедурами при решении η_n задачи рассматриваемого информационного элемента. В строке содержится информация о множестве входных и выходных данных, связанных с анализируемой процедурой. Анализ столбцов позволяет выявить входные и выходные информационные элементы рассматриваемой задачи η_n. Элементы $d_l \in D^n, l = \overline{1, L^n}$ являются входными при решении η_n задачи, если d_n столбец матрицы Y_n содержит единственную, отличную от нуля запись $y_{rl}^n = +1, r = \overline{1, R^n}$. Если l-й столбец содержит запись $y_{rl}^n = -1$, то соответствующий ему элемент $d_l \in D^n, l = \overline{1, L^n}$ является выходным. Информационной матрицей смежности $C_{\text{и}}^n$ при решении η_n задачи назовем квадратную бинарную матрицу, проиндексированную по обеим осям множествами $D^n \cup R^n$. Матрица $C_{\text{и}}^n$ имеет четыре подматрицы: C^n, C_1^n, C_2^n, C_3^n с размерами $D^n \times D^n$, $R^n \times D^n$ $D^n \times R^n$, $R^n \times R^n$.

Используя матрицу $C_{\text{и}}^n$, можно определить матрицу $M_{\text{и}}^n$, которая содержит подматрицы $M_n, M_n^{(1)}, M_n^{(2)}, M_n^{(3)}$, проиндексированы соответственно: $D^n \times D^n$, $R^n \times D^n$ $D^n \times R^n$, $R^n \times R^n$.

Подматрица $M_n^{(1)}$ удовлетворяет соотношению $M_n^{(1)} = C_1^n * (C^n)^k = C_1^n \text{x} (C^n)^{k+1} \neq C_1^n \text{x} C_l^{k-1}$, где k – целое положительное число, не больше числа L^n элементов при решении η_n задачи, т. е. $k < L^n - 1$. Матрица $M_n^{(1)}$ содержит единичные элементы в позиции (p_r, d_l), если процедура входит в последовательность процедур, необходимую для получения элемента d_l при решении η_n задачи. В противном случае запись в позицию (p_r, d_l) подматрицы $M_n^{(1)}$ равна нулю. Подматрица $M_n^{(2)}$ определяется соотношением $M_n^{(2)} = M_n \times C_2^n$ и содержит единичный элемент в позиции (d_l, p_r), если элемент d_l является входным для последовательности процедур, в состав в которых входит процедура p_r. В противном случае элемент (d_l, p_r) равен 0. Подматрица $M_n^{(3)}$ является матрицей достижимости процедур обработки данных при решении η_n задачи и удовлетворяет соотношению

$$(C_3^n)^{k-1} \neq (C_3^n)^k = (C_3^n)^{k+1} = M_n^{(3)}.$$

Единичная запись в позиции $(p_r, p_{r'})$ подматрицы $M_n^{(3)}$ соответствует наличию направленного пути в ИИГ технологии решения η_n задачи от процедуры p_r к процедуре $p_{r'}$.

Построение единого ИИГ осуществляется путем выполнения операции «наложения» графов G^n и заключается в совмещении идентичных уровней каждого графа и идентичных вершин на каждом уровне. В результате формируется ИИГ G^0, которому соответствует матрица смежности $C^0 = \left\| C^0_{ll'} \right\|$, $l = \overline{1, L}$, $l' = \overline{1, L}$, полученная путем логического сложения матриц $C^n_{\text{и}}$:

$$C^0 = C^1_{\text{и}} \vee C^2_{\text{и}} \vee \ldots \vee C^n_{\text{и}}.$$

Анализ структур полученного ИИГ позволяет на заключительном этапе анализа определить следующие общесистемные требования к обслуживанию заявок в СОД РВ: множество требуемых задач обработки данных для обслуживания одного типа заявок и базовые задачи для каждого типа, взаимосвязи между заявками по решаемым задачам и между задачами по используемым процедурам и данным, рациональную дисциплину обслуживания заявок и оценку требуемой производительности вычислительной системы для заданной дисциплины обслуживания и др.

В качестве моделей описания и анализа задач обработки данных при создании типовых блочно-модульных СОД также используется аналогичная совокупность графовых и матричных моделей.

При формировании полного ИИГ технологии решения задачи обработки информационных потоков данных и управление ими учитывает всю необходимую информацию о взаимосвязях и отношениях между различными элементами отдельных задач.

Полный ИИГ и соответствующие ему матрицы смежности и достижимости позволят, применив далее ГА, более оптимально построить процесс обработки информационных потоков данных в СОД РВ и управление передачей данных между абонентами инфокоммуникационной системы.

ГА работают с совокупностью особей, каждая из которых представляет возможное решение поставленной задачи СОД РВ.

Поясним особенности реализации генетического алгоритма для решения поставленной задачи:

– индивидуум = вариант решения задачи = набор из 10 хромосом X_j;

– хромосома X_j = технический или экономический параметр, влияющий на j-ю структуру = 16-разрядная запись этого числа;

– так как в любой инфокоммуникационной системе имеются ограничения на технические средства так и на информационные потоки данных, то не все значения хромосом являются допустимыми. Это учитывается при генерации популяций.

Далее применяется алгоритм решения поставленной задачи, представленный на рис. 1.4.

1.6. Применение генетических алгоритмов для оптимизации процесса передачи данных в инфокоммуникационных системах

Оптимизация процессов передачи информации в РП ИКС П неразрывно связана с оценкой ее текущего состояния и может быть решена за несколько этапов и различными методами. Для решения этой задачи с помощью ГА наиболее эффективно представление РП ИКС П в виде графа, отражающего оптимальную стуктуру ИКС. Для формирования графа и оценки текущего состояния и оптимизации процессов передачи информации в РП ИКС П необходимо решить следующие задачи:

– провести вертикальную и горизонтальную декомпозиции, необходимые, поскольку РП ИКС П является сложной по архитектуре и по взаимосвязям подсистем;

– определить существенность связей в ИКС;

– определить алгоритм оптимального установления соединения, учитывающий нагрузку канала сети, временные характеристики передаваемой информации, кратчайший путь передачи информации и т. п.

– сформировать граф передачи информации в РП ИКС П;

– разработать алгоритм для оптимизации процесса передачи данных в РП ИКС П с использованием сформированного графа и ГА.

Для решения первой задачи производится вертикальная и горизонтальная декомпозиции РП ИКС П [20]. Специфика декомпозиции заключается в том, чтобы при моделировании любой подсистемы РП ИКС воспроизвести взаимодействие процессов, протекающих во всей сети и влияющих на характеристики исследуемой подсистемы. Это требование реализуется при декомпозиции РП ИКС П на классы направлений обмена информацией. Кластеризация направлений выполняется по признакам пути обмена информацией (ПОИ), по которому проложен основной маршрут. Факторными признаками выступают длина пути (число транзитных участков в ПОИ) и нагрузка, приходящаяся на узлы коммутации (УК) и каналы связи (КС), составляющие рассматриваемый путь. Поскольку значения нагрузки УК и КС путей могут существенно различаться, УК и КС сети ранжируются: канал или узел обладает бóльшим рангом, если на него приходится бóльшая нагрузка. Определение значений нагрузки для каждого элемента сети осуществляется в программе построения маршрутов на основе заданных графа сети и производительности μ_i каналов и узлов. Тогда в соответствии с матрицей нагрузки для каждого элемента сети подсчитываются суммарная интенсивность λ_i входящего потока и значения нагрузки $\rho_i = \lambda_i / \mu_i$. На полученном множестве значений нагрузки узлов и каналов каждому элементу назначается ранг.

Разбиение множества путей $\Pi = \{\pi_1, \ldots, \pi_{Nd}\}$ (N – количество путей; d – количество транзитных участков) проводится с помощью двухэтапной

классификации объектов ИКС. На первом этапе по признаку, характеризующему длину пути, образуются подмножества в соответствии с оптимальными по времени передачи информации по транзитным участкам ИКС. В результате получаются D подмножеств Π_d, $d=\overline{1,\ k_1}$, где k_1 – максимальное число транзитных участков для путей $\pi_i \in \Pi$, $i=\overline{1,\ N}$. Далее, внутри каждого подмножества Π_d выполняется разбиение на классы постепени значимости с использованием процедуры кластерного анализа. При этом элемент сети при наличии большей нагрузки обладает бóльшим значением ранга.

Группирование путей $\pi_j \in \Pi_d$ в непустые непересекающиеся подмножества, именуемые классами, осуществляется по критерию близости объектов. Мера близости представляет собой функцию, ставящую в соответствие каждой паре точек некоторое число S_{ij}, характеризующее степень сходства (близости) между объектами π_i и π_j. Для рассматриваемого случая в качестве меры близости целесообразно принять функцию [20]

$$S_{ij} = \begin{cases} 0,\ S'_{ij} \geq 1; \\ S_{ij} = 1 - S'_{ij},\ S'_{ij} < 1, \end{cases}$$

где $S'_{ij} = \left| \sum_{k=1}^{2d+1} \left(z_{ik} - z_{jk} \right) b \right|$ – расстояние между точками π_i и π_j, причем b – величина, характеризующая различие элементов, обладающих одинаковым рангом. Величина $\left(z_{ik} - z_{jk} \right) b$ выражает различие объектов π_i и π_j по k-му признаку.

В качестве критерия оптимальности при разбиении множества ПОИ используется сумма «внутренних» связей за вычетом некоторого порогового значения, характеризующего их существенность. Оптимальное по заданному критерию разбиение должно удовлетворять требованию: средняя длина связи внутри каждого класса превышает среднюю длину связи объектов этого класса как со всеми остальными классами объектов, так и с отдельными объектами каждого другого класса.

Показателем качества разбиения ПОИ при значении порога существенности a является сумма связей $\left(S_{ij} - a \right)$ между π_i и π_j внутри разбиения R:

$$f\left(a,\ R\right) = \sum_{h=1}^{L} \sum_{i,\ j \subset R_h} \left(S_{ij} - a \right), \qquad (1)$$

Здесь сумма $\sum_{i,\ j \subset R_h} \left(S_{ij} - a \right)$ характеризует связи внутри класса R_h. Оптимальным будет разбиение, максимизирующее $f(a,\ R)$ при заданном a по всем возможным разбиениям данного множества объектов.

К достоинствам показателя (1) относится также возможность выбирать значение порога существенности связей на основе содержательного смысла конкретной задачи. При различных значениях порога оптимальны, вообще говоря, разные разбиения. Формирование подмножеств оптимальных путей прохождения информации в зависимости от уровня существенности связи определяет, в конечном счете, точность оценки передачи информации в РП ИКС П в целом.

Рассмотренный алгоритм ранжирования декомпозиции РП ИКС П не позволяет полностью оптимальным образом определить путь прохождения информации в ИКС. Для этого необходимо рассмотреть также существенность связей в ИКС и провести оценку состояния РП ИКС П.

Определение порога существенности связей в инфокоммуникационной сети. Элементы сети, на которые приходится разная нагрузка, могут обладать одним и тем же значением ранга. Проблема состоит в том, как ранжировать элементы ИКС.

Для ранжирования элементов ИКС использован алгоритм, приведенный в [20]. Ранжирование производится раздельно, сначала для УК, затем для КС, по следующей схеме:

– выбирается УК (КС), имеющий наименьшую нагрузку. Всем УК (КС), для которых это условие выполняется, присваивается ранг 1;

– из оставшихся УК (КС) выбирается узел (канал) с наименьшей нагрузкой $T_{q\min}$, $q = 2$, и выполняется процедура предыдущего этапа. Группе выбранных УК (КС) присваивается ранг

$$z = \lfloor (T_{1\min} - T_{2\min}) / (bT_{1\min}) \rfloor,$$

где $\lfloor \cdot \rfloor$ – символ округления до целого в большую сторону;

– процедура повторяется для $q = 3,\ 4,\ \ldots$ до тех пор, пока не будет исчерпано все множество УК (КС).

Наиболее распространена оценка состояния РП ИКС П с использованием программных инструментов. Основными классами соответствующих программных инструментов являются системы управления сетью (Network management systems); средства управления системой (System management); встроенные системы диагностики и управления (Embedded systems); оборудование для диагностики и сертификации кабельных систем. Проведенный анализ средств поддержки администрирования сетей показал проблемы в их использовании:

– известные средства дают возможность проводить оценку значения параметров, но нет информации, действительно ли эти параметры адекватно и с наименьшими затратами отображают состояние РП ИКС П;

– использование средств не снимает проблемы избыточности или недостатка набора параметров для адекватной оценки состояния РП ИКС П;

– высокая стоимость средств оценки состояния РП ИКС П, особенно для интеллектуальных средств, доходящая до сотен тысяч долларов;

– все проанализированные средства не обладают инструментом выявления минимального набора информативных параметров для полнофакторного анализа и ограничены по функции идентификации состояний РП ИКС П.

Функциональная надежность инфокоммуникационной сети предприятия. Состояние РП ИКС П определяется в том числе и ее функциональной надежностью. Характеристики функциональной надежности РП ИКС П отражают возможность ИКС организовать электронный документооборот по мере запроса пользователей. Другими словами, функциональная надежность сети во многом определяется процессом установления соединения.

При оценке функциональной надежности сети количественным показателем служит вероятность установления (не установления) соединения при поступлении соответствующего запроса на передачу (получения) информации. В свою очередь процесс установления соединения зависит от сформированных D подмножеств путей и распределения потоков информации. При этом время установления соединения жестко ограничивается, как и вероятность превышения этого времени, ограничивается также вероятность не установления соединения за допустимое время.

В связи с этим вначале рассмотрим стратегию маршрутизации при передаче информации в промышленной ИКС.

Для оптимизации процесса передачи данных в РП ИКС П стратегию маршрутизации построим на применении совокупности маршрутных таблиц (МТ), которые распределяются по УК и указывают, как в зависимости от конечного адресата должен быть распределен по выходным линиям трафик, поступающий в данный узел. Получаемое решение МТ подлежит оценке с учетом динамики нагрузки и проверяется средствами имитационного моделирования. В качестве такого средства можно использовать имитационную модель установления соединения.

В общем случае РП ИКС П неоднородна. Она содержит, как отмечалось в начале настоящей статьи, разнообразное оборудование. При этом ИКС может содержать два типа станций: станции первого типа, допускающие альтернативные исходящие пучки каналов, и станции второго типа, которые допускают только один исходящий пучок каналов в соответствии с номером, содержащимся в поле передачи информации.

Пути между станциями состоят из цепочки ветвей. Каждая ветвь представляет собой пучок каналов, соединяющих два смежных узла в такой цепочке. Взаимное сочетание названных типов станций обусловливает особенности и влияет на характеристики процесса доставки информации: если в ветви все каналы требуемой скорости заняты, то информация через такую ветвь не пойдет и должна быть передана по другому пути, для чего за счет рестарта возвращается либо к одной из предшествующих станций первого типа, либо к станции-источнику для повторной посылки. Эта особенность передачи информации отражается в модели РП ИКС П.

Кроме фазы доставки информации до адресата процесс установления сквозного соединения включает фазу предустановки соответствующих каналов станций, через которые информация прошла при ее доставке к требуемому адресату, и тем самым на промежуточных станциях был зарегистрирован необходимый для сквозного соединения ресурс.

Таким образом, с точки зрения прохождения вызова от станции к станции в ИКС состояние ветви бинарно: либо ветвь полностью занята и информация по ней не проходит (состояние ветви «1»), либо ветвь имеет хотя бы один незанятый канал требуемой скорости, и передача информация по такой ветви произойдет (состояние ветви «0»). Поэтому установление соединения имеет три исхода:

– информация дойдет до адресата, причем время передачи информации, складывающееся из времени доставки информации и времени предустановки соответствующих каналов станций, не превысит допустимого значения;

– соединение не будет установлено, поскольку информация вообще не дойдет до адресата (все ветви каналов ИКС полностью заняты другой нагрузкой);

– информация дойдет до адресата, однако суммарное время доставки за счет рестартов и повторных попыток доставки информации по другим путям превысит установленное максимальное значение.

Ввиду возможности рестартов количество ветвей, пройденных информацией при ее доставке к адресату, оказывается случайным. Поэтому и весь процесс установления соединения можно считать стохастическим (случайным). Если времена прохождения информации по ветви, предустановки соответствующих каналов станций и рестартов считать фиксированными величинами, то время доставки информации по ветви определяется выражением

$$\tau_{\text{ус}} = n_g t_{\text{в}} + n_p t_{\text{п/у к}} + r t_r, \qquad (2)$$

где n_g – число ветвей, пройденных информацией при ее доставке адресату включая рестарты; $t_{\text{в}}$, $t_{\text{п/у к}}$, t_r – времена прохождения ветви, предустановки соответствующего канала станций и рестарта, соответственно; n_p – число ветвей в пути, по которому устанавливается соединение между абонентами; r – число рестартов, произошедших в процессе установления соединения. В общем случае $0 \le r < r_{\text{доп}}$, где $r_{\text{доп}}$ – допустимое число рестартов при установлении соединения. При $r > r_{\text{доп}}$ считается, что информация адресату не доставлена.

В задаче оценки передачи данных в ИКС входными данными являются структурная и потоковая метрики, множество кратчайших путей (МКП), полученные на их основе значения вероятностей полной занятости ветвей и ограничения, при которых должно выполняться соединение: максимально допустимое число ветвей в соединении, допустимое время установления

соединения, времена продвижения информации через ветвь и предустановки соответствующих каналов, число рестартов, время выполнения рестарта.

Выходными данными являются статистики, позволяющие оценить вероятность установления соединения (доставки информации адресату) за время, не превышающее допустимого, средние и среднеквадратические значения $\tau_{ус}$, характеризующие процесс установления соединения, которые могут быть использованы при упрощенных аналитических расчетах.

Алгоритм процесса установления соединения. Рассмотрим алгоритм процесса установления соединения на примере передачи информации между двумя станциями (i, j) (i – источник вызова; j – адресат) [20]. Такую схему передачи информации можно представить в виде графа $G=(A, L)$, где A – множество узлов, входящих в трассу обмена информацией; L – множество ветвей, отображающих линии связи, соединяющие узлы направления обмена. Множество узлов $A=(F, S)$ включает $F=\{F_k\}$, $k=\overline{1, p}$ – подмножество узлов, отображающих станции первого типа, допускающие альтернативные исходящие пути (p – количество узлов первого типа) и $S=\{S_n\}$, $n=\overline{1, z}$ – подмножество узлов, отображающих станции второго типа, допускающие единственный исходящий путь в соответствии с адресом узла (z – количество узлов второго типа). Множество соединительных ветвей $L=\{\boldsymbol{\eta}_v\}$, $v=\overline{1, m}$, включает $\boldsymbol{\eta}_v=(\eta_{v1}, \ldots, \eta_{vc})$ – упорядоченные векторы скоростей, на которых может осуществляться связь по v-й ветви, где m – количество узлов, входящих в ветвь, причем c – количество предусмотренных в сети скоростей передачи информации. Упорядоченность заключается в том, что $v_{m'} > v_m$ при $m < m'$, $m, m'=\overline{1, c}$.

Положим, что установление соединения с требуемой скоростью v_m может быть осуществлено, если свободны каналы со скоростями $v_{m'} \geq v_m$, $m, m'=\overline{1, c}$.

В качестве критерия эффективности МТ примем вероятность установления соединения за время, не превышающее допустимого. В общем случае, в силу большой размерности сети для решения задачи оценки вероятности установления (не установления) соединения в ИКС целесообразно использовать двухступенчатую выборку [20].

На первой ступени из станций сети $\{A_i\}$, $i=\overline{1, k+w}$, входящих в D подмножеств, случайным образом выбираются N_1 пар станций (i, j), $i, j=\overline{1, k+w}$, $i \neq j$, для которой должна быть оценена возможность установления соединения. Пара (i, j) задает для второй ступени входные данные, которыми являются записи файлов, содержащие состав МКП$_{ij}$ и соответствующие вероятности, характеризующие «закрытость» ветвей множества кратчайших путей.

При случайном выборе пары (i, j) в начале в соответствии с вероятностями готовностей узлов к приему или передаче данных $p(i)$, $i=\overline{1,\ k+w}$, выбирается номер узла-источника, готового принять информацию, а затем с учетом вероятностей готовности связей источника разыгрывается номер узла-адресата.

Файлы вероятностей $P(i)$ и $P\left(j|i\right)$, $i=\overline{1,\ k+w}$, $i \neq j$, формируются при решении задачи распределения потоков, причем

$$P(i)=\frac{\sum\limits_{\text{к}}\lambda_{i\text{к}}}{\sum\limits_{i}\sum\limits_{\text{к}}\lambda_{i\text{к}}};\ P\left(j|i\right)=\frac{\lambda_{ij}}{\sum\limits_{i,\ j\neq i}\lambda_{ij}},$$

где $\lambda_{i\text{к}}$ – интенсивность нагрузки, поступающей от к-го абонента, подключенного к i-му узлу-источнику; λ_{ij} – интенсивность обмена информацией между i-м источником и j-м адресатом, задаваемая матрицей нагрузки.

Выходом алгоритма случайного процесса установления соединения являются номера пар станций $\left(i,\ j\right)$ и значение вероятности $P\left(i,\ j\right)=$ $=P(i)(i)P\left(j|i\right)$.

Для ускорения формирования оценки вероятности установления (не установления) соединения в ИКС на первой ступени все множество МКП$_{ij}$, $i=\overline{1,\ k+w}$, $i \neq j$, декомпозируется на подмножества, используя известные алгоритмы декомпозиции. Основой для объединения в подмножества является оптимальное время передачи информации по альтернативным ветвям путей ИКС.

На второй ступени производится разделение по числу ветвей со скоростями $\eta \in \eta_{\min} ... \eta_{\max}$ в МКП$_{ij}$, полностью занятых нагрузкой при требуемой скорости. Случайным образом определяются количество полностью занятых ветвей в МКП$_{ij}$ и на полученной реализации имитируется процесс доставки информации адресату. По факту доставки вызова фиксируются значения n_g, n_p и r. Процедура при случайном выборе количества полностью занятых ветвей повторяется N_2 раз. По результатам экспериментов вычисляются необходимые статистики.

Исходя из того, что каждое генерирование конкретной реализации ИКС равновероятно, очередной номер определяется по формуле $j-\lfloor Ud \rfloor+1$, где $U \in [0,1]$ – число, формируемое генератором случайных чисел; $\lfloor \cdot \rfloor$ – символ округления в меньшую сторону. Составляющие МКП$_{ij}$ с этим номером полагаются равными единице.

Таким образом, быстродействие таких ИКС определяется в первую очередь пропускной способностью КС и пунктом ее размещения.

После того как произведена декомпозиция РП ИКС П, определены существенность связей в ИКС и параметры оптимального установления соединения, произведена оценка состояния РП ИКС П, рассмотрена стратегия маршрутизации, разработаем алгоритм оптимизации процесса передачи данных в ИКС.

Всю архитектуру РП ИКС П можно представить в виде полновзвешенного графа G с матрицей весов $\|C_{ij}\|$ $(i, j \in \text{МТ})$, где каждой вершине присваивается вес $\omega_j \geq \Lambda$ для всех $j \in \text{МТ}$. Граф G формируется из полученных при декомпозиции D подмножеств оптимальных путей. Для каждой вершины определим два числа, которые называются передаточными числами:

$$y_1 = \sum_{j \in \text{МТ}} \omega_j d\left(x_i, x_j\right); \; y_2 = \sum_{j \in \text{МТ}} \omega_j d\left(x_j, x_i\right),$$

где $d(x_i, x_j) = C_{ij}$ – кратчайшее расстояние от вершины x_i до вершины x_j. Числа y_1 и y_2 называются внешними и внутренними передаточными числами вершины x_i соответственно.

Если граф G имеет симметричную матрицу весов, т. е. по одним и тем же каналам связи осуществляется как передача, так и прием данных, то $y_1 = y_2$. Таким образом, в качестве пункта размещения коммутационной станции (КСт) можно взять вершину графа, участвующую в передаче информации от узла 1 к узлу 2.

Выбор варианта построения КСт производится таким образом, чтобы затраты на ее создание были минимальными и выполнялось условие:

$$\sum_{i \in \text{МТ}} \lambda_i / m_0 \leq \omega_{j(r)}, \;\; j \in \text{МТ}, \;\; r \in R, \tag{3}$$

где ω_{jr} – пропускная способность r-го варианта КСт, расположенного в j-м пункте. В этом случае алгоритм построения КСт будет иметь следующий вид.

1. Упорядочить все затраты на создание вариантов построения КСт по возрастанию: $C_1 < C_2 < \ldots < C_z < \ldots < C_n$.

2. Просмотреть варианты построения КСт в порядке возрастания затрат на их создание до тех пор пока не будет выполняться условие (3).

3. Если условие (3) выполняется, то за j-й КСт закрепляется r-е КСт с пропускной способностью производительностью ω_{jr}. Если условие (3) не выполняется, то значение j инкрементируется и выполняется переход к п. 1.

4. Если $j \leq m$, то перейти к п. 5, в противном случае инкрементировать j и перейти к шагу 1.

5. Вычислить передаточные числа y_1 или y_2.

Учитывая, что граф G имеет симметричную матрицу весов, и приняв $\omega_j = \omega = \text{const}$ для всех j, получим $y_1 = y_2 = \omega \sum_{j \in \text{МТ}} d\left(x_i, x_j\right)$. Тогда задача поиска пункта размещения КСт, используя традиционный подход, сводится к нахождению столбца (или строки) с минимальной суммой элементов.

Применение ГА. Традиционный подход для решения поставленной задачи неэффективен, так как требует длительного времени и рассмотрения большого числа вариантов формирования матрицы G. Поэтому рассмотрим применение для решения поставленной задачи ГА.

За основу разработки алгоритма оптимизации процесса передачи данных в инфокоммуникационных системах воспользуемся теорией ГА, изложенной в [11]. Рассмотрим задачу оптимального разбиения графа G с минимизацией K. Задача разбиения состоит в отыскании такого разбиения B_i из множества возможных разбиений B некоторого графа G, при котором минимизируется величина K, являющаяся целевой функцией разбиения, и учитываются все поставленные в задаче ограничения и граничные условия, если они существуют.

В задачах разбиения графов общее число вариантов решения равно числу перестановок из n вершин графов $C_n = n!$, а с учетом ограничений на формирование D подмножеств – числу сочетаний из n вершин по m частей:

$$C_n^m = \frac{n!}{m!(n-m)!}.$$

Решение задачи разбиения графа на основе полного перебора затруднительно из-за экспоненциальной сложности процесса. В этой связи разработаны различные эвристики решения данных задач [11]. Все они в каком-то смысле основаны на анализе и переборе вариантов решения, однако различаются по технологии и принципам реализации.

Построим схему алгоритма с определением оптимальных путей передачи информации в РП ИКС П на основе ГА (рис. 1.5). В блоке 1 строится текущая (начальная) популяция решений, т. е. определяется заданное подмножество $B_1, B_2, \ldots, B_k$, $k \le s$ (s – количество популяций решений). На построение популяции оказывает влияние внешняя среда в виде принимающего решение исследователя.

При большом количестве элементов для получения текущей популяции предпочтительными будут конструктивные, итерационные, случайные и поисковые алгоритмы. В качестве целевой функции выберем функцию

$$K = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} K_{ij}, \; i \ne j,$$

где K_{ij} – число связей между узлами B_i и B_j при построении графа G с оптимальными путями передачи информации; n – количество путей в графе.

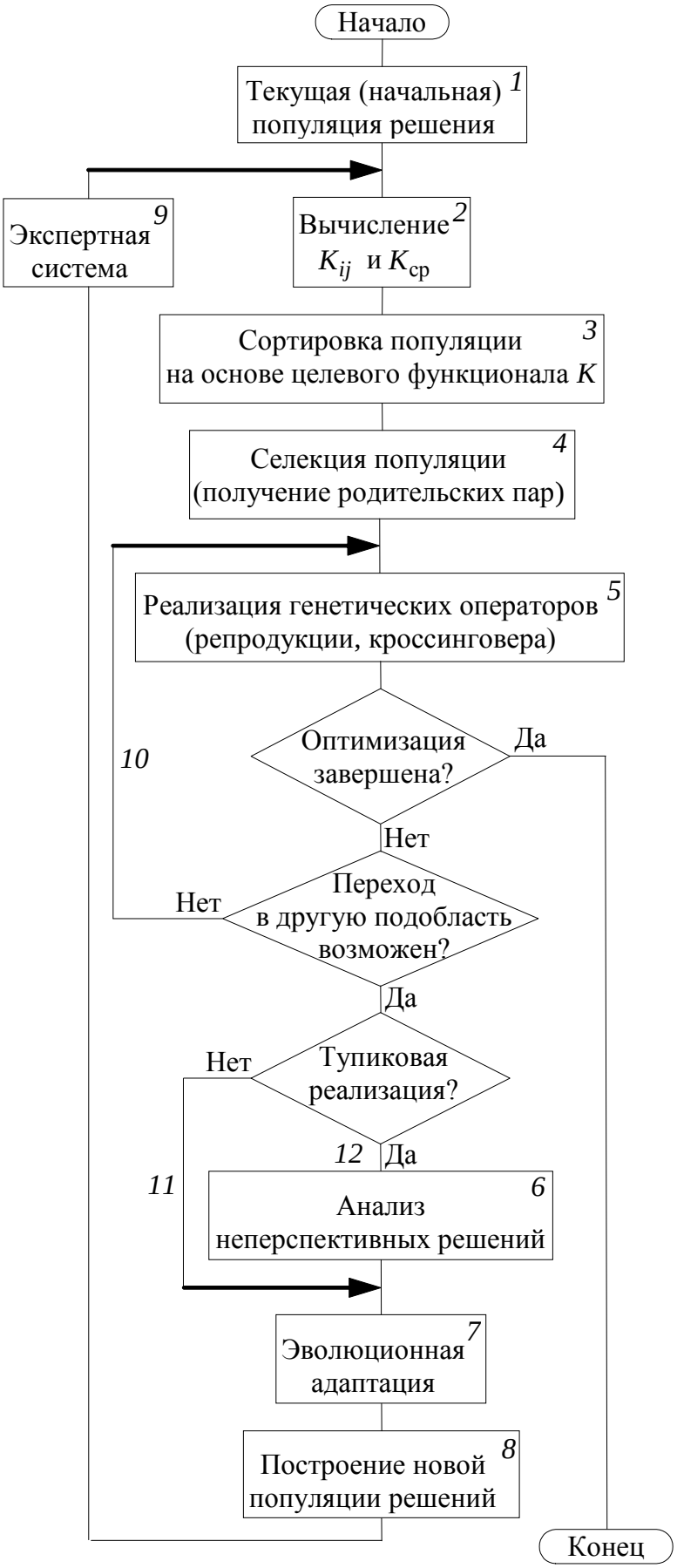

Рис. 1.5. Алгоритм синтеза функционала качества

Вычислим K для каждого элемента текущей популяции и определим $K_{ср}$ – среднее значение целевой функции на данной популяции (блок 2 на рис. 1.5).

Блок 3 осуществляет сортировку популяции на основе целевой функции. При этом могут быть применены все существующие методы сортировки. Сначала расставляются элементы с наименьшим значением K и так далее по возрастанию. Блок 4 выполняет селекцию популяции для получения родительских пар. Селекция выполняется одним из методов, описанных в [11]. Блок 5 реализует генетические операторы (репродукции, кроссинговера) и их модификации.

Существует большое число видов операторов репродукции: на основе рулетки; на основе заданной шкалы; элитная и турнирная [11].

В разработанном алгоритме применена турнирная репродукция. При этом виде репродукции некоторое число элементов (согласно размеру «турнира») выбирается случайно или направленно из популяции (D подмножеств) и лучшие элементы в этой группе на основе заданного турнира определяются для дальнейшего эволюционного поиска. Оператор репродукции считается эффективным, если он создает возможность перехода из одной подобласти альтернативных решений области поиска в другую, что повышает вероятность нахождения оптимального пути передачи информации. Если переход из одной подобласти альтернативных решений области поиска в другую невозможен, необходимо из блока 5 по связи 10 вернуться к блоку 4. Для решения оптимизационной задачи применяется «жадный» оператор кроссинговера (ЖОК) [11]. Алгоритм ЖОК нахождения минимального или максимального по времени пути передачи информации по РП ИКС П на графе заключается в следующем.

1. Для всех хромосом (составляющих) популяции вычисляется целевая функция (время передачи информации по пути между вершинами графа). Выбирается заданное число родительских хромосом и случайным образом на одной из хромосом определяется точка ЖОК.

2. В выбранной хромосоме для i-го гена (разряда кода хромосомы), расположенного слева от точки ЖОК, определяется значение целевой функции на i-м шаге. Аналогичные действия выполняются по определению времени передачи информации от i-го гена во всех остальных хромосомах, выбранных в родительской паре для ЖОК.

3. В хромосому «потомок» выбирают ген, имеющий минимальное значение текущей целевой функции при минимизации исходной целевой функции.

4. Процесс продолжается до построения хромосомы «потомок». После завершения отдельного цикла осуществляется переход из блока 5 по связи 11 к блоку 7. Если реализация заходит в тупик, то из блока 5 по связи 12 переходят к блоку 6.

При получении популяции, обеспечивающей минимальное время передачи информации по всем ветвям путей D подмножеств РП ИКС П ГА заканчивает свою работу (выход из блока 5). Блок 6 собирает и анализирует неперспективные решения задачи разбиения. Блок 7 реализует стратегии адаптации и на основе обратных связей выбирает модель эволюции, а также порядок использования и применения различных алгоритмов генетической оптимизации. В блоке 8 осуществляется построение новой популяции решений. Блок 9 позволяет управлять процессом поиска с помощью информирующих обратных связей. Далее для каждой хромосомы из новой популяции вычисляется K и выживают те элементы из старой и новой популяции, у которых $K \leq K_{\text{ср}}$. При этом в стандартном случае количество элементов в новой популяции не должно превышать число элементов в старой популяции.

Предложенная схема генетического поиска позволяет варьировать размер популяции от генерации к генерации, чем частично предотвращать преждевременную сходимость алгоритма в задачах построения оптимальных путей передачи информации в РП ИКС П. Можно предложить большое число аналогичных схем генетического поиска для решения задач разбиения. Их эффективность проверяется экспериментальным путем [11].

Описанная стратегия поиска позволяет ускорить определение оптимальных путей передачи информации в РП ИКС П. Это связано с параллельной обработкой множества альтернативных решений, причем в такой схеме возможно сконцентрировать поиск на получении наиболее перспективных из них. Отметим, что на каждой итерации ГА можно проводить различные изменения в перспективных, неперспективных и других решениях.

2. ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ В РП ИКС П

2.1. Область применения нейронных сетей

Нейронные сети – это раздел искусственного интеллекта, в котором для обработки сигналов используются явления, аналогичные происходящим в нейронах живых существ.

Если первый этап становления НС можно охарактеризовать, как попытку синтезировать из набора сравнительно просто функционирующих нейронов некоторую упорядоченную структуру, способную выполнять сложные нелинейные преобразования, то сейчас передний фронт исследований перемещается в область психологии, когнитивных наук. Задача состоит в том, чтобы понять, какие сверхструктуры нейронов и как могут моделировать элементарные акты мыслительной деятельности, а затем воплотить эти принципы в работу систем управления.

Нейронные сети предложены для задач, простирающихся от управления боем до присмотра за ребенком. Потенциальными приложениями являются те, где человеческий интеллект малоэффективен, а обычные вычисления трудоемки или неадекватны. Этот класс приложений, во всяком случае, не меньше класса, обслуживаемого обычными вычислениями, и можно предполагать, что искусственные нейронные сети займут свое место наряду с обычными вычислениями в качестве дополнения такого же объема и важности.

На практике, для того чтобы применение нейронной сети было оправдано, необходимо, чтобы задача обладала следующими признаками:

– отсутствует алгоритм или неизвестны принципы решения задач, но накоплено достаточное число примеров;

– проблема характеризуется большими объемами входной информации;

– данные неполны или избыточны, частично противоречивы.

Таким образом, НС хорошо подходят для управления техническими объектами, распознавания образов и решения задач классификации, оптимизации и прогнозирования (табл. 2.1) [18].

Анализ всевозможных источников информации позволил сформировать следующий перечень возможных промышленных применений нейронных сетей.

1. Промышленное производство:

– управление промышленными роботами (манипуляторами), адаптивная робототехника;

– управление технологическими процессами;

– обнаружение неисправностей (техническая диагностика);

– управление качеством производимой продукции;

– управление голосом;

– распознавание образов (техническое зрение) и др.

Таблица 2.1

Промышленность		
Управление технологическими процессами Оценка экологической обстановки Оптимальное планирование Идентификация вида полимеров Обнаружение повреждений Приложения аналитической химии	Идентификация химических компонентов Прогнозирование свойств синтезируемых полимеров Разработка нефти и газа Управление ценами и производством Оптимизация закупок сырья Анализ проблем функционирования заводов и магазинов	Контроль качества артезианских вод Управление водными ресурсами Управление работой пресса Оптимизация работы моторов Контроль качества изделий Прогнозирование потребления энергии
Высокие технологии		Оборона
Проектирование и оптимизация сетей связи Анализ и сжатие изображений Распознавание печатных и рукописных символов Фальсификации в пищевой и парфюмерной промышленности Обслуживание кредитных карт	Идентификация и верификация говорящего Видеонаблюдение Автоматизированное распознавание речевых команд Распознавание слитной речи с (без) настройкой на говорящего Речевой ввод текста в компьютер	Анализ визуальной аэрокосмической информации Отбор целей Обнаружение наркотиков и взрывчатых веществ Сличение изображений с криминальной базой данных Предсказание целесообразности условного освобождения
Наука и техника		Здравоохранение
Поиск неисправностей в научных приборах Диагностика печатных плат Идентификация продуктов Синтез новых видов стекла Автоматизированное проектирование Оптимизация биологических экспериментов Геофизические и сейсмологические исследования Распознавание ингредиентов	Спектральный анализ и интерпретация спектров Интерпретация показаний сенсоров Моделирование физических систем Анализ данных в ботанике Планирование химических экспериментов Отбор сенсоров для контроля химических процессов Прогноз температурного режима технологических процессов Диагностика сбоев сигнализации	Идентификация микробов и бактерий Диагностика заболеваний Интерпретация ЭКГ Анализ качества лекарств Обработка и анализ медицинских тестов Прогнозирование результатов применения методов лечения Оптимизация атлетической подготовки Диагностика слуха

2. Военная промышленность и аэронавтика:
– автоматическое пилотирование;
– управление системами наведения;

– обработка звуковых сигналов (шифрование, разделение, идентификация, локализация, устранение шума, интерпретация);

– обработка радарных сигналов (распознавание целей, идентификация и локализация источников) и др.

3. Нефтяная и химическая промышленность:

– управление процессами;

– идентификация неисправностей оборудования;

– анализ геологической информации;

– разведка залежей минералов по данным аэрофотосъемок;

– анализ составов примесей и др.

4. Телевидение и связь:

– адаптивное управление сетью связи;

– сжатие и восстановление изображения и др.

Представленный перечень использования нейронных сетей далеко не полон. Более подробно о применении и реализующих их структурах НС можно прочитать в журналах «Neural Computation», «Neural Computing and Applications», «Neural Networks», «IEEE Transactions on Neural Networks», «IEEE Transactions on System», «Man and Cybernetics» и др.

Существует два подхода к созданию искусственных нейронных сетей.

Информационный подход: безразлично, какие механизмы лежат в основе работы искусственных нейронных сетей, важно лишь, чтобы при решении задач информационные процессы в НС были подобны биологическим. *Биологический подход*: при моделировании важно полное биоподобие, и необходимо детально изучить работу биологического нейрона.

Однако исторически сложилось так, что первые искусственные нейронные сети были созданы в результате попыток создать компьютерную модель, воспроизводящую деятельность мозга в упрощенной форме. Конечно, возможности человеческого мозга неизмеримо больше, чем возможности самой мощной искусственной нейронной сети. Однако искусственные нейросети обладают рядом свойств, присущих биологическим нейросетям, в том числе и человеческому мозгу.

Важнейшая особенность сети, свидетельствующая о ее широких возможностях и огромном потенциале, состоит в параллельной обработке информации всеми звеньями. При громадном количестве межнейронных связей это позволяет значительно ускорить процесс обработки информации. Во многих случаях становится возможным преобразование сигналов в реальном времени. Кроме того, при большом числе межнейронных соединений сеть приобретает устойчивость к ошибкам, возникающим на некоторых линиях. Функции поврежденных связей берут на себя исправные линии, в результате чего деятельность сети не претерпевает существенных возмущений.

Другое не менее важное свойство – способность к обучению и обобщению накопленных знаний. Нейронная сеть обладает чертами искусственного

интеллекта. Натренированная на ограниченном множестве данных сеть способна обобщать полученную информацию и показывать хорошие результаты на данных, не использовавшихся в процессе обучения.

Характерная особенность сети состоит также в возможности ее реализации с применением технологии сверхбольшой степени интеграции. Различие элементов сети невелико, а их повторяемость огромна. Это открывает перспективу создания универсального процессора с однородной структурой, способного перерабатывать разнообразную информацию.

Использование перечисленных свойств на фоне развития устройств со сверхбольшой степенью интеграции (VLSI) и повсеместного применения вычислительной техники вызвало в последние годы огромный рост интереса к нейронным сетям и существенный прогресс в их исследовании. Создана база для выработки новых технологических решений, касающихся восприятия, искусственного распознавания и обобщения видеоинформации, управления сложными системами, обработки речевых сигналов и т. п. Искусственные нейронные сети в практических приложениях, как правило, используются в качестве подсистемы управления или выработки решений, передающей исполнительный сигнал другим подсистемам, имеющим иную методологическую основу. Функции, выполняемые НС в современных средствах автоматизации электромеханических объектов, подразделяются на несколько групп: 1) управления технологическими объектами; 2) прогнозирования; 3) идентификации и оценивания; 4) аппроксимации данных в системах управления (например, решение траекторных задач); 5) классификации и распознавания образов (например, техническое зрение роботов).

В задачах управления динамическими процессами НС выполняет, как правило, несколько функций. Во-первых, она представляет собой нелинейную модель этого процесса и идентифицирует его основные параметры, необходимые для выработки соответствующего управляющего сигнала. Во-вторых, сеть выполняет функции следящей системы, отслеживает изменяющиеся условия окружающей среды и адаптируется к ним. Она также может играть роль нейрорегулятора, заменяющего собой традиционные устройства. Важное значение, особенно при управлении роботами, имеют классификация текущего состояния и выработка решений о дальнейшем развитии процесса.

В области прогнозирования задача сети формулируется как предсказание будущего поведения системы по имеющейся последовательности ее предыдущих состояний. По информации о значениях переменной x в моменты времени, предшествующие прогнозированию, сеть вырабатывает решение о том, чему должно быть равно оцениваемое значение исследуемой последовательности в текущий момент времени.

Аппроксимирующая сеть играет роль универсального аппроксиматора функции нескольких переменных [24], который реализует нелинейную

функцию вида $y = f(x)$, где x – входной вектор; y – реализованная функция нескольких переменных. Множество задач моделирования, идентификации, обработки сигналов удается сформулировать в аппроксимационной постановке.

Для классификации и распознавания образов сеть накапливает в процессе обучения знания об основных свойствах этих образов, таких как геометрическое отображение структуры образа, распределение главных компонентов (РГК), или о других характеристиках. При обобщении акцентируются отличия образов друг от друга, которые и составляют основу для выработки классификационных решений.

Различные способы объединения нейронов между собой и организации их взаимодействия привели к созданию сетей разных типов. Каждый тип сети, в свою очередь, тесно связан с соответствующим методом подбора весов межнейронных связей (т. е. обучения). Среди множества существующих видов сетей в качестве важнейших можно выделить [7], [8], [18], [19] многослойный персептрон, радиальные сети RBF, сети с самоорганизацией в результате конкуренции нейронов, сети с самоорганизацией корреляционного типа, а также рекуррентные сети, в которых имеются сигналы обратной связи.

2.2. Топология нейронных сетей

Нейронная сеть представляет собой совокупность большого числа сравнительно простых элементов – нейронов, топология соединений которых зависит от типа сети. Некоторые нейроны связаны с «внешним миром», некоторые только с другими нейронами (скрытые нейроны). Существует два основных типа топологии НС: *feed-forward* (с прямой связью) и feedback (с обратной связью). В feed-forward-сети сигнал распространяется только в одну сторону, т. е. сеть не имеет петель.

В *feedback*-нейросети (рис. 2.1) выходной сигнал от нейрона передается на вход другого нейрона на том же или предыдущем уровне.

Термин feedback (обратная связь) указывает способ соединения нейронов. Но не следует путать его и back propagation. Back propagation – это метод обучения feed-forward-нейросетей. В методе back propagation ошибка на последнем (выходном уровне) используется для коррекции весов всей сети (распространения обратно через сеть). Feedback-нейросети не используют back propagation для обучения. Нейросеть back propagation – это не нейросеть с топологией feedback.

В feed-forward-нейросетях (рис. 2.2) сигнал распространяется только в одном направлении – от входов к выходам. Линейные feed-forward НС были созданы одними из первых. В настоящее время чаще всего используют нелинейные feed-forward-сети.

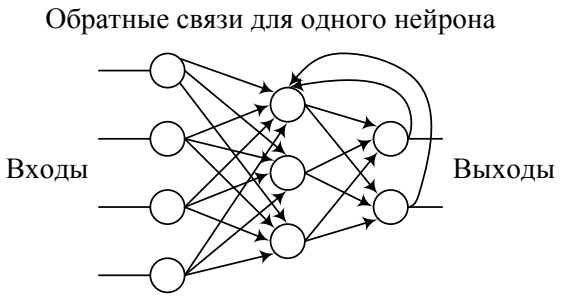

Рис. 2.1. НС feedback
(с обратной связью)

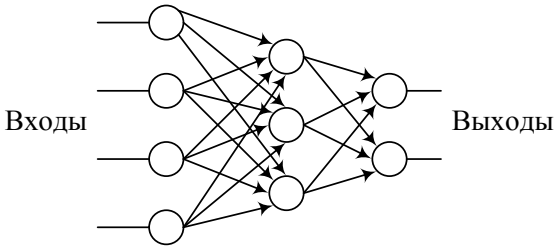

Рис. 2.2. НС feed-forward
(с прямой связью)

Feed-forward-сети считаются менее пригодными для реализации ассоциативной памяти, чем feedback-сети, хотя они могут выполнять такие же задачи. Важно знать: можно доказать математически, что любая feedback-сеть имеет эквивалентную feed-forward-сеть, которая может выполнять те же самые функции.

В настоящее время нейронные сети с архитектурой feed-forward работают лучше, чем сети с архитектурой feedback, по нескольким причинам. Емкость feedback оказалась не очень большой. Так, например, количество данных, которое может быть сохранено без ошибок в CAM (Hophield Content Addressable Memory), вычисляется по формуле $M \leq N/(4 \cdot \log N)$, где N – количество нейронов. Это значит, что для хранения 1000 записей потребуется 20 000 нейронов и 200 000 000 уникальных связей. Для сравнения NetTalk, приложение использующее feed-forward-модель, используя 300 нейронов и 20 000 уникальных соединений, может выучить правильное написание 1000 слов и около 7000 ассоциаций буква-звук.

Кроме того, feed-forward-сети работают быстрее, чем feedback, так как для того чтобы найти решение, им нужно сделать только один проход. Feedback-сети должны повторять цикл снова и снова, до тех пор, пока выходы не перестанут изменяться. Обычно для этого требуется от 3 до 1000 циклов.

Feed-forward НС могут быть контролируемые (supervised) или неконтролируемые (unsupervised). Контролируемые нейронные сети во время обучения сравнивают свой ответ с заранее известным, в то время как неконтролируемые не делают этого.

2.3. Модели нейронных сетей

Рассмотрим основные модели НС, применяемые в РП ИКС П для решения различных задач.

Модели НС могут быть программного и аппаратного исполнения. Несмотря на существенные различия, отдельные типы НС обладают несколькими общими чертами.

Во-первых, основу каждой НС составляют относительно простые, в большинстве случаев – однотипные, элементы (ячейки), имитирующие работу нейронов мозга. Далее под нейроном будет подразумеваться искусственный нейрон, т. е. ячейка НС. Каждый нейрон характеризуется своим текущим состоянием по аналогии с нервными клетками головного мозга, которые могут быть возбуждены или заторможены. Он обладает группой синапсов – однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон – выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Общий вид нейрона приведен на рис. 2.3.

На входы j-го нейрона поступает n сигналов $x_1, x_2, ..., x_n$, которые взвешиваются усилителями, реализующими синаптические веса, после чего взвешенные значения $w_{j1}x_1, w_{j2}x_2, ..., w_{jn}x_n$ вместе с пороговым значением θ_j, именуемым также сигналом смещения, подаются на сумматор $\sum$, в результате чего формируется внутренний сигнал u_j. Сома биологического нейрона моделируется с помощью некоторой нелинейной функции $\Psi(u_j)$, называемой в теории ИНС либо активационной, либо передаточной функцией формального нейрона.

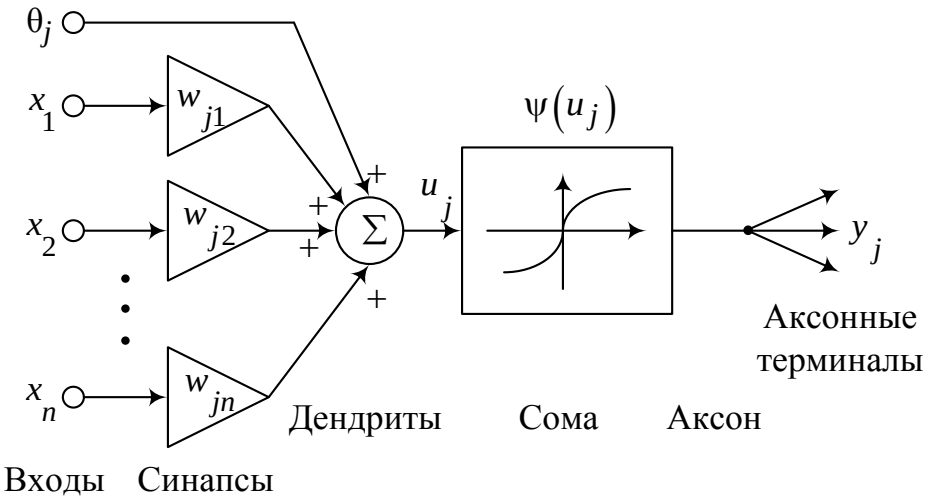

Рис. 2.3. Общий вид нейрона

Каждый синапс характеризуется величиной синаптической связи или ее весом ω_i, который по физическому смыслу эквивалентен электрической проводимости.

Таким образом, математическая модель МакКалока – Питтса может быть записана в виде

$$y_j = \Psi\left(\sum_{i=1}^{n} x_i \cdot w_{ji} + \vartheta_j\right),$$

или

$$y_j = \Psi\left(\sum_{i=0}^{n} x_i \cdot w_{ji}\right), \tag{2.1}$$

где $w_{j0} = \vartheta_j$, $x_0 \equiv 1$.

Введя вектор синаптических весов j-го нейрона $w_j = (w_{j0}, w_{j1}, ..., w_{jn})^{\text{т}}$ и вектор входов $x = (1, x_1, x_2, ..., x_n)^{\text{т}}$, можно переписать уравнение (2.1) в векторной форме

$$y_j = \Psi(x \cdot w_j^{\text{т}}).$$

Функциональные характеристики отдельных нейронов и сетей в целом определяются видом используемых активационных функций. В настоящее время используется множество преобразований, некоторые из которых приведены на рис. 2.4 ($\theta(\alpha) = [\alpha \geq 0]$ – ступенчатая функция Хэвисайда (1); $\sigma(\alpha) = \left(1 + e^{-\alpha}\right)^{-1}$ – сигмоидная функция (2); $\text{th}\,\alpha = 2\sigma(2\alpha) - 1$ – гиперболический тангенс (3); $\ln\left(\alpha + \sqrt{\alpha^2 + 1}\right)$ – логарифмическая функция (4); $\exp\left(-\alpha^2 / 2\right)$ – гауссовская функция (5); α – линейная функция (6)).

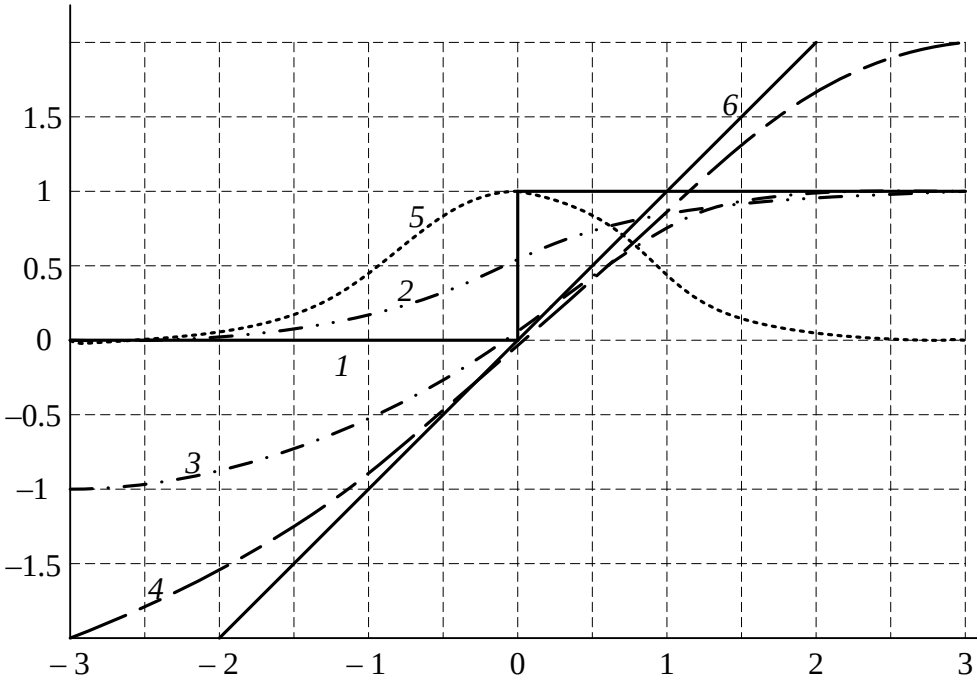

Рис. 2.4. Функции активации

Одной из наиболее распространенных является нелинейная функция с насыщением, так называемая логистическая функция или сигмоид (т. е. функция S-образного вида) [28].

При уменьшении α сигмоид становится более пологим, в пределе при α = 0 вырождаясь в горизонтальную линию на уровне 0,5, при увеличении α сигмоид приближается по внешнему виду к функции единичного скачка с порогом T в точке $x = 0$. Из выражения для сигмоида очевидно, что выходное значение нейрона лежит в диапазоне [0; 1]. Одно из ценных свойств сигмоидной функции – простое выражение для ее производной.

Следует отметить, что сигмоидная функция дифференцируема на всей оси абсцисс, что используется в некоторых алгоритмах обучения. Кроме того она обладает свойством усиливать слабые сигналы лучше, чем большие, и предотвращает насыщение от больших сигналов, так как они соответствуют областям аргументов, где сигмоид имеет пологий наклон.

Проведенный анализ практического применения нейронных сетей в ИКС П показал, что для более стабильной работы ИКС следует применять НС, структура которой показана на рис. 2.5.

Параметр γ_j управляет крутизной функции активации, а Γ_j – коэффициент усиления, определяющий максимальные и минимальные значения выходного сигнала γ_j.

Несложно видеть, что выходной сигнал такого нейрона может быть записан в виде

$$y_j = \Gamma_j \Psi\left(\gamma_j \cdot x \cdot w_j^{\mathrm{T}}\right).$$

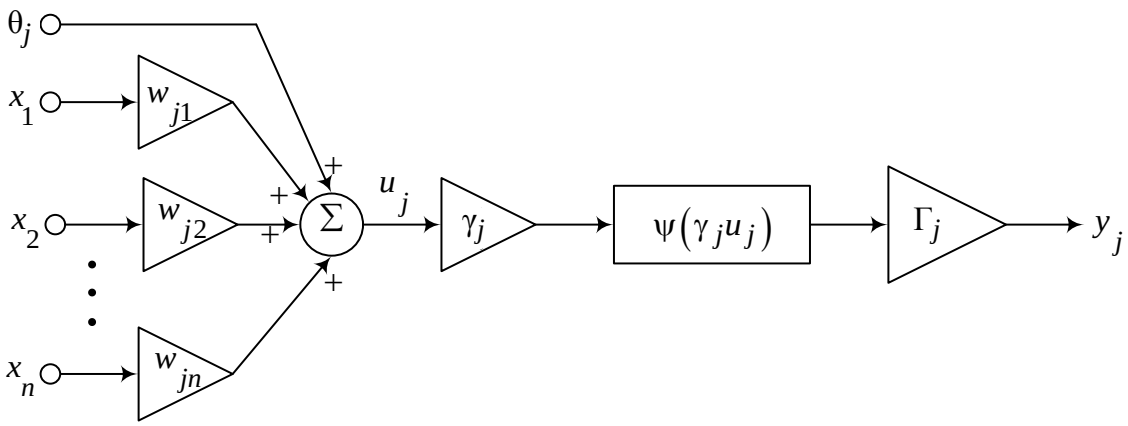

Рис. 2.5. Обобщенный формальный нейрон

Нейронным сетям присущ принцип параллельной обработки сигналов, который достигается путем объединения большого числа нейронов в так называемые слои и соединения определенным образом нейронов различных слоев, а также, в некоторых конфигурациях, и нейронов одного слоя между собой, причем обработка взаимодействия всех нейронов ведется послойно.

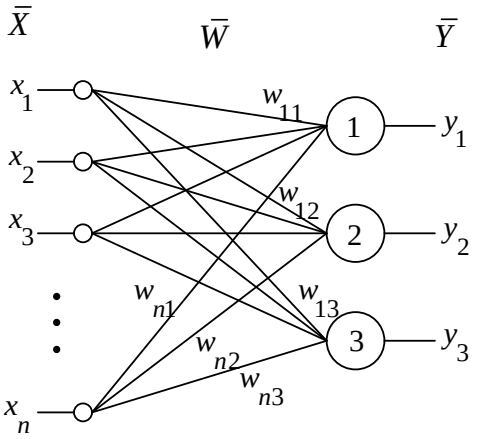

Рис. 2.6. Трехнейронный персептрон

В качестве примера простейшей НС рассмотрим трехнейронный персептрон (рис. 2.6), т. е. такую сеть, нейроны которой имеют активационную функцию в виде единичного скачка. На n входов поступают некие сигналы, проходящие по синапсам на 3 нейрона, образующие единственный слой этой НС и выдающие три выходных сигнала:

$$y_j = \Psi\left(\sum_{i=1}^{n} x_i \cdot w_{ji}\right),\ j = 1...3.$$

Очевидно, что все весовые коэффициенты синапсов одного слоя нейронов можно свести в матрицу W, в которой каждый элемент w_{ij} задает величину i-й синаптической связи j-го нейрона. Таким образом, процесс, происходящий в НС, может быть записан в матричной форме:

$$\mathbf{Y} = \mathbf{F}(\mathbf{XW}),$$

где **X** и **Y** – соответственно, входной и выходной сигнальные векторы: **F**(**V**) – активационная функция, применяемая поэлементно к компонентам вектора **V**.

Теоретически число слоев и число нейронов в каждом слое может быть произвольным, однако фактически оно ограничено ресурсами компьютера или специализированной микросхемы, на которых обычно реализуется НС. Чем сложнее НС, тем масштабнее задачи, подвластные ей.

Выбор структуры НС осуществляется в соответствии с особенностями и сложностью задачи. Для решения некоторых отдельных типов задач уже существуют оптимальные на сегодняшний день конфигурации, описанные, например, в [13], [28], [32] и других изданиях. Если же задача не может быть сведена ни к одному из известных типов, разработчику приходится решать сложную проблему синтеза новой конфигурации.

При этом он руководствуется несколькими основополагающими принципами: возможности сети возрастают с увеличением числа ячеек сети, плотности связей между ними и числом выделенных слоев; введение обратных связей наряду с увеличением возможностей сети поднимает вопрос о динамической устойчивости сети; сложность алгоритмов функционирования сети (в том числе, например, введение нескольких типов синапсов – возбуждающих, тромозящих и др.) также способствует усилению мощи НС.

Вопрос о необходимых и достаточных свойствах сети для решения того или иного рода задач представляет собой целое направление нейрокомпьютерной науки. Так как проблема синтеза НС сильно зависит от решаемой

задачи, дать общие подробные рекомендации затруднительно. В большинстве случаев оптимальный вариант получается на основе интуитивного подбора.

Очевидно, что процесс функционирования НС, т. е. сущность действий, которые она способна выполнять, зависит от величин синаптических связей, поэтому, задавшись определенной структурой НС, отвечающей какой-либо задаче, разработчик сети должен найти оптимальные значения всех переменных весовых коэффициентов (некоторые синаптические связи могут быть постоянными). Этот этап называется обучением НС, и от того, насколько качественно он будет выполнен, зависит способность сети решать поставленные перед ней проблемы во время эксплуатации. На этапе обучения, кроме параметра качества подбора весов, важную роль играет время обучения. Как правило, эти два параметра связаны обратной зависимостью и их приходится выбирать на основе компромисса.

Обучение НС может вестись с учителем или без него. В первом случае сети предъявляются значения как входных, так и желательных выходных сигналов, и она по некоторому внутреннему алгоритму подстраивает веса своих синаптических связей. Во втором случае выходы НС формируются самостоятельно, а веса изменяются по алгоритму, учитывающему только входные и производные от них сигналы.

Существует великое множество различных алгоритмов обучения, которые делятся на два больших класса: детерминистские и стохастические. В первом из них подстройка весов представляет собой жесткую последовательность действий, во втором классе она производится на основе действий, подчиняющихся некоторому случайному процессу.

Развивая дальше вопрос о возможной классификации НС, важно отметить существование бинарных и аналоговых сетей. Первые из них оперируют с двоичными сигналами, и выход каждого нейрона может принимать только два значения: логический ноль («заторможенное» состояние) и логическая единица («возбужденное» состояние). К этому классу сетей относится и рассмотренный персептрон, так как выходы его нейронов, формируемые функцией единичного скачка, равны либо 0, либо 1. В аналоговых сетях выходные значения нейронов способны принимать непрерывные значения, что могло бы иметь место после замены активационной функции нейронов персептрона на сигмоид.

Еще одна классификация делит НС на синхронные и асинхронные [13]. В первом случае в каждый момент времени свое состояние меняет лишь один нейрон, во втором – состояние меняется сразу у целой группы нейронов, как правило, у всего слоя. Алгоритмически ход времени в НС задается итерационным выполнением однотипных действий над нейронами. Далее будут рассматриваться только синхронные НС.

Сети также можно классифицировать по числу слоев. На рис. 2.7 представлен двухслойный персептрон, полученный из персептрона с рис. 2.6

путем добавления второго слоя, состоящего из двух нейронов. Здесь уместно отметить важную роль нелинейности активационной функции, так как, если бы она не обладала данным свойством или не входила в алгоритм работы каждого нейрона, результат функционирования любой p-слойной НС с весовыми матрицами $W^{(i)}$, $i = 1, 2, ..., p$ для каждого слоя i сводился бы к перемножению входного вектора сигналов **X** на матрицу

$$W^{(\Sigma)} = W^{(1)} \cdot W^{(2)} \cdots W^{(p)},$$

т. е. фактически такая p-слойная НС эквивалентна однослойной НС с весовой матрицей единственного слоя $W^{(\Sigma)}$:

$$Y = XW^{(\Sigma)}.$$

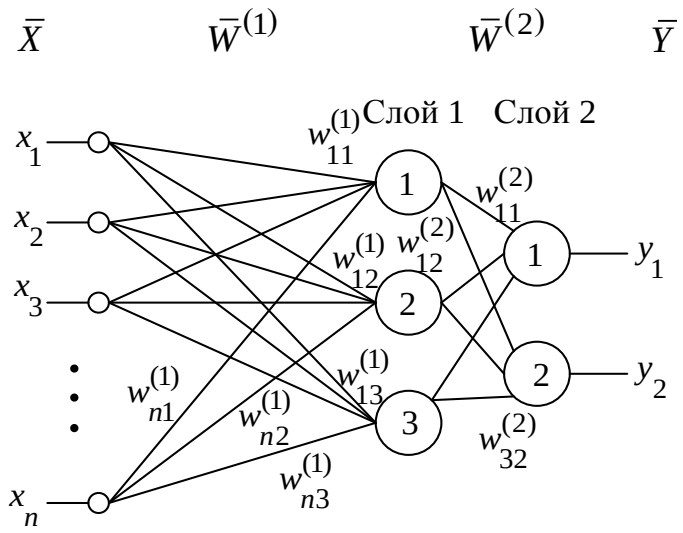

Рис. 2.7. Двухслойный персептрон

Основной особенностью НС и образующих их нейронов является способность к обучению, в процессе которого синаптические веса настраиваются с помощью того или иного адаптивного алгоритма с целью наиболее эффективного решения поставленной проблемы.

Одним из простейших обучающихся нейронов является АДаптивный ЛИНейный Элемент (Adaline), предложенный Б. Уидроу и приведенный на рис. 2.8.

Структурно Adaline состоит из двух основных частей: адаптивного линейного ассоциатора и нелинейной активационной функции. Adaline имеет $(n + 1)$ входов – $x_0, x_1, ..., x_n$ и два выхода – аналоговый u_j и бинарный y_j. Кроме того, имеется дополнительный вход, на который подается обучающий сигнал d_j, показывающий, какой должна быть желаемая реакция нейрона на каждый конкретный набор входных сигналов.

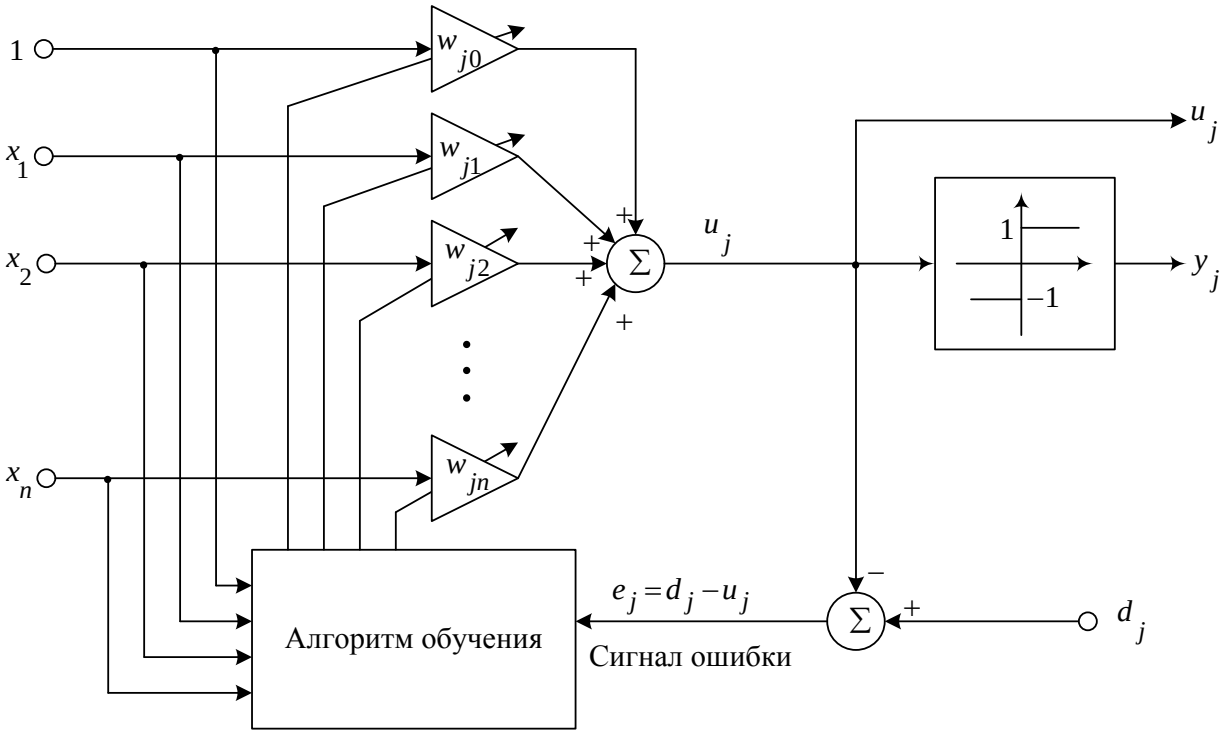

Рис. 2.8. Adaline

Аналоговый выход u_j представляет собой взвешенную сумму входов x_j

$$u_j = \sum_{i=0}^{n} w_{ji} \cdot x_i = x \cdot w_j^{\mathrm{T}},$$

а бинарный выход y_j может принимать значения +1 или –1 в зависимости от полярности аналогового сигнала u_j. Выходной сигнал u_j сравнивается с внешним обучающим сигналом d_j, и возникающий при этом сигнал ошибки $e_j = d_j - u_j$ поступает в алгоритм обучения, который перестраивает синаптические веса так, чтобы минимизировать некоторую функцию ошибки e_j, называемую критерием обучения.

В качестве такой функции чаще всего используют квадратичную, что позволяет использовать для обучения не только «родной» алгоритм, синтезированный Б. Уидроу и М. Хоффом специально для Adaline, но и множество рекуррентных процедур адаптивной идентификации.

Adaline может использоваться как в качестве элементарного нейрона в составе НС, так и самостоятельно в задачах распознавания образов, обработки сигналов, реализации логических функций [11].

Кроме Adaline для обучения можно применить элементарный персептрон Ф. Розенблатта, который структурно подобен Adaline. Основное отличие состоит в алгоритме обучения, поскольку в Adaline ошибка $e_j = d_j - u_j$ является линейной функцией от векторов входов x, в то время как ошибка

обучения персептрона $e_j = d_j - y_j$ от входов зависит нелинейно, причем характер этой нелинейности определяется активационной функцией Ψ(•), которая в общем случае может быть недифференцируемой (например, сигнум-функция). Данное обстоятельство существенно затрудняет процесс обучения персептрона и требует использования специальных процедур.

Многослойный персептрон. В настоящее время большинство существующих нейронных сетей в зависимости от архитектуры могут быть разбиты на три большие категории:

– многослойные с прямой передачей информации, в которых отдельные нейроны объединены в слои, между которыми информация передается в одном направлении от входа к выходу (рис. 2.9, а);

– рекуррентные (с обратной связью), в которых сигналы с выхода могут подаваться на вход или внутренние слои сети (рис. 2.9, б);

– клеточные, в которых каждый нейрон связан только со своими соседями (рис. 2.9, в).

В общем случае каждый нейрон данного слоя однонаправлено связан со всеми нейронами последующего слоя. Эти связи организованы через синаптические веса, которые действуют как усилители в соответствующих каналах.

Например, в трехслойном персептроне, схема которого приведена на рис. 2.9, а (сигналы смещения θ_j для простоты изображения опущены), нейроны сгруппированы в последовательные слои: нулевой, первый, второй и третий. Нейроны нулевого слоя, иногда называемого входным, не производят никаких вычислений, а служат лишь для передачи вектора входных сигналов $x = (x_1, x_2, ..., x_n)^{\mathrm{T}}$ на нейроны слоя, называемого первым скрытым слоем.

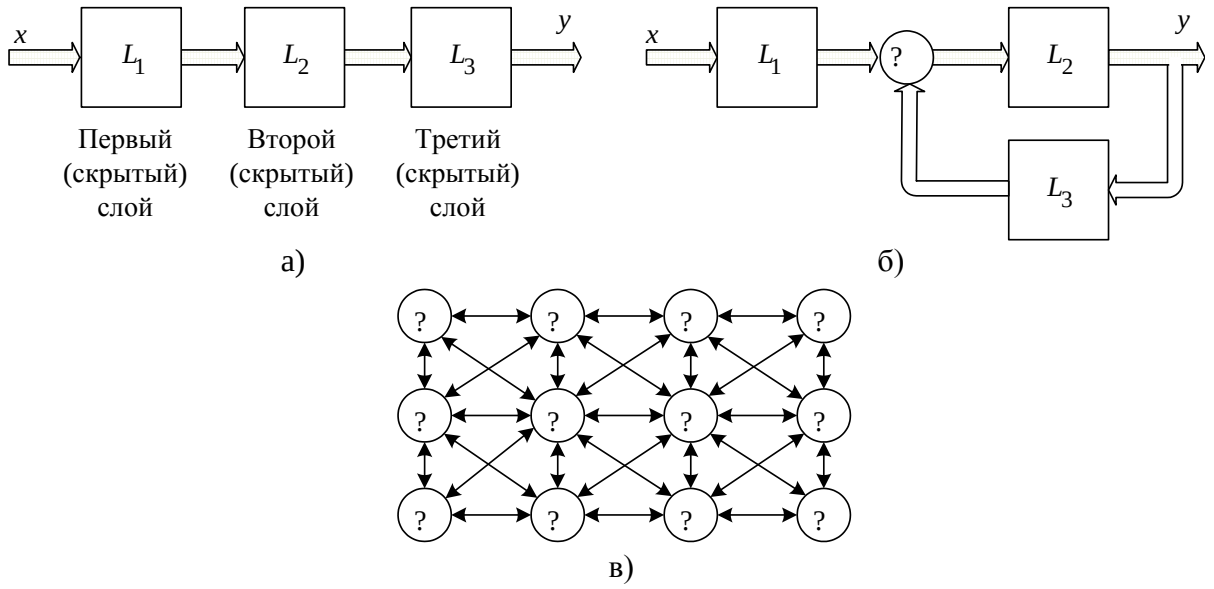

Рис. 2.9. Архитектуры нейронных сетей

Сигналы с первого слоя поступают на второй скрытый слой и далее на третий. Последний третий слой является выходным, и в нем формируется вектор выходных сигналов y НС в целом. Теоретически количество слоев может быть произвольно большим, однако рядом авторов было доказано [40]–[42], что для решения сколь угодно сложных задач автоматизированного управления достаточно лишь трех слоев, т. е. трехслойный персептрон является универсальным аппроксиматором.

Заметим, что в многослойных персептронах связи между нейронами внутри слоя или от высшего к низшему слою запрещены, при этом количество нейронов, а соответственно, и синаптических весов в каждом слое различно. Каждый нейрон (как правило, элементарный персептрон Розенблатта) имеет один выход и множество входов, которые, в свою очередь, являются выходами всех нейронов предыдущего слоя.

Обозначив внутренний сигнал j-го нейрона 5-го слоя, содержащего n_s нейронов, как $u_i^{[s]}$ (s = 1, 2, 3; j = 1, 2, ..., n_s), несложно представить его в виде взвешенной суммы выходов предыдущего слоя:

$$u_i^{[s]} = \sum_{i=0}^{n_{s-1}} w_{ji}^{[s]} \cdot o_i^{[s-1]} ,$$

где $w_{ji}^{[s]}$ – синаптические веса, связывающие каждый нейрон s-го слоя со всеми нейронами предыдущего слоя так, что $o^{[s-1]} = x^{[s]}$, $o^{[0]} = x_1$, $o^{[3]} = y_t$.

Тогда выход каждого нейрона есть нелинейное преобразование вида

$$o_j^{[s]} = \psi_j^{[s]} \cdot \left(u_j^{[s]}\right) = \psi_j^{[s]} \left(\sum_{i=0}^{n_{s-1}} w_{ji}^{[s]} \cdot o_i^{[s-1]} \right),$$

где для сокращения записи величина смещения $\theta_j^{[s]}$ обозначается как синаптический вес $w_{j0}^{[s]} = \theta_j^{[s]}$, а $o_0^{[s]} \equiv 1$.

Используя векторно-матричную форму, можно записать преобразование, осуществляемое каждым слоем, в компактном виде

$$o_j^{[s]} = \psi^{[s]} \cdot \left(W^{[s]} \cdot x^{[s]}\right) = \psi^{[s]}\left(W^{[s]} \cdot o^{[s-1]}\right),$$

где $o^{[s]} = \left(o_0^{[s]}, o_1^{[s]}, o_2^{[s]}, ..., o_{n_s}^{[s]}\right)^{\mathrm{T}} - \left((n_s + 1) \times 1\right)$ – выходной вектор сигналов; $x^{[s]} = \left(x_0^{[s]}, x_1^{[s]}, ..., x_{n_{s-1}}^{[s]}\right)^{\mathrm{T}} = o^{[s-1]} - \left((n_{s-1} + 1) \times 1\right)$ – входной вектор;

$W^{[s]} - \left((n_s+1)\times(n_{s-1}+1)\right)$ – матрица синаптических весов; $\psi^{[s]}(\bullet)$ – диагональный нелинейный оператор, образованный активационными функциями всех нейронов $\psi_j^{[s]}$.

С учетом введенных обозначений уравнение трехслойного персептрона принимает вид

$$y = \psi(x) = \psi^{[3]}(W^{[3]}\psi^{[2]}(W^{[2]}\psi^{[1]}(W^{[2]}x))),$$

а сама НС может быть представлена в компактной форме, приведенной на рис. 2.10.

На сегодня многослойный персептрон – пожалуй, наиболее распространенная нейросеть, нашедшая применение во многих областях. Однако, для того чтобы обеспечить требуемое поведение НС в процессе преобразования входных сигналов в выходные, необходимо соответствующим образом организовать процедуру обучения.

Наконец, можно более подробно рассмотреть вопрос обучения НС, для начала – на примере персептрона, структура которого представлена на рис. 2.6.

Рассмотрим алгоритм обучения с учителем [15], [28].

1. Проинициализировать элементы весовой матрицы (обычно небольшими случайными значениями).

2. Подать на входы один из входных векторов, которые сеть должна научиться различать, и вычислить ее выход.

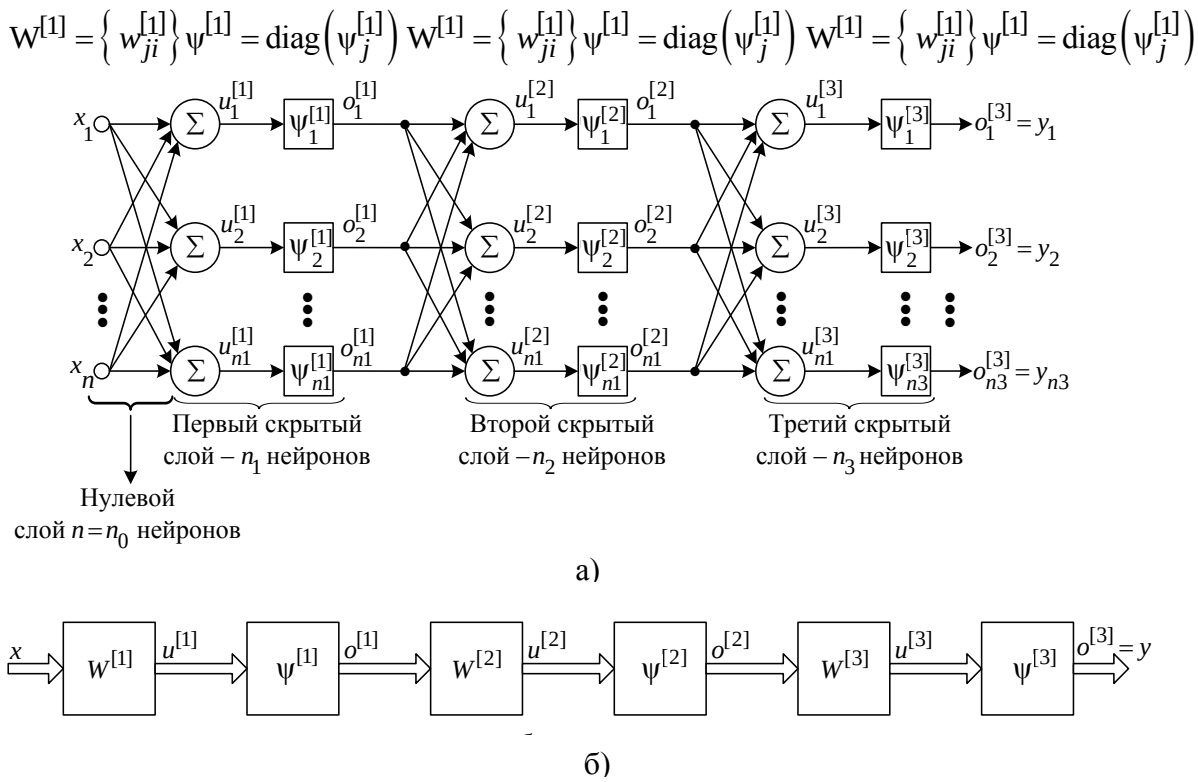

Рис. 2.10. Трехслойный персептрон

3. Если выход правильный, перейти на шаг 4. Иначе – вычислить разницу между идеальным и полученным значениями выхода:

$$\delta = Y_I - Y.$$

Модифицировать веса в соответствии с формулой

$$w_{ij}(t+1) = w_{ij}(t) + \nu \cdot \delta \cdot x_i,$$

где t и $(t + 1)$ – номера, соответственно, текущей и следующей итераций; $0 < \nu < 1$ – коэффициент скорости обучения; i – номер входа; j – номер нейрона в слое.

Очевидно, что если $Y_I > Y$, весовые коэффициенты будут увеличены и тем самым уменьшат ошибку. В противном случае они будут уменьшены, и Y тоже уменьшится, приближаясь к Y_I.

4. Цикл с шага 2, пока сеть не перестанет ошибаться.

На втором шаге на разных итерациях поочередно в случайном порядке предъявляются все возможные входные векторы. К сожалению, нельзя заранее определить число итераций, которые потребуется выполнить, а в некоторых случаях и гарантировать полный успех.

2.4. Алгоритм обратного распространения

В задаче обучения многослойных сетей возникают трудности с настройкой весов скрытых слоев, которые могут быть преодолены с помощью специальной процедуры, получившей название алгоритма обратного распространения ошибок или обобщенного дельта-правила.

Среди различных структур НС одной из наиболее известных является многослойная структура, в которой каждый нейрон произвольного слоя связан со всеми аксонами нейронов предыдущего слоя или, в случае первого слоя, со всеми входами НС. Такие НС называются полносвязными.

Когда в сети только один слой, алгоритм ее обучения с учителем довольно очевиден, так как правильные выходные состояния нейронов единственного слоя заведомо известны, и подстройка синаптических связей идет в направлении, минимизирующем ошибку на выходе сети. По этому принципу строится, например, алгоритм обучения однослойного персептрона. В многослойных же сетях оптимальные выходные значения нейронов всех слоев, кроме последнего, как правило, неизвестны, и двух- или более слойный персептрон уже невозможно обучить, руководствуясь только значениями ошибок на выходах НС.

Один из вариантов решения этой проблемы – разработка наборов выходных сигналов, соответствующих входным, для каждого слоя НС, что, конечно, является очень трудоемкой операцией и не всегда осуществимо.

Второй вариант – динамическая подстройка весовых коэффициентов синапсов, в ходе которой выбираются, как правило, наиболее слабые связи и изменяются на малую величину в ту или иную сторону, а сохраняются только те изменения, которые повлекли уменьшение ошибки на выходе всей сети. Очевидно, что данный метод «тыка», несмотря на кажущуюся простоту, требует громоздких рутинных вычислений.

И, наконец, третий, более приемлемый вариант – распространение сигналов ошибки от выходов НС к ее входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы. Этот алгоритм обучения НС получил название процедуры обратного распространения. Название «сети обратного распространения» (back propagation) они получили из-за используемого алгоритма обучения, в котором ошибка распространяется от выходного слоя к входному, т. е. в направлении, противоположном направлению распространения сигнала при нормальном функционировании сети.

Нейронная сеть обратного распространения состоит из нескольких слоев нейронов, причем каждый нейрон слоя i связан с каждым нейроном слоя (i + 1), т. е. речь идет о полносвязной НС.

Данный метод обучения многослойной нейронной сети называется обобщенным дельта-правилом или правилом error backpropagation (обратного распространения ошибки). Это ознаменовало возрождение интереса к нейронным сетям, который стал угасать в начале 70-х гг. Позже было обнаружено, что Паркер опубликовал подобные результаты в 1982 г., а Вербос выполнил такую работу в 1984 г. Однако такова природа науки, что ученые, работающие независимо друг от друга, не могут использовать все то прогрессивное, что есть в других областях, и поэтому часто случается повторение уже достигнутого. Однако статья Руммельхарта и др., опубликованная в журнале «Nature» (1986), является до сих пор наиболее цитируемой в этой области.

Обучение сети начинается с предъявления образа и вычисления соответствующей реакции. Сравнение с желаемой реакцией дает возможность изменять веса связей таким образом, чтобы сеть на следующем шаге могла выдавать более точный результат. Обучающее правило обеспечивает настройку весов связей. Информация о выходах сети является исходной для нейронов предыдущих слоев. Эти нейроны могут настраивать веса своих связей для уменьшения погрешности на следующем шаге.

Когда ненастроенной сети предъявляется входной образ, она выдает некоторый случайный выход. Функция ошибки представляет собой разность между текущим выходом сети и идеальным выходом, который необходимо получить. Для успешного обучения сети требуется приблизить выход сети к желаемому выходу, т. е. последовательно уменьшать величину функции ошибки. Это достигается настройкой межнейронных связей. Обобщенное дельта-правило обучает сеть путем вычисления функции ошибки

для заданного входа с последующим ее обратным распространением от каждого слоя к предыдущему. Каждый нейрон в сети имеет свои веса, которые настраиваются, чтобы уменьшить величину функции ошибки. Для нейронов выходного слоя известны их фактические и желаемые значения выходов. Поэтому настройка весов связей для таких нейронов является относительно простой. Однако для нейронов предыдущих слоев настройка не столь очевидна. Интуитивно ясно, что нейроны внутренних слоев, которые связаны с выходами, имеющими большую погрешность, должны изменять свои веса значительно сильнее, чем нейроны, соединенные с почти корректными выходами. Другими словами, веса данного нейрона должны изменяться прямо пропорционально ошибке тех нейронов, с которыми данный нейрон связан. Вот почему обратное распространение этих ошибок через сеть позволяет корректно настраивать веса связей между всеми слоями. В этом случае величина функции ошибки уменьшается и сеть обучается.

В общем случае задача обучения НС сводится к нахождению некой функциональной зависимости $Y = F(X)$, где X – входной, а Y – выходной векторы.

Согласно методу наименьших квадратов, минимизируемой целевой функцией ошибки НС является величина

$$E(w) = \frac{1}{2} \sum_{j,p} (y_{j,p}^{(N)} - d_{j,p})^2,$$

где $y_{j,p}^{(N)}$ – реальное выходное состояние нейрона j-го выходного слоя N нейронной сети при подаче на ее входы p-го образа; $d_{j,p}$ – идеальное (желаемое) выходное состояние этого нейрона.

Суммирование осуществляется по всем нейронам выходного слоя и по всем обрабатываемым сетью образам. Минимизация ведется методом градиентного спуска, что означает подстройку весовых коэффициентов следующим образом:

$$\Delta w_{ij}^{(n)} = -\eta \frac{\partial E}{\partial w_{ij}}. \qquad (2.2)$$

Здесь w_{ij} – весовой коэффициент синаптической связи, соединяющей i-й нейрон слоя $(n - 1)$ с j-м нейроном слоя n; $0 < \eta < 1$ – коэффициент скорости обучения.

Как показано в [8],

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \frac{\mathrm{d}y_j}{\mathrm{d}s_j} \frac{\partial s_j}{\partial w_{ij}}. \qquad (2.3)$$

Здесь под y_j, как и раньше, подразумевается выход j-го нейрона, а под s_j – взвешенная сумма его входных сигналов, т. е. аргумент активационной

функции. Так как множитель $\frac{\mathrm{d}y_j}{\mathrm{d}s_j}$ является производной этой функции по ее аргументу, из этого следует, что производная активационной функции должна быть определена на всей оси абсцисс. В связи с этим функция единичного скачка и прочие активационные функции с неоднородностями не подходят для рассматриваемых НС. В них применяются такие гладкие функции, как гиперболический тангенс или классический сигмоид с экспонентой. В случае гиперболического тангенса

$$\frac{\mathrm{d}\,y}{\mathrm{d}\,s} = 1 - s^2.$$

Третий множитель $\frac{\partial s_j}{\partial w_{ij}}$, очевидно, равен выходу нейрона предыдущего слоя $y_i^{(n-1)}$.

Первый множитель в (2.3) легко раскладывается следующим образом [8]:

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \frac{\mathrm{d}y_k}{\mathrm{d}s_k} \frac{\partial s_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \frac{\mathrm{d}y_k}{\mathrm{d}s_k} w_{jk}^{(n+1)}.$$

Здесь суммирование по k выполняется среди нейронов слоя (n + 1).

Введя новую переменную

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \frac{\mathrm{d}y_j}{\mathrm{d}s_j},$$

получим рекурсивную формулу для расчетов величин $\delta_j^{(n)}$ слоя n из величин $\delta_k^{(n+1)}$ более старшего слоя (n + 1):

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} w_{jk}^{(n+1)} \right] \frac{\mathrm{d}y_j}{\mathrm{d}s_j}. \qquad (2.4)$$

Для выходного же слоя

$$\delta_l^{(N)} = (y_l^{(N)} - d_l) \frac{\mathrm{d}y_l}{\mathrm{d}s_l}. \qquad (2.5)$$

Теперь можно записать (2.2) в раскрытом виде:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot y_i^{(n-1)}. \qquad (2.6)$$

Иногда для придания процессу коррекции весов некоторой инерционности, сглаживающей резкие скачки при перемещении по поверхности целевой функции, (2.6) дополняется значением изменения веса на предыдущей итерации:

$$\Delta w_{ij}^{(n)}(t) = -\eta\left(\mu \cdot \Delta w_{ij}^{(n)}(t-1) + (1-\mu)\delta_j^{(n)} y_i^{(n-1)}\right), \qquad (2.7)$$

где μ – коэффициент инерционности; t – номер текущей итерации.

Таким образом, полный алгоритм обучения НС с помощью процедуры обратного распространения строится следующим образом.

1. Подать на входы сети один из возможных образов и в режиме обычного функционирования НС, когда сигналы распространяются от входов к выходам, рассчитать значения последних. Напомним, что

$$s_j^{(n)} = \sum_{i=0}^{M} y_i^{(n-1)} \cdot w_{ij}^{(n)},$$

где M – число нейронов в слое (n – 1) с учетом нейрона с постоянным выходным состоянием +1, задающего смещение; $y_i^{(n-1)} = x_{ij}^{(n)}$ – i-й вход j-го нейрона слоя n; $y_j^{(n)} = f\left(s_j^{(n)}\right)$, где $f()$ – сигмоид $y_q^{(0)} = I_q$, где I_q – q-я компонента вектора входного образа.

2. Рассчитать $\delta^{(N)}$ для выходного слоя по формуле (2.4).

Рассчитать по формуле (2.6) или (2.7) изменения весов $\Delta w^{(N)}$ слоя N.

3. Рассчитать по формулам (2.4) и (2.5) (или (2.4) и (2.6)), соответственно, $\delta^{(n)}$ и $\Delta w^{(n)}$ для всех остальных слоев, $n = N - 1, ..., 1$.

4. Скорректировать все веса в НС

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t). \qquad (2.8)$$

5. Если ошибка сети существенна, перейти на шаг 1. В противном случае – конец.

Сети на шаге 1 попеременно в случайном порядке предъявляются все тренировочные образы, чтобы сеть, образно говоря, не забывала одни по мере запоминания других.

Из выражения (2.6) следует, что, когда выходное значение $y_i^{(n-1)}$ стремится к нулю, эффективность обучения заметно снижается. При двоичных входных векторах в среднем половина весовых коэффициентов не будет корректироваться, поэтому область возможных значений выходов нейронов [0, 1] желательно сдвинуть в пределы [–0,5; +0,5], что достигается простыми модификациями логистических функций. Например, сигмоид с экспонентой преобразуется к виду

$$f(x) = -0{,}5 + \frac{1}{1 + \mathrm{e}^{-\alpha \cdot x}}.$$

Теперь коснемся вопроса емкости НС, т. е. числа образов, предъявляемых на ее входы, которые сеть способна научиться распознавать. Для сетей с числом слоев больше двух он остается открытым. Для НС с двумя слоями, т. е. выходным и одним скрытым слоями, детерминистская емкость сети C_d оценивается как

$$\frac{N_w}{N_y} < C_d < \frac{N_w}{N_y}\log\frac{N_w}{N_y},$$

где N_w – число подстраиваемых весов; N_y – число нейронов в выходном слое.

Следует отметить, что данное выражение получено с учетом некоторых ограничений. Во-первых, число входов N_x и нейронов в скрытом слое N_h должно удовлетворять неравенству $N_x + N_y > N_y$. Во-вторых, $\frac{N_w}{N_y} > 1000$.

Как видно из формул (2.2)–(2.8), описывающих алгоритм функционирования и обучения НС, весь этот процесс может быть записан и затем запрограммирован в терминах и с применением операций матричной алгебры, что сделано, например, в [8]. На рис. 2.10, б показан трехслойный персептрон [8], обучаемый с помощью обратного распространения ошибок.

2.5. Нейронные сети Хопфилда и Хэмминга

Все введенные ранее искусственные нейроны описываются алгебраическими уравнениями и являются статическими моделями. Это означает, что фактор времени явно не влияет на поведение нейрона: выходная реакция в текущий k-й момент времени $y_j(k)$ определяется только входными сигналами в этот же момент времени $x_1(k), x_2(k), ..., x_n(k)$ и никак не зависит от прошлых состояний.

В задачах оптимизации и цифровой обработки сигналов широкое распространение получили динамические нейроны, описываемые разностными или дифференциальными уравнениями, поведение которых существенным образом определяется их предысторией. К ним относятся нейроны Хопфилда, Гроссберга, нейроны-осцилляторы и другие модели.

Простейшим динамическим нейроном является нейрон Хопфилда. Среди различных конфигураций искуственных нейронных сетей (НС) встречаются такие, при классификации которых по принципу обучения, строго говоря, не подходят ни обучение с учителем [14], ни обучение без учителя [15]. В таких сетях весовые коэффициенты синапсов рассчитываются только однажды перед началом функционирования сети на основе информации об обрабатываемых данных, и все обучение сети сводится именно к этому расчету. С одной стороны, предъявление априорной информации можно

расценивать, как помощь учителя, но с другой – сеть фактически просто запоминает образцы до того, как на ее вход поступают реальные данные, и не может изменять свое поведение, поэтому говорить о звене обратной связи с «миром» (учителем) не приходится.

Из сетей с подобной логикой работы наиболее известны сети Хопфилда и Хэмминга, которые обычно используются для организации ассоциативной памяти.

Структурная схема сети Хопфилда приведена на рис. 2.11. Она состоит из единственного слоя нейронов, число которых является одновременно числом входов и выходов сети. Каждый нейрон связан синапсами со всеми остальными нейронами, а также имеет один входной синапс, через который осуществляется ввод сигнала. Выходные сигналы, как обычно, образуются на аксонах.

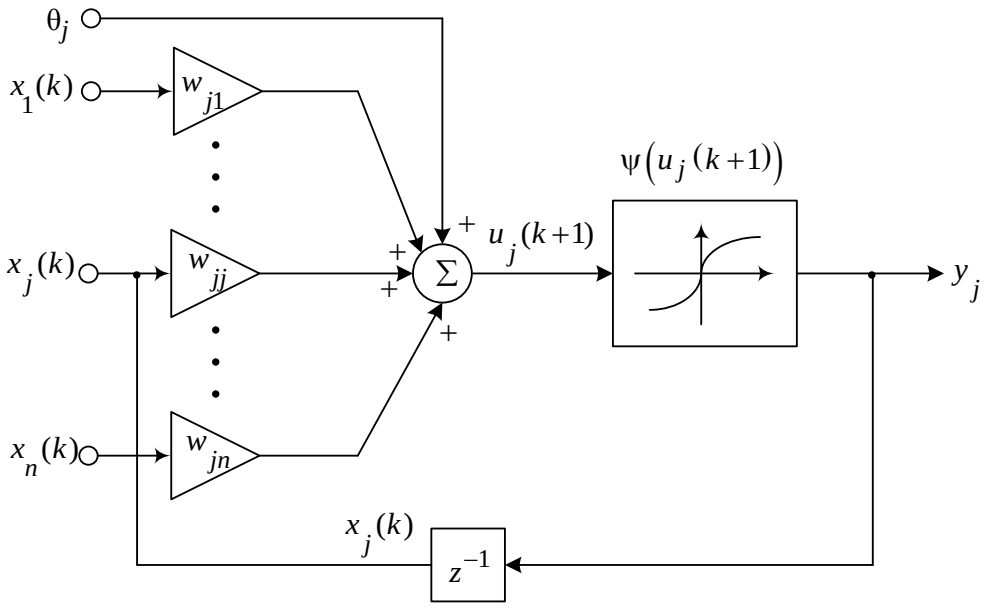

Рис. 2.11. Структурная схема сети Хопфилда

Задача, решаемая данной сетью в качестве ассоциативной памяти, как правило, формулируется следующим образом.

Известен некоторый набор двоичных сигналов (данных, описывающих некие объекты или характеристики процессов), которые считаются образцовыми. Сеть должна уметь из произвольного неидеального сигнала, поданного на ее вход, выделить («вспомнить» по частичной информации) соответствующий образец (если такой есть) или «дать заключение» о том, что входные данные не соответствуют ни одному из образцов. В общем случае любой сигнал может быть описан вектором $\mathbf{X} = \{x_i : i = 0, ..., n-1\}$, n – число нейронов в сети и размерность входных и выходных векторов. Каждый элемент x_i равен либо +1, либо –1. Обозначим вектор, описывающий k-й образец, через $\mathbf{X}^k$, а его компоненты, соответственно, $-x_i^k$; $k = 0, ..., m-1$, m – число образцов.

Когда сеть распознает (или «вспомнит») какой-либо образец на основе предъявленных ей данных, ее выходы будут содержать именно его, т. е. $\mathbf{Y} = \mathbf{X}^k$, где $\mathbf{Y} = \{y_i; i = 0, ..., n-1\}$ – вектор выходных значений сети. В противном случае, выходной вектор не совпадет ни с одним образцовым.

На стадии инициализации сети весовые коэффициенты синапсов устанавливаются следующим образом [28]:

$$w_{ij} = \begin{cases} \sum_{k=0}^{m-1} x_i^k x_j^k, \; i \neq j; \\ 0, \qquad i = j, \end{cases}$$

где i и j – индексы, соответственно, предсинаптического и постсинаптического нейронов; x_i^k, x_j^k – i-й и j-й элементы вектора k-го образца.

Алгоритм функционирования сети следующий (p – номер итерации).

1. На входы сети подается неизвестный сигнал. Фактически его ввод осуществляется непосредственной установкой значений аксонов:

$$y_i(0) = x_i, \; i = 0, ..., n-1,$$

поэтому обозначение на схеме сети входных синапсов в явном виде носит чисто условный характер. Ноль в скобках справа от y_i означает нулевую итерацию в цикле работы сети.

2. Рассчитывается новое состояние нейронов

$$s_j(p+1) = \sum_{i=0}^{n-1} w_{ij} y_i(p), \; j = 0, ..., n-1$$

и новые значения аксонов

$$y_j(p+1) = f\left[s_j(p+1)\right],$$

где f – активационная функция в виде скачка, приведенная на рис. 2.4.

3. Проверяется, изменились ли выходные значения аксонов за последнюю итерацию: если да – переход к п. 2, иначе (если выходы застабилизировались) – конец. При этом выходной вектор представляет собой образец, наилучшим образом сочетающийся с входными данными.

Как отмечалось ранее, иногда сеть не может провести распознавание и выдает на выходе несуществующий образ. Это связано с проблемой ограниченности возможностей сети. Для сети Хопфилда число запоминаемых образов m не должно превышать величины, примерно равной $0{,}15n$.

Когда нет необходимости, чтобы сеть в явном виде выдавала образец, т. е. достаточно, скажем, получать номер образца, ассоциативную память успешно реализует сеть Хэмминга. Данная сеть характеризуется, по сравнению с сетью Хопфилда, меньшими затратами на память и объемом вычислений, что становится очевидным из ее структуры (рис. 2.12).

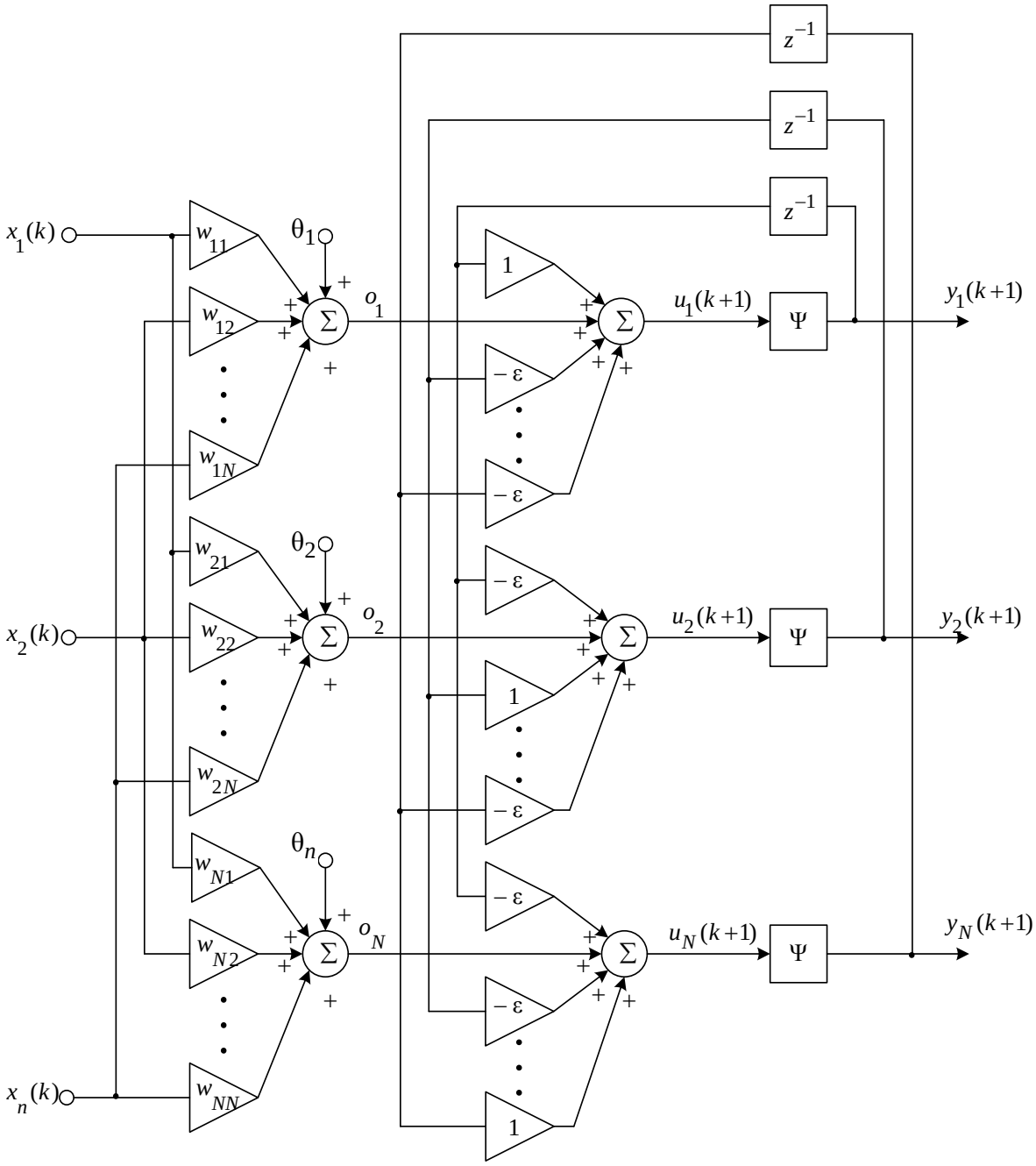

Рис. 2.12. Структурная схема сети Хэмминга

Сеть состоит из двух слоев. Первый и второй слои имеют по m нейронов, где m – число образцов. Нейроны первого слоя имеют по n синапсов, соединенных со входами сети (образующими фиктивный нулевой слой). Нейроны второго слоя связаны между собой ингибиторными (отрицательными обратными) синаптическими связями. Единственный синапс с положительной обратной связью для каждого нейрона соединен с его же аксоном.

Идея работы сети состоит в нахождении расстояния Хэмминга от тестируемого образца до всех образцов. Расстоянием Хэмминга называется число отличающихся битов в двух бинарных векторах. Сеть должна выбрать

образец с минимальным расстоянием Хэмминга до неизвестного входного сигнала, в результате чего будет активизирован только один выход сети, соответствующий этому образцу.

На стадии инициализации весовым коэффициентам первого слоя и порогу активационной функции присваиваются следующие значения:

$$w_{ik} = \frac{x_i^k}{2},\ i = 0, ..., n-1,\ k = 0, ..., m-1;$$
$$T_k = n/2,\ \ k = 0, ..., m-1.$$

где x_i^k – i-й элемент k-го образца.

Весовые коэффициенты тормозящих синапсов во втором слое берут равными некоторой величине $0 < \varepsilon < 1/m$. Синапс нейрона, связанный с его же аксоном, имеет вес +1.

Алгоритм функционирования сети Хэмминга следующий.

1. На входы сети подается неизвестный вектор $\mathbf{X} = \{x_i : i = 0, ..., n-1\}$, исходя из которого рассчитываются состояния нейронов первого слоя (верхний индекс в скобках указывает номер слоя):

$$y_j^{(1)} = s_j^{(1)} = \sum_{i=0}^{n-1} w_{ij} x_i + T_j,\ j = 0, ..., m-1.$$

После этого полученными значениями инициализируются значения аксонов второго слоя:

$$y_j^{(2)} = y_j^{(1)},\ j = 0, ..., m-1.$$

2. Вычисляются новые состояния нейронов второго слоя:

$$s_j^{(2)}(p+1) = y_j(p) - \varepsilon \sum_{k=0}^{m-1} y_k^{(2)}(p),\ k \neq j,\ j = 0, ..., m-1$$

и значения их аксонов:

$$y_j^{(2)}(p+1) = f\left[s_j^{(2)}(p+1)\right],\ j = 0, ..., m-1.$$

Активационная функция f имеет вид порога (рис. 2.4), причем величина F должна быть достаточно большой, чтобы любые возможные значения аргумента не приводили к насыщению.

3. Проверяется, изменились ли выходы нейронов второго слоя за последнюю итерацию: если да – переход к шагу 2, иначе – конец.

Из оценки алгоритма видно, что роль первого слоя весьма условна: воспользовавшись один раз на шаге 1 значениями его весовых коэффициентов, сеть больше не обращается к нему, поэтому первый слой может быть вообще исключен из сети (заменен на матрицу весовых коэффициентов).

2.6. Требования к алгоритму обучения

Обучение нейронных сетей можно представить как задачу оптимизации в тех случаях, когда удается оценить работу сети. Строится функция оценки. Простейший и самый распространенный пример оценки – сумма квадратов расстояний от выходных сигналов сети до их требуемых значений:

$$E = \frac{\sum (z - z^*)^2}{2},$$

где z и z^* – полученное и требуемое значения выходного сигнала.

Для обучения используется оценка, усредненная по примерам с известным ответом. Предлагается рассматривать обучение нейронных сетей как задачу оптимизации. Это означает, что весь мощный арсенал методов оптимизации может быть испытан для обучения. Существует, однако, ряд специфических ограничений. Они связаны с огромной размерностью задачи обучения. Число параметров может достигать 10 и даже более. Уже в простейших программных имитаторах на персональных компьютерах подбирается 10^3–10^4 параметров.

Из-за высокой размерности возникает два требования к алгоритму:

1) ограничение по памяти. Пусть n – число параметров. Если алгоритм требует затрат памяти порядка n^2, то он вряд ли применим для обучения. Вообще говоря, желательно иметь алгоритмы, которые требуют затрат памяти порядка K^n, K = const;

2) возможность параллельного выполнения наиболее трудоемких этапов алгоритма и желательно – нейронной сетью.

Еще два обстоятельства связаны с нейрокомпьютерной спецификой:

1) обученный нейрокомпьютер должен с приемлемой точностью решать все тестовые задачи. Поэтому задача обучения становится по существу многокритериальной задачей оптимизации: надо найти точку общего минимума большого числа функций. Обучение нейрокомпьютера исходит из гипотезы о существовании такой точки. Основания гипотезы – очень большое число переменных и сходство между функциями. Само понятие «сходство» здесь трудно формализовать, но опыт показывает, что предположение о существовании общего минимума или, точнее, точек, где значения всех оценок мало отличаются от минимальных, часто оправдывается;

2) обученный нейрокомпьютер должен иметь возможность приобретать новые навыки без утраты старых. Возможно более слабое требование: новые навыки могут сопровождаться потерей точности в старых навыках, но эта потеря не должна быть особо существенной, а качественные изменения должны быть исключены. Это означает, что в достаточно большой

окрестности найденной точки общего минимума оценок значения этих функций незначительно отличаются от минимальных. Мало того, что должна быть найдена точка общего минимума, так она еще должна лежать в достаточно широкой низменности, где значения всех минимизируемых функций близки к минимуму. Для решения этой задачи нужны специальные средства.

2.7. Выбор эталонов для обучения нейросети

Эталонные технологические режимы (ТР) необходимы для обучения нейросети. Перечислим основные положения, которые нужно соблюдать при их выборе.

1. Технологические данные для обучения должны быть представительными (репрезентативными) как в совокупности, так и каждое в отдельности.

2. Будущее не похоже на прошлое. Данные, собранные за прошлый период, могут не действовать в будущем. Это важно при определении размера «временного окна» и частоты обновления технологической информации.

3. Данные должны быть типовыми для рассматриваемого ТР. Нейросеть не может правильно решать задачи управления для резко отличающихся ТР, данные которых не входили в диапазон обучающего множества.

4. Набор данных должен быть сбалансирован. Необходимо стремиться к тому, чтобы информация для различных ТР в обучающей выборке была представлена равномерно, без выбросов.

В производственных условиях перечисленные требования практически реализуются в правильном выборе и обработке целевой функции, применяемой, например, при решении задачи оптимального управления. В этом случае ТР, соответствующий минимуму материальных, энергетических и информационных затрат, выбирается в качестве эталона обучения. Кроме того, вид целевой функции для выбора эталонов должен отвечать следующим требованиям.

1. *Согласованность*. Целевая функция должна строиться на известных связях между входными и выходными параметрами, приведенных к одной координате объекта и моменту времени.

2. *Однозначность*. Одна целевая функция должна отражать изменение только одного критерия и иметь четкий физический смысл.

3. *Экстремальность*. У целевой функции должна быть ярко выраженная экстремальная область, распознаваемая экспертом.

4. *Ориентированность* на конкретный критерий, определяющий экономическую эффективность ТР.

5. *Соответствие*. Характер целевой функции и количественные характеристики ТР должны обеспечивать успешную оптимизацию технологического процесса.

6. *Управляемость*. В качестве аргументов должны фигурировать технологические параметры, поддающиеся управлению с помощью нейросети.

Общий алгоритм действий специалиста для определения необходимых эталонов обучения нейросети приведен на рис. 2.13.

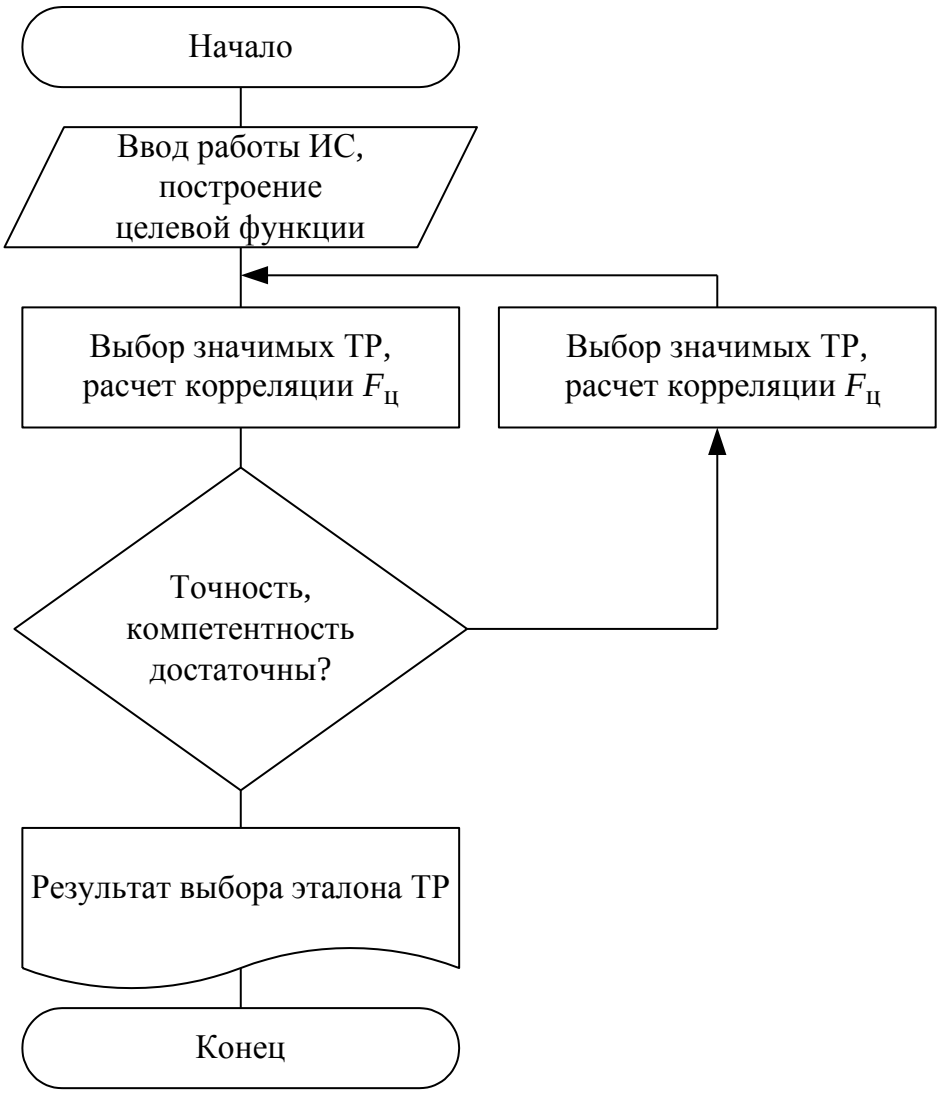

Рис. 2.13. Алгоритм выбора эталонов для обучения нейросети

Для удовлетворения перечисленных требований целевую функцию $F_ц$ располагают на графике в координатах X–Y (например, производительность – расход электроэнергии), задавая последовательный ряд текущих ТР с условными номерами $N1$, ..., $N12$. Далее фиксируют точку оптимума с координатами $X_{opt}Y_{opt}$ и выбирают $ТР_{opt}$ наиболее подходящий, по мнению эксперта, в данном случае к оптимальным условиям.

В реальности часто приходится наблюдать большой разброс местоположений текущих ТР на графике относительно кривой целевой функции $F_ц$. Фактически определение положения $ТР_{opt}$ также представляет собой задачу распознавания объекта, каковым является $ТР_{opt}$, на фоне изображения траектории технологического процесса с производственными шумами.

При выборе $ТР_{opt}$ в качестве эталона для обучения еще не созданной нейросети часто прибегают к производственному опыту и инженерной интуиции специалистов. Так, опытный производственник может уверенно выбрать не только $ТР_{opt}$, но и указать пределы изменения *X*, *Y*, соответствующие их оптимальному, низкому и высокому значениям.

Построение кривой $F_ц$ в условиях производственного шума необходимо дополнять корреляционной обработкой технологической информации. При выборе эталонов следует руководствоваться коэффициентом корреляции *r* для $F_ц$, значения которого должны быть не ниже граничных значений.

Если требуется несколько эталонов, их выбирают в зависимости от сочетания технологических факторов, от которых зависят эти режимы работы.

Количество факторов зависит от объема базы знаний в системе автоматизированного контроля технологического процесса, а количество различаемых лингвистических переменных определяется опытом и степенью компетентности специалистов. Таким образом, уже на стадии определения эталонов в обучение нейросети закладывается коллективный опыт ведения производства обслуживающим персоналом.

2.8. Повышение эффективности обучения НС обратного распространения

Простейший метод градиентного спуска очень неэффективен в случае, когда производные по различным весам сильно отличаются. Это соответствует ситуации, когда значение функции *S* для некоторых нейронов близко по модулю к 1 или когда модуль некоторых весов много больше 1. В этом случае для плавного уменьшения ошибки надо выбирать очень маленькую скорость обучения, но при этом обучение может занять непозволительно много времени.

Простейшим методом усовершенствования градиентного спуска является введение момента *m*, когда влияние градиента на изменение весов изменяется со временем. Тогда формула (2.8) примет вид

$$w_{ij}^{(n)}(t) = -\eta \cdot \delta_j^{(n)} \cdot x_i^n + \mu \cdot \Delta w_{ij}^{(n)}(t-1).$$

Дополнительным преимуществом от введения момента является способность алгоритма преодолевать мелкие локальные минимумы.

Эволюционные алгоритмы обучения. Алгоритмы обучения, основанные на градиентных и ньютоновских процедурах оптимизации, реализуют так называемый регулярный подход, в рамках которого вычисление на каждом шаге синаптических весов осуществляется на основе достаточно строго формализованных правил. Однако, как отмечалось в [8], регулярные процедуры хороши, но годятся не для всякой целевой функции и не для всякой архитектуры сети. В ситуациях, когда целевая функция либо выходной сигнал сети или являются недифференцируемыми или многоэкстре-

мальными, или просто вычисление производных по каким-либо причинам нежелательно или затруднительно, на первый план выходят эволюционные алгоритмы обучения, развивающиеся в трех относительно независимых направлениях: случайный поиск, эволюционное планирование и генетическое программирование.

В общем случае обучение на основе эволюционного подхода основывается на элементарном методе проб и ошибок, когда решение ищется случайно и при удаче принимается, а при неудаче отвергается с тем, чтобы немедленно снова обратиться к случайному выбору как к источнику возможностей. Такая случайная основа поиска решений опирается на уверенность, что именно случайность содержит в себе все возможности, в том числе и наилучшее решение. В общем случае метод проб и ошибок является универсальным подходом (хотя и не всегда достаточно «быстрым») к решению задач в условиях дефицита априорной и текущей информации, к которым в полной мере могут быть отнесены задачи обучения нейронных сетей.

Одним из проявлений эффективной реализации проб и ошибок в природе являются законы эволюции, представляющие собой итеративный процесс, который включает воспроизведение в условиях мутаций, естественный отбор, случайные рекомбинации, индивидуальное обучение [44]. Наиболее ярко элемент случайности в пробах и ошибках проявляется в виде мутаций, которые имеют стохастический характер. Эти мутации могут либо улучшить, либо ухудшить приспособительные свойства биологической особи. В первом случае эта особь имеет больше шансов на выживание и, следовательно, на закрепление полученной мутации в потомстве.

Использование идей биологической эволюции в технике и породило то, что сегодня называется эволюционными алгоритмами, которые наряду с нейронными сетями и фаззи-системами сформировали новое научное направление – вычислительный интеллект.

Оценка эффективности способов обучения НС. В качестве критерия эффективности способов обучения НС можно принять выражение [25]

$$K = \frac{\text{Затраты на поиск}}{\text{Положительный эффект}},$$

где затраты определяются числом проб на один цикл поиска, а эффект измеряется величиной смещения к точке оптимума.

На рис 2.14 [8] (где ρ – расстояние до экстремума целевой функции, измеренное в рабочих шагах поиска n) показана граница изменения потерь на поиск между алгоритмами случайного поиска и регулярными градиентными.

Потери на случайный поиск имеют параболический характер, т. е. изменяются пропорционально $4n$. Таким образом, случайный поиск имеет преимущество перед градиентным по быстродействию при значительном количестве настраиваемых параметров n, причем это преимущество тем более, чем больше n.

Заштрихована область, где быстродействие случайного поиска превышает быстродействие градиентного по критерию потерь на поиск.

Таким образом, случайный поиск более эффективен при оптимизации объектов с большим числом варьируемых параметров (НС именно такими объектами и являются) вдали от экстремума. В ситуациях малой размерности ($n < 3$) и в окрестности экстремума ($p < 4$) целесообразнее использовать градиентные методы.

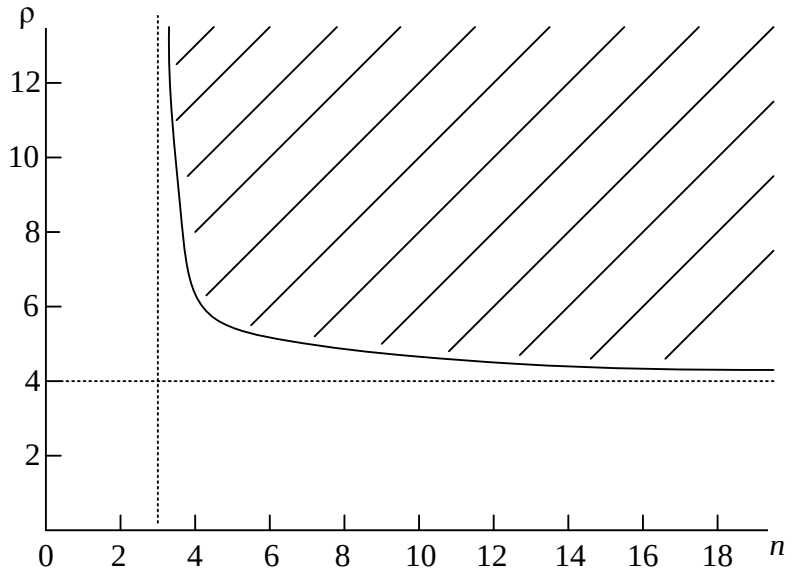

Рис. 2.14. График потерь на поиск

Свойства случайного поиска можно существенно улучшить, обеспечив конкретным алгоритмам способность к самообучению и самосовершенствованию в процессе работы. Таким образом, возникает самообучение алгоритмов обучения. Это приводит к тому, что в каждом новом состоянии система обучения либо начинает поиск сначала, либо «доверяет» предыдущему результату и пробует продвинуться к оптимуму по направлению, которое в предыдущем состоянии оказалось удачным.

Самообучение естественно вводится в алгоритмы случайного поиска в форме перестройки его вероятностных характеристик, т. е. в целенаправленном воздействии на вектор **Z**, который перестает быть равновероятным и приобретает некоторые преимущественные направления, например, в сторону предыдущего наилучшего шага.

Обучение алгоритма обучения, как правило, начинается в обстановке равновероятного поиска. В процессе накопления опыта работы система обучения должна приобретать «представление» о решаемой задаче в виде некоторых вероятностных гипотез о наилучшем направлении движения в пространстве настраиваемых параметров. По мере накопления опыта дисперсность этих гипотез должна уменьшаться, и в пределе система обучения должна «знать» вектор наилучшего направления.

Таким образом, процесс самообучения стремится «задетерминировать» поиск в наилучшем направлении, т. е. таким образом перестроить его вероятностные характеристики, чтобы процесс настройки НС стал неслучайным. С другой стороны, алгоритм самообучения должен обладать способностью переучиваться, если ситуация каким-либо образом изменилась.

Введение самообучения не должно исключать применение алгоритмов случайного поиска в «чистом виде», т. е. оно должно затрагивать только вероятностные свойства выбора направления рабочего шага, но не менять структуру алгоритма. Это дает возможность комбинировать самообучение с различными алгоритмами, рассмотренными ранее.

Наиболее простым является самообучение методом исключения. Этот подход опирается на следующее очевидное положение.

Более эффективным является алгоритм покоординатного самообучения, во многом схожий с алгоритмом Гаусса – Зайделя. Недостаток этого алгоритма связан с его организацией, требующей в каждый момент движения только по одной переменной. Отсутствие промежуточных направлений снижает скорость сходимости. Более эффективным представляется подход, основанный на непрерывном самообучении.

Алгоритмы обучения на основе случайного поиска решают ту же задачу, что и алгоритмы, в основе которых лежат процедуры градиентной и ньютоновской оптимизаций. Вместе с тем существует широкий класс задач, где детерминированные методы не работают вообще – это задачи глобальной оптимизации, когда целевая функция имеет множество экстремумов. Задача отыскания глобального экстремума функции многих переменных является проблемой значительно более сложной и трудоемкой, чем определение локального экстремума. Дело в том, что многоэкстремальная функция почти не дает возможности судить о поведении критерия качества по нескольким наблюдениям, что возможно при унимодальности. Естественно, что число наблюдений при поиске глобального экстремума должно быть значительно большим.

Простейшей процедурой поиска глобального экстремума является так называемый блуждающий глобальный поиск.

На рис. 2.15 для сравнения показаны три схемы обучения при различных комбинациях немедленной реакции и самообучения.

На рис. 2.15, а показан случайный поиск без самообучения, который работает только с учетом немедленной реакции. В этом случае вероятностные характеристики случайного шага ζ неизменны.

Рис. 2.15, б отражает применение самообучения совместно с алгоритмом поиска. Здесь вероятностные характеристики случайных шагов перестраиваются соответствующим образом по каналу обратной связи при одновременной работе алгоритма поиска.

Рис. 2.15, в соответствует обучению сети только на основе алгоритма самообучения, когда немедленная реакция исключена.

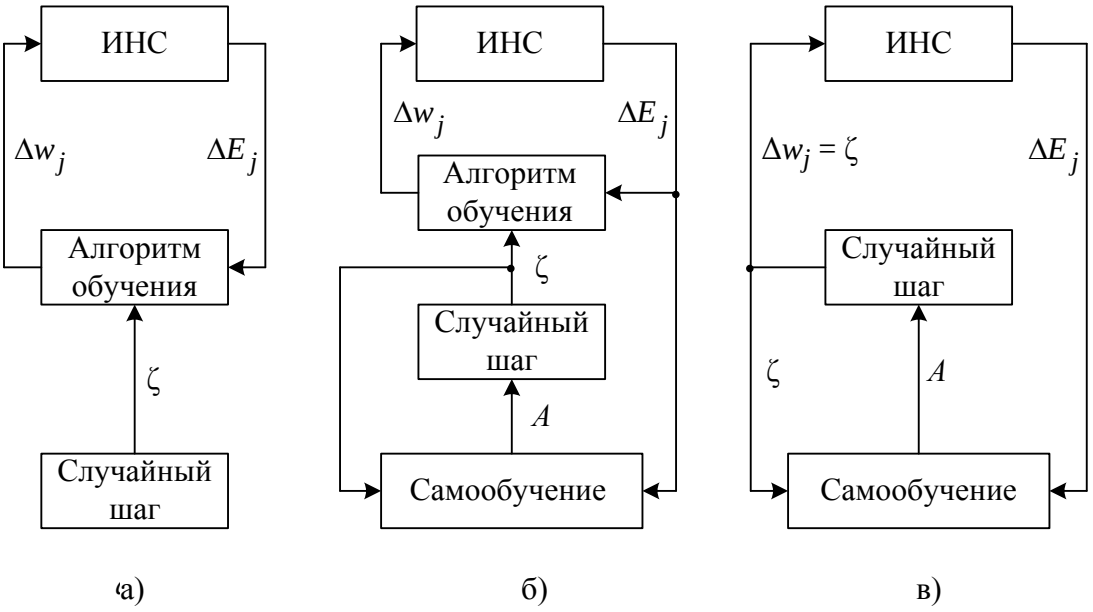

Рис. 2.15. Схемы обучения на основе случайного поиска

В этом случае настройка сети осуществляется только за счет перестройки вероятностных характеристик поиска.

Список литературы

1. Анализ и синтез систем управления. Теория. Методы. Примеры решения типовых задач с использованием персонального компьютера / Д. Х. Имаев, З. Ковальски, Н. Н. Кузьмин [и др.]. – Гданьск ; СПб. ; Сургут ; Томск, 1997.

2. Батищев, Д. А. Генетические алгоритмы решения экстремальных задач / Д. А. Батищев. – Воронеж : Изд-во ВГТУ, 1995.

3. Белов, Ю. В. Индустриальные средства проектирования и оценки эффективности программных систем, работающих в реальном времени / Ю. В. Белов, В. С. Проценко, В. В. Федоров, А. А. Хижняк // Вычисл. системы и вопр. Принятия решений. – М., 2001.

4. Бэлэни, Н. Будущее Web – за семантикой / Н. Бэлэни. – 2005, http://www.iso.ru/rus/document6130.phtml

5. Бэлэни, Н. Будущее Web – за семантикой/ Н. Бэлэни. http://www.iso.ru/rus/document6130. phtml#2

6. Божко, А. Н. Структурный синтез на элементах с ограниченной сочетаемостью / А. Н. Божко, А. Ч. Толпаров. http: // technonew. developer. stack.net / doc/ 44191.html.

7. Бодянский, Е. В. Искусственные нейронные сети: архитектуры, обучение, применения / Е. В. Бодянский, О. Г. Руденко. – Харьков : ХГПУ, 2008.

8. Бодянский, Е. В. Искусственные нейронные сети : архитектуры, обучение, применения / Е. В. Бодянский, О. Г. Руденко. – Харьков : Телетех, 2004. – 372 с.

9. Валидатор разметки Яндекса http://webmaster.yandex.ru/ microtest.xml.

10. Валидатор разметки Google http://www.google.com /webmasters/tools/richsnippets.

11. Гладков, Л. А. Генетические алгоритмы / Л. А. Гладков, В. В. Курейчик, В. М. Курейчик // под. ред. В. М. Курейчика. – 2-е изд., испр. и доп. – М. : Физматлит, 2010. – 386 с.

12. Джонс, М. Т. Программирование искусственного интеллекта в приложениях / М. Т. Джонс. – М. : ДМК Пресс, 2004.

13. Итоги науки и техники: физические и математические модели нейронных сетей. – Т. 1. – М. : Изд. ВИНИТИ, 1990.

14. Короткий, С. Нейронные сети: алгоритм обратного распространения / С. Короткий // http://ииклуб.рф/neur-2.html

15. Короткий, С. Нейронные сети: обучение без учителя / С. Короткий // http:// www.neuronos.ru / Нейронные-сети-обучение-без-учителя.

16. Кошелев, В. А. Некоторые задачи синтеза оптимальных модульных СОД РВ / В. А. Кошелев // Теоретические и прикладные задачи оптимизации. – М. : Наука, 1995.

17. Кротюк, Ю. М. Синтез оптимальных модульных СОД РВ с относительными приоритетами / Ю. М. Кротюк, В. А. Кошелев // Вопросы кибернетика. Автоматизация проектирования систем обработки данных. – М. : Научный совет комплексной проблеме «Кибернетика», 1985.

18. Круг, П. Г. Нейронные сети и нейрокомпьютеры: учеб. пособие по курсу «Микропроцессоры» / П. Г. Круг. – М. : Изд-во МЭИ, 2002. – 176 с.

19. Круглов, В. В. Искусственные нейронные сети. Теория и практика / В. В. Круглов, В. В. Борисов. – М. : Горячая линия – Телеком, 2001. – 382 с.

20. Кутузов, О. И. Моделирование телекоммуникационных сетей / О. И. Кутузов, Т. М. Татарникова ; СПбГУТ. – СПб., 2001. – 76 с.

21. Мюллер, И. Эвристические методы в инженерных разработках / И. Мюллер. – М. : Радио и связь, 1984.

22. Нестеренко, В. Д. Управление инфокоммуникационными сетями / В. Д. Нестеренко. – СПб. : Политехника, 2007. – 249 с.

23. Обогащенная структурная разметка для web-документов, W3C рабочий проект, 2011.

24. Осовский, С. Нейронные сети для обработки информации / С. Осовский // пер. с пол. И. Д. Рудинского. – М. : Финансы и статистика, 2002. – 344 с.

25. Растригин, Л. А. Системы экстремального управления / Л. А. Растригин. – М. : Наука, 1974. – 632 с.

26. Тауберер, Дж. Краткое введение в RDF / Дж. Тауберер. – 2006, http:// xmlhack.ru/texts/06/rdf-quickintro/rdf-quickintro.html

27. Тауберер, Дж. Краткое введение в RDF / Дж. Таубер. – http://rdfabout.com/ quickintro.xpd

28. Уоссермен, Ф. Нейрокомпьютерная техника / Ф. Уоссермен. – М. : Мир, 1992.

29. Хартьян, Д. Ю. Результаты корреляционного анализа совокупности параметров инфокоммуникационной инфраструктуры / Д. Ю. Хартьян, В. А. Шапцев // Матер. 8-й междунар. науч.-практ. конф. «Связь-2004: Проблемы функционирования информационных сетей». Новосибирск, 26–30 июня 2004 г. Новосибирск : ЗАО РИЦ Прайс Курьер, 2004. – Т. 2. – С. 298–306.

30. Химмельблау, Д. Прикладное нелинейное программирование / Д. Химмельблау. – М. : Мир, 1975.

31. Язык запросов SPARQL для RDF, рекомендация W3C. 2008,http://www.w3.org/ TR/rdf-sparql-query/

32. Artificial Neural Networks: Concepts and Theory // IEEE Computer Society Press. – 1992.

33. Creative Commons: About Licenses, http://creativecommons. org/about/ licenses/

34. Beckett, D. Turtle: Terse RDF Triple Language / D. Beckett, T. Berners-Lee. – January 2008. W3C Team Submission. http://www.w3.org /TeamSubmission/turtle/

35. Brickley, D. FOAF Vocabulary Specification 0.98 / D. Brickley, L. Miller. – August 2010, http://xmlns.com/foaf/spec/.

36. Brickley, D. RDF Vocabulary Description Language 1.0: RDF Schema / D. Brickley, V. G. Ramanathan. – http://www.w3.org/TR/2004/REC-rdf-schema-20040210

37. Dublin Core metadata initiative: Dublin Core metadata element set, http://purl.oclc.org/ docs/core/documents/rec-dces-19990702.htm

38. Dublin Core metadata initiative, http://purl.oclc.org/docs/core /documents/ rec-dces-19990702.htm

39. Google, Inc., Yahoo, Inc., Microsoft Corporation, 2011, http://schema.org/ docs/gs.

40. Hecht-Nielsen, R. Theory of the backpropagation neural network / R. Hecht-Nielsen // Int. Joint Conf. on Neural Networks. – Washington, DC, 1989. – № 1. – P. 593–606.

41. Hornik, K. Multilayer feedforward networks are universal approximators / K. Hornik, M. Stinchcombe, H. White // Neural Networks. – 1989. – № 2. – P. 359–366.

42. Hornik, K. Approximation capabilities of multiplayer feedforward networks / K. Hornik // Neural Networks. – 1991. – № 4. – P. 251–257.

43. Sporny, M. RDFa API Latest. W3C Working Draft / M. Sporny, B. Adriaan, M. Birbeck. – URL: http://www.w3.org/TR/RDFA-API/

44. Prokhorov, D. Adaptive Critic Designs / D. Prokhorov, D. Wunsch // IEEE Transactions on Neural Networks. – 1997. – Vol. 8, № 5. – P. 997–1007.

45. SPARQL Query Results XML Format http://www.w3.org/TR /rdf-sparql-XMLres/

46. The Open Graph Protocol. 2010, http://www.zencoder.pro /open-graph-protocol-opisanie-socialnih-media

47. http://www.w3.org/2001/sw/interest/

48. http://my-eventos.com/solution/ontoquad/

Белов Михаил Петрович
Филиппов Феликс Васильевич

ИНТЕЛЛЕКТУАЛИЗАЦИЯ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ

Часть 1

Учебное пособие

Редактор *Л. К. Паршина*

Компьютерная верстка *Н. А. Ефремовой*

План 2014–2015 гг., п. 43, а

Подписано к печати 10.11.2014
Объем 5,25 усл.-печ. л. Тираж 35 экз. Заказ 524

Редакционно-издательский центр СПбГУТ
191186 СПб., наб. р. Мойки, 61

Отпечатано в СПбГУТ