

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

1(173)
2011

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Талалай М. С., Трушин К. В. Обзор подходов к проектированию регулярных ИС 2
Бибило П. Н., Логинова И. П., Романов В. И., Черемисинова Л. Д. Система CLTT проектирования топологии функциональных блоков заказных цифровых СБИС 8

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

- Евгеньев Г. Б. Синергетическая методология интеграции знаний 15
Колесникова С. И. Особенности применения линейной свертки критериев в методе парных сравнений 24
Комаров Е. Г. Построение модели эталонного объекта и рейтинговых оценок на его основе 30

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Мандрикова О. В., Соловьев И. С. Вейвлет-технология анализа вариаций магнитного поля Земли 34
Герасимов Ю. Ю., Соколов А. П., Катаров В. К. Разработка системы оптимального проектирования сети лесовозных автомобильных дорог 39

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Джашитов В. Э., Панкратов В. М., Голиков А. В. Мультимедийные информационно-компьютерные технологии в научно-образовательных курсах лекций и экспресс-контроле знаний по точным наукам 44
Пузанков Д. В., Пантелеев М. Г., Жукова Н. А., Семенов В. С. Информационная среда проектирования компетентностных моделей на основе технологий семантического Web 49

Журнал в журнале

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- От главного редактора 56
Галушкин А. И. Нейрокомпьютеры в решении задач обеспечения информационной безопасности 58
Аведьян Э. Д., Пантюхин Д. В. Алгоритм нелинейного преобразования аргументов в нейронной сети СМАС 64
Скрибцов П. В., Казанцев П. А., Долгополов А. В. Виртуальные нейросетевые процессоры — метод кроссплатформенного ускорения расчета нейросетевых задач на современных многоядерных процессорах 73
Contents 79
Приложение. Малыхин В. М., Малыхина Г. Ф., Меркушева А. В. Концепция, методы, алгоритмы и нейросетевые структуры для восстановления формы сигналов при регистрации их смеси

Главный редактор
НОРЕНКОВ И. П.

Зам. гл. редактора
ФИЛИМОНОВ Н. Б.

Редакционная
коллегия:

АВДОШИН С. М.
АНТОНОВ Б. И.
БАТИЩЕВ Д. И.
БАРСКИЙ А. Б.
БОЖКО А. Н.
ВАСЕНИН В. А.
ГАЛУШКИН А. И.
ГЛОРИОЗОВ Е. Л.
ДОМРАЧЕВ В. Г.
ЗАГИДУЛЛИН Р. Ш.
ЗАРУБИН В. С.
ИВАННИКОВ А. Д.
ИСАЕНКО Р. О.
КОЛИН К. К.
КУЛАГИН В. П.
КУРЕЙЧИК В. М.
ЛЬВОВИЧ Я. Е.
МАЛЬЦЕВ П. П.
МЕДВЕДЕВ Н. В.
МИХАЙЛОВ Б. М.
НЕЧАЕВ В. В.
ПАВЛОВ В. В.
ПУЗАНКОВ Д. В.
РЯБОВ Г. Г.
СОКОЛОВ Б. В.
СТЕМПКОВСКИЙ А. Л.
УСКОВ В. Л.
ФОМИЧЕВ В. А.
ЧЕРМОШЕНЦЕВ С. Ф.
ШИЛОВ В. В.

Редакция:

БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://www.informika.ru/text/magaz/it/> или <http://novtex.ru/IT>.

Журнал включен в систему Российского индекса научного цитирования.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

УДК 004.942

М. С. Талалай, аспирант, инженер-исследователь,
К. В. Трушин, аспирант, инженер-практикант,
ЗАО "ИНТЕЛ АО"
e-mail: mikhail.s.talalay@intel.com,
konstantin.trushin@intel.com

Обзор подходов к проектированию регулярных ИС

Обсуждаются современные технологии производства интегральных схем, накладывающие ряд дополнительных требований на процесс проектирования и производства, в том числе требование повышения регулярности фотошаблонов, т. е. наличия периодичности в топологии. Описаны существующие подходы к повышению регулярности топологии интегральных схем, которые разрабатываются и используются сегодня.

Ключевые слова: регулярная топология, регулярные элементные блоки, конфигурируемая межслойными переходами логическая матрица

Введение

Жесткая конкурентная борьба на рынке микроэлектроники заставляет производителей снижать стоимость интегральных схем (ИС). Одним из способов достижения этой цели является масштабирование размеров МОП-транзистора за счет перехода на технологический процесс производства с меньшим размером топологических элементов. Минимальный размер L элемента топологии, который можно изготовить на кремниевой пластине с помощью фотолитографии, определяется критерием Релея

$$L = k_1 \frac{\lambda}{A},$$

где λ — длина волны излучающего устройства; A — числовая апертура линзы; k_1 — числовой коэффициент, зависящий от используемой технологии. Его теоретический предел при использовании равномерных сеток составляет 0,25. Значение коэффициента при использовании современных технологий приблизительно равно 0,4.

В прошлом уменьшение L осуществлялось за счет снижения длины волны λ . Уменьшение λ с ны-

нешнего значения 193 нм до более низких значений связано как с техническими трудностями (поглощение излучения материалом линзы, слишком высокая энергия кванта излучения), так и с экономическими (высокая стоимость такого решения [1]). Начиная с технологического процесса 180 нм, λ стала больше L , и в дальнейшем отношение L/λ только уменьшалось [2], что порождает дифракционные эффекты, которые искажают вид элемента топологии на кремниевой пластине (рис. 1).

Технология оптической коррекции дифракционных искажений (ОКДИ, *Optical Proximity Correction, OPC*) частично решает эту проблему за счет внесения в фотошаблон изменений, которые после литографии приближают вид получаемого топологического элемента на кремниевой пластине к спроектированному [3]. При переходе на новый технологический процесс число и сложность необходимых ОКДИ-поправок в фотошаблоне растет. После применения методов ОКДИ объем информации, описывающий фотошаблон, увеличивается в 4–8 раз по сравнению с первоначальной топологией и растет быстрее, чем линейно [4, 5]. Это увеличивает время изготовления фотошаблона и повышает его стоимость.

При разработке каждого нового технологического процесса в первую очередь изготавливают схемы памяти, а сложные вычислительные блоки могут быть реализованы только через некоторое время [6, 7]. Схема памяти имеет регулярную структуру из повторяющихся блоков, что упрощает ее изготовление. Далее в течение примерно года производители развивают технологию изготовления ИС с произвольной топологией, затрачивая много средств. Поэтому становится актуальным проектирование регулярной топологии и для логических блоков. Регулярность топологии также понижает коэффициент k_1 в формуле Релея и упрощает мас-

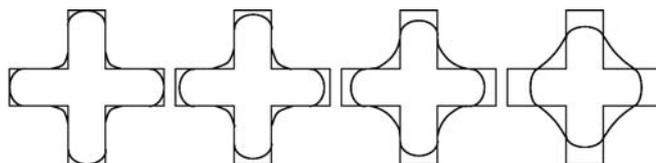


Рис. 1. Нарастание искажений элемента топологии при переходе к более тонким технологическим процессам

штабирование без уменьшения λ . Регулярность упрощает процедуру ОКДИ в топологии, что сокращает время, а как следствие, и стоимость изготовления фотошаблонов и снижает стоимость ИС.

Классификация интегральных схем по типу регулярности

Регулярность топологии ИС оценивается, например, преобразованием Фурье для характеристических функций нижних слоев [8]. Непосредственно на уровень регулярности влияет набор базисных блоков, из которых составляется схема. Чем разнообразней набор базисных блоков, тем менее регулярна топология схемы. В данной работе существующие методы проектирования ИС рассматриваются на предмет регулярности топологии схемы. В зависимости от целей проектировщика используются следующие подходы:

- ручное проектирование (*custom design*) — регулярность минимальна;
- проектирование на стандартных ячейках (*CBD — Cell Based Design*);
- техника проектирования на основе регулярных топологических заготовок (*Regular Fabrics*), разделяющаяся на физическом уровне на использование стандартных логических матриц (*Gate Arrays*) и использование логических матриц, конфигурируемых межслойными переходами (*VPGA*);
- проектирование структурных ИС (*structural ASICs*);
- подход с логическими матрицами, программируемыми пользователем (*FPGA*), — регулярность максимальна (рис. 2).

Проектировщик при выборе метода проектирования должен учитывать существующее соотношение между эксплуатационными характеристиками и регулярностью получаемой топологии. Как правило, чем выше регулярность, тем хуже характеристики. Таким образом, задачей для исследователя становится выявление методов проектиро-

вания с оптимальным соотношением. Сравнение электрических характеристик и занимаемой площади для разных подходов проведено в работах [9, 10]. Авторы отмечают, что подход *VPGA* реализует схему, в два раза большую по площади и в четыре раза большую по времени работы, чем подход *CBD*. В свою очередь, схема на блоках *FPGA* в среднем "медленнее" в 3,2 раза и больше по площади в 18 раз, чем *VPGA*.

Программируемые логические матрицы

Схемы, спроектированные на основе программируемых логических матриц, имеют вид повторяющихся логических блоков, соединенных посредством *SRAM*-переключателей (*static random access memory*). В этом подходе логическая функциональность каждого блока конфигурируется с помощью перевода соответствующего переключателя в нужное состояние. Количество оборудования в программируемой логической матрице фиксировано. Перед разработчиком стоит задача запрограммировать нужную функциональность в изготовленной схеме на кристалле, минимизируя используемое оборудование, что дает возможность снизить стоимость и повысить производительность получаемого решения [11].

Сравнивая данный подход проектирования с подходом *CBD*, исследователи отмечают простоту и скорость проектирования на уровне физического синтеза [12]. Также не требуется сложный и трудоемкий процесс разработки фотошаблона. К минусам получаемых схем относятся большая площадь и высокая потребляемая мощность [9].

Структурные ИС

Структурные ИС, как методология, являются компромиссным решением по стоимости, функциональности и производительности относительно метода программируемых логических матриц и подхода *CBD* [13]. Структурная ИС составляется

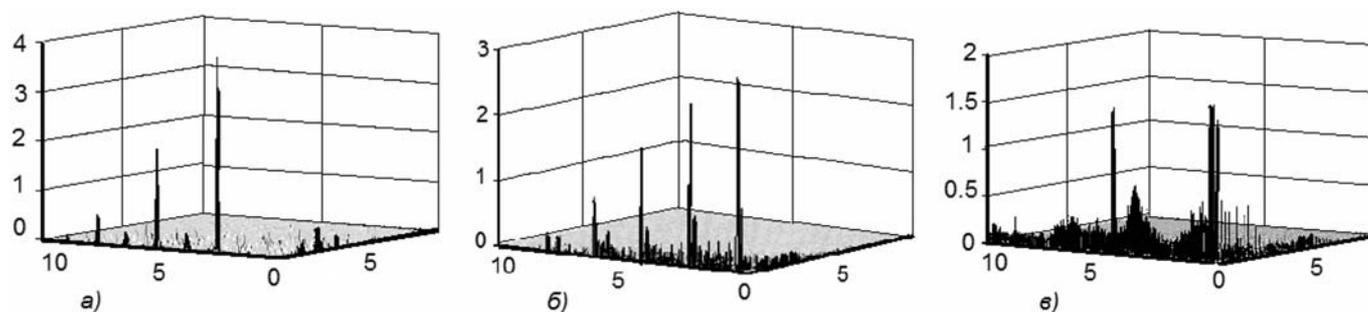


Рис. 2. Преобразование Фурье для *FPGA* (а), *Regular Fabrics* (б), *CBD* подходов (в) [8]

на фиксированной матрице одинаковых элементарных блоков, каждый из которых может реализовать весьма ограниченный набор логических функций. Обычно структурные ИС также содержат заранее изготовленные устройства ввода-вывода и встроенную память. Различные архитектуры структурных ИС используют элементарные блоки различной величины: от группы частично соединенных транзисторов и резисторов до сложных иерархических блоков, содержащих подблоки с логическими вентилями и регистрами. Проектировщику доступны для редактирования слои металла выше тех, которые заняты трассировкой внутри функционального блока. Матрица элементарных блоков может быть неоднородной, предоставляя разную площадь для различных блоков [14].

Маршрут проектирования структурных ИС схож с маршрутом проектирования с помощью библиотеки стандартных ячеек. Из-за небольшого объема рынка структурных ИС важным аспектом проектирования структурных ИС является минимизация числа используемых слоев металла для снижения стоимости одной ИС [14]. Также может возникать задача оптимизации размещения с учетом заранее изготовленной схемы синхронизации (*clock-tree*) [15, 16]. Поскольку в структурных ИС задержка проводников имеет большее влияние, чем для ИС на основе библиотеки стандартных ячеек, исследуется задача разбиения проводников заранее изготовленными буферами [17, 18]. Маршрут проектирования в основном опирается на алгоритмы, разработанные для проектирования схем с помощью библиотеки стандартных ячеек, которые не оптимизированы под блочную природу структурных ИС, что снижает качество результата [13, 14]. Для исправления ситуации предлагается использовать алгоритмы, которые применяются при проектировании программируемых логических матриц [13]. Например, размещение должно учитывать заранее запланированную структуру ИС и помещать элементы определенного типа в отведенные для них места [14].

К преимуществам данного подхода в сравнении с подходом *CBD* относятся меньшее время и стоимость разработки, а также значительное снижение затрат на фотошаблоны. Недостатки получаемой схемы заключаются в большей площади, большей стоимости и меньшей производительности.

По отношению к реализациям схемы на программируемых логических матрицах структурные ИС имеют меньшую площадь и стоимость в совокупности с большей производительностью. Но разработка схем на структурных ИС сложнее и занимает больше времени.

Существует ряд подходов проектирования, в которых ИС составляется из заранее спроектированных регулярных топологических блоков. Перед исследователями стоят задачи выбора набора базовых блоков с точки зрения логической функциональности и регулярности физической топологии каждого из них. При этом разработчик должен учитывать регулярность физической топологии всей схемы, которая будет построена из выбранных элементов. Для достижения регулярности в локальном и глобальном смысле используются регулярные заготовки в нижних слоях как основа, на которой выстраивается вся схема. Часто запланированные заготовки в нижних слоях предполагают использование полос полупроводник *p*- или *n*-типа одинаковой ширины, равномерные сетки как для затворов транзисторов, так и для межслойных переходов. Чтобы добиться нужной регулярности, добавляются дополнительные ограничения. Такими ограничениями могут являться, например, использование равномерной сетки для трассировки и однонаправленность нижних слоев металла. Указанные ограничения приводят к специальным задачам на уровне как логического синтеза, так и физического. Разнородность запланированных ограничений определяет разные задачи для проектировщика и, как следствие, разные методы решения исходной задачи физического синтеза логического блока. К подобным решениям относятся подходы на основе конфигурируемой межслойными переходами (*VPGA*) или проводниками логической матрицы. Каждый из указанных подходов рассмотрен ниже. Решение задачи логического синтеза для схем на регулярных заготовках реже дифференцируется исследователями по типу установленных ограничений на слои.

Логический синтез для регулярных схем

Задачей регулярного логического синтеза является получение графа, в котором вершинами являются логические блоки выбранного типа (стандартная ячейка, матрица, программируемая пользователем и т.д.). Ребрами являются электрические соединения между блоками. Схема, описанная таким графом после логического синтеза, должна иметь повторяющуюся структуру в результате физического синтеза. Решение этой задачи в методологии логических матриц получается автоматически в силу близкой топологии и логического описания каждого из блоков. Для подходов, которым свойственно большое разнообразие базисных эле-

ментов и разнородность топологий каждого из них, указанная задача является актуальной.

Одним из подходов получения регулярной схемы является метод построения регулярных блоков на основе имеющегося размещения, использующего стандартные ячейки. Авторы [19] предлагают для заранее подготовленных, имеющих регулярную структуру логических единиц (буфера, инвертора, минимального мультиплексора и вентиля, реализующего штрих Шеффера) реализовать все логические функции от трех переменных. Далее проводится разбиение исходной реализации схемы на блоки с числом входов, не превосходящим трех. Итеративным алгоритмом, введя понятие логической "близости" для блоков разбиения, полученные блоки объединяются в небольшое число групп. Далее для каждой группы подбирается физическая реализация таким образом, чтобы для каждого конкретного представителя группы необходимая для него функциональность достигалась минимальными изменениями реализации для группы, например, определением точки снятия сигнала либо постановкой межслойного перехода. Данный подход является довольно трудоемким. Также к минусам подхода можно отнести явную зависимость от исходного синтеза блоков и их дальнейшего размещения. Похожая методология предложена в работе [20] с идеей регулярно нацеленной кластеризации; предлагаемый алгоритм был расширен в работе [21].

Одним из направлений в области регулярного синтеза для проектирования на стандартных ячейках является метод оценки регулярности на логическом уровне [22]. Авторы описывают два способа оценки регулярности логической схемы: на глобальном и на локальном уровнях. Глобальный уровень опирается на наличие повторяющихся логических подсхем, которые часто используются для повышения эксплуатационных характеристик. Схожесть, а иногда и идентичность указанных подсхем, позволяют модифицировать их одновременно, что благоприятно отражается на характеристиках схемы после этапа физического синтеза. Способ выявления физической схожести логических подсхем описывается набором эвристических формул. Локальные же изменения нацелены на ускорение процедуры синтеза в рамках классических методов синтеза путем применения различных трансформаций схемы, а также на поддержание глобальной регулярности после отмеченных трансформаций. Другой набор формул для оценки регулярности подсхемы, а также ряд схем оптимизации регулярности вместе с эксплуатационными характеристиками предложены в работах [20, 21, 23, 24].

Для задач логического синтеза существует ряд подходов, нацеленных на снижение доступной элементной базы. Такие подходы основаны на выявлении логических схем, составленных из транзисторов или, чаще, из стандартных ячеек. Важно при этом оценить функциональность (мощность множества функций, которое можно реализовать, используя указанный блок) полученного набора базисных блоков [19, 25, 26]. Для выявления функциональности рассматриваются функции с точностью до инверсий, симметрий по входам, что уменьшает набор различных логических функций в рассматриваемом классе.

Одним из способов уменьшения доступной элементной базы является также простое сокращение числа библиотечных элементов. Исследователи, применяя такой подход, отмечают увеличивающуюся регулярность схемы при небольшом ухудшении эксплуатационных характеристик [27, 28], а иногда и улучшении [29].

Физический синтез регулярных блоков

После получения графа соединений на этапах логического синтеза решается задача о регулярной физической реализации базового конструктивного блока. Способ построения регулярной реализации зависит от правил проектирования, которые определяются технологией, и от поставленных ограничений на использование нижних слоев металлизации.

Конфигурируемая межслойными переходами логическая матрица. Конфигурируемая межслойными переходами логическая матрица состоит из заранее изготовленных одинаковых базовых топологических блоков и блоков трассировки без межслойных переходов. Как топологические блоки, так и трассировка могут иметь различную архитектуру и изначально не реализуют какую-либо логическую функцию. Задача физического проектирования заключается в том, чтобы из графа соединений получить размещение и трассировку с помощью межслойных переходов.

В [30] авторы установили, что среди всех ими рассмотренных оптимальным является топологический блок, состоящий из конфигурируемых проводниками мультиплексорных блоков (*hardware LUT*) и логического вентиля И-НЕ с тремя входами. Позже было показано, что уменьшение гранулярности функционального блока улучшает качество получаемой схемы [31]. Авторы [32] пришли к выводу о выигрыше ортогональной (*crossbar*) трассировки у блока переключений (*switch box*) по площади, хотя при этом возникают проблемы висячих емкостей.

В [25] предлагается топологический блок, состоящий из трех логических вентилях ИЛИ-НЕ и одного И. Представлено семейство различных по размерам блоков и, соответственно, логических матриц на их основе. Процедура физического проектирования обеспечивает выбор оптимального с точки зрения производительности и площади варианта матрицы. Во время размещения периодически проверяется, предоставляет ли выбранная на данный момент матрица лучшую производительность. Если нет, то делается выбор из остальных. Инструментом сравнения является статический временной анализ. Размещение основано на алгоритме моделирования отжига и оптимизирует временные характеристики [33].

Авторы [34—36] используют топологический блок, состоящий из отдельных транзисторов. Они определили зависимость максимального необходимого числа дорожек второго слоя металла от числа транзисторов в функциональном блоке и предложили следующий индуктивный метод трассировки топологического блока. Все возможные конфигурации соединений разбиваются на пять вариантов. Для каждого варианта создают одно или два соединения, удаляют один проводник из второго слоя и два транзистора. За счет этого происходит переход к задаче меньшей размерности. Таким образом, процедура дойдет до тривиальной задачи, которая уже решена. Сложностью данного подхода является необходимость заранее выбирать число транзисторов в базовом блоке и возможные из-за этого потери площади.

В [37] авторы предлагают структуру регулярных сегментов поперечной ортогональной трассировки. Очень высокая регулярность слоев металла приводит к трудностям с трассировкой. Авторы используют процедуру физического проектирования, которая оптимизирует трассируемость для такой фиксированной структуры. Первым этапом выполняется кластеризация схемы алгоритмом [38] в регулярные топологические блоки для увеличения плотности размещения и снижения числа межблочных соединений. Затем выполняется первичное размещение. Для получения более правдоподобной оценки дефицита трассировочных ресурсов межблочной (глобальной) трассировки используют быструю глобальную трассировку на основе построения деревьев Штейнера вида L и Z . Для снижения дефицита вставляется горизонтальный ряд дополнительных топологических блоков в необходимых местах. Далее используют инкрементальное размещение на основе алгоритма [39]. При этом происходит обновление соответствующих деревьев Штейнера и значения дефицита трасси-

ровочных ресурсов глобальной трассировки. Для построения окончательной трассировки используется алгоритм волновой трассировки [40]. Его основная идея — избегать проведения проводников через участки с дефицитом трассировочных ресурсов. Для этого используется функция, оценивающая загруженность трассировки.

Конфигурируемая проводниками логическая матрица. Конфигурируемая проводниками логическая матрица отличается от *VPGA* отсутствием заранее изготовленной трассировки. Трассировка и размещение транзисторов точными комбинаторными методами описаны в работе [4]. Здесь авторы используют каждый трассировочный слой металла либо в горизонтальном, либо в вертикальном направлениях (для каждого слоя доступно лишь одно направление), каждый слой имеет одинаковый фиксированный шаг трассируемой сетки, все транзисторы реализуются на полосах полупроводника p - и n -типа одинаковой ширины. Размещение транзисторов выполняется алгоритмом "ветвей и границ", а трассировка, оставаясь NP-полной задачей, решается полиномиальной редукцией к задаче выполнимости булевых формул. Данный подход при моделировании показывает небольшой проигрыш в основных схемных характеристиках по отношению к характеристикам, получаемым при синтезе на стандартных ячейках, но, несомненно, такое размещение является более регулярным, чем на стандартных ячейках, и, следовательно, потенциально предпочтительнее для использования в новейших технологиях.

Интересное решение задачи синтеза блока предлагается в работе [5]. Авторы решают обратную задачу, а именно: для максимально регулярной структуры найти минимальное удаление в слоях, при этом добившись нужной логической функциональности. После удаления различных кусков в слоях остаются неиспользуемые части, которые не нарушают логическую функциональность, но оставляют регулярность получаемых структур. Наличие указанного решения позволяет построить процесс литографии на основе двух совмещенных масок фотошаблона, первая из которых является начальной, т. е. максимально регулярной, а вторая содержит лишь те минимальные изменения, которые необходимы для получения нужной функциональности. Предполагается, что использование второго фотошаблона с минимальной площадью открытых для экспонирования областей останется физически реализуемым на этапе литографии. Данный подход находится только в своей начальной стадии развития и требует исследования.

Заклучение

Регулярность как метрика спроектированной схемы вводит ряд ограничений на задачи логического и физического синтеза, что позволяет выделить данное направление в САПР для ИС.

В статье рассмотрены существующие методы проектирования на предмет регулярности получаемой топологии схемы. С точки зрения соотношения регулярности и электрических характеристик наиболее оптимальным на сегодняшний день является подход на основе заготовленных конструкций геометрических примитивов в нижних слоях металлизации, имеющих периодическое расположение (*Regular Fabrics*). В статье рассмотрены задачи трассировки и размещения, которые возникают при использовании данных ограничений, и описаны различные способы решения этих задач.

Вопрос об использовании классических оптимизационных алгоритмов (масштабирование транзисторов, буферизация критических путей и др.) остается открытым при использовании их на подобных регулярных структурах.

Список литературы

1. **Borodovsky Y.** Lithography 2009. Overview and opportunities // SEMICON West. 2009. URL: [http://www.semi.org/cms/groups/public/documents/web_content/ctr_030805.pdf].
2. **Scheffer L. K.** Physical CAD Challenges to Incorporate Design for Lithography and Manufacturability // Proc. of ASP-DAC. 2004. P. 768–773.
3. **Cobb N.B.** Fast Optical and Process Proximity Correction Algorithms for Integrated Circuit Manufacturing // PhD thesis, University of California, Berkeley. 1998.
4. **Gupta P., Kahng A. B., Sylvester D., Yang J.** Performance-Driven Optical Proximity Correction for Mask Cost Reduction // Proc. of ISQED. 2005. P. 270–275.
5. **Teh S. H., Heng C. H., Tay A.** Design-Process Integration for Performance-based OPC Framework // Proc. of DAC. 2008. P. 522–527.
6. **Intel** First to Demonstrate Working 45nmChips 2006. URL: http://www.intel.com/pressroom/archive/releases/20060125comp.htm.
7. **Intel's** Transistor Technology Breakthrough Represents Biggest Change to Computer Chips in 40 Years, 2007. URL: [http://www.intel.com/pressroom/archive/releases/2007/20070128comp.htm].
8. **Kheterpal V.** Logic synthesis for regular fabrics // PhD thesis, Carnegie University. 2006.
9. **Kuon I., Rose J., Rogers S.** Measuring the Gap Between FPGAs and ASICs // Proc. of FPGA. 2006. P. 23–30.
10. **Ran Y., Marek-Sadowska M.** An integrated Design Flow for a Via-Configurable Gate Array // Proc. of DAC. 2004. P. 582–589.
11. **Chapman K.** Get your Priorities Right — Make your Design Up to 50 % Smaller // Xilinx Incorporated. 2007. URL: [http://www.xilinx.com/support/documentation/white_papers/wp275.pdf].
12. **Standard** Cell ASIC to FPGA Design Methodology and Guidelines. Altera Corporation 2009. URL: [http://www.altera.com/literature/an/an311.pdf].
13. **Zahiri B.** Structured ASIC: Opportunities and Challenges // Proc. of ICCD. 2003. P. 404–409.
14. **Wu K. C., Tsai Y. W.** Structured ASIC, Evolution or Revolution? // Proc. of ISPD. 2004. P. 103–106.
15. **Okamoto T., Kimoto T., Maeda N.** Design Methodology and Tools for NEC Electronics' Structured ASIC ISSP // Proc. of ISPD. 2004. P. 90–96.
16. **Schmit H., Gupta A., Ciabanu R.** Placement Challenges for Structured ASICs // Proc. of ISPD. 2008. P. 84–86.
17. **Hsu P. Y., Lee S. T., Chen F. W.** Buffer Design and Optimization for LUT-based Structured ASIC Design Styles. // Proc. of GLSVLSI. 2009. P. 377–380.
18. **Zhang T., Sapatnekar S. S.** Buffering Global Interconnects in Structured ASIC Design // Proc. of ASP-DAC. 2005. P. 23–26.
19. **Kheterpal V.** Logic synthesis for regular fabrics // Proc. of DAC'06. 2006. P. 112–119.
20. **Sreenivasa D. R., Kurdahi F. J.** Partitioning by regularity extraction // Proc. of DAC. 1992. P. 235–238.
21. **Chawdhary A., Sudhakar K., Saripella P.** A General Approach for Regularity Extraction in Datapath Circuits // Proc. of ICCAD. 1998. P. 332–339.
22. **Kutzschebauch T.** Regularity Driven Logic Synthesis // Proc. of ICCAD. 2000. P. 439–446.
23. **Nardi A., Sangiovanni-Vincentelli A.** Logic Synthesis for Manufacturability // IEEE Design and Test of Computers. 2004. Vol. 21. N 3. P. 192–199.
24. **Chakraborty A., Pandini D., Macii A., Poncino M.** Evaluating Regularity Extraction in Logic Synthesis // Proc. of ISSCS. 2005. P. 641–644.
25. **Hu B. J., Hailin, Liu Q., Marek-Sadowska M.** Synthesis and Placement Flow for Gain-Based Programmable Regular Fabrics // Proc. of ISPD. 2003. P. 197–203.
26. **Koorapaty A., Chandra V., Patel C., Pileggi L., Schmit H.** Heterogeneous Programmable Logic Block Architectures // Proc. of DATE. 2003. P. 1118–1119.
27. **Pawlowski D., Deng L., Wong M.** Fast and Accurate OPC for Standard-Cell Layouts // Proc. of ASP-DAC. 2007. P. 7–12.
28. **Jiao H., Chen L.** Cellwise OPC Based on Reduced Standard Cell Library // Proc. of ISQED. 2008. P. 810–814.
29. **Ueno K., Murakami H., Yano N., Okuda R.** A Design Methodology Realizing an Over GHz Synthesizable Streaming Processing Unit // Proc. of SVCDDTP. 2007. P. 48–49.
30. **Koorapaty A., Pileggi L., Schmit H.** Heterogeneous Logic Block Architectures for Via-Patterned Programmable Fabrics // Proc. of ICFPLA'03. 2003. P. 426–436.
31. **Koorapaty A., Kheterpal V., Gopalakrishnam P., Fu M., Pileggi L.** Exploring Logic Block Granularity for Regular Fabrics // Proc. of PATE'04. 2004. P. 10468–10474.
32. **Patel C., Cozzie A., Schmit H., Pileggi L.** An Architectural Exploration of Via Patterned Gate Arrays // Proc. of ISPD'03. 2003/ P. 184–189.
33. **Betz V., Rose J.** VPR: A New Packing, Placement and Routing Tool for FPGA Research // Proc. of ICFPLA'97. 1997. P. 213–222.
34. **Ran Y., Marek-Sadowska M.** The Magic of a Via-Configurable Regular Fabric // Proc. of ICCD'04. 2004. P. 1–6.
35. **Ran Y., Marek-Sadowska M.** On Designing Via-Configurable Cell Blocks for Regular Fabrics // Proc. of DAC'04. M. 2004. P. 198–203.
36. **Ran Y., Marek-Sadowska M.** Designing Via-Configurable Cell Blocks for Regular Fabrics // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2006. Vol. 14. № 1. P. 1–14.
37. **Ran Y., Marek-Sadowska M.** Via-Configurable Routing Architectures and Fast Design Mappability Estimation for Regular Fabrics // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2006. Vol. 14. № 9. P. 998–1009.
38. **Hu B., Marek-Sadowska M.** Wire Length Prediction based clustering and its application in placement // Proc. of DAC. 2003. P. 800–805.
39. **Singh D. P., Brown S. D.** Incremental Placement for Layout-Driven Optimizations on FPGAs // Proc. of ICCAD. 2002. P.752–759.
40. **Ebeling C., McMurchie L., Hauck S. A., Burns S.** Placement and Routing Tools for the Triptych FPGA // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 1995. Vol. 3. № 4. P. 473–482.

П. Н. Бибило, д-р техн. наук, проф., зав. лаб.,
И. П. Логинова, канд. техн. наук, ст. науч. сотр.,
В. И. Романов,
 канд. техн. наук, доц. вед. науч. сотр.,
Л. Д. Черемисинова,
 д-р техн. наук, доц., гл. науч. сотр.,
 Объединенный институт проблем информатики
 Национальной академии наук Беларуси, г. Минск
 e-mail: bibilo@newman.bas-net.by

Система CLTT проектирования топологии функциональных блоков заказных цифровых СБИС

Описывается система CLTT проектирования функциональных блоков заказных цифровых сверхбольших интегральных схем, выполненных на основе КМОП-технологии. Маршрут проектирования позволяет по исходному описанию функциональных блоков на языке VHDL получить послойный топологический чертеж с компактным размещением макроэлементов, информационных связей, шин питания и заземления в соответствии с технологическими требованиями.

Ключевые слова: сверхбольшая интегральная схема, макроэлементы с регулярной структурой, комбинационная логика, программируемые логические матрицы, VHDL, размещение элементов, трассировка соединений, описание топологии

Введение

Разработка систем автоматизированного проектирования заказных цифровых СБИС представляет собой сложную проблему, обусловленную постоянно увеличивающейся степенью интеграции и большой размерностью решаемых задач. Если для разработки операционных блоков СБИС могут применяться уже готовые решения из ранее созданных проектов СБИС [1], то проектирование управляющей логики каждый раз ведется заново, поэтому актуальной проблемой является разработка эффективных средств автоматизированного проектирования управляющей логики заказных цифровых СБИС.

Проблемам создания автоматизированных систем проектирования (САПР) СБИС посвящена обширная литература, имеются обзоры [2] САПР, используемых в промышленных целях для разработки проектов микросхем. Описываемая в данной работе система CLTT является развитием системы CL [3] и предназначена для проектирования то-

пологии функциональных управляющих блоков заказных СБИС на базе программируемых макроэлементов трех типов: ПЛМ (программируемые логические матрицы); РМОП-схемы (регулярные МОП-схемы с последовательными соединениями транзисторов) и ПЗУ (постоянные запоминающие устройства). Функциональный блок может быть реализован в виде одного макроэлемента либо сети макроэлементов указанных типов. Регулярные структуры типа ПЛМ, ПЗУ давно используют при разработке СБИС, для них успешно разрабатывают "кремниевые компиляторы" [4], позволяющие получать описание послойной топологии по описанию их поведения на высокоуровневом языке проектирования. С переходом на субмикронные нормы проектирования интерес к таким структурам не ослабевает, в работе [5], например, предлагаются новые схмотехнические решения для ПЛМ.

Кроме "кремниевой компиляции" послойной топологии макроэлементов, система CLTT имеет следующие особенности: переход на новые нормы проектирования касается только разработки новых библиотек фрагментов послойной топологии, из которых "собираются" макроэлементы; задачи размещения макроэлементов решаются как в автоматическом, так и диалоговом режиме; в окрестности каждого макроэлемента осуществляется локальная разводка шин питания и заземления этого макроэлемента; встроенный топологический редактор ориентирован на работу с заданными типами макроэлементов. Что касается размещения логических блоков СБИС, то за последние годы был достигнут значительный прогресс в исследовании задачи размещения. В настоящее время широко используются следующие алгоритмы размещения: KraftWerk, Capo, Dragon и FengShui [6]. В литературе известно большое число эвристических алгоритмов размещения логических блоков на кристалле СБИС. Наиболее исследованной задачей является размещение стандартных блоков, имеющих одинаковую высоту. Представительные обзоры таких методов приведены в работах [7—9]. Более скромные результаты получены для случая размещения разногабаритных блоков на кристалле СБИС [10]. В основном предполагается, что "больших" нестандартных блоков немного и потому можно вначале разместить их, а дальше применять алгоритмы размещения стандартных блоков.

Предлагаемая система CLTT нацелена на решение задачи размещения для более общего случая, когда все блоки (или большинство) разногабаритные. Используемая в системе CLTT стратегия размещения основана на предположении, что полу-

ченное размещение существенно зависит от начального размещения и что наиболее эффективно поручить переразмещение человеку, снабдив его удобными инструментальными средствами.

1. Архитектура системы CLTT

Архитектура системы CLTT представлена на рис. 1. Исходными для этой системы могут быть описания цифровых блоков на языках VHDL (*Very high speed integrated circuits Hardware Description Language*) [11] и SF (*Structural and Functional description language*) [3]. Язык SF, являясь также и внутренним языком системы, ориентирован на иерархические структурно-функциональные описания, комбинационные блоки в которых задаются либо в виде логических уравнений, либо в матричном виде (парой матриц, описывающих систему ДНФ — дизъюнктивных нормальных форм булевых функций). Система CLTT содержит четыре подсистемы.

Подсистема "Формирование проекта" позволяет получить представление иерархически организованного проекта во внутренней форме, т. е. в виде иерархических SF-описаний.

Подсистема "Оптимизация проекта" предназначена для решения задач логического проектирования и обеспечивает оптимизацию представлений комбинационной части проекта (систем булевых функций) с учетом разных ограничений и критериев оптимизации. Подсистема включает средства:

- устранения иерархии описания, т. е. получение функционального описания блока из многоуровневого структурно-функционального описания;
- преобразования многоуровневых представлений систем булевых функций в двухуровневые И-ИЛИ представления (системы ДНФ);
- минимизации булевых функций в классе ДНФ, настраиваемой согласно требованиям проектировщика;
- разбиения многоуровневых представлений на блоки с ограниченными числами входных и выходных переменных, элементарных конъюнкций, на которых заданы ДНФ функций каждого из блоков.

Подсистема "Верификация проекта" решает задачу определения функциональной эквивалентности различных состояний проекта. Поскольку в процессе проектирования оптимизация проекта сводится к последовательному решению трудоемких логико-комбинаторных задач разными программами и при этом имеется возможность внесения поправок проектировщиком, необходимо иметь уверенность в том, что функции проекта не изменились.

Подсистема "Компиляция проекта" предназначена для формирования послойных топологических описаний подсхем, полученных после оптимизации в подсистеме "Оптимизация проекта". На входе подсистемы "Компиляция проекта" все оптимизированные описания подсхем преобразуются проектировщиком в структурные описания трех типов программируемых макроэлементов — ПЛМ, РМОП-схемы или ПЗУ. Формирование топологии каждой подсхемы проходит следующие фазы компиляции:

- преобразование структурного описания подсхемы в структурное описание макроэлемента заданного типа;
- преобразование структурного описания макроэлемента в символьное описание его топологии;
- преобразование символьного описания топологии макроэлемента в его послойную топологию.

Формирование конечного описания послойной топологии подсхемы опирается на ее компиляцию из параметризованных топологических фрагментов. Для каждого типа макроэлемента разработана библиотека фрагментов, из которых по принципу мозаики собирается его послойная топология. Топология макроэлемента любого из упомянутых типов на символьном уровне представляется в виде матрицы или совокупности нескольких матриц. Элементом матрицы символьной топологии является имя топологического фрагмента, размещенное в месте, определяемом схмотехнической и топологической реализацией прототипа макроэлемента. Программируемое ядро матриц символьной топологии определяется элементами матриц структурных описаний подсхемы, преобразованных в матрицы структурного описания макроэлемента данного типа.

Подсистема компиляции проекта позволяет также создавать SPICE-модели подсхем, реализованных в виде макроэлементов ПЛМ и РМОП-схем (соответствующие файлы имеют расширение **.cir**). Эти модели могут быть экспортированы в систему аналогового схмотехнического моделирования, например, в систему "AccuSim" (Mentor Graphics) (рис. 1) для определения схмотехнических характеристик проектируемой подсхемы.

Результатом работы подсистемы "Компиляция проекта" являются собранные в месте хранения проекта следующие топологические описания (соответствующие файлы имеют расширение **.sou**):

- послойной топологии макроэлементов схемы;
- послойной топологии макроэлементов схемы и разводки внутренних шин с оптимизацией числа портов подвода силовых шин;
- топологии огибающих многоугольников макроэлементов схемы с реализацией разводки по двум

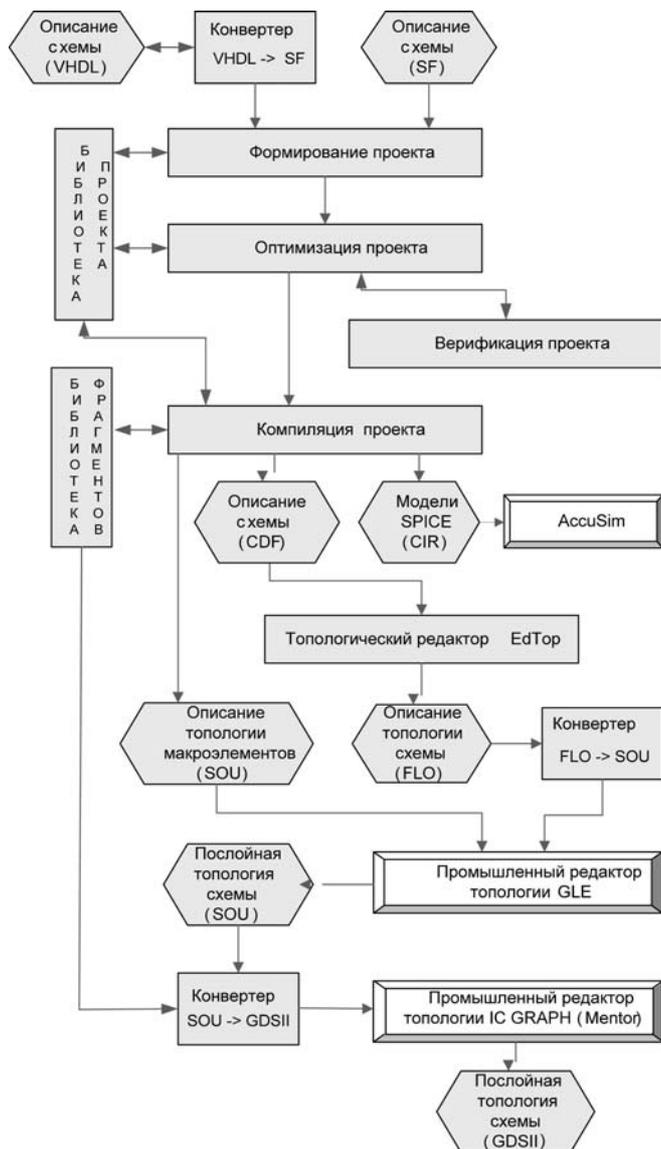


Рис. 1. Архитектура системы CLTT

слоям металлизации внутри огибающего прямоугольника и указанием местоположения портов.

Автоматическая разводка внутри огибающих прямоугольников макроэлементов, определение размеров огибающих прямоугольников и координат их портов проводятся одновременно с компиляцией топологии макроэлементов из ячеек топологических библиотек с соблюдением всех правил МОП-технологии. Подсистема "Компиляция проекта" формирует также информацию для организации процесса расположения макроэлементов на поле кристалла заказной СБИС, который осуществляется встроенным топологическим редактором "EdTop".

Информация о топологических размерах огибающих многоугольников макроэлементов, коор-

динатах расположения портов для подвода информационных и силовых шин, о связях макроэлементов с другими макроэлементами проектируемой схемы формируется в виде текстового файла, который имеет расширение **.cdf** и находится в месте хранения проекта.

Топологический редактор *EdTop* предназначен для решения основных задач топологического проектирования — размещения элементов и макроэлементов (и их связей) с учетом технологических ограничений. Он выполняет различные процедуры размещения макроэлементов, реализованные как в автоматическом, так и в интерактивном режимах работы проектировщика. Результатом работы топологического редактора *EdTop* является информация о размещении элементов и их связей — соответствующий файл имеет расширение **.flo**.

Конвертер FLO → SOU реализует этап получения описания послойной топологии в формате **.sou**. Он проводит замену абстрактного представления огибающих многоугольников макроэлементов и всех связей, представленных в файле формата **.flo**, следующими ссылками: 1) на соответствующие описания послойной топологии макроэлементов схемы; 2) на топологические описания всех связей в двух слоях металлизации; 3) на ячейки межслойных переходов.

Конвертер SOU → GDS, входящий в состав GLE (промышленного редактора топологии), реализует заключительный этап получения послойной топологии в формате **.gds**. Он осуществляет "раскрытие" иерархии многоуровневого описания послойной топологии схемы в формате **.sou** с использованием библиотек топологических фрагментов макроэлементов.

Программное обеспечение системы CLTT содержит: 1) оболочку, обеспечивающую проведение сеанса проектирования; 2) программные модули, реализующие проектные процедуры в четырех подсистемах; 3) графический редактор *EdTop*, обеспечивающий в автоматическом и интерактивном режимах создание эскиза топологии с оптимальным размещением макроэлементов схемы и трассировкой всех соединений.

2. Компиляция топологии макроэлементов

Этап логического синтеза завершается получением структурных описаний функциональных блоков в виде троичных и булевых матриц, на основе которых составляются описания символьной топологии макроэлементов схемы. Элементы символьной топологии, определяющие "программируемые" МОП-транзисторы, соответствуют элементам матриц в структурных описаниях макроэлементов.

Топологическая структура фрагментов с "программируемыми" МОП-транзисторами для каждого типа макроэлемента определяется схематической реализацией данного типа макроэлемента на кристалле СБИС. Топологии "программируемых" ячеек и, соответственно, их размеры для ПЛМ, ПЗУ и РМОП-схем различаются. Однако принцип образования из "программируемых" ячеек матриц в макроэлементах разных типов одинаков (примыкание ячеек по горизонтали и по вертикали). Совокупности топологических фрагментов "программируемых" и других функциональных частей макроэлемента параметризованы по вертикальным и горизонтальным размерам, поэтому они собираются в автоматическом режиме в искомую послойную топологию согласно описанию символьной топологии. Вид послойной топологии (представлены не все технологические слои) макроэлементов РМОП-схемы и ПЛМ небольшой размерности приведены на рис. 2 и 3.

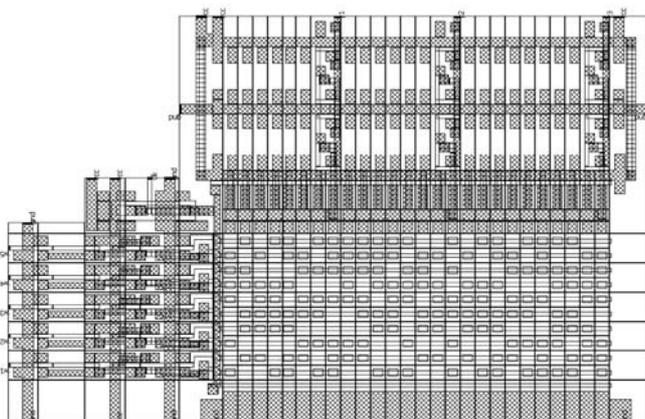


Рис. 2. Пример послойной топологии РМОП-схемы

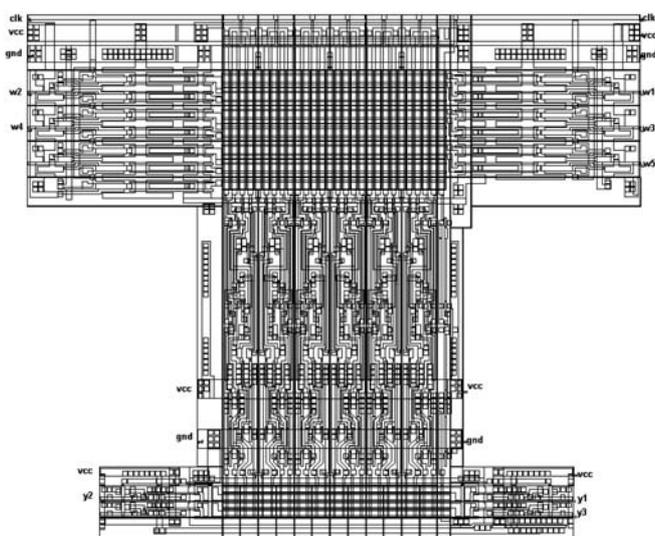


Рис. 3. Пример послойной топологии ПЛМ

Побочной стороной обеспечения максимально возможной параметризации библиотечных топологических фрагментов является наличие в макроэлементах многократного дублирования шин "земли", питания и синхронизации, подводимых к различным функциональным блокам макроэлемента. В целях эффективного решения задач размещения и трассировки при работе с редактором *EdTop* проводится устранение дублирующего подвода силовых и синхронизирующих линий путем автоматического формирования и оптимального размещения единственного порта вне макроэлемента с соблюдением всех технологических правил для каждого типа силовых линий. Эти порты размещаются на новой границе макроэлемента, на которой также расположены порты для подвода входных и выходных информационных связей. Вся инфраструктура внутренней трассировки к портам на новой границе макроэлемента, определяющей форму огибающего многоугольника макроэлемента, формируется автоматически и добавляется к послойной топологии макроэлемента.

Вся совокупность сформированной таким образом топологии объявляется оптимизированной послойной топологией макроэлемента. Описание последней организовано иерархическим образом в виде ссылок на послойную топологию макроэлемента и ссылок на послойную топологию внутренней трассировки макроэлемента, оформленную в виде отдельного элемента библиотеки. Такая организация описания топологии макроэлемента позволяет в дальнейшем осуществлять различную ориентацию и зеркальную трансформацию топологии, которая определяется итогами размещения и трассировки. Таким образом, подготовка к проектированию прототипа эскиза топологии в редакторе *EdTop* состоит из следующих этапов:

- преобразование системы координат описания топологии (SOU-формат);
- проведение оптимизирующих преобразований топологии макроэлементов;
- импортирование в редактор *EdTop* списков макроэлементов схемы; координат огибающих прямоугольников; координат, имен и типов портов макроэлементов; связей макроэлементов.

3. Размещение и трассировка соединений макроэлементов

При проектировании топологии заказных СБИС одной из наиболее важных и сложных задач является задача размещения на кристалле спроектированных макроэлементов электрической схемы. Основными целями процесса размещения являются минимум занимаемой площади кристалла и

создание наилучших условий для последующей трассировки соединений схемы.

В разработанном в рамках системы CLTT топологическом редакторе *EdTop*, в отличие от известного редактора OT-TO Layout Editor [12], приняты следующие соглашения о классификации контактов и связей проектируемой схемы. Контакты, расположенные на границе огибающего прямоугольника для всей схемы, соответствующие ее входам и выходам, называются *внешними*. Внешние контакты могут располагаться в любом месте границы прямоугольника схемы, но их положение фиксируется при проведении процедур размещения и трассировки. Внешние контакты участвуют в образовании *внешних* информационных связей. *Внутренние* информационные связи образуются между парами контактов макроэлементов или элементов схемы.

Проблема размещения интенсивно изучается последние 30 лет. Наибольшие успехи получены для случая размещения одногабаритных элементов и элементов (таких как в базовых матричных кристаллах), имеющих одну и ту же высоту, а также для случая, когда имеются крупные макроэлементы, но их немного. В редакторе *EdTop* реализован общий случай задачи размещения, когда на размеры макроэлементов не накладывается никаких ограничений, за исключением общей площади проектируемого кристалла. Решение задачи размещения предполагает:

- задание критериев размещения;
- начальное размещение элементов на кристалле;
- дальнейшую оптимизацию начального размещения.

Макроэлементы на эскизе представляют собой прямоугольники, имеющие контактные площадки внешних полюсов на границах. Вводится модель площади кристалла, выделяемой под элементы и называемой коммутационным полем. Это поле представляет собой координатную сетку, служащую для задания размеров площадей макроэлементов и их позиций. Макроэлементы сначала размещают плотно друг к другу, но после начального размещения, перед трассировкой, их раздвигают на некоторое расстояние, устанавливаемое параметрически.

Под задачей начального размещения элементов понимают поиск такого отображения O_{best} элементов схемы $e_i \in E$ на множества S_i позиций коммутационного поля, при котором минимизируется функция штрафов:

$$F(O_{best}) = \min \sum_{e_i \in E} F(O_i),$$

при условии, что

$$S_i \cap S_j = \emptyset \text{ (для всех } i \neq j) \text{ и } \sum S_i = \min.$$

Здесь $F(O_i)$ — количественная оценка штрафа за вносимые на коммутационное поле "белые пятна" (неиспользуемые области) при размещении элемента e_i на прямоугольнике S_i позиций коммутационного поля (прямоугольники не могут перекрываться).

Вначале для размещения выделяется квадрат, размеры которого вычисляются в зависимости от размеров размещаемых макроэлементов. Размещение ведется, начиная от левого верхнего угла, имеющего координаты (0,0). Привязка размещенных элементов задается также координатами их левых верхних углов. Макроэлементы перед размещением упорядочиваются по невозрастанию их площадей, именно из такой последовательности они выбираются для размещения.

Реализованный алгоритм размещения основывается на последовательном заполнении прямоугольных областей размещения $K_i((X_i, Y_i), (W_i, H_i), (W_i^d, H_i^d))$ коммутационного поля, характеризующихся координатами (X_i, Y_i) левого верхнего угла, фактически возможными шириной и высотой (W_i, H_i) фактически возможной области размещения, шириной и высотой (W_i^d, H_i^d) подобласти "предпочтения". Последние определяются числом позиций поля, занятых макроэлементами выше и

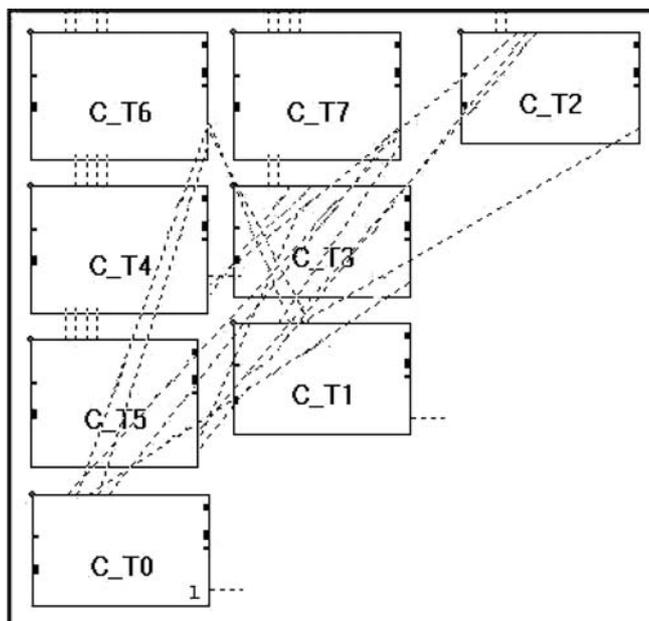


Рис. 4. Начальное размещение макроэлементов

слева от K_i . Аналогично задается размещаемый макроэлемент $e_j(x_j, y_j), (w_j, h_j)$.

На каждом шаге алгоритма для размещения выбирается очередной еще не размещенный макроэлемент e_j , для него подбирается та из областей $K_i((X_i, Y_i), (W_i, H_i), (W_i^d, H_i^d))$ с $w_j \leq W_i, h_j \leq H_i$, выбор которой минимизирует площадь появляющихся на поле "белых пятен". При этом рассматриваются варианты размещения с поворотом макроэлемента на 90° . После привязки макроэлемента параметры "затронутых" областей размещения пересчитываются, включая деление одной области на две новые — справа и снизу от размещенного элемента. После получения начального размещения макроэлементов (пример которого приведен на рис. 4) осуществляется автоматическая разводка всех связей. Такая разводка допускает появление на эскизе "некорректных" (пересекающих другие соединения или макроэлементы) соединений, которые впоследствии должны быть поправлены в интерактивном режиме.

В рамках обслуживания проекта редактором *EdTop* ручная работа проектировщика сочетается с автоматически выполняемыми процедурами начального размещения макроэлементов и последующей разводки силовых и сигнальных шин, обеспечивающих требуемые связи межсоединений. При реализации разводки шин редактор *EdTop* обеспечивает два режима исполнения: с фиксированным набором позиций внешних контактов и их произвольным размещением. Вариант фиксированных позиций предполагает, что все внешние входы проектируемой схемы расположены на левой сто-

роне прямоугольной площадки кристалла сверху, в порядке их присутствия в файле описания схемы.

В основе алгоритма разводки межсоединений лежит анализ взаимного расположения на плоскости исходного и конечного положений соединяемых контактов. На основе выполненного анализа выбирается один из заранее перечисленных планов топологического размещения сегментов линии межсоединения. Естественно, что построенный вариант разводки, скорее всего, не является конечным и требует вмешательства проектировщика для устранения некорректных разводок — разнесения пересекающихся линий соединений по разным слоям металлизации.

Технологически корректное размещение макроэлементов и связей является результатом работы в топологическом редакторе (рис. 5).

В процессе интерактивного проектирования возможно и проведение оптимизационных преобразований, направленных на сокращение суммарной длины линий соединения схемы. Для этого проектировщику предоставляются удобные инструменты коррекции автоматически выполненного начального размещения. К ним относятся средства:

- настройки конфигурационных параметров редактора *EdTop*;
- переразмещения макроэлементов и их совокупностей;
- перетрассировки построенных соединений;
- демонстрации свойств выделенных компонентов схемы с возможностью их частичной модификации;
- масштабируемого отображения эскиза проектируемой схемы;
- навигации видимого на экране фрагмента эскиза по всей площади будущего кристалла.

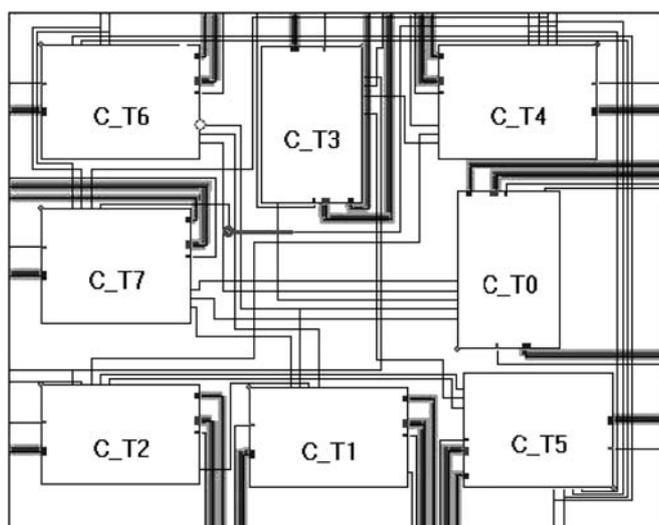


Рис. 5. Результат работы редактора *EdTop*

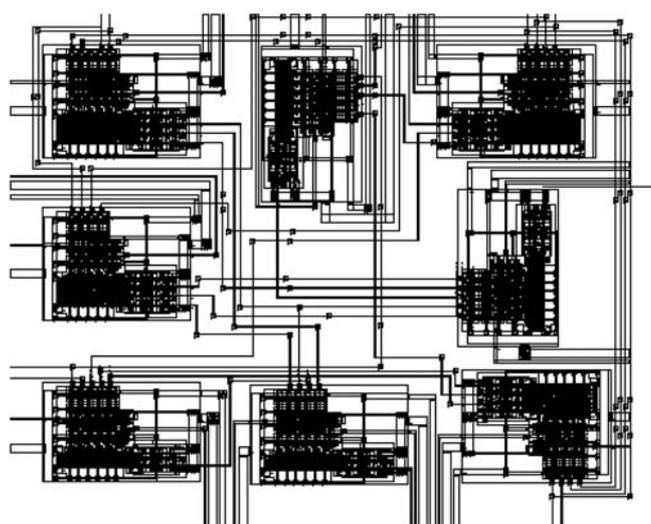


Рис. 6. Послойная топология функционального блока

Получение послойной топологии проектируемого функционального блока осуществляется конвертером $FLO \rightarrow SOU$. Результат конвертации эскиза в послойную топологию схемы представлен на рис. 6.

4. Пример размещения и трассировки макроэлементов управляющего блока контроллера

Для демонстрации качественной оценки эффективности технологии проектирования в рамках системы CLTT можно привести следующий выполненный с помощью предлагаемой системы CLTT проект. Функциональный блок *example* — таблица микрокоманд контроллера, заданного на проектирование, содержала 2130 строк и была задана системой из 63 частично определенных булевых функций, зависящих от 18 аргументов и заданных на 2130 наборах булева пространства. Задание на проектирование контроллера было представлено в виде VHDL-описания системы частичных булевых функций, примеры таких описаний представлены в работе [11].

В процессе проектирования VHDL-описание было преобразовано в матричное представление системы частичных булевых функций в формате SF. Функции были отдельно минимизированы в классе ДНФ, в результате все 63 функции оказа-

лись заданными 710 элементарными конъюнкциями. Разбиение минимизированной системы на четыре подсистемы позволило получить одноуровневую логическую сеть из четырех макроэлементов, представляющих собой РМОП-схемы. Результат топологического проектирования полученной одноуровневой сети макроэлементов представлен на рис. 7.

Заключение

Система CLTT по своим функциональным возможностям соответствует последним достижениям в области разработки отечественных систем автоматизации проектирования заказных СБИС, она обеспечивает сокращение сроков проектирования и увеличивает надежность процесса проектирования. Разработанная система создает предпосылки для формирования комплексной системы автоматизации проектирования управляющей логики за счет использования других типов макроэлементов и готовых топологических решений IP-блоков.

Список литературы

1. Shung B., Rajeev J., Ken R. et al. An Integrated CAD System for Algorithm-Specific IC Design // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 1991. Vol. 10, № 4. P. 447—461.
2. Аньшаков Г. Г. Системы автоматизированного проектирования электронных устройств // Информационные технологии. 2005. № 4. С. 63—71.
3. Система "Custom Logic" автоматизированного проектирования управляющей логики заказных цифровых СБИС / П. Н. Бибило, И. В. Василькова, С. Н. Кардаш и др. // Микроэлектроника, 2004. Т. 32, № 5. С. 379—398.
4. Эйрис Р. Проектирование СБИС. Метод кремниевой компиляции: пер. с англ. М.: Наука, 1988. 456 с.
5. Wang J., Chang C., Yeh C. Analysis and design of high-speed and low-power CMOS PLAs // IEEE Journal of Solid-State Circuits. 2001. Vol. 36, № 8. P. 1250—1262.
6. Caldwell A. E., Kahng A. B., Markov I. L. Can Recursive Bisection Alone Produce Routable Placements // DAC 2000. P. 477—482.
7. Shahookar K., Mazumber P. VLSI Cell Placement Techniques // J. ACM Computing Surveys. 1991. Vol. 23, No. 2. P. 143—220.
8. Рыбин Д. А. Сравнение различных подходов к глобальному размещению стандартных элементов, основанному на разбиении // Информационные технологии. 2005. № 5. С. 9—18.
9. Wang M., Yang X., Sarrafzadeh M. Dragon2000: Fast Standard-Cell placement for large Circuits // Intern. Conf. On Computer-Aided Design, 2000. P. 260—263.
10. Ковалев А. В., Коноплев Б. Г. Метод мозаичного синтеза топологии заказных СБИС / Известия вузов. Электроника. 1999. № 4. С. 52—57.
11. Бибило П. Н., Авдеев Н. А. VHDL. Эффективное использование при проектировании цифровых систем, М.: СОЛОН-Пресс, 2006. 344 с.
12. Бобовский В., Потапов Ю. Профессиональный редактор топологий микросхем OT_TO Layout Editor 2002 // EDA Expert. Инженерная микроэлектроника. 2002. № 2 (65). С. 69—73.

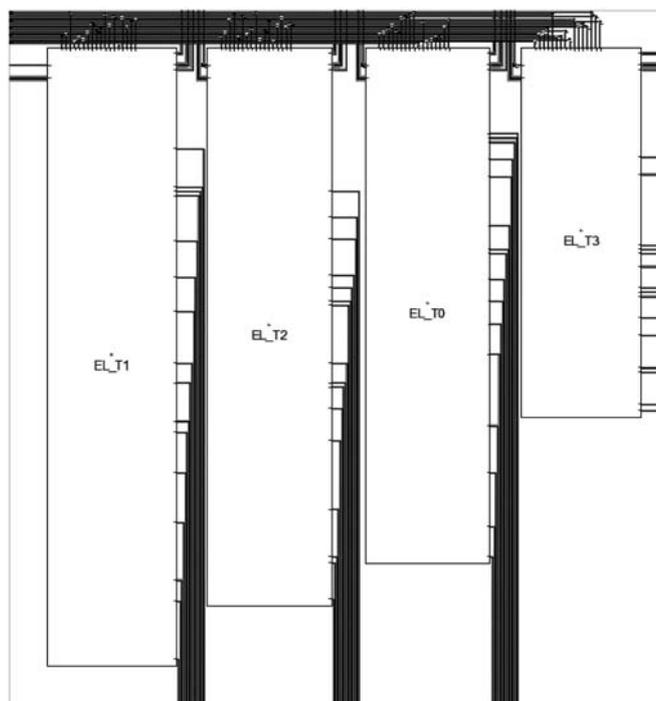


Рис. 7. Конечный результат размещения макроэлементов схемы *example* в редакторе *EdTop*

УДК 004.416.6

Г. Б. Евгеньев, д-р техн. наук, проф.,
МГТУ им. Н.Э. Баумана,
e-mail: egb@sprut.ru

Синергетическая методология интеграции знаний

Описывается новая информационная технология создания интеллектуальных прикладных систем с использованием геометрических, математических, четких и нечетких знаний.

Ключевые слова: информационная технология, искусственный интеллект, гибридная система, теория программирования, четкие знания, нечеткие знания, модуль знаний, геометрические знания, математические знания, нейросетевые модели, лингвистическая переменная, нечеткие базы знаний

Введение

Актуальным направлением дальнейшего развития информационных технологий создания интеллектуальных систем является совместное использование методов и средств различных дисциплин. В настоящее время широко развиты работы по созданию гибридных систем [1]. Гибридизация представляет собой интеграцию методов и технологий на глубинном, а не на внешнем уровне, когда различные блоки системы реализуют какой-то один метод решения интеллектуальных задач и взаимодействуют между собой [1]. Глубинный уровень объединения предполагает создание новых методов, использующих понятия объединяемых базовых методов. Однако пока гибридизация охватывает методы, объединяемые под зонтичным термином "мягкие вычисления" (*soft computing*) [1].

Представляется целесообразным значительно расширить круг объединяемых методов, выйдя на междисциплинарный уровень интеграции. Это означает включение в этот круг не только методов мягких вычислений, но и четких вычислений классического искусственного интеллекта, а также математики и теории программирования. В таком случае системы с полным основанием можно будет назвать синергетическими, так как **синергетика** (от греч. συν — совместно и греч. εργος — действующий) — это междисциплинарное направление научных исследований.

Пионером работ в области синергетического искусственного интеллекта (ИИ) является В. Б. Тарасов [2]. Синергетические системы ИИ обладают такими основными характеристиками как множество неоднородных компонентов, активность, автономность, множество гибких взаимосвязей между компонентами, семиотическая природа взаимосвязей, открытость, высокий эволюционный потенциал. Архитектура и технологии синергетического ИИ отличаются от классического неоднородностью, гибридностью, объединением различных информационных технологий [2].

1. Гибридизация методов программирования и классического ИИ

Для создания синергетической системы ИИ необходимо, прежде всего, рассмотреть возможность гибридизации передовых методов теории программирования и классического ИИ. Такой гибрид может стать технологической основой для создания синергетических систем. Далее на этой основе нужно разработать методы компьютеризации трех уровней знаний: математических, классических для четких вычислений и нечетких для мягких вычислений.

Наиболее перспективной технологией в области программирования для создания прикладных систем является *объектно-ориентированное программирование* (ООПр) [3]. Искусственный интеллект изначально исповедовал декларативные, т. е. непроцедурные, неалгоритмические методы представления знаний. В этой связи ООПр представляет собой большой шаг в направлении сближения технологий программирования и искусственного интеллекта.

В развитии процедурного программирования был ряд этапов, к числу важнейших из них следует отнести технологии *структурного* и *модульного программирования*.

В *структурном программировании* фундаментальной является теорема о структурировании [4], в которой доказано, что любая простая программа функционально эквивалентна структурной программе, составленной из элементов базисного множества {последовательность, *ifthenelse*, *whiledo*}. Это означает, что любая программа, представляющая собой один функциональный узел, может быть составлена из структур упомянутых трех базовых типов.

В модульном программировании фундаментальной является рекомендация выделения групп подпрограмм, использующих одни и те же глобальные данные, в отдельно компилируемые модули. Связь между модулями осуществляется через специальный интерфейс, в то время как доступ к реализации модуля запрещен [5].

Следующим принципиально важным шагом в развитии технологии был переход к *объектно-ориентированному программированию* (ООПр). Этот шаг позволяет разрабатывать качественные программные средства повышенной сложности за счет лучших механизмов разделения данных, увеличения повторяемости кодов, использования стандартизованных интерфейсов пользователя и т. д.

В ООПр программа представляется в виде совокупности *объектов*, каждый из которых является экземпляром определенного класса, а классы образуют иерархию с наследованием свойств [3]. Следует подчеркнуть, что при концептуальном (понятийном) моделировании прикладных областей (Про) используют эквивалентную объекту категорию *понятия*, которая является основной единицей интеллектуальной деятельности, базовой конструкцией представления *знаний*. Объект — это абстракция множества предметов реального мира, в которой: 1) все предметы множества (экземпляры) имеют одни и те же характеристики (свойства); 2) все экземпляры подчинены и согласовываются с одним и тем же набором *правил поведения*. Состояние объекта характеризуется перечнем его свойств и текущим значением каждого из этих свойств, а *поведение объекта* определяется *методом*, состоящим из набора правил.

Начало развития технологии ИИ относится ко второй половине 50-х годов прошлого века, когда начались интенсивные поиски моделей и алгоритмов человеческого мышления и разработка первых программ на их основе [6]. Переход от теоретических поисков исследовательского характера к первым системам, обеспечивающим получение практических результатов, произошел в середине 1970-х годов, когда на смену поискам универсального алгоритма мышления пришла идея моделировать конкретные знания специалистов-экспертов. В это время появились первые системы, основанные на знаниях, или *экспертные системы* (ЭС) [6].

Большинство экспертных систем базируется на понятии "формальная продукционная система". Психологические исследования процессов принятия решений человеком показали, что рассуждая, человек использует правила, аналогичные продукциям, т. е. правила вида "условие → действие". Чтобы превратить продукционную систему ИИ в систему программирования, необходимо разработать методику, при которой осуществлялась бы

компиляция как правил вывода, так и экземпляра решателя, осуществляющего вывод на конкретном наборе правил. На этой основе и был создан гибридный вариант, получивший название экспертного программирования [7].

Продукционные правила, рассматриваемые как модули, естественным образом реализуют структуры "Следование" и "Альтернатива". Следование правил устанавливается по мере определения переменных, необходимых для их исполнения, а альтернативы реализуются условиями правил. Цикл "пока" в описываемой технологии реализован с помощью выделенной переменной *Fincalc*, при появлении которой организуется циклическое выполнение набора правил вплоть до выполнения определенного условия, изменяющего в одном из правил значение *Fincalc* из исходного 0 на 1.

Новым в описываемой технологии является рассмотрение продукционного правила в качестве динамической объект-функции [7]. Механизмы этой функции включают основной набор действий, необходимых для создания прикладных систем: вычисление по формулам (в том числе присвоение значений переменным), определение значений по таблицам, выбор значений из баз данных, обновление значений в базах данных, занесение значений в базы данных, вычисление значений с использованием подпрограмм, вычисление значений с помощью методов, сгенерированных из модулей знаний, вычисление значений с помощью исполняемых *exe*-модулей или *dll*-библиотек, сгенерированных другими системами.

Понятие "объект" имеет две взаимосвязанные и в то же время относительно самостоятельные стороны: набор свойств, значения которых определяют состояние объекта, и набор правил поведения, из которых формируется метод объекта. В данном случае набор свойств включает все переменные, входящие во входы, выходы и управление, а правила поведения определяются условиями и механизмами.

Описанная объект-функция относится к классу динамических, поскольку она обеспечивает выполнение действия, т. е. атомарного вычисления, которое приводит к изменению состояния системы или возврату значения.

На базе описанных принципов разработан новый вид программирования, получивший название *экспертного программирования* [7]. В экспертном программировании динамическая объект-функция получила название *модуль знаний*.

Выше была рассмотрена методика, которая обеспечивает компиляцию правил вывода. Для превращения продукционной системы ИИ в систему программирования была разработана методика компиляции экземпляра решателя, осуществляющего вывод на конкретном наборе правил [7].

Как показывает опыт, метод экспертного программирования доступен для непрограммирующих экспертов, обеспечивая генерацию эффективных программных кодов при повышении производительности процесса в 7–10 раз в сравнении с опытными программистами. Для реализации экспертного программирования разработана инструментальная система SprutExPro [7].

2. Структурная диаграмма синергетической системы ИИ

На рис. 1 представлена диаграмма классов объектов синергетической системы генерации знаний. Диаграмма разделена на три уровня знаний: математических, четких и нечетких. Ядром для построения синергетической системы ИИ являются четкие знания классического ИИ. Математика, оперируя только числовой информацией, не содержит в явном виде семантику своих моделей. Одни и те же математические модели могут быть применены в самых разных прикладных областях. В соответствии с трактовкой ИИ математические знания являются четкими, так как нечеткие знания строятся на базе лингвистических переменных. Для наполнения математических знаний семантикой над ними нужно надстроить систему четких знаний. При этом формальные математические переменные, связанные с переменными словаря базы знаний, получают смысловое содержание.

Вместе с тем, чтобы оперировать понятиями нечетких знаний, мы должны дать им четкие описания.

Фундаментом для построения четких знаний является словарь (см. рис. 1). Словарь имеет имя и методы, обеспечивающие сортировку и поиск слов, а также импорт слов из текстовых документов. Словарь состоит из слов, каждое из которых имеет имя-идентификатор общепринятое наименование и тип (целый, действительный или символьный). Существуют методы добавления и удаления слов, а также определения входимости слов в модули знаний. Со словами могут быть связаны ассоциативные списки допустимых значений. Ассоциативные списки, так же как и слова, имеют имя-идентификатор, наименование списка и тип значений. С ассоциативными списками связаны методы добавления, удаления, сортировки и поиска списка. Списки состоят из элементов, каждый из которых должен иметь значение и может быть добавлен или удален.

На основе словаря строятся модули знаний (см. рис. 1). Каждый модуль знаний (МЗ) имеет литературное наименование, имя-идентификатор, имя предусловия и версию. С модулем связаны методы добавления, выбора модуля-аналога, трансляции и тестирования модуля, определения входимости

модуля знаний в базы знаний и другие модули, а также удаления модуля.

Модуль имеет свой словарь, представляющий собой подмножество терминов из словаря базы знаний и включающий входные и выходные переменные. Кроме того, модуль может иметь предусловие, определяющее область определения модуля и содержащее набор взаимосвязанных логических выражений.

Как следует из структурной диаграммы на рис. 1, модуль знаний может быть составным и включать в себя другие модули. В качестве механизма здесь выступает метод, представляющий собой скомпилированную реализацию решателя для данного набора МЗ. При необходимости этот структурированный модуль может иметь собственное предусловие.

Особое положение в составе математических знаний имеют знания, получаемые с помощью технологий Data Mining (раскопки данных). В основу этих технологий положена концепция шаблонов (паттернов) и зависимостей, отражающих **многоаспектные взаимоотношения** в данных. Поиск паттернов проводится **автоматическими** методами, не ограниченными рамками априорных предположений о структуре выборки и виде распределений значений анализируемых показателей. Паттерны представляют собой разновидность МЗ и могут быть импортированы в базу знаний (см. рис. 1). Важное положение Data Mining — нетривиальность разыскиваемых паттернов. Это означает, что они должны отражать неочевидные, неожиданные (*unexpected*) регулярности в данных, составляющие так называемые скрытые знания (*hidden knowledge*).

3. Гибридизация четких и геометрических знаний

Геометрические знания представляют собой параметризованные геометрические образы. С точки зрения технологии программирования наиболее технологичным является представление таких знаний в форме подпрограмм на геометрическом языке. С точки зрения технологии ИИ знания должны представляться в формате, максимально приближенном к общепринятому в данной области. Таким образом, желательно иметь инструментальное средство, с помощью которого можно было бы строить геометрические образы с параллельным формированием геометрических подпрограмм. Подобным средством является система SprutCAD [7], пример интерфейса которой представлен на рис. 2 (см. вторую сторону обложки). Здесь мы видим чертеж зубчатого колеса, построенный средствами графического редактора, и программу на геометрическом языке, сгенерированную автоматически в процессе построения чертежа.

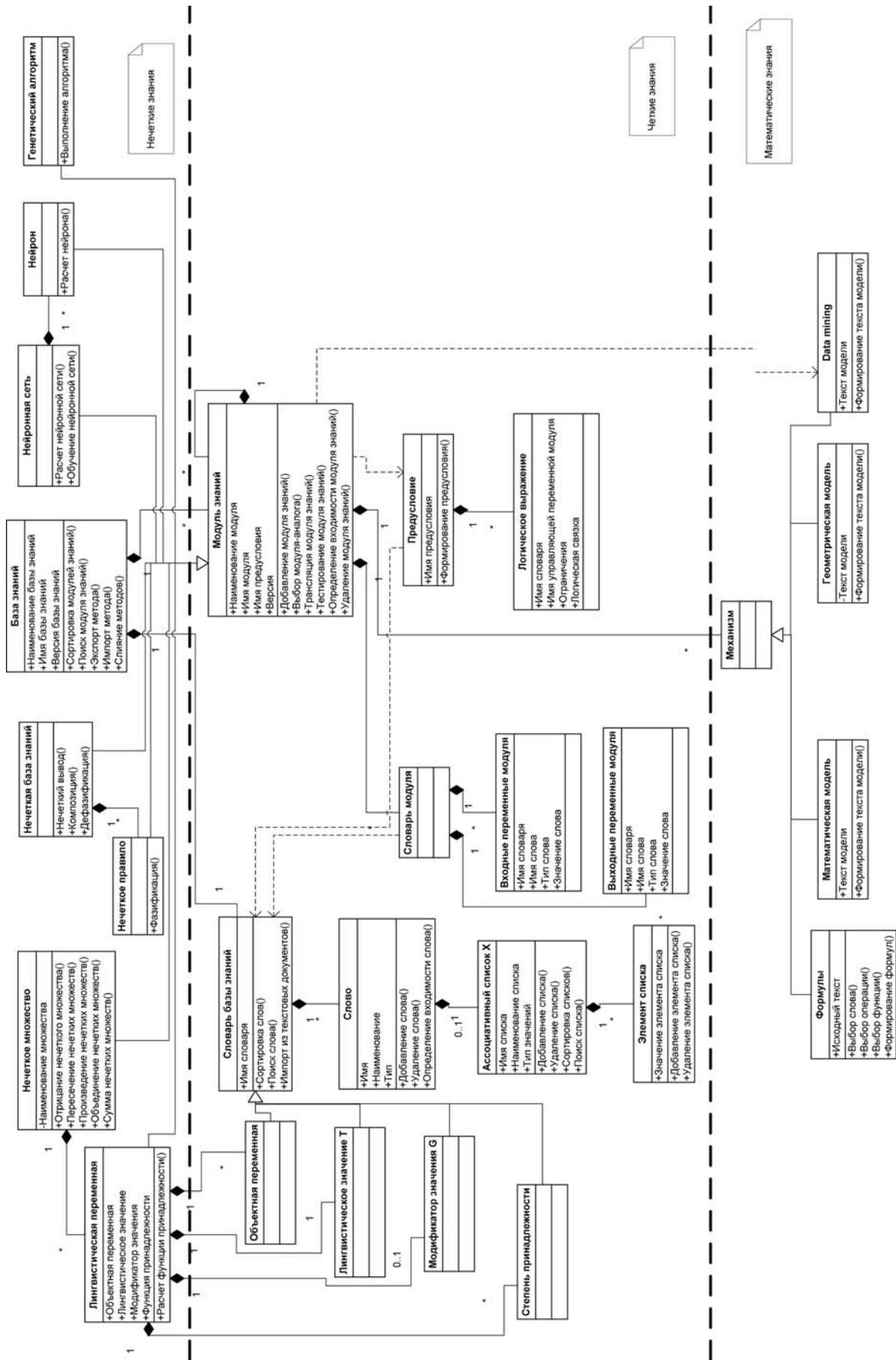


Рис. 1. Структурная диаграмма

МЗ: "GenDraw" - Генерация чертежа

Предусловия запуска

имя	наименование	тип	условие
-----	--------------	-----	---------

Входные свойства

имя	наименование	тип	значение
B_	Ширина венца, мм	REAL	30
dv_	D отверстия	REAL	10
d_	Делительный диаметр, мм	REAL	100
f_	Фаска	REAL	3
df_	D впадин	REAL	97
da_	D вершин	REAL	103
Alfa	Угол альфа	REAL	30

Механизм - Prt-модуль

Имя файла PRT-модуля: [каталог ресурсов\koleso.prt](#)

Согласование PRT-переменных и свойств МЗ

b_	вход	B_
d_	вход	d_
da_	вход	da_
df_	вход	df_
dv_	вход	dv_
f_	вход	f_
alfa_	вход	Alfa

Рис. 3. Символьное представление объект-функции вывода графических данных

Программа состоит из описаний точек (P), прямых (L) и контуров (K), которые сопровождаются комментариями, содержащими семантику геометрических элементов. Буквенные параметры, заключенные в круглых скобках (b_, da, d_, df, dv, f_), представляют собой входные переменные геометрической подпрограммы.

Для построения чертежа конкретной детали эти переменные должны быть заданы или рассчитаны с помощью модулей четких знаний.

Для вывода графической информации используются модули знаний с механизмом типа Prt-подпрограммы. На рис. 3 приведено внешнее представление такого модуля вывода чертежа зубчатого колеса. В этом случае модуль имеет только входные переменные, значения которых определены другими модулями.

4. Гибридизация четких и математических знаний

Геометрические знания представляют собой одну из разновидностей математических знаний, необходимых для построения интеллектуальных систем. Другой разновидностью математических знаний, необходимых для выполнения расчетов, являются модели непрерывных систем, основанные на дифференциально-алгебраических системах уравнений.

Для включения в состав интеллектуальной системы проектирования инструментальное средство, обладающее возможностями генерации упомянутых моделей, должно обеспечивать: поддержку технологии объектно-ориентированного моделирования, совместимую с языком UML; удобное и адекватное описание модели на общепринятом математическом языке без написания какого-либо программного кода; автоматическое построение компьютерной модели, соответствующей заданной математической, с возможностью автономного использования этой компьютерной модели.

В наибольшей мере этим требованиям отвечает пакет Model Vision Studium (MVS) [8]. Основным элементом языка MVS является *активный динамический объект* (АДО). Активный динамический объект MVS обеспечивает моделирование не только непрерывного поведения, но также дискретного или гибридного [8].

Каждый АДО является экземпляром соответствующего класса. Описание АДО включает в себя описание параметров, переменных (внешних и внутренних) и описание поведения. Внешние переменные являются видимой частью объекта (интерфейсом), а внутренние переменные и поведение инкапсулированы внутри объекта. Значения внутренних переменных могут изменяться только изнутри объекта. Значения внешних переменных могут изменяться как извне объекта, так и изнутри него. Значения параметров указываются при создании экземпляра объекта и далее являются неизменными на протяжении всего времени существования объекта. С каждой непрерывной переменной X (имеющей тип значения скалярный, вещественный, векторный или матричный) автоматически

связывается набор производных $\frac{dX}{dt}$, $\frac{d^2X}{dt^2}$, ..., не требующих специального описания.

По отношению к локальным объектам, которые здесь рассматриваются, содержащий их объект является объектом-контейнером. При построении экспертных баз знаний таким контейнером является объект-функция. Внешнее представление математического модуля знаний представлено на рис. 4. Модуль знаний с идентификатором **RsDP** предназначен для расчета дальности полета тела единичной массы, брошенного под углом к горизонту θ_0 со скоростью V_0 . Скорость должна принимать значения больше нуля, а угол — быть в пределах от нуля до $\pi/2$.

Для моделирования непрерывных систем необходим минимальный набор типов данных: скалярный вещественный тип, а также типы "вектор" и "матрица" со своими традиционными операциями, а кроме этого целые числа для вычисления индексов векторов и матриц.

МЗ: " RsDP " - Расчет дальности полета Предусловия запуска

имя	наименование	тип	условие
m	Масса, кг	REAL	1

Входные свойства

имя	наименование	тип	значение
V0	Скорость начальная, м/мин	REAL	(0,)
Teta0	D отверстия	REAL	(0,1.57)
g	Ускорение, м/сек^2	REAL	100

Механизм - DLL-модуль

Имя файла DLL-модуля: каталог ресурсов\pol.dll

Согласование PRT-переменных и свойств МИЗ

V0	ВХОД	V0_
Teta0	ВХОД	Teta0
g	ВХОД	g_
x	ВЫХОД	x_
y	ВЫХОД	y_

Выходные свойства

имя	наименование	тип	значение
x	Дальность полета, м	REAL	
y	Высота полета, м	REAL	

Рис. 4. Текстовое представление модуля знаний для решения системы дифференциальных уравнений

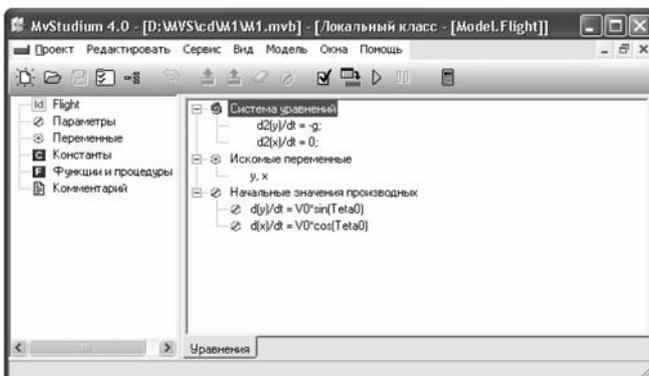


Рис. 5. Окно редактора формул пакета MVS

Собственное поведение непрерывного объекта задается системой дифференциально-алгебраических уравнений общего вида, т. е. системой, допускающей использование производных порядка выше первого и не разрешенной относительно производных. Уравнения могут задаваться как в скалярной, так и в матричной форме. Современный инструмент моделирования должен позволять вводить и редактировать уравнения в естественной математической форме (рис. 5). На этом рисунке при-

ведено окно редактора формул для представления системы уравнений тела единичной массы, брошенного под углом к горизонту, в скалярном виде на входном языке пакета MVS. Язык моделирования должен также позволять явно указывать набор искомых переменных. Например, в пакете MVS транслятор сам определяет допустимый набор искомых переменных, однако в ряде случаев этот набор может быть не единственным. Кроме того, для систем уравнений с производными порядка выше первого необходимо иметь возможность явно указывать начальные значения "младших" производных (по умолчанию им присваивается нулевое начальное значение).

В общем случае может потребоваться автоматическое преобразование исходных уравнений к форме, пригодной для численного решения [8].

5. Гибридизация четких знаний и нейросетевых моделей

Формальным нейроном называется элемент, используемый в узлах нейронной сети. Математическую модель формального нейрона можно представить уравнением [9]

$$y = f(g) = f\left(\sum_{i=2}^n w_i x_i + w_0\right),$$

где y — выходной сигнал нейрона; $f(g)$ — функция выходного блока нейрона; w_i — постоянный коэффициент — вес i -го входа; x_i — i -й входной сигнал; w_0 — начальное состояние (возбуждение) нейрона; $i = 1, 2, 3, \dots, n$ — номер входа нейрона; n — число входов.

Формальный нейрон может быть представлен с помощью описанных выше двух модулей знаний. Первый реализует входные блоки и сумматор, а второй — функцию активации. В качестве простого примера модуля, осуществляющего расчет взвешенной суммы входов нейрона, на рис. 6 приведено внешнее представление модуля знаний RSNeur11. Модуль не имеет предусловия запуска. Такой модуль запускается на исполнение, когда будут определены все входные свойства. В качестве входных свойств выступают входные переменные и синаптические коэффициенты. Выходом модуля является взвешенная сумма входов нейрона, которая вычисляется по формуле.

Вычисление значения функции активации осуществляется модулем RFNeur11 (рис. 7) Этот модуль имеет предусловие, которое определяет вид выбранной функции. В данном случае используется логистическая функция. Входные свойства модуля включают параметр функции активации и взвешен-

М3: "RSNeur11" - Расчет взвешенной суммы входов нейрона 1 слоя 1

Предусловия запуска

имя	наименование	тип	условие

Входные свойства

имя	наименование	тип	значение
w14	Вес связи 14	REAL	
x_2	Входная переменная 2	REAL	
x_0	Входная переменная 0	REAL	
x_3	Входная переменная 3	REAL	
x_8	Входная переменная 8	REAL	
x_9	Входная переменная 9	REAL	
w11	Вес связи 11	REAL	
w13	Вес связи 13	REAL	
w12	Вес связи 12	REAL	
x_1	Входная переменная 1	REAL	
w16	Вес связи 16	REAL	
w17	Вес связи 17	REAL	
w15	Вес связи 15	REAL	
w19	Вес связи 19	REAL	
w18	Вес связи 18	REAL	
x_6	Входная переменная 6	REAL	
x_4	Входная переменная 4	REAL	
x_5	Входная переменная 5	REAL	
x_7	Входная переменная 7	REAL	

Механизм - Формула

s11=x_0*x_1*w11+x_2*w12+x_3*w13+x_4*w14+x_5*w15+x_6*w16+x_7*w17+x_8*w18+x_9*w19

Выходные свойства

имя	наименование	тип	значение
s11	Взвешенная сумма входов нейрона 1 слоя 1	REAL	

Рис. 6. Внешнее представление модуля значений RSNeur11

М3: "RFNeur11" - Расчет логистической активационной функции нейрона 1 слоя 1

Предусловия запуска

имя	наименование	тип	условие
VidAktFn\$	Вид активационной функции	STRING	логистическая

Входные свойства

имя	наименование	тип	значение
a_	Параметр функции активации	REAL	
s11	Взвешенная сумма входов нейрона 1 слоя 1	REAL	

Механизм - Формула

y11 = 1 / (1 + 2.71828^(-a_*s11))

Выходные свойства

имя	наименование	тип	значение
y11	Активационная функция нейрона 1 слоя 1	REAL	

Рис. 7. Внешнее представление модуля RFNeur11

ную сумму входов нейрона. Параметр функции активации выбирается из базы данных со свойствами сети нейронов, а взвешенная сумма входов вычисляется предыдущим модулем.

Таким образом, приведенные выше два модуля знаний полностью реализуют классическую формальную модель нейрона. С помощью описанной технологии не составляет труда генерировать модели с другим числом входных переменных и синтаксических коэффициентов, а также с другими активационными функциями и даже с другими моделями нейронов.

Хотя один нейрон и способен выполнять простейшие процедуры распознавания, сила нейронных вычислений проистекает от соединения нейронов в сети.

Для построения продукционной модели нейронной сети генерируется метод, состоящий из набора модулей знаний. Этот метод в дальнейшем может использоваться в качестве модуля в более сложных базах знаний.

6. Гибридизация четких знаний и нечетких знаний. Лингвистическая переменная

Системы нечетких продукций строят на основе понятия "лингвистическая переменная" [1]. Лингвистической переменной называется пятерка $\{x, T(x)X, G, M\}$, где x — имя переменной; $T(x)$ — множество имен лингвистических значений переменной x , каждое из которых является нечетким множеством на множестве X ; G — синтаксическое правило для образования имен значений x ; M — семантическое правило для ассоциирования каждой величины значения с ее понятием.

Например, рассмотрим лингвистическую переменную, описывающую возраст человека, тогда: x — "возраст"; X — множество целых чисел из интервала $[1, 120]$; $T(X)$ — значения "молодой", "зрелый", "старый"; G — "очень", "не очень". Такие добавки позволяют образовывать новые значения: "очень молодой", "не очень старый" и пр. M : математическое правило, определяющее вид функции принадлежности для каждого значения из множества T .

В структуре четких знаний лингвистическая переменная представляется, в виде структурированного модуля знаний (метода). Ниже представлен словарь метода (см. таблицу)

Входной является числовая переменная x , представляющая собой в данном случае возраст. Символьная переменная "Значения возраста" $\{T(X)\}$ имеет ассоциативный список: "молодой", "зрелый" и "старый". Символьная переменная "Модификаторы возраста" $\{G\}$ имеет ассоциативный список: "очень" и "не очень". Помимо этих символьных переменных в числовых выходов входит "Функция принадлежности". Остальные переменные метода носят вспомогательный характер и являются локальными.

Словарь метода

Имя	Наименование	Тип	Вид
x_	Переменная x	REAL	вход
Age_G\$	Модификатор возраста	STRING	выход
Age_TS	Значения возраста	STRING	выход
FnPr	Функция принадлежности	REAL	выход
a_	Константа функции принадлежности	REAL	локал.
dax	Разность a - x	REAL	локал.
dxb	Разность x - b	REAL	локал.
b_	Константа 2 функции принадлежности	REAL	локал.

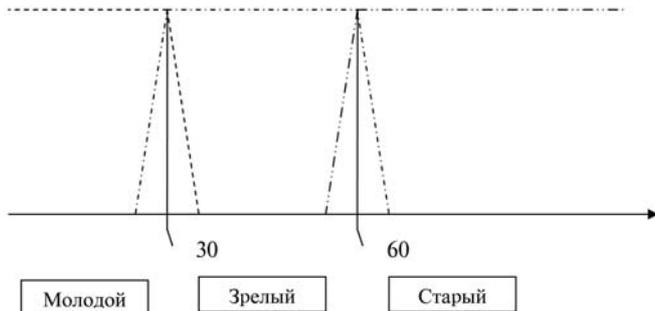


Рис. 8. Шкала возрастов

В описываемом модуле была использована шкала возрастов, представленная на рис. 8. Функция принадлежности равна 1: для значения "молодой" — до 30 лет, для значения "зрелый" — от 30 до 60 лет, для значения "старый" — от 60 лет. Модификатор возраста действует при отклонениях от этих значений в пределах 5 лет. Например, до 35 лет — не очень молодой.

Для значения возраста "молодой" используется функция принадлежности класса L , которая определяется как:

$$L(x; a, b) = \begin{cases} 1, & x \leq a, \\ \frac{b-x}{b-a}, & a \leq x \leq b, \\ 0, & x \geq b. \end{cases}$$

Расчет этой функции осуществляется структурированным модулем (рис. 9).

Используемые здесь константы в соответствии с принятой шкалой назначаются с помощью модуля, приведенного на рис. 10.

М3: "MRsFPPrL" - Метод расчета функции принадлежности класса L

Предусловия запуска

имя	наименование	тип	условие
Age	TSЗначения возраста	STRING	молодой

Входные свойства

имя	наименование	тип	значение
x	Переменная x	REAL	
a	Константа функции принадлежности	REAL	
b	Константа 2 функции принадлежности	REAL	

Механизм - Внешний Метод

База Знаний:	текущая База Знаний - Fuzzy.SDB
Метод:	RsFPPrLРасчет функции принадлежности класса L

Согласование свойств внешнего метода и свойств МИЗ

a	Константа функции принадлежности	=a	Константа функции принадлежности
b	Константа 2 функции принадлежности	=b	Константа 2 функции принадлежности
x	Переменная x	=x	Переменная x
FnPr	Функция принадлежности	=FnPr	Функция принадлежности

Выходные свойства

имя	наименование	тип	значение
FnPr	Функция принадлежности	REAL	

Рис. 9. Структурированный модуль MRsFPPrL

М3: "NzPrAge" - Назначение параметров возраста

Предусловия запуска

имя	наименование	тип	условие
-----	--------------	-----	---------

Механизм - Формула

$$\begin{aligned} a_ &= 30 \\ b_ &= 60 \end{aligned}$$

Выходные свойства

имя	наименование	тип	значение
b	Константа 2 функции принадлежности	REAL	
a	Константа функции принадлежности	REAL	

Рис. 10. Модуль NzPrAge

Для значения возраста "зрелый" используется функция принадлежности трапециoidalного класса:

$$MF(x) = \begin{cases} 1 - \frac{b-x}{b-a}, & a \leq x \leq b, \\ 1, & b \leq x \leq c, \\ 1 - \frac{x-c}{d-c}, & c \leq x \leq d, \\ 0, & \text{в остальных случаях.} \end{cases}$$

Для возраста "старый" используется функция принадлежности класса γ , которая определяется как

$$\gamma(x; a, b) = \begin{cases} 0, & x \leq a, \\ \frac{x-a}{b-a}, & a \leq x \leq b, \\ 1, & x \geq b. \end{cases}$$

Расчет этих функций осуществляется структурированными модулями, аналогичными приведенному выше для возраста "молодой".

Из числа выходных переменных осталось рассмотреть модули генерации модификаторов возраста. Наиболее интересным является назначение модификаторов в окрестностях точек 30 и 60 (см. рис. 8). В этих окрестностях база знаний задает вопрос пользователю: к какому возрасту отнести человека. Например, когда $x = 31$ задается вопрос, отображенный на рис 11.

При выборе значения "молодой" имеем функцию принадлежности 0,8 и модификатор "не очень", а при выборе "зрелый" — функцию принадлежности 1 и модификатор "очень".

Назначение модификатора проводится нижеприведенным правилом с механизмом типа "таблица". Модификатор "очень" присваивается, когда функция принадлежности равна 1, а "не очень", когда эта функция меньше 1 (рис. 12).



Рис. 11. Вопрос базы знаний

МЗ: "NzAgG" - Назначение модификатора возраста

Предусловия запуска

имя	наименование	тип	условие
FnPr	Функция принадлежности	REAL	

Входные свойства

имя	наименование	тип	значение
FnPr	Функция принадлежности	REAL	

Механизм - Таблица

Конфигурация свойств в таблице

FnPr	Age_G\$
------	---------

Таблица

1	очень
[0,1)	не очень

Выходные свойства

имя	наименование	тип	значение
Age_G\$	Модификаторы возраста	STRING	

Рис. 12. Модуль "Назначение модификатора возраста"

Основой для проведения операции нечеткого логического вывода является база правил, содержащая нечеткие высказывания в форме "Если—то" и функции принадлежности для соответствующих лингвистических переменных (термов). При этом должны соблюдаться приведенные ниже условия.

1. Существует хотя бы одно правило для каждого лингвистического термина выходной переменной.

2. Для любого термина входной переменной имеется хотя бы одно правило, в котором этот терм используется в качестве предпосылки (левая часть правила).

В противном случае имеет место неполная база нечетких правил.

Пусть в базе правил имеется m правил вида:

R_1 : ЕСЛИ x_1 это A_{11} ... И ... x_n это A_{1n} , ТО y это B_1
...

R_i : ЕСЛИ x_1 это A_{i1} ... И ... x_n это A_{in} , ТО y это B_i

...

R_m : ЕСЛИ x_1 это A_{m1} ... И ... x_n это A_{mn} , ТО y это B_m ,

где x_k , $k = 1...n$ — входные переменные; y — выходная переменная; A_{ik} — заданные нечеткие множества с функциями принадлежности.

Результатом нечеткого вывода является четкое значение переменной y^* , на основе заданных четких значений x_k , $k = 1...n$.

В общем случае механизм логического вывода включает четыре этапа: введение нечеткости (фазификация), нечеткий вывод, композиция и приведение к четкости, или дефазификация. Фазификация с использованием предлагаемой технологии приведена выше на примере лингвистической переменной "возраст". Дефазификация проводится аналогично путем построения соответствующего набора модулей знаний. Нечеткий вывод в предлагаемой технологии в принципе не отличается от четкого аналога.

Заключение

Разработаны теоретические основы синергетической методологии интеграции основных видов знаний. Показано, что методы гибридизации являются средством создания синергетических систем, представляющих собой междисциплинарное направление в области создания прикладных интеллектуальных программных средств. Выполнена практическая реализация разнообразных синергетических систем в форме демонстрационных прототипов с применением созданной информационной технологии, доступной для использования непрограммирующими пользователями.

Список литературы

1. Ярушкина Н. Г. Основы теории нечетких и гибридных систем: учеб. пособие. М. Финансы и статистика, 2004. 320 с.
2. Тарасов В. Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. — М.: Эдиториал УРСС. 2002. 352 с.
3. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. Пер. с англ. М.: Бином, СПб.: Невский диалект, 2000. 560 с.
4. Лингер Р. Миллс Х., Уитт Б. Теория и практика структурного программирования: Пер. с англ. М.: Мир, 1982. 406 с.
5. Иванова Г. С., Ничушкина Т. Н., Пугачев Е. К. Объектно-ориентированное программирование: учеб. для вузов / Под ред. Г. С. Ивановой М.: Изд-во МГТУ им. Н. Э. Баумана, 2001. 320 с.
6. Попов Э. В. Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ. М.: Наука, 1987. 288 с.
7. Евгеньев Г. Б. Интеллектуальные системы проектирования: учеб. пособие для вузов М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. 320 с.
8. Колесов Ю. Б. Объектно-ориентированное моделирование сложных динамических систем. СПб.: Изд-во СПбГПУ, 2004. 239 с.
9. Комарцова Л. Г., Максимов А. В. Нейрокомпьютеры: учеб. пособие для вузов. М.: Изд-во МГТУ им. Н. Э. Баумана, 2004. 400 с.

С. И. Колесникова, канд. физ.-мат. наук, доц.,
Томский госуниверситет
систем управления и радиоэлектроники
e-mail: skolesnikova@yandex.ru

Особенности применения линейной свертки критериев в методе парных сравнений¹

Рассматривается проблема, касающаяся нежелательных эффектов метода парных сравнений Саати Т., связанных с применением линейной свертки при получении весовых коэффициентов альтернатив. Некорректное применение свертки может привести к неточности принятия итогового решения. Исследуются свойства модифицированной процедуры парных сравнений, позволяющей обойти указанные противоречия классического метода, даются иллюстративные примеры.

Ключевые слова: метод анализа иерархий, весовые коэффициенты альтернатив, интеллектуальная система, тестовое распознавание образов, качество принятия решения

Введение в проблему

Одним из самых популярных методов, используемых в информационно-аналитических и интеллектуальных системах поддержки принятия решений [1], является метод анализа иерархий (МАИ) — простое и удобное средство, позволяющее структурировать проблему, оценивать построенный набор альтернатив по каждому из заданных критериев, находить неточности и противоречия в суждениях эксперта, ранжировать альтернативы [2].

В общем виде постановка задачи, решаемой МАИ [2], включает цель, альтернативы и критерии оценки альтернатив; требуется выбрать наилучшую альтернативу. После построения иерархической структуры (цели—критерии—альтернативы) система парных сравнений элементов каждого уровня приводит к результату, который может быть представлен в виде обратно-симметричной матрицы \mathbf{A} , элемент которой a_{ij} есть интенсивность проявления альтернативы (элемента иерархии) i относительно альтернативы j в смысле выбранного фиксированного критерия. Главные собственные векторы матрицы \mathbf{A} парных сравнений интерпретируются как векторы приоритетов сравниваемых элементов. Для элементов каждого уровня вычис-

ляются коэффициенты важности для каждой из альтернатив в виде произведения в случае, когда оценки элементов заданы в виде отношения (мультипликативный МАИ).

Обозначим формируемую на каждом этапе матрицу парных сравнений (МПС) $\mathbf{A} = \|a_{ij}\|_{g \times g}$ альтернатив (далее в контексте работы будем предполагать понятия альтернативы и признака идентичными) $\Theta = (z_1, z_2, \dots, z_g)$. Матрица $\mathbf{A} = \|a_{ij}\|_{g \times g}$, $a_{ij} = w_i w_j^{-1}$ относительных весовых коэффициентов w_i, w_j — компонент весового вектора $\mathbf{w} = (w_1, w_2, \dots, w_g)^T$, где g — число сравниваемых альтернатив, должна обладать следующими свойствами [2]:

- 1) $a_{ij} > 0$; 2) $a_{ij} = a_{ji}^{-1}$;
- 3) $a_{ij} = a_{ik} a_{kj}$, $i, j = \overline{1, g}$; 4) $\mathbf{A}\mathbf{w} = g\mathbf{w}$,

Число g является максимальным собственным значением матрицы \mathbf{A} , и для некоторого единственного (нормированного) вектор-столбца $\mathbf{w} = (w_1, \dots, w_g)^T$ с положительными компонентами выполняется равенство 4).

Обозначим $X = \{x\}$ — множество выбираемых решений-альтернатив x , $f(x)$ — числовой критерий на множестве X .

Решение $x' \in X$ называется оптимальным по Парето (Парето-оптимальным) [3, 4], если не существует такого возможного решения $x \in X$, для которого имеет место неравенство $f(x) > f(x')$. Все Парето-оптимальные решения образуют множество Парето, обозначаемое $P_f(X)$.

Бинарное отношение предпочтительности (доминирования) альтернатив (признаков) обозначается знаком " $>$ ", например, $z_2 > z_1$.

Как известно [2], в МАИ используется линейная свертка критериев. В ряде работ [3, 4] отмечено, что в определенном классе многокритериальных задач с помощью линейной свертки в принципе невозможно найти "наилучшее" решение, и поэтому ее использование становится некорректным. Отмечено также, что при выборе наилучшего способа решения задачи многокритериального выбора следует руководствоваться следующим принципом: всякое решение, которое потенциально может оказаться выбранным в результате решения многокритериальной задачи, может быть получено в результате максимизации используемой свертки критериев с некоторыми неотрицательными коэффициентами c_1, c_2, \dots, c_m , $\sum_i c_i = 1$. Другими словами у каждого элемента, принадлежащего опти-

¹ Работа поддержана РФФИ (проекты № 09-01-99014-р_офи, № 10-01-00462-а).

мальной (по Парето) области множества X , должна существовать возможность быть выбранным в качестве наилучшего решения. В противном случае "наилучшее" решение может и не быть найдено вообще в смысле экстремума выбранного способа скаляризации, в частности, в результате максимизации линейной свертки критериев в МАИ.

Нежелателен также и эффект традиционного применения МАИ, приводящего к тому, что предпочтения, выявленные на всем множестве альтернатив, могут не совпадать с "частными" предпочтениями на подмножестве альтернатив. В работах [5, 6] предлагается следующая схема решения проблемы единичной нормировки, согласно которой порядок предпочтения альтернатив на множестве метрических критериев строится на основе относительного (взаимного) превосходства.

Для каждой пары альтернатив с номерами (i, j) определяются нормированные по s -му критерию значения весовых коэффициентов альтернатив (ВКА) v_{is}, v_{js} (по терминологии работы [5] значения s -го критерия в i -й и j -й альтернативах), т. е. $v_{is} + v_{js} = 1$. Далее вычисляются выражения

$$D_i = \sum_{s=1}^v c_s v_{is}, \quad D_j = \sum_{s=1}^v c_s v_{js}$$

и строится обратносимметричная матрица из отношений $\gamma_{ij} = D_j/D_i$, а именно: $\mathbf{B} = \|b_{ij}\|$, $b_{ij} = \gamma_{ij}$, $b_{ji} = 1/b_{ij}$. Величина γ_{ij} называется коэффициентом относительного превосходства i -й альтернативы. Далее рекомендуется ненормированный вектор приоритетов альтернатив $\mathbf{w} = (w_1, \dots, w_g)^T$ находить как собственный вектор (как и в МАИ [2]) из решения матричного уравнения: $\mathbf{Bw} = \lambda_{\max} \mathbf{w}$, где λ_{\max} — максимальное собственное значение матрицы \mathbf{B} .

Заметим, однако, что выполнение свойства транзитивности ($b_{ij} = b_{ik}b_{kj}$), являющееся весьма важным условием для данного вывода (см. ниже утверждение), в данном случае не гарантируется. Действительно, обозначив через v'_{is}, v'_{js} ненормированные по s -му критерию значения ВКА, получаем выражения: $v_{is} = v'_{is}/(v'_{is} + v'_{js})$, $v_{js} = v'_{js}/(v'_{is} + v'_{js})$. Отсюда следует, величина γ_{ij} — это не просто коэффициент относительного превосходства i -й альтернативы, а по сравнению с j -й альтернативой, т. е. $D_i = D_{i/j}, D_j = D_{j/i}$:

$$D_{i/j} = \sum_{s=1}^v c_s v_{is} = \sum_{s=1}^v c_s \frac{v'_{is}}{v'_{is} + v'_{js}},$$

$$D_{j/i} = \sum_{s=1}^v c_s v_{js} = \sum_{s=1}^v c_s \frac{v'_{js}}{v'_{is} + v'_{js}}.$$

Тогда получаем: $b_{ik}b_{kj} = \gamma_{ik}\gamma_{kj} = (D_{i/k}/D_{k/i}) \times (D_{k/j}/D_{j/k})$; $b_{ij} = (D_{i/j}/D_{j/i})$ и вопрос о справедливости $b_{ij} = b_{ik}b_{kj}$, таким образом, остается открытым.

Утверждение. Вектор ВКА является собственным вектором, соответствующим максимальному собственному значению λ_{\max} , только в случае совместной (согласованной) матрицы парных сравнений \mathbf{A} .

Доказательство утверждения непосредственно следует из вида характеристического уравнения для \mathbf{A} , имеющего место при выполнении свойств матрицы парных сравнений 1)–4) [2, 4]:

$$\det(\mathbf{A} - \lambda \mathbf{E}) = \det \begin{bmatrix} (1-\lambda) & 1 & \dots & 1 \\ 1 & (1-\lambda) & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & (1-\lambda) \end{bmatrix} =$$

$$= (-1)^g \lambda^{g-1} (\lambda - g) = 0,$$

из которого следует, что матрица \mathbf{A} будет иметь только два собственных значения (при выполнении условий 1)–4)), при этом значение g является максимальным. В противном случае собственный вектор несовместной МПС будет соответствовать максимальному собственному значению, которое строго больше g , а не равно g .

Настоящая работа является продолжением исследований, проводившихся в работах [5–7]. В статье даны свойства приведенной и иллюстрированной модифицированной процедуры МАИ (ММАИ), а именно, доказано, что:

- предпочтения, индуцированные на множествах (альтернатив, признаков) Θ_1 и Θ_2 ($\Theta_1 \subset \Theta_2$) посредством применения классической процедуры МАИ Саати, в общем случае не совпадают;
- в модифицированной процедуре МАИ указанное противоречие устраняется;
- "наилучшее" решение, выбираемое по ММАИ, принадлежит Парето-оптимальному множеству.

Постановка задачи

Требуется построить процедуру оценки значимости альтернатив по совокупности заданных критериев, во-первых, сохраняющей бинарные отношения (предпочтения) при изменении (добавлении дополнительной альтернативы, или удалении альтернативы из исследуемой совокупности); во-вторых, гарантирующей корректное применение линейной свертки, а также определить ее свойства и класс решаемых задач.

Прежде чем перейти к решению задачи, обсудим примеры из работ [4–8], иллюстрирующие негативные стороны классического применения МАИ, приводящие к неточности в принятии решения.

Пример 1. Пусть задача состоит в выборе прямоугольного участка [4] из следующих трех вариантов: (I) 7×15 , (II) 10×10 , (III) 5×20 (измерение проводится в некоторых единицах длины). Первый участок, очевидно, имеет максимальную площадь. Однако при использовании МАИ с равновесными метрическими критериями (длина и ширина участка) и аддитивной сверткой критериев, ошибочный выбор будет сделан в пользу второго участка. Действительно, соответствующие МПС имеют вид [2]:

$$A_1 = \begin{bmatrix} 1 & 7/10 & 7/5 \\ 10/7 & 1 & 10/5 \\ 5/7 & 5/10 & 1 \end{bmatrix}; A_2 = \begin{bmatrix} 1 & 15/10 & 15/20 \\ 10/15 & 1 & 10/20 \\ 20/15 & 20/10 & 1 \end{bmatrix}.$$

Оценки собственных векторов МПС A_1 и A_2 равны $w^1 = (0,318, 0,455, 0,227)$, $w^2 = (0,333, 0,222, 0,444)$, соответственно, итоговый весовой вектор равен $w = (0,326, 0,338, 0,336)$.

Заметим, что при использовании мультипликативной свертки весовой вектор по равновесным критериям в примере 1 будет равен $(0,318^{0,5} \cdot 0,333^{0,5}, 0,455^{0,5} \cdot 0,222^{0,5}, 0,227^{0,5} \cdot 0,444^{0,5}) = (0,326, 0,319, 0,319)$, или в нормализованном виде: $w = (0,338, 0,331, 0,331)$, что тоже соответствует верному выбору. Известно [3, 4], что максимальное значение мультипликативной свертки на некотором решении (в данном случае точки с координатами (ширина, длина)) является достаточным условием Парето-оптимальности этого решения.

Пример 2. Пусть оцениваются две альтернативы (признака) z_1, z_2 по двум равновесным мерам относительной важности со следующими МПС [5, 6]:

$$A_1 = \begin{bmatrix} 1 & 1/2 \\ 2 & 1 \end{bmatrix}; A_2 = \begin{bmatrix} 1 & 4 \\ 1/4 & 1 \end{bmatrix}.$$

В результате применения МАИ получим для альтернатив z_1, z_2 вектор ВКА $w = (w_1, w_2) = (0,567; 0,433)$. Так как $w_1 > w_2$, то $z_1 > z_2$. При оценивании трех альтернатив z_1, z_2, z_3 по тем же двум мерам относительной важности со следующими МПС:

$$A_1 = \begin{bmatrix} 1 & 1/2 & 3 \\ 2 & 1 & 6 \\ 1/3 & 1/6 & 1 \end{bmatrix}; A_2 = \begin{bmatrix} 1 & 4 & 1/2 \\ 1/4 & 1 & 1/8 \\ 2 & 8 & 1 \end{bmatrix}.$$

В результате классического применения МАИ по-

лучаем $w = (w_1, w_2, w_3) = (0,304, 0,338, 0,358)$, т. е. $z_2 > z_1$.

Пример 3. Третий пример касается тестовых методов распознавания образов (основы одного из наиболее эффективных подходов к созданию интеллектуальных систем), использующих для принятия решений наборы (тесты) [1], содержащие меньшее число признаков и с большим весом, где под весом теста понимается сумма весовых коэффициентов признаков. Новый метод определения "весов" сравниваемых признаков, изложенный в работе [9], учитывает вклад признаков в распознающую способность теста с учетом их взаимозависимости. При этом для вычисления весовых коэффициентов признаков используется метрический вариант МАИ с мерами относительной важности признаков, выбранных специальным образом.

Нижеприведенный пример связан с целесообразностью корректного оценивания признаков, входящих в тест, при удалении одного из признаков из анализируемого набора (с целью повышения обобщающей способности теста).

Пусть матрица Q (объект-признак) содержит описание шести объектов, в матрице R указывается на соответствие номеров объектов и классов, которым объекты принадлежат. Для данного примера тупиковыми тестами (матрица T) являются наборы признаков $\Theta_1 = \{z_1, z_2, z_3\}$, $\Theta_2 = \{z_1, z_2, z_4\}$ и $\Theta_3 = \{z_2, z_3, z_4\}$:

$$Q = \begin{matrix} & z_1 & z_2 & z_3 & z_4 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 0 & 2 \\ 1 & 1 & 1 & 2 \end{bmatrix} \end{matrix}; R = \begin{matrix} & & & & & \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \end{bmatrix} \end{matrix}; T = \begin{matrix} & z_1 & z_2 & z_3 & z_4 \\ \begin{matrix} 1 \\ 1 \\ 0 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \end{matrix}.$$

Корректное применение тестового алгоритма по принципу совпадения значений всех признаков [8] не позволит новый объект $O = (0, 1, 2, 1)$ отнести к одному из выделенных классов, однако фрагмент $(0, 1)$, порождаемый набором $\Theta = \{z_1, z_2\}$, содержится в O и соответствующих объектах из первого класса и не содержится в объектах из второго класса, что дает нам основание полагать, что распознаваемый объект более близок к первому классу. Непосредственное применение МАИ в методе из работы [9] может привести к ошибочному определению весовых коэффициентов признаков, а, следовательно, и к неточности в принятии решения.

Решение задачи. Модифицированная процедура МАИ и ее свойства

Согласно работам [5, 6] укажем основные этапы ММАИ и перечислим ее свойства.

I. Построим МПС на каждом из ν этапов (по числу ν мер относительной важности одной альтернативы над другой). Результатом каждого s -го уровня иерархии является g -компонентный вектор нормализованных значений ВКА — $\mathbf{w}_s = (w_1^s, w_2^s, \dots, w_g^s)$. Введем весовые коэффициенты мер относительной важности альтернатив c_s ,

$$s \in \{1, 2, \dots, \nu\}, \quad \sum_{s=1}^{\nu} c_s = 1.$$

II. Сформируем всевозможные векторы $\mathbf{w}_{ij}^s = (w_{ij}^s(i), w_{ij}^s(j))$ локальных ВКА уровня I по формулам строки III приведенной ниже таблицы.

III. Сформируем матрицу $\mathbf{W} = \|\mathbf{w}_{ij}\| = \|(w_{ij}(i), w_{ij}(j))\|$, где векторы $(w_{ij}(i), w_{ij}(j))$ — локальные нормализованные ВКА z_i, z_j уровня 2 относительно всей совокупности мер относительной важности признаков, компоненты которых находим по формулам строки IV таблицы.

IV. Глобальные значения ВКА считаем по одной из формул [2]:

$$V_i^{(1)} = \frac{1}{g} \sum_{j=1}^g w_{ij}(i), \quad V_i^{(2)} = \left(\prod_{j=1}^g w_{ij}(i) \right)^{1/g}, \quad i = \overline{1, g}.$$

Свойства обобщенной процедуры МАИ сформируем в виде трех теорем. Обозначим $\rho_1 = \{(z_i, z_j): z_i \succ z_j\}$ ($\rho_2 = \{(z_i, z_j): z_j \succ z_i\}$).

Теорема 1. Пусть заданы множества (наборы, тесты) признаков $\Theta_1 = \{z_1, z_2, \dots, z_{g-1}\}$ и $\Theta_2 = \{z_1, z_2, \dots, z_{g-1}, z_g\}$ ($\Theta_1 \subset \Theta_2$). Бинарные отношения (предпочтения) $z_i \rho_1 z_j, z_i \rho_2 z_j, z_i, z_j \in \Theta_1 \subset \Theta_2, i \neq j$, индуцированные на множествах Θ_1 и Θ_2 посредством применения классической процедуры МАИ, в общем случае не совпадают.

Теорема 2. Пусть заданы множества (наборы, тесты) альтернатив Θ_1 и $\Theta_2, \Theta_1 \subset \Theta_2$. Бинарные отношения (предпочтения) $z_i \rho_1 z_j, z_i \rho_2 z_j, z_i, z_j \in \Theta_1 \subset \Theta_2, i \neq j$, индуцированные посредством применения классической процедуры МАИ на множестве Θ_1 и модифицированной процедуры ММАИ на множестве Θ_2 , совпадают.

Из приведенной выше таблицы следует, что функция, на основе которой принимается решение в ММАИ (назовем ее функцией относительного сходства i -й альтернативы по всей совокупности альтернатив и критериев), имеет вид:

$$f_i = \frac{1}{g} \sum_{j=1}^g w_{ij}(i) \left(f_i = \left[\prod_{j=1}^g w_{ij}(i) \right]^{1/g} \right),$$

$$w_{ij}(i) = \sum_{s=1}^{\nu} c_s w_{ij}^s(i) = \sum_{s=1}^{\nu} c_s \frac{w_i^s}{w_i^s + w_j^s},$$

$$w_i^s = \left(\prod_{l=1}^g a_{il}^s \right)^{1/g} \left[\sum_{k=1}^g \left(\prod_{l=1}^g a_{kl}^s \right)^{1/g} \right]^{-1},$$

величины $a_{kl}^s = w_k^s / w_l^s$ — элементы МПС по s -му критерию МАИ.

Этапы модифицированного МАИ

Промежуточные исходные данные s -го уровня иерархии, $(w_1^s, w_2^s, \dots, w_g^s)$	Относительные ВКА z_i, z_j	Вектор относительных нормализованных ВКА
I $w_i^s = \left(\prod_{l=1}^g a_{il}^s \right)^{1/g}, i = \overline{1, g}$	$w_i^s = w_i^s / \sum_{l=1}^g w_l^s, i = \overline{1, g}$	$(w_1^s, w_2^s, \dots, w_g^s), s \in \{1, 2, \dots, \nu\}$
II w_i^s, w_j^s	$w_{ij}^s(i) = \frac{w_i^s}{w_i^s + w_j^s}; w_{ij}^s(j) = \frac{w_j^s}{w_i^s + w_j^s}$	$\mathbf{w}_{ij}^s = (w_{ij}^s(i), w_{ij}^s(j))$
III $w_{ij}(i) = \sum_{s=1}^{\nu} c_s w_{ij}^s(i), w_{ij}(j) = \sum_{s=1}^{\nu} c_s w_{ij}^s(j), \sum_{s=1}^{\nu} c_s = 1$		$(w_{ij}(i), w_{ij}(j))$
IV $u_i = \sum_{l=1}^g w_{il}(i)$	$V_i = u_i / \sum_{l=1}^g u_l, i = \overline{1, g}$	(V_1, V_2, V_g)

Сформулируем достаточное условие Парето-оптимальности функции относительного сходства в терминах множества выбираемых решений-альтернатив X и множества, оптимального по Парето $P_f(X)$.

Теорема 3. Если функция относительного сходства в точке x' достигает максимального значения на множестве выбираемых решений, то x' входит в Парето-оптимальное множество.

Доказательства теорем 1 и 2 следуют из работы [7], свойств МПС 1)–4) и формул из таблицы; доказательство теоремы 3 опирается на исследования [3, 4] и вид функции относительного сходства f .

Следует также отметить, что свойства МПС 1)–4) являются условиями применения МАИ и считаются по умолчанию выполненными. Невыполнение указанных свойств приводит к некорректному использованию МАИ (см. утверждение и работу [4]).

Иллюстративный пример применения ММАИ

Покажем этапы ММАИ на данных примера 2.

Этап 1. Пусть МПС альтернатив (признаков) составлена по каждому из двух данных равновесных критериев и посчитаны весовые коэффициенты. Матрицы парных сравнений \mathbf{B}_I^1 и \mathbf{B}_{II}^1 представлены для удобства в расширенном виде, включая и оценки собственных векторов \mathbf{w}_I^j и их нормализованные значения $\mathbf{w}_{I\text{норм}}^j$ (верхний индекс указывает номер критерия, нижний — номер этапа). Вычисления проведем с точностью до трех знаков.

$$\mathbf{B}_I^1 = \begin{array}{ccccc} & z_1 & z_2 & z_3 & \mathbf{w}_1^1 & \mathbf{w}_{I\text{норм}}^1 \\ \begin{bmatrix} 1,000 & 0,500 & 3,000 & 1,145 & 0,300 \\ 2,000 & 1,000 & 6,000 & 2,289 & 0,600 \\ 0,333 & 0,167 & 1,000 & 0,382 & 0,100 \end{bmatrix}; \end{array}$$

$$\mathbf{B}_I^2 = \begin{array}{ccccc} & z_1 & z_2 & z_3 & \mathbf{w}_1^2 & \mathbf{w}_{I\text{норм}}^2 \\ \begin{bmatrix} 1,000 & 4,000 & 0,500 & 1,260 & 0,308 \\ 0,250 & 1,000 & 0,125 & 0,315 & 0,077 \\ 2,000 & 8,000 & 1,000 & 2,520 & 0,615 \end{bmatrix}. \end{array}$$

По двум равновесным критериям оценки альтернатив, следуя традиционному МАИ, будут равны $\mathbf{w}^0 = (0,316, 0,352, 0,332)$. Согласно модифицированному МАИ далее следуют действия.

Этап 2. По двум критериям формируем МПС $\mathbf{B}_{II}^1 = \|\mathbf{w}_{ij}^1\| = \|(w_i^1, w_j^1)\|$ и $\mathbf{B}_{II}^2 = \|\mathbf{w}_{ij}^2\| = \|(w_i^2, w_j^2)\|$:

$$\mathbf{B}_{II}^1 = \begin{bmatrix} (1,000; 1,000) & (0,300; 0,600) & (0,300; 0,100) \\ (0,600; 0,300) & (1,000; 1,000) & (0,600; 0,100) \\ (0,100; 0,300) & (0,100; 0,600) & (1,000; 1,000) \end{bmatrix};$$

$$\mathbf{B}_{II}^2 = \begin{bmatrix} (1,000; 1,000) & (0,308; 0,077) & (0,308; 0,615) \\ (0,077; 0,308) & (1,000; 1,000) & (0,077; 0,615) \\ (0,615; 0,308) & (0,615; 0,077) & (1,000; 1,000) \end{bmatrix}.$$

Далее нормализуем по каждому критерию каждую пару элементов сравнения альтернатив и получаем относительные локальные ВКА (по формулам строки II 3-го и 4-го столбцов в таблице), например, $w_{12}^1 = \left(\frac{0,300}{0,300 + 0,600}; \frac{0,600}{0,300 + 0,600} \right) = (0,333; 0,667)$, и оформляем их в матрицы нормализованных локальных ВКА по каждому критерию:

$$\mathbf{B}_{II}^1 = \begin{bmatrix} (0,500; 0,500) & (0,333; 0,667) & (0,750; 0,250) \\ (0,667; 0,333) & (0,500; 0,500) & (0,857; 0,143) \\ (0,250; 0,750) & (0,143; 0,857) & (0,500; 0,500) \end{bmatrix};$$

$$\mathbf{B}_{II}^2 = \begin{bmatrix} (0,500; 0,500) & (0,800; 0,200) & (0,333; 0,667) \\ (0,200; 0,800) & (0,500; 0,500) & (0,111; 0,889) \\ (0,667; 0,333) & (0,889; 0,111) & (0,500; 0,500) \end{bmatrix}.$$

Этап 3. Формируем обобщенную матрицу $\mathbf{W} = \|\mathbf{w}_{ij}\| = \|(w_{ij}(i), w_{ij}(j))\|$, где векторы $(w_{ij}(i), w_{ij}(j))$ — локальные ВКА z_i, z_j уровня 2 относительно *всей совокупности критериев* сравнения, компоненты которых находим по формулам строки IV таблицы, например,

$$(u_{12}^1; u_{12}^2) = \left(\frac{1}{2}(0,333 + 0,800); \frac{1}{2}(0,667 + 0,200) \right) = (0,567; 0,433).$$

$$\mathbf{W} = \begin{bmatrix} (0,500; 0,500) & (0,567; 0,433) & (0,542; 0,458) \\ (0,433; 0,567) & (0,500; 0,500) & (0,484; 0,516) \\ (0,458; 0,542) & (0,516; 0,484) & (0,500; 0,500) \end{bmatrix}.$$

Этап 4. Глобальные значения ВКА считаем по формуле $V_i^{(1)} = \sum_{j=1}^g w_{ij}$, и затем нормализуем: $V = (0,357; 0,315; 0,328)$.

Таким образом, согласно модифицированной процедуре ВКА первых двух альтернатив будут равны 0,357; 0,315, что согласуется с оценками, полученными по двум критериям в первоначальном наборе альтернатив.

Продолжая рассмотрение примера 1, получим следующие значения относительных весовых коэффициентов:

$$\mathbf{V}'_1 = \begin{bmatrix} (7;7) & (7;10) & (7;5) \\ (10;7) & (10;10) & (10;5) \\ (5;7) & (5;10) & (5;5) \end{bmatrix};$$

$$\mathbf{V}'_2 = \begin{bmatrix} (15;15) & (15;10) & (15;20) \\ (10;15) & (10;10) & (10;20) \\ (20;15) & (20;10) & (20;20) \end{bmatrix};$$

$$\mathbf{V}_1 = \begin{bmatrix} (0,500; 0,500) & (0,412; 0,588) & (0,583; 0,417) \\ (0,588; 0,412) & (0,500; 0,500) & (0,667; 0,333) \\ (0,417; 0,583) & (0,333; 0,667) & (0,500; 0,500) \end{bmatrix};$$

$$\mathbf{V}_2 = \begin{bmatrix} (0,500; 0,500) & (0,600; 0,400) & (0,429; 0,571) \\ (0,400; 0,600) & (0,500; 0,500) & (0,333; 0,667) \\ (0,571; 0,429) & (0,667; 0,333) & (0,500; 0,500) \end{bmatrix}.$$

Ненормализованные оценки собственных векторов, согласно ММАИ, будут равны $\mathbf{w}^1 = (0,493, 0,580, 0,411)$, $\mathbf{w}^2 = (0,505, 0,405, 0,575)$, соответственно. Итоговый весовой вектор по равновесным критериям равен $\mathbf{w} = (0,336, 0,332, 0,332)$, что отвечает верному выбору.

Приведенные примеры свидетельствуют о том, что, по-видимому, не существует некоей универсальной функции (полезности) скаляризации критериев в целях выбора в качестве наилучшего произвольного элемента из оптимального множества Парето. Обсуждаемая модификация известного метода позволяет надеяться, что на основе определенной выше функции относительного сходства возможно непротиворечивое упорядочение анализируемых элементов (альтернатив, признаков). На основе данной модификации метода парных сравнений, реализованного в новой интеллектуальной системе распознавания состояний динамических систем [10], успешно решен ряд прикладных задач (например, [11]).

Заключение

В статье рассмотрены свойства модификации метода анализа иерархий, расширяющие возможности классического метода и обосновывающие способ применения линейной свертки, некорректное использование которой в традиционном методе приводит к ошибкам и неточности принятия решения в многокритериальных задачах. Приведенные иллюстративные примеры применения модифицированного варианта известного метода свидетельствуют о его непротиворечивости.

Список литературы

1. Журавлев Ю. И., Гуревич И. Б. Распознавание образов и анализ изображений // Искусственный интеллект в 3-х кн. Кн. 2. Модели и методы: Справочник / Под ред. Д.А.Поспелова. М: Радио и связь. 1990. С. 149—190.
2. Саати Т. Л. Принятие решений. Метод анализа иерархий. М.: Радио и связь. 1989. 311 с.
3. Подиновский В. В., Ногин В. Д. Парето-оптимальные решения многокритериальных задач. Н.: Наука. 2007.
4. Ногин В. Д. Упрощенный вариант метода анализа иерархий на основе нелинейной свертки критериев // Журнал вычислительной математики и математической физики. 2004. Т. 44, № 7. С. 1259—1268.
5. Самохвалов Ю. Я. Особенности применения метода анализа иерархий при оценке проблем по метрическим критериям // Кибернетика и системный анализ. 2004. № 5. С. 15—19.
6. Самохвалов Ю. Я. Групповой учет относительного превосходства альтернатив в задачах принятия решений // Кибернетика и системный анализ. 2003. № 6. С. 141—145.
7. Колесникова С. И. Системный подход к оцениванию взаимного влияния признаков в тестовом распознавании // Кибернетика и системный анализ. 2009. № 3. С. 127—135.
8. Дюкова Е. В., Песков Н. В. Построение распознающих процедур на базе элементарных классификаторов. URL: www.ccas.ru/frc/papers/djukova05constmction.pdf.
9. Колесникова С. И., Янковская А. Е. Оценка значимости признаков для тестов в интеллектуальных системах // Известия РАН. Теория и системы управления, 2008. № 6. С. 135—148.
10. Колесникова С. И., Букреев В. Г., Мертвецов А. Н., Цой Ю. Р., Лаходынов В. С. Интеллектуальная модель распознавания состояний динамических систем "IReDDS": Свидетельство об официальной регистрации программы ЭВМ № 2010610441, 11.01.2010.
11. Колесникова С. И., Янковская А. Е. Выбор оптимального подмножества тестов с применением редукции многокритериального выбора и метода анализа иерархий // Третья Международная конференция "Системный анализ и информационные технологии" САИТ—2009 (14—18 сентября 2009 г., Звенигород, Россия): Труды конференции. М., 2009. С. 278—284.

Е. Г. Комаров, канд. техн. наук, доц., зав. каф.,
Московский государственный университет леса
e-mail: komarov@mgul.ac.ru

Построение модели эталонного объекта и рейтинговых оценок на его основе

Разработана модель эталонного объекта в виде совокупности нечетких чисел и лингвистических значений шкал, используемых для оценивания характеристик объектов. Построена модель рейтингового оценивания на основе эталонного объекта, которая дополняет информацию, полученную на основе традиционных рейтинговых оценок.

Ключевые слова: неопределенность, эталонный объект, рейтинговые оценки

Введение

При сравнительном анализе объектов с нечисловыми (качественными) характеристиками существенную сложность вызывает отсутствие эталонного объекта, в связи с чем проводятся трудоемкие сравнительные процедуры объектов друг с другом [1–3]. Если объектов много, то трудоемкость этих процедур многократно возрастает. Результатами оценивания качественных характеристик, как правило, являются слова естественного (профессионального) языка или наборы таких слов, например, сравнительно небольшая вероятность банкротства, удовлетворительные знания, высокая технологичность и т. д. Использование слов естественного языка при оценивании качественных характеристик объектов является причиной возникновения неопределенности нечеткого характера и соответственно сложности количественного оценивания процессов функционирования и управления. Определяя рейтинговые оценки объектов с качественными характеристиками и принимая на их основе управленческие решения, как правило, поступают по принципу — чем больше (или соответственно меньше) значения характеристик, тем больше (меньше) рейтинговая оценка объекта и, соответственно, его рейтинг. Однако подобный подход не позволяет использовать информацию мониторинга объектов с похожими характеристиками в прошлом для повышения успешности функционирования оцениваемых объектов в будущем. Поэтому целью данной статьи является

разработка модели эталонного объекта и рейтингового оценивания на его основе. Рейтинговые оценки, рассчитанные на основе эталонного объекта, являются количественными показателями близости характеристик оцениваемых объектов и эталонного объекта, построенного на основе реальных апостериорных данных. Исходя из этого можно считать, что рейтинговые оценки дополняют и уточняют информацию, полученную на основе традиционных рейтинговых оценок, и, как следствие, повышают достоверность прогноза состояний объектов и управляющих рекомендаций по их улучшению.

Построение модели эталонного объекта

Рассмотрим N объектов, у которых оцениваются характеристики $X_j, j = \overline{1, m}$, оказывающие существенное влияние на успешность их функционирования в будущем — Y .

Пусть $X_{lj}, l = \overline{1, m}$, — уровни вербальных шкал, применяемых для оценивания соответственно характеристик $X_j, j = \overline{1, m}$, а $Y_l, l = \overline{1, k}$, — уровни вербальной шкалы, используемой для оценивания характеристики Y . Уровни расположены в порядке возрастания интенсивности проявления этих характеристик.

Обозначим $a_l^j, l = \overline{1, m_j}, j = \overline{1, m}$, относительное число объектов рассматриваемой совокупности, отнесенных при оценивании характеристики $X_j, j = \overline{1, m}$, к уровню $X_{lj}, l = \overline{1, m_j}, j = \overline{1, m}$:

$$\sum_{l=1}^{m_j} a_l^j = 1, j = \overline{1, m}.$$

Опираясь на эти данные и метод, описанный в работе [4], построим m лингвистических переменных с названиями $X_j, j = \overline{1, m}$ и терм-множествами $X_{lj}, l = \overline{1, m_j}, j = \overline{1, m}$. Обозначим $\mu_{lj}(x)$ функцию принадлежности нечеткого числа \tilde{X}_{lj} , соответствующего l -му терм-множеству, j -й лингвистической переменной, $l = \overline{1, m_j}, j = \overline{1, m}$. Будем называть оценками объектов нечеткие числа $\tilde{X}_{lj}, l = \overline{1, m_j}, j = \overline{1, m}$, или их функции принадлежности $\mu_{lj}(x), l = \overline{1, m_j}, j = \overline{1, m}$. Обозначим оценку n -го объекта в рамках характеристика X_j с функцией принадлеж-

ности \tilde{X}_j^n как $\mu_j^n(x) \equiv (a_{j1}^n, a_{j2}^n, a_{jL}^n, a_{jR}^n)$, $n = \overline{1, N}$, $j = \overline{1, m}$. Нечеткое число \tilde{X}_j^n с функцией принадлежности $\mu_j^n(x)$ равно одному из нечетких чисел \tilde{X}_{lj} , $l = \overline{1, m_j}$, $j = \overline{1, m}$.

Обозначим a_l , $l = \overline{1, k}$ относительное число объектов, отнесенных при оценивании успешности их функционирования к уровню Y_l , $l = \overline{1, k}$.

Опираясь на эти данные и метод, описанный в работе [4], построим лингвистическую переменную с названием Y (успешность функционирования) и терм-множеством Y_l , $l = \overline{1, k}$. Обозначим $\mu_l(x)$ функцию принадлежности нечеткого числа \tilde{Y}_l , соответствующему терму Y_l , $l = \overline{1, k}$. Будем называть оценками объектов нечеткие числа \tilde{Y}_l , $l = \overline{1, k}$, или их функции принадлежности $\mu_l(x)$, $l = \overline{1, k}$.

Среди N объектов выделяем те, которые получили от экспертов высшие оценки успешности их функционирования или оценки $\mu_k(x) \equiv (y_{k1}, y_{k2}, y_{kL}, y_{kR})$. Не ограничивая общности, будем считать, что это объекты с номерами $i = \overline{1, M}$ и функциями принадлежности $\{\mu_j^i(x) \equiv (a_{j1}^i, a_{j2}^i, a_{jL}^i, a_{jR}^i)\}$, $i = \overline{1, M}$, $j = \overline{1, m}$, значений характеристик X_j , $j = \overline{1, k}$.

Модель эталонного объекта будем строить в виде совокупности нечетких чисел (или их функций принадлежности), соответствующих проявлению характеристик X_j , $j = \overline{1, m}$, т. е.

$$\{\mu_j(x) \equiv (x_{j1}, x_{j2}, x_{jL}, x_{jR})\}, j = \overline{1, m}.$$

После построения эти нечеткие числа предлагается распознать в рамках уровней вербальных шкал X_j , $l = \overline{1, m_j}$, применяемых для оценивания соответственно характеристик X_j , $j = \overline{1, m}$. Исходя из этого эталонный объект может быть представлен в двух видах — формализованном, т. е. в виде совокупности нечетких чисел, и в виде совокупности значений характеристик, описанных в рамках профессионального языка специалистов исследуемой области. Первое представление эталонного объекта предлагается использовать специалистам в области обработки информации, в том числе и нечеткой. Второе представление предлагается использовать лицам, принимающим решения, но при этом не являющимся специалистами в области обработки информации.

Для построения модели эталонного объекта в статье используется регрессионная модель, описанная в работе [5]:

$$\tilde{Y} = \tilde{a}_0 + \tilde{a}_1 \tilde{X}_1 + \dots + \tilde{a}_m \tilde{X}_m;$$

$$\tilde{Y} = \begin{pmatrix} \tilde{Y}_1 \\ \dots \\ \tilde{Y}_n \end{pmatrix}, \tilde{Y}_i \equiv (y_1^i, y_2^i, y_L^i, y_R^i), y_1^i - y_L^i \geq 0,$$

$i = \overline{1, n}$ — выходные T -числа;

$$\tilde{X}_j = \begin{pmatrix} \tilde{X}_j^i \\ \dots \\ \tilde{X}_j^n \end{pmatrix}, \tilde{X}_j^i \equiv (x_1^{ji}, x_2^{ji}, x_L^{ji}, x_R^{ji}), x_1^{ji} - x_L^{ji} \geq 0,$$

$j = \overline{1, m}$, $i = \overline{1, n}$ — входные T -числа; $a_j \equiv (b^j, b_L^j, b_R^j)$,

$j = \overline{0, m}$, — неизвестные коэффициенты регрессионной модели — треугольные (нормальные) числа.

Как известно [6], T -числом (нечетким числом T -типа) называется нечеткое число \tilde{A} с функцией принадлежности

$$\mu_{\tilde{A}}(x) = \begin{cases} 1 + \frac{x - a_1}{a_L}, & a_1 - a_L \leq x < a_1, a_L > 0; \\ 1 - \frac{x - a_2}{a_R}, & a_2 < x \leq a_2 + a_R, a_R > 0; \\ 1, & a_1 \leq x \leq a_2; \\ 0, & x < a_1 - a_L \text{ или } x > a_2 + a_R. \end{cases}$$

Символически T -число записывается в виде $\tilde{A} \equiv (a_1, a_2, a_L, a_R)$. Нормальное треугольное число является частным случаем T -числа при $a_1 = a_2$ и символически записывается в виде $\tilde{A} \equiv (a_1, a_L, a_R)$. При $a_L = a_R$ треугольное число называется симметричным, а при $a_L \neq a_R$ — несимметричным.

В работе [5] дано определение взвешенного отрезка $[A_1, A_2]$ для T -числа $\tilde{A} \equiv (a_1, a_2, a_L, a_R)$:

$$A_1 = a_1 - \frac{1}{6} a_L, A_2 = a_2 + \frac{1}{6} a_R.$$

Обозначим $[A_{j1}^i, A_{j2}^i]$, $i = \overline{1, M}$, $j = \overline{1, m}$, взвешенные отрезки нечетких чисел с функциями принадлежности $\{\mu_j^i(x) \equiv (a_{j1}^i, a_{j2}^i, a_{jL}^i, a_{jR}^i)\}$, $i = \overline{1, M}$, $j = \overline{1, m}$. Тогда получим

$$A_{j1}^i = a_{j1}^i - \frac{1}{6} a_{jL}^i, A_{j2}^i = a_{j2}^i + \frac{1}{6} a_{jR}^i,$$

$$i = \overline{1, M}, j = \overline{1, m}.$$

Обозначим $[B_{j1}, B_{j2}], j = \overline{1, m}$, взвешенные отрезки нечетких чисел с функциями принадлежности $\{\mu_j(x) \equiv (x_{j1}, x_{j2}, x_{jL}, x_{jR})\}, j = \overline{1, m}$.

Тогда получим

$$B_{j1} = x_{j1} - \frac{1}{6} x_{jL}, B_{j2} = x_{j2} + \frac{1}{6} x_{jR}, j = \overline{1, m}.$$

Обозначим $[C_1, C_2]$ взвешенный отрезок нечеткого числа с функцией принадлежности $\mu_k(x) \equiv (y_{k1}, y_{k2}, y_{kL}, y_{kR})$, а $[D_1, D_2]$ — взвешенный отрезок нечеткого числа, которое получается подстановкой нечетких чисел с функциями принадлежности $\{\mu_j(x) \equiv (x_{j1}, x_{j2}, x_{jL}, x_{jR})\}, j = \overline{1, m}$, в регрессионную модель, описанную в работе [5]:

$$D_1 = b^0 - \frac{1}{6} b_L^0 + \sum_{j=1}^m \theta_{\tilde{a}_j \tilde{X}_j}^1(b^j, b_L^j, b_R^j);$$

$$D_2 = b^0 + \frac{1}{6} b_R^0 + \sum_{j=1}^m \theta_{\tilde{a}_j \tilde{X}_j}^2(b^j, b_L^j, b_R^j).$$

Если $\tilde{a}_j \equiv (b^j, b_L^j, b_R^j), j = \overline{1, m}$, — неотрицательное нечеткое число ($b^j + b_R^j \geq 0$), то

$$\begin{aligned} & \theta_{\tilde{a}_j \tilde{X}_j}^1(b^j, b_L^j, b_R^j) = \\ & = b^j \left(x_1^{ji} - \frac{1}{6} x_L^{ji} \right) - b_L^j \left(\frac{1}{6} x_1^{ji} - \frac{5}{12} x_L^{ji} \right); \\ & \theta_{\tilde{a}_j \tilde{X}_j}^2(b^j, b_L^j, b_R^j) = \\ & = b^j \left(x_2^{ji} + \frac{1}{6} x_R^{ji} \right) + b_R^j \left(\frac{1}{6} x_2^{ji} + \frac{5}{12} x_R^{ji} \right). \end{aligned}$$

Если $\tilde{a}_j \equiv (b^j, b_L^j, b_R^j), j = \overline{1, m}$, — отрицательное нечеткое число ($b^j + b_R^j < 0$), то

$$\begin{aligned} & \theta_{\tilde{a}_j \tilde{X}_j}^1(b^j, b_L^j, b_R^j) = \\ & = b^j \left(x_2^{ji} + \frac{1}{6} x_R^{ji} \right) - b_L^j \left(\frac{1}{6} x_2^{ji} + \frac{5}{12} x_R^{ji} \right); \\ & \theta_{\tilde{a}_j \tilde{X}_j}^2(b^j, b_L^j, b_R^j) = \\ & = b^j \left(x_1^{ji} - \frac{1}{6} x_L^{ji} \right) + b_R^j \left(\frac{1}{6} x_1^{ji} - \frac{5}{12} x_L^{ji} \right). \end{aligned}$$

Обозначим $(C_1 - D_1)^2 + (C_2 - D_2)^2$ через ρ_1^2 , а $\sum_{i=1}^M \sum_{j=1}^m [(A_{j1}^i - B_{j1})^2 + (A_{j2}^i - B_{j2})^2]$ — через ρ_2^2 .

Неизвестные параметры функций принадлежности $\{\mu_j(x) \equiv (x_{j1}, x_{j2}, x_{jL}, x_{jR})\}, j = \overline{1, m}$, эталонного образа находятся из решения оптимизационной задачи

$$\rho_1^2 + \rho_2^2 \rightarrow \min$$

при условиях:

$$x_{j1} - x_{jL} \geq 0; x_{j2} + x_{jR} \leq 1; x_{jL} \geq 0; x_{jR} \geq 0, j = \overline{1, m}.$$

Задача решается известными методами [7–8].

Построенный эталонный образ объекта $\{\mu_j(x) \equiv (x_{j1}, x_{j2}, x_{jL}, x_{jR})\}, j = \overline{1, m}$, является совокупностью формализованных представлений характеристик, т. е. совокупностью нечетких чисел, и подобное его представление хорошо понятно специалистам области обработки информации (в том числе и нечеткой), однако хотелось бы, чтобы этот образ был понятен также лицам, которые принимают решения, но не занимаются обработкой информации. Для этой цели мы идентифицируем полученные нечеткие числа с функциями принадлежности $\{\mu_j(x) \equiv (x_{j1}, x_{j2}, x_{jL}, x_{jR})\}, j = \overline{1, m}$, с соответствующими нечеткими числами \tilde{X}_{lj} с функциями принадлежности $\mu_{lj}(x), l = \overline{1, m_j}, j = \overline{1, m}$. Эти числа являются формализациями лингвистических значений шкал, которые используются для оценивания характеристик $X_j, j = \overline{1, m}$, объектов. После идентификации эталонный объект будет представлен в виде совокупности значений характеристик, описанных на профессиональном языке специалистов рассматриваемой предметной области. Подобное представление значительно облегчает процесс принятия решений.

Взвешенные отрезки нечетких чисел с функциями принадлежности $\{\mu_j(x) \equiv (x_{j1}, x_{j2}, x_{jL}, x_{jR})\}, j = \overline{1, m}$, были обозначены $[B_{j1}, B_{j2}], j = \overline{1, m}$. Обозначим $[Q_{j1}^l, Q_{j2}^l], l = \overline{1, m_j}, j = \overline{1, m}$, взвешенные отрезки нечетных чисел \tilde{X}_{lj} с функциями принадлежности $\mu_{lj}(x), l = \overline{1, m_j}, j = \overline{1, m}$.

Пусть

$$\begin{aligned} f^2(\mu_j(x), \mu_{lj}(x)) &= (Q_{j1}^l - B_{j1})^2 + (Q_{j2}^l - B_{j2})^2, \\ & l = \overline{1, m_j}, j = \overline{1, m}. \end{aligned}$$

Нечеткое число с функцией принадлежности $\mu_j(x) \equiv (x_{j1}, x_{j2}, x_{jL}, x_{jR})$ идентифицируется с лингвистическим значением X_{sj} характеристики X_j , если

$$f^2(\mu_j(x), \mu_{sj}(x)) = \min_l f^2(\mu_j(x), \mu_{lj}(x)), l = \overline{1, m_j}.$$

Эталонный образ, определенный в виде совокупности нечетких чисел $\{\mu_j(x) \equiv (x_{j1}, x_{j2}, x_{jL}, x_{jR})\}$, $j = \overline{1, m}$, или лингвистических значений характеристик, предлагается использовать для сравнительного анализа с реальными объектами, определения рейтинговых оценок и выработки управляющих воздействий, направленных на успешность функционирования объектов в будущем.

Модель рейтингового оценивания на основе эталонного объекта

Определим рейтинговые оценки объектов в рамках нескольких качественных характеристик на основе эталонного объекта, построенного в предыдущем разделе.

В этом разделе через $[B_{j1}, B_{j2}]$, $j = \overline{1, m}$, были обозначены взвешенные отрезки нечетких чисел с функциями принадлежности $\{\mu_j(x) \equiv (x_{j1}, x_{j2}, x_{jL}, x_{jR})\}$, $j = \overline{1, m}$ (т. е. нечетких чисел, которые образуют эталонный образ), где

$$B_{j1} = x_{j1} - \frac{1}{6}x_{jL}; B_{j2} = x_{j2} + \frac{1}{6}x_{jR}; j = \overline{1, m}.$$

Обозначим $[A_{j1}^n, A_{j2}^n]$, $n = \overline{1, N}$, $j = \overline{1, m}$, взвешенные отрезки оценок n -го объекта или нечетких чисел \tilde{X}_j^n , $n = \overline{1, N}$, с функциями принадлежности $\{\mu_j^n(x) \equiv (a_{j1}^n, a_{j2}^n, a_{jL}^n, a_{jR}^n)\}$, $n = \overline{1, N}$, $j = \overline{1, m}$. Тогда получим

$$A_{j1}^n = a_{j1}^n - \frac{1}{6}a_{jL}^n; A_{j2}^n = a_{j2}^n + \frac{1}{6}a_{jR}^n; n = \overline{1, N}, j = \overline{1, m}.$$

Рейтинговую оценку n -го объекта, $n = \overline{1, N}$, определим следующим образом:

$$r_n = 1 - \sum_{j=1}^m \omega_j \sqrt{\frac{1}{2}(B_{j1} - A_{j1}^n)^2 + \frac{1}{2}(B_{j2} - A_{j2}^n)^2},$$

где ω_j , $j = \overline{1, m}$, — весовые коэффициенты характеристик X_j , $j = \overline{1, m}$.

Разработана модель эталонного объекта при мониторинге функционирования объектов с качественными характеристиками. Эталонный объект, определенный в виде совокупности нечетких чисел или лингвистических значений характеристик, предлагается использовать для сравнительного анализа объектов, определения рейтинговых оценок и выработки управляющих воздействий, направленных на успешность функционирования объектов в будущем.

Построена модель многокритериального оценивания на основе эталонного объекта, позволяющая для объектов с качественными характеристиками находить числовые рейтинговые оценки. Рейтинговые оценки, построенные на основе эталонного объекта, являются количественными показателями близости характеристик оцениваемых объектов и характеристик эталонного объекта, построенного на основе реальных апостериорных данных.

Исходя из этого совокупный анализ данных мониторинга функционирования объектов на основе традиционных рейтинговых оценок и рейтинговых оценок на основе эталонного объекта повышают достоверность прогноза состояний объектов и управляющих рекомендаций по их улучшению.

Список литературы

1. **Пфанцгаль И.** Теория измерений: Пер. с англ. М.: Мир, 1976. 166 с.
2. **Смоляк С. А., Тигаренко Б. П.** Устойчивые методы оценивания. Статистическая обработка неоднородных совокупностей. М.: Статистика, 1980. 208 с.
3. **Джини К.** Средние величины. М.: Статистика, 1970. 556 с.
4. **Домрачев В. Г., Комаров Е. Г., Полещук О. М.** Мониторинг функционирования объектов на основе нечеткого описания их состояний // Информационные технологии. 2007. № 11. С. 46—52.
5. **Domrachev V. G., Poleshuk O. M.** A regression model for fuzzy initial data // Automation and Remote Control. 2003. V. 64, № 11. P. 1715—1724.
6. **Аверкин А. Н., Батыршин И. З., Блишун А. Ф., Силов В. Б., Тарасов В. Б.** Нечеткие множества в моделях управления и искусственного интеллекта. М.: Наука. Гл. ред. физ.-мат. лит., 1986. 312 с.
7. **Coleman T. F., Li Y.** A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables // SIAM J. Optim. 1996, V. 6. № 4. P. 1040—1058.
8. **Комаров Е. Г.** Обработка экспертных оценок качества образовательных информационных ресурсов на основе лингвистических переменных // Обозрение прикладной и промышленной математики. 2008. Т. 16. Вып. 1. С. 150—151.

УДК 519.6

О. В. Мандрикова^{1, 2},

д-р техн. наук, вед. науч. сотр.,

И. С. Соловьев, аспирант²,

¹ Институт космических исследований
и распространения радиоволн ДВО РАН,

² Камчатский государственный

технологический университет

e-mail: oksanam1@mail.kamchatka.ru,

e-mail: kamigsol@yandex.ru

Вейвлет-технология анализа вариаций магнитного поля Земли

Для изучения особенностей геомагнитных сигналов предложена вейвлет-технология их обработки и анализа, позволяющая в автоматическом режиме выделить возмущения в вариациях магнитного поля Земли, определить их интенсивность, сформировать спокойную суточную вариацию и вычислить индекс геомагнитной активности K .

Ключевые слова: вейвлет-преобразование, магнитное поле Земли, индекс геомагнитной активности K

Введение

Изучение характера вариаций магнитного поля Земли служит основой оценки свойств и состояния электромагнитных полей в околоземном про-

странстве. Вследствие воздействия солнечного ветра на магнитосферу Земли изменяются параметры магнитного и электрического полей, интенсивности токов и потоков высыпавшихся частиц, возбуждаются разного рода волны [1]. Все эти величины используются для оценки силы магнитной бури, характеризующей воздействие Солнца на околоземное пространство.

Временной ход компоненты вектора напряженности геомагнитного поля будем называть магнитным сигналом. В моменты магнитной бури в сигналах наблюдаются резкие выбросы и пульсации разных частотных спектров и периодов. На рис. 1 в качестве примера показаны спокойная и возмущенная составляющие магнитного поля (горизонтальные компоненты вектора напряженности геомагнитного поля — H -компоненты). Из-за сильной изменчивости, нерегулярности и сложной внутренней структуры сигналов задача их обработки и анализа таких сигналов в автоматическом режиме является весьма сложной. Существующие методы анализа вариаций магнитного поля основаны на операциях сглаживания, что дает большое усреднение, маскирует динамику возмущений и влечет существенную потерю важной информации [2]. Эти методы [1, 3] позволяют выделить характерные периоды, но не дают информации о локальных изменениях, протекающих в физическом процессе, и их масштабных характеристиках.

Современные методы цифровой обработки сигналов позволяют справиться с данной проблемой, разработанные на их основе адаптивные алгоритмы дают возможность на основе небольшого числа параметров получить аппроксимации требуемой точности [4, 5], идентифицировать локальные структуры сигнала и подавить шум. Полученные на основе этих методов представления сигналов имеют многокомпонентную структуру, включающую трендовые составляющие и локальные разномасштабные компоненты. Трендовые составляющие характеризуют регулярные изменения процес-

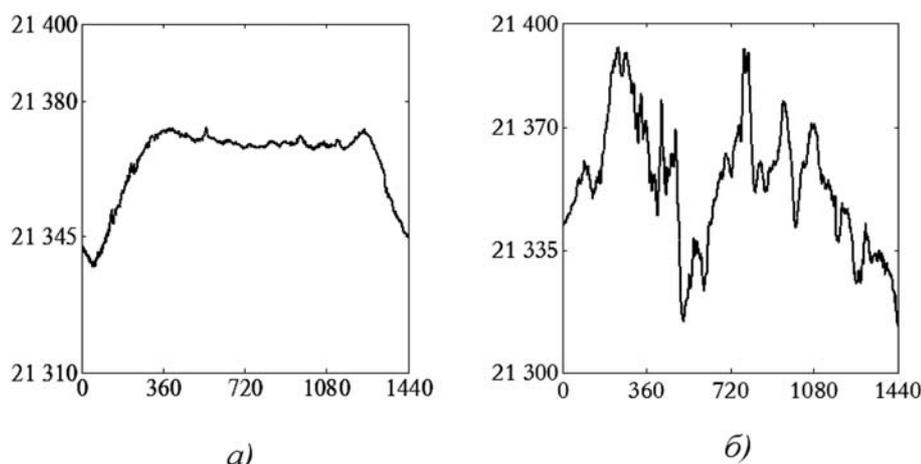


Рис. 1. Регистрируемые данные магнитного поля Земли:

a — спокойная суточная вариация; b — возмущенная суточная вариация. На оси ox отмечены отсчеты регистрируемых сигналов, на оси oy — значения H -компоненты магнитного поля Земли (в нТл)

са, другие носят локальный характер и в случае магнитных данных несут информацию о возмущенности магнитного поля. Инструментом, лежащим в основе построения таких схем, является вейвлет-преобразование [4, 5]. С использованием этих методов в данной статье осуществляется декомпозиция магнитных сигналов на составляющие и выполняется их детальный анализ. Это позволяет в автоматическом режиме выделить геомагнитные возмущения и реализовать процедуру определения спокойной суточной вариации, что, в свою очередь, обеспечивает возможность вычисления индекса геомагнитной активности K (K -индекса). K -индекс характеризует геомагнитные возмущения в месте регистрации и используется в широком круге научных задач физики атмосферы, ионосферы, распространения радиоволн.

Применение вейвлет-технологии

Выделение компонент магнитного сигнала и идентификация геомагнитных возмущений

Если h, q — пара сопряженных зеркальных фильтров, определяемых формулами

$$\Psi_{j-1}^0(t) = \sum_n h_n \Psi_j(2^j t - n) \text{ и } \Psi_{j-1}^1(t) = \sum_n q_n \Psi_j(2^j t - n),$$

где Ψ — вейвлет; n — отсчеты сигнала; j — масштабный уровень; t — время, то семейство $\{\Psi_{j-1}^0(2^{j-1}t - n), \Psi_{j-1}^1(2^{j-1}t - n)\}_{n \in \mathbb{Z}}$, \mathbb{Z} — множество целых чисел, является ортонормированным базисом пространства W_j , где $W_j := \text{clos}_{L^2(\mathbb{R})}(\Psi_j(2^j t - n); n \in \mathbb{Z})$, $L^2(\mathbb{R})$ — пространство Лебега.

Таким образом, сопряженные зеркальные фильтры преобразуют ортогональный базис $\{\Psi_j(2^j t - n)\}_{n \in \mathbb{Z}}$ в два ортогональных семейства $\{\Psi_{j-1}^0(2^{j-1}t - n)\}_{n \in \mathbb{Z}}$ и $\{\Psi_{j-1}^1(2^{j-1}t - n)\}_{n \in \mathbb{Z}}$. Пространства W_{j-1}^0 и W_{j-1}^1 , порожденные каждым из этих семейств, ортогональны и

$$W_{j-1}^0 \oplus W_{j-1}^1 = W_j.$$

Функция f при этом представляется в виде суммы компонент:

$$f_0(t) = f_{-1}(t) + d_{-1}^1,$$

где $f_{-1} \in W_{-1}^0$, $d_{-1}^1 \in W_{-1}^1$, $f_{-1}(t) = \sum_n c_n^{-1} \Psi_{-1}^0(2^{-1}t - n)$,

коэффициенты разложения $c_n^{-1} = \langle f_0, \Psi_{-1}^0(2^{-1}t - n) \rangle$,

$d_{-1}^1(t) = \sum_n s_n^{-1} \Psi_{-1}^1(2^{-1}t - n)$, коэффициенты разложения $s_n^{-1} = \langle f_0, \Psi_{-1}^1(2^{-1}t - n) \rangle$.

Фильтр h пропускает низкие частоты, тем самым осуществляет сглаживание сигнала и выделяет *аппроксимирующую компоненту сигнала*. Фильтр q пропускает высокие частоты, тем самым выделяет *высокочастотную компоненту сигнала*. Эту компоненту называют также *детализирующей компонентой сигнала*. Рекурсивное расщепление пространств на основе этой процедуры в вейвлет-теории называют деревом пространств вейвлет-пакетов:

$$W_j = \bigoplus_{i=-1}^I W_j^p,$$

где I — число пространств разложения; p — порядковый номер пространства. Схема представления сигнала на основе его разложения в вейвлет-пакеты показана на рис. 2. Компоненты f_j в этом разложении являются аппроксимирующими, они характеризуют регулярные изменения процесса, компоненты d_j^k являются детализирующими и содержат информацию о локальных особенностях сигнала.

Локальные особенности магнитного сигнала несут информацию о содержащихся в нем флуктуациях, вызванных геомагнитными возмущениями. Они могут быть идентифицированы на основе превышения абсолютных значений коэффициентов $|s_n^j|$ детализирующих компонент d_j^k некоторого порогового значения T_j [4, 5]. Геомагнитные возмущения имеют многомасштабную структуру и,

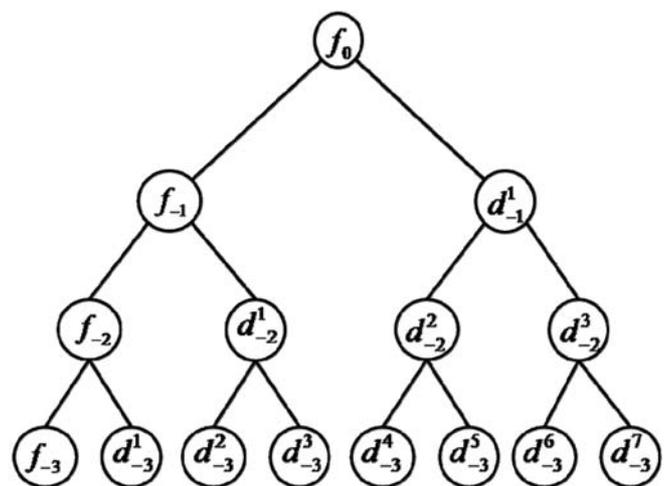


Рис. 2. Схема представления сигнала f_0 в пространстве вейвлет-пакетов

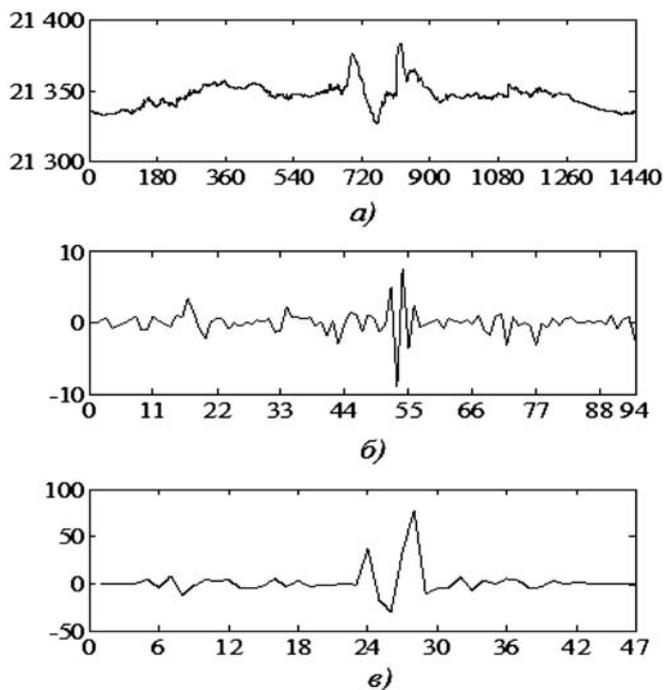


Рис. 3. Разложение магнитного сигнала на вейвлет-компоненты: *a* — суточная вариация: на оси *ox* отмечены отсчеты сигнала; на оси *oy* — значения *H*-компоненты магнитного поля Земли (в нТл); *б* — компонента d_{-4}^1 ; на оси *ox* отмечены отсчеты сигнала; на оси *oy* — значения вейвлет-коэффициентов s_n^{-4} ; *в* — компонента d_{-5}^1 ; на оси *ox* отмечены отсчеты сигнала; на оси *oy* — значения вейвлет-коэффициентов s_n^{-4} .

следовательно, могут проявлять себя на различных масштабных уровнях j (рис. 3).

Построим процедуру их выделения на основе проверки следующих условий:

1) если $|s_n^j| > T_j$, где T_j — пороговое значение, соответствующее уровню j , то в сигнале в момент времени $t = n$ имеем возмущение, соответствующее масштабу j ;

2) если $\sum_{j \in I_M} |s_n^j| > T_{\text{сумм}}$, где $T_{\text{сумм}}$ — наперед заданное пороговое значение, определяющее наличие многомасштабной флуктуации, то в сигнале имеем многомасштабное геомагнитное возмущение, проявляющее себя на масштабах $j \in I_M$ (I_M — множество индексов).

Каждая детализирующая компонента d_j^k (см. рис. 2) определяет свой частотно-временной прямоугольник в частотно-временной плоскости [4]. Таким образом, разномасштабные составляющие исходного сигнала f_0 рассортировываются и хранятся в различных пространствах W_{j_i} . Иден-

тификацию компонент магнитного сигнала d_j^k , которые могут содержать геомагнитные возмущения, и пороговых значений T_j и $T_{\text{сумм}}$, определяющих наличие возмущений, выполним на основе следующего, алгоритма.

1. Построим полное дерево разложения:

$$W_j^0 : W_j^0 = \oplus_{j=-1}^I W_j^P.$$

2. На основе обработки *спокойных* дней определим пороговые значения T_j и $T_{\text{сумм}}$, определяющие наличие в сигнале возмущений:

$$T_j = \max_k \max_n |s_n^j|,$$

k — номер анализируемого спокойного дня,

$$T_{\text{сумм}} = \sum_j T_j$$

3. На основе обработки *возмущенных* дней идентифицируем детализирующие компоненты сигнала d_j^k , которые могут содержать геомагнитные

возмущения $d_j^k, j \in I_M$. Будем называть такие компоненты информативными.

Для каждой детализирующей компоненты сигнала $d_j^k = \sum_n s_n^j \Psi_j^k(2^j t - n)$ на основе проверки ус-

ловия $\max_n |s_n^j| > T_j$ определим возмущенные компоненты. Из индексов j полученных возмущенных компонент сформируем множество индексов I_M .

Интенсивность $I_j, j \in I_M$ возмущений на масштабе j может быть определена из соотношения:

$$I_j = \max_n |s_n^j|.$$

Процедура идентификации спокойного дня может быть реализована на основе проверки следующего условия для каждого текущего дня: если $\forall j \max_n |s_n^j| > T_j$, то текущий день является спокойным днем, в противном случае — возмущенным днем.

Формирование спокойной суточной вариации и вычисление *K*-индекса

K-индекс был введен Бартельсом в 1938 г. [7]. Он определяет меру магнитной возмущенности и вычисляется из магнитограмм по специальной методике [1], которая предполагает высокий уровень профессиональной подготовки персонала магнитной обсерватории. Основные требования данной методики предъявляются к выделению не возмущенной вариации геомагнитного поля, которая называется *Sq*-кривой. *Sq* (*Solar quiet*)-ва-

риация обусловлена токовой системой, возникающей на освещенной стороне Земли на высоте слоя E ионосферы. Она является средней сглаженной кривой нескольких, наиболее близких к текущей дате, спокойных составляющих напряженности магнитного поля. Далее, на основе разницы между наибольшим и наименьшим отклонениями в течение трехчасового интервала реальной магнитограммы от значения Sq -кривой определяют амплитуду возмущения (в нТл), которая переводится по квазилогарифмической шкале в K -индекс.

Существующие методы автоматизации вычисления K -индекса не удовлетворяют требованиям, поскольку не содержат средства адаптации к изменчивости Sq -кривой с течением времени и не соответствуют методике, предложенной Бартельсом. Используя описанную выше процедуру идентификации спокойного дня, K -индекс можно вычислить в автоматическом режиме по методике, соответствующей методике Бартельса, путем реализации следующего алгоритма.

1. В память загружается магнитный сигнал текущего дня и осуществляется проверка, является ли он спокойным.

2. Если текущий день спокойный, то он замещает первый день в тройке спокойных дней, и с учетом изменений рассчитывается Sq -вариация.

3. Для расчета Sq -вариации используются значения трех спокойных дней, ближайших по дате к текущему дню: их соответствующие значения складываются и делятся на 3.

4. На основе полученной Sq -вариации далее определяем разницу между наибольшим и наименьшим отклонениями в течение трехчасового интервала реальной магнитограммы от значения Sq -кривой и переводим ее по квазилогарифмической шкале в K -индекс.

Результаты экспериментов

В качестве экспериментальных данных использовали вариации магнитного поля Земли, полученные на обсерватории "Паратунка" (с. Паратунка, Камчатская область). При проведении анализа применяли ортонормированные вейвлеты класса Добеши. Разложение осуществляли до пятого уровня полного вейвлет-дерева. На основе описанного выше метода были определены *информативные* компоненты магнитных сигналов, они представлены на рис. 4.

Далее, используя эти компоненты, была выполнена процедура их вейвлет-восстановления. В результате были получены возмущенные состав-

ляющие магнитных сигналов. Также эта операция была выполнена для спокойных дней, и проведено сравнение составляющих спокойных и возмущенных дней (рис. 5). Анализ рис. 5 показывает, что в периоды геомагнитных бурь в сигнале наблюдается существенное увеличение амплитуды флуктуаций.

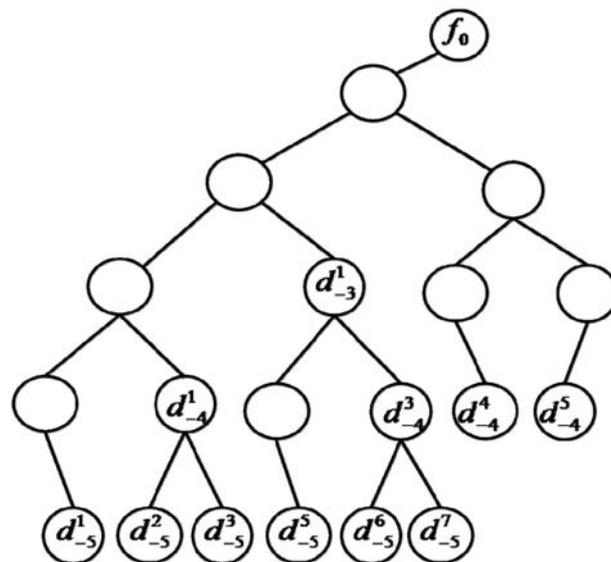


Рис. 4. Компоненты магнитного сигнала, содержащие возмущения

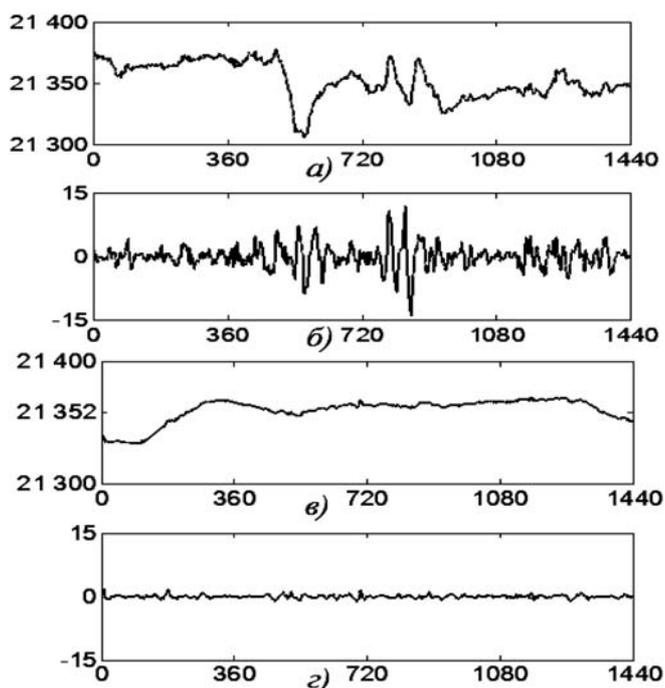


Рис. 5. Магнитные данные за возмущенный день (05.01.2008) (а); магнитные данные за спокойный день (03.01.2008) (б); восстановленные составляющие сигналов за 05.01.2008 и 03.01.2008, характеризующие возмущенность магнитного поля Земли (в, г). На оси ox отмечены отсчеты регистрируемых сигналов (один отсчет равен одной минуте), на оси oy — значения H -компоненты магнитного поля Земли (в нТл)

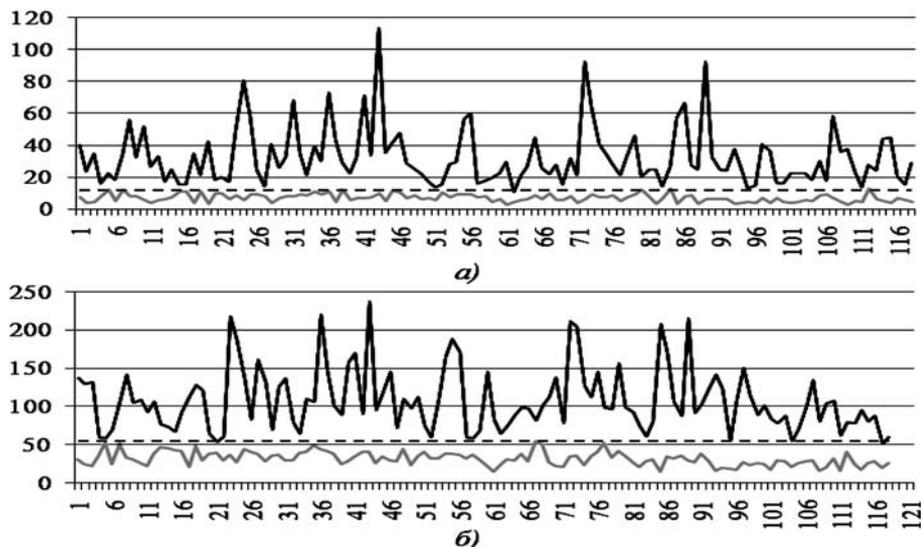


Рис. 6. Результат обработки магнитных данных за 2008 г. Черной линией отмечены результаты обработки возмущенных дней, серой — спокойных дней. На оси oy отмечены:

a — значения функции $\max_n |s_n^j|$; b — значения функции $\sum_j \max_n |s_n^j|$. На оси ox отмечены номера анализируемых дней

Результаты расчета К-индекса методом intermagnet и предложенным методом

Результат на основе intermagnet								Результат, полученный предложенным способом									
2008.11								2008.11									
1.	1	0	1	0	0	0	0	1	1.	0	0	1	0	0	0	1	0
2.	0	0	1	2	1	0	0	1	2.	0	0	1	2	1	1	1	1
3.	0	0	0	0	0	0	0	1	3.	1	0	0	0	0	0	0	1
4.	0	0	0	0	0	0	0	0	4.	0	0	0	0	0	0	0	0
5.	0	0	0	0	1	0	0	0	5.	0	0	0	0	0	0	0	1
6.	1	0	0	0	0	0	1	0	6.	1	0	0	0	0	0	1	1
7.	0	2	2	2	2	1	2	4	7.	1	2	2	3	2	2	2	3
8.	3	2	3	4	3	2	1	1	8.	3	2	3	4	3	2	2	1
9.	3	2	2	3	1	2	1	3	9.	2	3	2	3	1	1	2	2
10.	0	1	1	1	2	2	0	0	10.	0	1	1	0	2	1	0	0
11.	0	0	0	1	0	0	0	0	11.	0	0	0	0	0	0	0	1
12.	0	0	1	1	1	0	0	0	12.	0	0	2	1	1	1	0	1
13.	0	0	0	0	1	0	0	0	13.	1	1	0	0	0	0	0	1
14.	0	0	0	0	0	1	0	0	14.	1	0	1	0	1	2	1	0
15.	0	0	0	0	0	1	1	2	15.	1	0	1	0	1	1	2	2
16.	2	2	2	1	3	0	0	1	16.	2	3	2	1	3	1	0	1
17.	0	1	0	0	0	0	0	0	17.	2	1	0	0	0	0	1	2

Результат расчета значений функций $\max_n |s_n^j|$ и

$\sum_j \max_n |s_n^j|$ для спокойных и возмущенных дней (рис. 6) подтвердил эффективность предложенной в данной статье процедуры идентификации спокойного дня. Анализ результатов вычисления К-индекса показал, что наименьшую погрешность расчетов обеспечивает функция Добеши 3-го порядка. В таблице приведены результаты определения К-индекса предлагаемым методом и методом intermagnet [2] за первые 17 дней ноября 2008 г. Серым цветом отмечены полученные значения К-индекса, которые отличаются от метода intermagnet на значение 1, черным цветом — на значение 2. Полученные результаты подтверждают эффективность предложенной методики.

Список литературы

1. Будько Н., Зайцев А., Карпачев А., Козлов А., Филиппов Б. Космическая среда вокруг нас. Троицк: ТРОВАНТ, 2005. 231 с.
2. Nowozynski K. Calculate geomagnetic activity K indices using the Adaptative Smoothing method. URL: http://www.intermagnet.org/Software_e.html.2007.
3. Кей С. М., Марпл С. Л. Современные методы спектрального анализа: Обзор // ТИИЭР. 1981. Т. 69. № 11. С. 3—5.
4. Малла С. Вейвлеты в обработке сигналов / Пер. с английского, М.: Мир, 2005. 672 с.
5. Мандрикова О. В. Моделирование геохимических сигналов на основе вейвлет-преобразования. Владивосток: Дальнаука, 2007. 123 с.

Ю. Ю. Герасимов, д-р техн. наук, ст. науч. сотр. НИИ леса Финляндии,
А. П. Соколов, канд. техн. наук, доц.,
В. К. Катаров, канд. техн. наук, преподаватель Петрозаводский государственный университет
 e-mail: katarov@psu.karelia.ru

Разработка системы оптимального проектирования сети лесовозных автомобильных дорог¹

Рассмотрены вопросы применения новых информационных технологий и математических методов для решения задачи проектирования сетей лесовозных автомобильных дорог (ЛАД). Приведены общая постановка задачи, применяемые модели, обоснованы используемые методы и средства решения поставленных задач.

Ключевые слова: лесозаготовки, сеть лесных дорог, планирование лесных дорог, строительство дорог, объектные пространственные модели, математические модели, оптимизация, географические информационные системы

Введение

Актуальность проблемы модернизации сетей лесовозных автомобильных дорог (ЛАД) в России на сегодняшний день очень велика. Низкая обеспеченность транспортной инфраструктурой сдерживает развитие лесопромышленного комплекса, так как существенно ограничиваются возможности более полного освоения эксплуатационных лесов и снижается экономическая доступность лесных ресурсов. В этой связи особое значение имеют мероприятия, позволяющие повысить эффективность строительства и эксплуатации ЛАД [1, 2].

Одним из направлений повышения эффективности строительства и эксплуатации ЛАД является всестороннее обоснование планов строительства дорог, а также транспортных планов, которое может быть организовано с применением новых инфор-

¹ Работа выполнена при финансовой поддержке Федерального агентства по науке и инновациям в рамках ФЦП "Научные и научно-педагогические кадры инновационной России" и Европейского Союза в рамках международного проекта "Лесозаготовки и логистика в России", финансируемого через финское агентство по развитию технологий и инноваций (TEKES).

мационных технологий и современных методов поддержки принятия решений, основанных на имитационном моделировании и математическом программировании. Эти методики отличаются от традиционно применяемых тем, что они позволяют учитывать гораздо большее число различных факторов и делать это точнее. Кроме того, благодаря применению современных средств они способствуют сокращению расходов на проектирование, строительство и эксплуатацию ЛАД [3—8].

Постановка цели и задач

Главной целью при проектировании оптимальной сети ЛАД следует считать обеспечение транспортной доступности основных лесосырьевых баз при условии минимальных затрат на строительство дорог. При этом должны быть приняты во внимание пространственное расположение наиболее перспективных лесных массивов, природно-производственные факторы, такие как распределение различных типов грунтов по территории лесосырьевой базы, расположение русел рек, наличие озер, болот, других непреодолимых препятствий или особо охраняемых зон, где по каким-либо причинам запрещено дорожное строительство. Кроме того, должны быть учтены возможные источники материалов для дорожного строительства (песчано-гравийные карьеры и т. п.), а также существующая сеть дорог.

Таким образом, все влияющие факторы можно разделить на четыре основные группы (рис. 1).

1. **Лесосырьевая база:** пространственное распределение лесных ресурсов, основные характеристики древостоев (запас, распределение по породам, средний объем хлыста и т. п.), местоположение перспективных для назначения в рубку лесных участков и т. д.

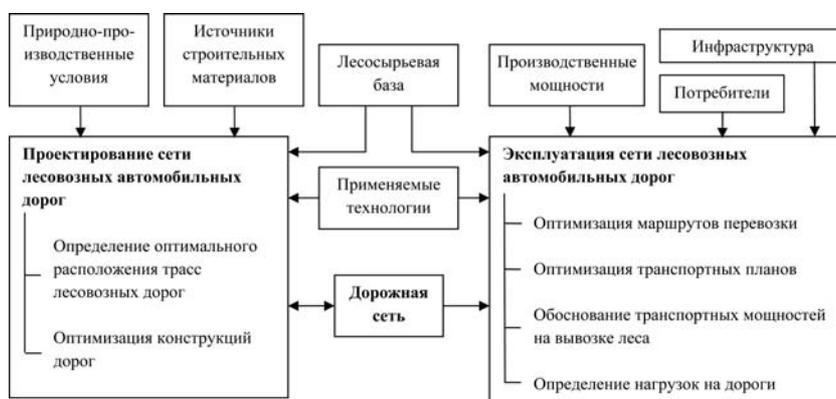


Рис. 1. Постановка задачи

2. *Природно-производственные условия:* пространственное распределение различных типов грунтов, расположение рек, озер и болот, рельеф, особо охраняемые территории и т. д.

3. *Источники строительных материалов:* расположение, доступность, характеристики карьеров.

4. *Имеющаяся сеть дорог и ее состояние.*

Основными **задачами** разработки системы проектирования сети ЛАД являются определение характеристик оптимальной сети дорог, предназначенных для вывозки круглых лесоматериалов и древесного топливного сырья (древесины) большегрузными автомобилями-сортиментовозами, а также определение оптимальных планов вывозки древесины. Эти задачи решаются для условий определенной лесосырьевой базы с заданными природно-производственными характеристиками, характеристиками производственных мощностей, существующей дорожной сети, требованиями и пространственным расположением потребителей древесины и другими внешними условиями (рис. 1). Кроме того, решаются некоторые сопутствующие задачи, связанные, например, с расчетом затрат на строительство дорог, определением оптимальных маршрутов по имеющейся сети, обоснованием производственных мощностей на вывозе древесины и т. п.

В настоящей статье описан подход к разработке компьютерной информационно-вычислительной системы (КИВС) для проектирования сети ЛАД, предназначенной для использования на предприятиях отрасли.

Обоснование применяемых методов и средств

При обосновании применяемых методов и средств для решения поставленных задач, прежде всего, следует принять во внимание, что создаваемые КИВС предназначены для применения непосредственно на лесозаготовительных предприятиях. Поэтому одним из важных требований должно стать обеспечение дружественной среды для пользователя. Объектные базы данных и модели должны быть разработаны в естественной наглядной форме, а результаты функционирования оптимизационных алгоритмов должны выводиться в форме, удобной для дальнейшего использования в соответствующих производственных процессах.

В этой связи в качестве основы для создания проектируемых систем целесообразно использовать географические информационные системы (ГИС). Это обусловлено тем, что весьма значительный объем исходных данных, необходимых для решения поставленных задач, носит пространственный характер. Из всего спектра средств, предоставляемых современными информационными

технологиями, только ГИС позволяют эффективно хранить и обрабатывать подобного рода информацию. Кроме того, использование ГИС позволит упростить работу конечного пользователя благодаря естественной наглядной форме представления обрабатываемой информации в виде слоев электронной карты [4, 9].

В качестве основной группы методов, которые необходимо применять для решения задач оптимизации проектирования и эксплуатации ЛАД, будут выступать методы математического программирования. При этом будет необходимо построение целого ряда моделей и решение нескольких оптимизационных задач, для каждой из которых следует подобрать наиболее эффективный алгоритм.

Для обеспечения процессов планирования сети и проектирования ЛАД должны быть разработаны следующие модели:

- объектные пространственные компьютерные ГИС-модели лесосырьевой базы; местности; карьеров; дорожной сети;
- математические модели, в том числе и оптимизационные — оптимизационная модель принятия решений о выборе конструкций земляного полотна и дорожных одежд ЛАД; модель оценки альтернативных участков ЛАД; модель выбора ключевых точек лесосырьевой базы; оптимизационная модель принятия решений о размещении сети ЛАД.

Все объектные модели должны быть реализованы в среде ГИС.

Структура системы. Исходные данные

С учетом сказанного, структура КИВС, предназначенной для решения задачи проектирования сети ЛАД, выглядит следующим образом (рис. 2).

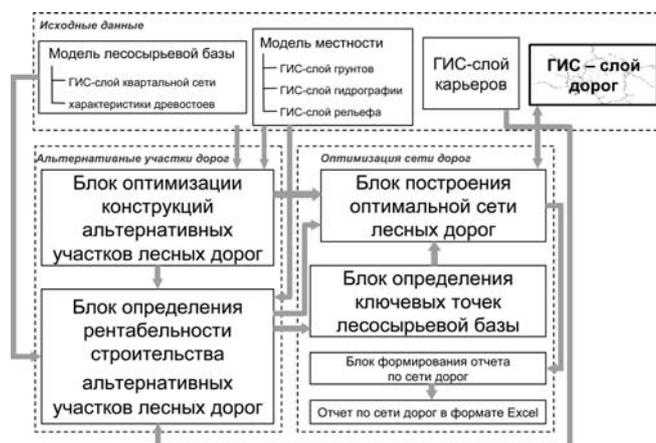


Рис. 2. Структура компьютерной информационно-вычислительной системы для решения задачи проектирования сети лесовозных автомобильных дорог

Исходными данными служат четыре объектных пространственных модели в формате ГИС. Каждая модель представляет собой комбинацию слоев электронной карты и баз данных, содержащих информацию о соответствующих объектах.

Модель лесосырьевой базы содержит таксационное описание основных характеристик древостоя для каждого объекта (лесного квартала). Основными, абсолютно необходимыми, характеристиками в данном случае являются общий запас спелых и перестойных насаждений и запас с подразделением по породам деревьев.

Модель местности состоит, как минимум, из двух слоев — слоя грунтов и слоя гидрографии. Для пересеченной местности желательно включить ГИС-слой рельефа. База данных слоя грунтов содержит идентификаторы каждого типа грунта для определения оптимальной конструкции дороги.

ГИС-слой карьеров содержит информацию о пространственном расположении потенциальных источников материалов для дорожного строительства. В соответствующую базу данных заносятся характеристики материалов, добываемых в карьере.

Кроме того, необходимо иметь слой, содержащий текущую дорожную сеть, для привязки к ней вновь проектируемых участков.

Таким образом, исходные данные представляют собой пространственную модель, содержащую описание различных аспектов, связанных с дорожным строительством, реализованную в среде географических информационных систем (рис. 3, см. вторую сторону обложки).

Методика построения оптимальной сети лесовозных автомобильных дорог

Предлагаемая методика базируется на решении задачи Штейнера на графах. Этот подход достаточно хорошо зарекомендовал себя при решении подобных задач [10—14]. Такая постановка требует задания, во-первых, множества вершин и множества дуг, и, во-вторых, множества целевых вершин, являющегося подмножеством множества вершин.

Каждой вершине из множества вершин должен быть присвоен вес, прямо пропорциональный выигрышу, получаемому от заготовки древесины в некоторой окрестности этой вершины. Каждой дуге также присваивается вес, прямо пропорциональный рентабельности строительства участка дороги, совпадающего с данной дугой.

При практической реализации предлагается привязывать каждую вершину графа к одному из кварталов рассматриваемой лесосырьевой базы и помещать ее в геометрическом центре (центроиде) лесного квартала. При этом исходными данными

для расчета веса вершины являются характеристики древостоя в данном квартале, которые содержатся в соответствующей базе данных объектной пространственной модели.

Множество дуг предлагается заполнять отрезками прямых, соединяющих каждую вершину с вершинами, находящимися в центроидах тех кварталов, которые имеют хотя бы одну пограничную точку с кварталом, в центроиде которого расположена рассматриваемая вершина. Другими словами, центроид каждого квартала соединяется с центроидами всех соседних кварталов, причем соседними считаются кварталы, имеющие хотя бы одну пограничную (общую) точку.

Полученные таким образом дуги будут являться альтернативными участками проектируемых дорог. Для каждого из них должен быть определен вес. Для этого служит подсистема "Альтернативные участки дорог" (см. рис. 2).

Данная подсистема состоит из двух блоков, работающих последовательно: блока оптимизации конструкций альтернативных участков лесных дорог и блока определения рентабельности строительства альтернативных участков лесных дорог.

В блоке оптимизации конструкций альтернативных участков лесных дорог в результате решения оптимизационной задачи выбирается конструкция дорожного полотна и определяются затраты строительства данного альтернативного участка с учетом грунтовых условий по оси трассы, а также с учетом необходимости строительства мостов и других водопропускных сооружений.

В общем виде эта оптимизационная модель выглядит следующим образом [2]:

$$\begin{cases} C_d = f(c, B, h_{п.ср}, h_{о.ср}, E_{гр}, C_{\Sigma}, l_o, l_{п}, l_{зп}, \\ C_{41}, C_{42}, C_{ГМ}) \rightarrow \min; \\ E_{ф}(h_{п.ср}, h_{о.ср}, E_{гр}, E_o, E_{п}, \alpha) \geq K_{пр}^{ТР} E_{ТР}(N); \\ h_{п.ср}, h_{о.ср} \geq 0,05 \text{ м}; \\ C_{\Sigma}, l_o, l_{п}, l_{зп}, C_{ГМ}, \alpha \geq 0, \end{cases}$$

где C_d — общие затраты на строительство участка дороги, руб.; c — ширина обочины, м; B — ширина проезжей части, м; $h_{п.ср}$ — средняя толщина покрытия дорожной одежды (ПДО), м; $h_{о.ср}$ — средняя толщина основания дорожной одежды (ОДО), м; $E_{гр}$ — модуль упругости грунта земляного полотна (ЗП); C_{Σ} — общая стоимость строительных материалов дорожной одежды, руб.; l_o — расстояние транспортировки материала ОДО, км; $l_{п}$ — расстояние транспортировки материала ПДО, км; $l_{зп}$ —

расстояние транспортировки материала ЗП, км; C_{41} — стоимость устройства водопропускных сооружений — труб, руб.; C_{42} — стоимость устройства водопропускных сооружений — мостов, руб.; $C_{ГМ}$ — стоимость армирования геоматериалами, руб.; E_{ϕ} — фактический модуль упругости дорожной одежды; E_0 — модуль упругости материала ОДО; E_{Π} — модуль упругости материала ПДО; α — коэффициент упрочнения дорожной одежды при армировании геоматериалами; $K_{\text{пр}}^{\text{тр}}$ — требуемый коэффициент прочности дорожной одежды по критерию упругого прогиба; $E_{\text{тр}}$ — минимальный требуемый общий модуль упругости конструкции; N — число приложений нагрузки от осей с нагрузкой 10 т в весенний период (апрель, май).

Управляемыми переменными в данном случае являются $h_{\text{п.ср}}$, $h_{\text{о.ср}}$, C_{Σ} , l_0 , $l_{\text{п}}$, $l_{\text{зп}}$, $C_{\text{ГМ}}$, α . Таким образом, в результате решения поставленной задачи можно определить оптимальные толщины слоев дорожной одежды, подобрать карьеры для строительства и, если необходимо, способ упрочнения дорожной одежды, соответствующие минимальным затратам на строительство участка с учетом начальных условий местности.

Если на протяжении альтернативного участка грунтовые условия изменяются, участок разбивается на более мелкие части, для которых расчет затрат выполняется по отдельности с последующим суммированием по всему участку. Если участок является непроходимым по каким-либо причинам (например, на пути встречается озеро), затраты на строительство принимаются условно бесконечными. Если участок дороги уже существует, тогда затраты приравниваются к нулю или к значению, равному затратам на ремонт участка, если таковой потребуется.

Для обеспечения работы алгоритма построения оптимальной сети лесных дорог каждый альтернативный участок оценивается в соответствии с перспективностью его строительства. Это осуществляется в блоке определения рентабельности строительства альтернативных участков лесных дорог (см. рис. 2). В качестве соответствующей оценки применяется стоимостной критерий, равный разности между выигрышем, получаемым от возможности заготовки в соответствующем квартале, и затратами, связанными со строительством этого участка дороги:

$$K_{\text{а.д}} = C_{\text{к}} - C_{\text{д}},$$

где $C_{\text{к}}$ — выигрыш, получаемый от возможности заготовки в соответствующем квартале,

$$C_{\text{к}} = \sum_{i=1}^n \sum_{j=1}^m V_i (h_{ij} C_{ij} - L_{\text{т}ij} C_{\text{т}ij}),$$

где n — число пород деревьев, заготавливаемых на рассматриваемом участке; m — число видов продукции, получаемой на делянках; V_i — объем разрешенной к заготовке в расчетном периоде спелой и перестойной древесины i -й породы в квартале, м^3 ; h_{ij} — средняя доля j -го вида продукции, получаемого из i -й породы; C_{ij} — прогнозируемая на расчетный период средняя цена j -го вида продукции, получаемого из i -й породы, руб.; $C_{\text{т}ij}$ — средние затраты на транспортировку 1 м^3 j -го вида продукции, получаемого из i -й породы на расстояние 1 км , руб./ $\text{м}^3/\text{км}$; $L_{\text{т}ij}$ — среднее расстояние перевозки j -го вида продукции, получаемого из i -й породы для данного квартала, км.

Для более точного учета местонахождения наиболее привлекательных с точки зрения заготовки участков леса предлагается провести корректировку значения $C_{\text{к}}$ каждого квартала с учетом значений $C_{\text{к}}$ соседних кварталов внутри круга заданного радиуса R . Скорректированное значение определяется по следующей формуле:

$$C'_{\text{к}} = C_{\text{к}}(1 - g) + \frac{g \sum_{i=1}^n \frac{C_{\text{к}i}}{d_i^N}}{\sum_{i=1}^n \frac{1}{d_i^N}},$$

где $C_{\text{к}}$ — исходное значение выигрыша, получаемого от возможности заготовки в квартале, руб.; g — степень влияния соседних кварталов ($g = 0, 1 \dots 0,5$); n — число соседних кварталов, попавших внутрь круга заданного радиуса R ; $C_{\text{к}i}$ — исходные значения выигрыша, получаемого от возможности заготовки в соседних кварталах, попавших во внутрь круга радиуса R , руб.; d_i — расстояние от центроида рассматриваемого квартала до центроида i -го соседнего квартала, км; N — параметр интерполяции ($N = 1 \div 3$).

На следующем этапе вступает в действие подсистема "Оптимизация сети дорог" (см. рис. 2). Для решения задачи Штейнера на графе определяется множество целевых вершин, соответствующих ключевым точкам лесосырьевой базы. Для

этого служит "Блок определения ключевых точек лесосырьевой базы".

В этой стадии процесса участвует пользователь системы. Сначала он задает число целевых точек. Система предлагает ему соответствующее число лесных кварталов с наибольшими значениями C'_k . Далее пользователь может скорректировать число и расположение целевых точек. Для его удобства с помощью метода IDW-интерполяции строится тематическая карта (поверхность), где цветом выделяются зоны, более благоприятные и менее благоприятные с точки зрения заготовки.

После определения целевых вершин запускается "Блок построения оптимальной сети лесных дорог". Размещение оптимальной сети ЛАД осуществляется в процессе решения задачи Штейнера на сгенерированном графе [10—14]. Данная задача заключается в определении частичного подграфа, содержащего пути в каждую из целевых вершин при условии максимума суммы весов всех задействованных при этом дуг. В качестве весов при этом выступают значения $K_{ад}$.

Поставленная задача решается методом динамического программирования [15—18] со следующим основным рекуррентным соотношением:

$$W(s, j) = \max(\max(W(s', j) + W(s \setminus s', j) | s' \subset s), \max(c_u + W(s, j_{e(u)}) | j_{b(u)} = j)),$$

где s — подмножество множества целевых вершин; j — вершина из множества целевых вершин; u — управление, заключающееся в выборе дуги, соединяющей вершину j с вершинами вне подмножества s ; c_u — вес дуги, соответствующей управлению u ; $j_{e(u)}$ — конечная вершина дуги, соответствующей управлению u ; $j_{b(u)}$ — начальная вершина дуги, соответствующей управлению u .

Полученный подграф и будет являться искомой оптимальной сетью ЛАД в рассматриваемых условиях.

По результатам работы блока построения оптимальной сети ЛАД в ГИС-слой дорог вносятся соответствующие изменения — он дополняется вновь спроектированными дорогами. Кроме того, в системе предусматриваются инструменты для автоматизированной генерации отчетов по разработанным проектам, которые в числе прочего должны содержать информацию по протяженности новых дорог, предварительных затратах на строительство в целом и по их элементам, дополнительных объемах древесины, доступной для заготовки, и т. д. Эту функцию выполняет "Блок формирования отчета по сети дорог".

Основными результатами функционирования представленной КИВС являются пространственные и конструктивные характеристики сети ЛАД, оптимальные с точки зрения рентабельности ее строительства, а также долгосрочные планы вывозки древесины, оптимальные с точки зрения эффективности использования транспортных средств и маршрутов движения.

Список литературы

1. Развитие транспортной инфраструктуры лесной отрасли — опыт Финляндии / Ю. Ю. Герасимов, С. Карвинен, В. С. Сютёв, А. П. Соколов, В. К. Катаров // Транспортное дело России. 2009. № 7(68). С. 99—102.
2. Создание инфраструктуры лесных дорог и управление системами транспортировки древесины для лесопромышленного комплекса и биоэнергетики. Первый этап: Разработка концепции оптимального проектирования и эксплуатации сети лесовозных дорог: отчет по НИР / Ю. Ю. Герасимов, В. С. Сютёв, А. П. Соколов, А. А. Селиверстов, В. К. Катаров, Д. В. Рожин. Петрозаводск, 2009. 72 с.
3. Герасимов Ю. Ю., Костюкевич В. М. Логистика в лесном комплексе: управление снабжением, транспортом и запасами: учеб. пособие. Петрозаводск: Изд-во ПетрГУ, 2001. 108 с.
4. Перспективы применения новых информационных технологий в лесном комплексе / Ю. Ю. Герасимов, Г. А. Давыдов, С. А. Кильпелайнен, А. П. Соколов, В. С. Сютёв // ИВУЗ, Лесной журнал. 2003. № 5. С. 122—128.
5. Герасимов Ю. Ю., Соколов А. П., Сютёв В. С. Логистика лесозаготовок: программа поиска оптимального лесотранспортного плана // Лесная Россия. 2008. № 5—6. С. 54—61.
6. Gerasimov Yu. Yu., Sokolov A. P., Karjalainen T. GIS-based decision-support program for short-wood transport in Russia // The Nordic-Baltic Conference on Forest Operations. Copenhagen September 23-25, 2008. Forest & Landscape Working Papers. 2008. No. 30.
7. Gerasimov Yu. Yu., Sokolov A. P., Karjalainen T. GIS-based Decision-Support Program for Planning and Analyzing Short-Wood Transport in Russia // Croatian Journal of Forest Engineering. 2008. Vol. 29, Is. 2. Zagreb: University of Zagreb. P. 163—175.
8. Соколов А. П., Герасимов Ю. Ю. Геоинформационная система для решения оптимизационной задачи транспортной логистики круглых лесоматериалов // ИВУЗ. Лесной журнал. 2009. № 3. С. 78—85.
9. На коллегии Рослесхоза // Лесное хозяйство. 2000. № 4. С. 54—56.
10. Романовский И. В. Дискретный анализ. СПб.: Невский диалект, 2003. 320 с.
11. Кристофидес Н. Теория графов. Алгоритмический подход. М.: Мир, 1978. 432 с.
12. Левин А. Ю. Алгоритм кратчайшего соединения группы вершин графа // Доклады АН СССР. 1971. Т. 200. № 4. С. 773—776.
13. Лотарев Д. Т. Задача Штейнера для транспортной сети на поверхности, заданной цифровой моделью // Автоматика и телемеханика, 1980. Т. 10. С. 104—115.
14. Солтан П. С., Присакару К. Ф. Задача Штейнера на графах // Доклады АН СССР. 1971. Т. 198, № 1. С. 46—49.
15. Андреев В. Н., Герасимов Ю. Ю. Принятие оптимальных решений в лесном комплексе. Йоэнсуу: Изд-во университета Йоэнсуу, 1999. 200 с.
16. Беллман Р. Динамическое программирование. М.: Иностранная литература, 1960. 400 с.
17. Вентцель Е. С. Исследование операций. Задачи, принципы, методология: учеб. пособие для студ. вузов. 2-е изд., стер. М.: Высш. шк., 2001. 208 с.
18. Вентцель Е. С. Элементы динамического программирования. М.: Наука, 1964. 176 с.

УДК 531.383

В. Э. Джашитов, д-р техн. наук, проф., зав. лаб.,
В. М. Панкратов, д-р техн. наук, проф.,
зам. директора по науке,
А. В. Голиков, канд. техн. наук, вед. науч. сотр.
Институт проблем точной механики
и управления РАН, г. Саратов
e-mail: iptmuran@san.ru

Мультимедийные информационно-компьютерные технологии в научно-образовательных курсах лекций и экспресс-контроле знаний по точным наукам

Предложен инновационный подход к разработке научно-образовательных курсов лекций по точным наукам с последующим экспресс-контролем знаний. Рассмотрены методические аспекты и практический опыт создания и применения мультимедийных компьютерных технологий при реализации нового подхода. Лекции представлены в виде пакетов презентационных слайдов в PowerPoint и встроенных программ динамической визуализации. Предусмотрено компьютерное моделирование и динамическая трехмерная визуализация принципов действия, уравнений движения, физических процессов и полей рассматриваемых типов измерительных приборов, кинематики и динамики различных механизмов и феноменов механического движения. Для динамической визуализации разработано оригинальное программное обеспечение.

Ключевые слова: информационно-компьютерные технологии в образовании, мультимедийные лекции по точным наукам, приборы, инерциальные датчики, механизмы, физические поля и процессы, динамическая визуализация

Введение

В процессе подготовки в университетах будущих создателей и исследователей в области динамических мехатронных систем и систем управления техническими объектами им преподаются разнообразные курсы лекций по точным наукам.

К основным специальным курсам относятся:

1. Теоретическая механика.
2. Общая и прикладная теория гироскопов.
3. Математические модели температурно-возмущенных датчиков инерциальной информации и датчиков других физических величин.
4. Микроэлектромеханические системы и методы обеспечения их термоинвариантности.
5. Математические модели и методы компьютерной визуализации физических процессов и полей, геометрических структур и феноменов механического движения в динамических системах и многие другие.

Современные требования к таким научно-образовательным курсам лекций чрезвычайно высоки. Теоретические положения и практические примеры в этих лекциях весьма сложны по содержанию и по форме.

Вместе с тем, возможности информационно-компьютерных технологий позволяют проводить обучение не только с помощью "мела и доски", но и с помощью специально разработанных и стандартных программных средств, компьютера и мультимедийного проектора.

Цель статьи: обмен опытом по подготовке научно-образованных создателей современной техники и систем управления, углубленному и наглядному изучению фундаментальных и прикладных научных аспектов исследования технических систем и подвижных объектов, внедрению современных информационных технологий в образовательный процесс.

Ключевые задачи:

- существенное повышение эффективности обучения в точных науках;
- разработка новой методики применения теоретических и программных средств;
- создание необходимого мультимедийного программного обеспечения для чтения лекций и проведения экспресс-контроля знаний.

Материал статьи основывается на общей концепции и положениях современных информационных технологий в образовании, представленных в работах [1–3] отечественных ученых.

Далее рассмотрены основные положения и особенности предлагаемого подхода, методические аспекты и практический опыт создания, реализации и преподавания этих курсов лекций [1–7].

Основные положения и особенности предлагаемого подхода

В настоящее время полностью разработаны и внедрены в учебный процесс первые четыре из вышеперечисленных курсов лекций.

Пятый курс находится в стадии разработки теоретического и программного обеспечения.

В основе мультимедийных лекций — книги и учебники выдающихся отечественных и зарубежных ученых и оригинальные работы [4] авторов этой статьи.

Основные особенности единой методики создания и преподавания всех разработанных курсов лекций следующие.

1. Все лекции представлены в виде *пакетов презентационных слайдов* в стандартной программной среде PowerPoint и банков специально разработанных *программ динамической визуализации* движения механизмов, мехатронных датчиков и систем, принципов действия гироскопов и т. д.

Для примера на рис. 1 показана структура мультимедийных компьютерных лекций по теоретической механике.

Содержание и объем мультимедийных курсов лекций определяется государственными образовательными стандартами и программами обучения по той или иной специальности.

Например, для студентов строительных специальностей Саратовского государственного технического университета объем лекций по *теоретической механике* ≈ 50 ч. *Содержание* лекций включает базовые разделы (статика, кинематика и динамика) и дополнительные разделы (элементы аналитической механики, теории удара и нелинейной динамики).

Для студентов специальности "Приборостроение" объем лекций по *теории гироскопов* ≈ 34 ч. *Содержание лекций* включает:

- основные положения теоретической механики, теории теплообмена, теории упругости и термоупругости, аэрогидромеханики и оптики, используемые в теории современных гироскопов;
- физические основы функционирования основных типов гироскопов (гироскопы в кардановом подвесе, поплавковые, динамически настраиваемые, электростатические, волновые твердотельные, микромеханические, волоконно-оптические гироскопы);
- специальные задачи теории возмущенных гироскопов.

2. В лекциях предусмотрено *компьютерное моделирование* и *динамическая двух- и трехмерная визуализация* кинематики и динамики механизмов и феноменов механического движения, принципов действия всех основных типов гироскопов, функционирования мехатронных датчиков физических величин и мехатронных систем, физических полей и процессов.

Для *динамической визуализации* разработано специализированное программное обеспечение.

При этом использован программный комплекс Borland C++ Builder и графический интерфейс Open GL. Вызов специализированных программ из презентаций PowerPoint осуществляется с помощью гиперссылок.

Важнейшей особенностью и отличием от существующих подходов является то, что динамическая визуализация осуществляется на основе полученных уравнений движения, разработанных алгоритмов и других математических моделей приборов, датчиков, механизмов и физических процессов, в них протекающих.

Это не 3D-, не Flash и не какая-либо другая анимация, а визуализация построенных математических моделей функционирования технических систем и физических процессов, в них протекающих.

Разработанные оригинальные программы динамической визуализации и составляют "Know How" предлагаемого подхода.

Созданные программы динамической визуализации полностью автономны, используют стандартные возможности Windows и не требуют установки на компьютер никаких универсальных программных систем аналитических и других вычислений или специализированных программных систем с возможностями визуализации и анимации.

В конечном итоге пользователь работает только с исполняемыми файлами (*.exe) и файлами ис-

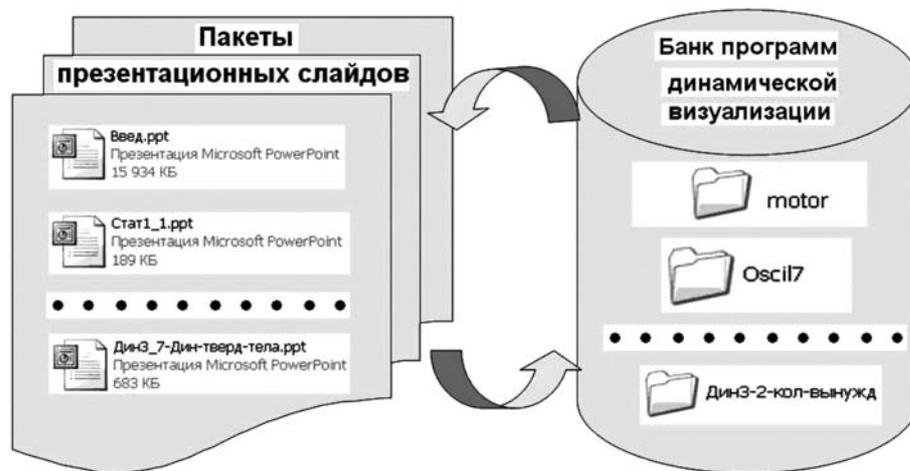


Рис. 1. Структура компьютерных лекций по теоретической механике

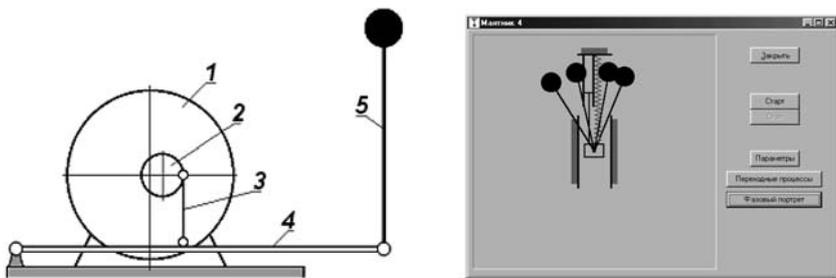


Рис. 2. Прибор П. Л. Капицы (1 — электромотор; 2 — подшипник; 3 — тяга; 4 — рычаг; 5 — маятник) и диалоговое окно программы визуализации устойчивых колебаний маятника около верхнего положения при вертикальных вибрациях основания

ходных данных, где задаются необходимые параметры для моделирования и визуализации.

В необходимых случаях использованы возможности и встроенная анимация программной среды PowerPoint.

Математическое моделирование и динамическая визуализация позволяют не только изучить физические процессы и разнообразные феномены механического движения, но и оценить влияние параметров на работу механизмов и мехатронных датчиков и систем, наглядно представить функционирование различных типов гироскопов и т. д.

На рис. 2 в качестве примера из теоретической механики показан реальный прибор П. Л. Капицы и его компьютерная модель, демонстрирующие в соответствии с уравнениями движения феномен устойчивости верхнего положения математического маятника с вертикально колеблющейся точкой подвеса.

Уравнения движения (математическая модель) маятника имеют вид

$$\ddot{\theta} + a_2 \sin \theta - a_3 \ddot{x} \sin \theta = 0;$$

$$\ddot{x} + b_2 \dot{x} - b_3 (\ddot{\theta} \sin \theta + \dot{\theta}^2 \cos \theta) = H \cos \omega t,$$

где x, θ — обобщенные координаты маятника; H, ω — амплитуда и частота вертикальных колебаний точки подвеса; $a_2 = g/l, a_3 = 1/l, b_2 = c/m, b_3 = l; l, m$ — длина и масса маятника; c — жесткость подвеса маятника.

В качестве примера из теории гироскопов на рис. 3 (см. четвертую сторону обложки) показана реальная конструкция роторного микромеханического гироскопа [8] и его компьютерная модель, демонстрирующая первичные колебания (вокруг оси z) и возникающие при наличии измеряемой угловой скорости Ω основания вторичные колебания (вокруг оси y) ротора датчика в соответствии с его уравнениями движения.

На рис. 4 (см. четвертую сторону обложки) представлены диалоговые окна программ динамической визуализации движений и принципов действия классических и современных гироскопов и механизмов, изучаемых в курсах теории гироскопов и теоретической механики.

На рис. 5 (см. четвертую сторону обложки) в качестве примера из специального курса "Математические модели температурно-возмущенных датчиков" представлена визуализация нестационарных температурных полей и тепловых процессов, полученных при моделировании датчика температуры для ракетно-космической техники.

3. Сложные рисунки, формулы, фотографии, графики, таблицы, программа лекций и т. п. распечатываются и выдаются обучаемым слушателям в качестве *раздаточного материала* (рис. 6).

4. Для экспресс-контроля степени усвоения материала лекций разработан универсальный программный комплекс и специализированные содержательные тесты по читаемым курсам лекций.

Программный комплекс для тестирования представляет собой систему управления базами данных и состоит из двух программ: редактора банка для создания, наполнения и редактирования заданий для тестирования и тестера для проведения тестирования. Редактор банка предназначен для использования только администратором (преподавателем).

Пример 3.11. Электромотор массы m_1 на горизонтальной идеально гладкой плоскости и прикреплен к ней болтами K, L . На валу мотора закреплен одним концом невесомый стержень длиной r . На другом конце стержня точечная масса m_2 . Вал вращается с угловой скоростью $\omega = \text{const}$. Ускорение свободного падения g .
Определить как функции времени $0 \leq t \leq t_*$:

- Суммарную нормальную силу реакции $R_y(t)$ плоскости и горизонтальную силу реакции $R_x(t)$ болтов K, L .
- Критическую угловую скорость $\omega_{кр}$ вращения, при которой мотор оторвется от пола, если болты срежутся.
- Закон движения $x_1 = x_1(t), y_1 = y_1(t)$ и траекторию $y_1 = y_1(x_1)$ центра масс C мотора при отсутствии болтов.

Решение

I. Электромотор прикреплен к плоскости болтами K, L .
 Исходные уравнения и соотношения из теоремы о движении ц. м.

$$(m_1 + m_2)\ddot{x}_C = \sum F_x^E = R_x, \quad (m_1 + m_2)\ddot{y}_C = \sum F_y^E = R_y - m_1 g - m_2 g$$

$$(m_1 + m_2)\ddot{x}_C = m_1 \ddot{x}_1 + m_2 \ddot{x}_2, \quad (m_1 + m_2)\ddot{y}_C = m_1 \ddot{y}_1 + m_2 \ddot{y}_2$$

$$x_1 = 0, \quad y_1 = CO = \text{const}, \quad x_2 = r \cos \omega t, \quad y_2 = CO + r \sin \omega t$$

1. Силы реакции $R_x(t) = -m_2 \omega^2 \cos \omega t, \quad R_y(t) = (m_1 + m_2)g - m_2 r \omega^2 \sin \omega t$

2. Критическая угловая скорость вращения вала $\omega_{кр} = \sqrt{\frac{m_1 + m_2}{m_2} g}$

II. Болты K, L внезапно срезаются
 Исходные уравнения $\ddot{x}_C = 0, \quad \ddot{y}_C = -g, \quad y_2(t) = y_1(t) + r \sin \omega t$

Угловая скорость вращения вала $\omega < \omega_{кр}$ $x_1(t) = \frac{m_2 r}{m_1 + m_2} (1 - \cos \omega t)$

Угловая скорость вращения вала $\omega > \omega_{кр}$ $x_1(t) = \frac{m_2 r}{m_1 + m_2} (1 - \cos \omega t)$

$$y_1(t) - b = -\frac{gt^2}{2} + \frac{m_2 r}{m_1 + m_2} (\omega t - \sin \omega t)$$

Рис. 6. Пример слайда из курса лекций по теоретической механике со сложными рисунками и формулами в качестве раздаточного материала

Возможности предлагаемого подхода, проблемы, перспективы

К ним можно отнести следующее.

Существенное повышение производительности и качества обучения. Один и тот же объем сложного учебного материала преподается с высоким качеством и за значительно меньшее время (по предварительным оценкам — в 2 и более раз), чем при традиционной методике чтения лекций.

Гибкая и эффективная система модернизации лекций, позволяющая добавлять, изменять и исключать те или иные фрагменты.

Возможность демонстрации (в виде фотографий, видеофильмов, дополнительных презентаций и т.д.) реальных конструкций приборов и механизмов, их работы, исторических, документальных и других материалов.

Применение разработанной методики и программных средств позволяет решить следующие важные *научные и педагогические задачи*:

- для учащихся — углубленное изучение, понимание и усвоение сложного материала, повышение интереса к предмету и возможность изучения самых новых достижений в рассматриваемых областях науки и техники;
- для лектора — оптимизация и повышение эффективности обучения.

Основная проблема — большая трудоемкость (интеллектуального и временного характера) при разработке сценариев лекций, при проектировании и создании слайдов и программного обеспечения для динамической визуализации.

Главная перспектива — это разработка *новых* теоретических, методических и программных средств для чтения лекций, реализации и контроля практических и учебно-исследовательских работ по основным и специальным курсам точных наук.

Заключение

Разработанные компьютерные курсы лекций по теоретической механике, общей и прикладной теории гироскопов и др. прошли первичную апробацию в Саратовском государственном техническом университете при чтении лекций для студентов приборостроительных, строительных и других специальностей.

При этом авторами накоплен определенный опыт использования предлагаемых информационных технологий и методики проведения занятий, позволяющий поделиться с читателями некоторыми практическими рекомендациями, которые приведены ниже.

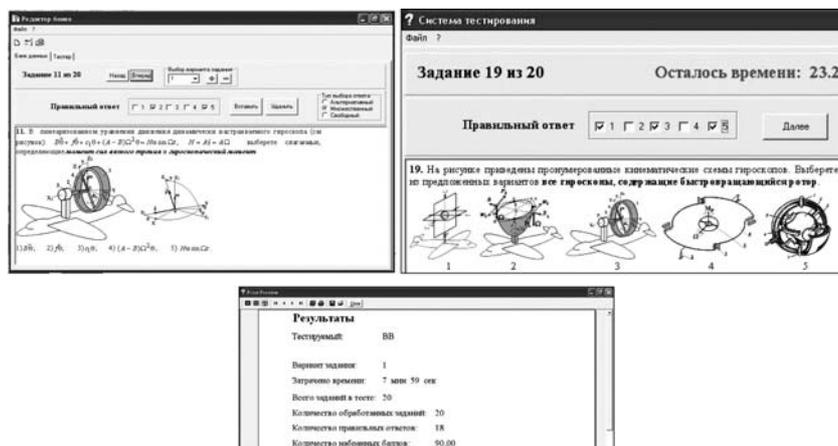


Рис. 7. Диалоговые окна программ экспресс-контроля знаний по теории гироскопов

Программный комплекс является автономной системой, не требует инсталляции, распространяется посредством обычного копирования и работает в любой операционной системе семейства Windows.

На рис. 7 показан вид диалогового окна такой программы по курсу "Общая и прикладная теория гироскопов".

5. При чтении лекций из аппаратуры используется ноутбук и мультимедийный проектор.

Практическая реализация

Мультимедийные курсы лекций разработаны каждый в электронном и печатном вариантах, неразрывно связанных друг с другом.

Электронный вариант записывается на CD и является неотъемлемой частью всего курса. Печатный вариант издается в виде книги, к которой прилагается CD с электронным вариантом.

В электронном варианте представлены основной теоретический материал в виде презентационных слайдов с анимацией и программы динамической визуализации принципов действия, уравнений движения, физических процессов и полей рассматриваемых типов измерительных приборов, кинематики и динамики различных механизмов и феноменов механического движения.

В печатном варианте представлены основные текстовый и графический материалы лекций, изложены основные особенности методики проведения лекций и приведены описание программных модулей, реализующих динамическую визуализацию, а также правила работы с программами.

В целом разработанные мультимедийные курсы лекций предназначены для преподавателей, читающих эти лекции, и могут быть использованы для дистанционного обучения и самообучения студентов и аспирантов.

Желательно ограничить число студентов в группе (<15...20) для лучшего усвоения ими курса, организации в течение занятия помощи слушателям и контроля со стороны лектора, а также для минимизации объема раздаточного материала (в основном это сложные рисунки).

В отличие от традиционной методики, когда лектор может практически *одновременно со студентами* (параллельно) рисовать на доске иллюстрации и излагать материал лекций, при чтении компьютерных лекций целесообразно действовать несколько по-другому (последовательно).

Предположим, на экране студент видит текущий слайд, на котором изложен материал конкретного раздела, приведены необходимые рисунки и, возможно, расположена кнопка вызова по гиперссылке программы динамической визуализации. Лектор сначала раздает заранее заготовленные рисунки, затем дает возможность студентам записать со слайда представленный материал, обращая внимание на главные и менее главные положения и вспомогательный материал (который можно не записывать). В это время лектор контролирует процесс в аудитории, в необходимых случаях подсказывает и помогает студентам. Когда слайд обработан студентами, лектор может дать свободные комментарии по представленному материалу и рисункам, продемонстрировать, например, принцип действия гироскопа или рассматриваемого механизма с помощью программ динамической визуализации. Далее осуществляется переход к следующему слайду.

Такая методика, безусловно, не единственно возможная. Лектор сам может опытным путем, в зависимости от степени подготовленности и других особенностей слушателей, выбрать оптимальный вариант чтения лекций.

Один из первых вопросов, которые задают студенты: "А нельзя ли "скинуть" им на "флэшку" или CD материал лекций и программы динамической визуализации?" На наш взгляд, это делать не только нецелесообразно, но и просто вредно по многим причинам. Назовем только одну из них.

Присутствие лектора, разъясняющего и помогающего понять и запомнить суть представленного сложного материала, является необходимым условием успешного обучения для очной его формы.

Вместе с тем, авторам представляется безусловно необходимым издание разработанных компьютерных курсов лекций (это может быть книга и прилагаемый к ней CD с компьютерным курсом лекций и программами динамической визуализации). Такое издание должно помочь преподавателям оптимизировать и повысить эффективность

обучения. Это издание может быть полезно и в системе дистанционного обучения.

И последнее.

При чтении компьютерных лекций должны *обязательно* быть выполнены основные требования к аппаратному оборудованию:

- наличие портативного и обладающего необходимыми ресурсами компьютера (ноутбука);
- размеры экрана должны обеспечивать видимость изображения с любого места в аудитории;
- необходимы достаточные *яркость* (от 2500 лм) и *контрастность* изображения на экране от мультимедийного проектора даже при ярком дневном свете.

Невыполнение этих простых технических требований может существенно снизить эффект применения самых передовых информационных технологий в образовании, чего, по мнению авторов, никак нельзя допускать.

Список литературы

1. **Норенков И. П., Зимин А. М.** Информационные технологии в образовании. М.: Изд-во МГТУ им. Э. Н. Баумана, 2004. 352 с. (Сер. Информатика в техническом университете).
2. **Информатизация образования: направления, средства, технологии: пособие для системы повышения квалификаций /** Под общ. ред. С. И. Маслова. М.: Изд-во МЭИ, 2004. 868 с.
3. **Соловов А. В.** Электронное обучение: проблематика, дидактика, технология. Самара: Новая техника, 2006. 462 с.
4. **Джашитов В. Э., Панкратов В. М.** Датчики, приборы и системы авиакосмического и морского приборостроения в условиях тепловых воздействий // Под общей ред. акад. РАН В. Г. Пешехонова. С.-Петербург: ГНЦ РФ ЦНИИ "Электроприбор", 2005. 404 с.
5. **Свидетельство** об отраслевой регистрации разработки № 9773 "Компьютерный курс лекций по общей и прикладной теории гироскопов". Федеральное агентство по образованию, ФГНУ "Государственный координационный центр информационных технологий", Отраслевой фонд алгоритмов и программ. 20.01.2008 /авторы: Джашитов В. Э., Панкратов В. М., Голиков А. В. № государственной регистрации 50200800190.
6. **Свидетельство** об отраслевой регистрации разработки № 10367 "Компьютерный курс лекций по теоретической механике". Федеральное агентство по образованию, ФГНУ "Государственный координационный центр информационных технологий", Отраслевой фонд алгоритмов и программ. 08.04.2008 / авторы: Джашитов В. Э., Панкратов В. М., Голиков А. В. № государственной регистрации 50200800840.
7. **Джашитов В. Э., Панкратов В. М., Голиков А. В.** Компьютерные научно-образовательные курсы лекций по теоретической механике, общей и прикладной теории гироскопов // Гироскопия и навигация. 2009. № 1. С. 113—121.
8. **Пешехонов В. Г., Несенюк Л. П., Грязин Д. Г., Некрасов Я. А., Евстигнеев М. И., Блажнов Б. А., Аксененко В. Д.** Инерциальные модули на микромеханических датчиках. Разработка и результаты испытаний // Тр. Юбилейной XV С.-Петербургской Международ. конф. по интегрированным навигационным системам. С.-Петербург, ЦНИИ "Электроприбор". 2008. С. 9—15.

Д. В. Пузанков, д-р техн. наук, проф., зав. каф.,
М. Г. Пантелеев, канд. техн. наук, доц.,
Н. А. Жукова, канд. техн. наук, ассистент,
В. С. Семенов, аспирант,
Санкт-Петербургский государственный
электротехнический университет "ЛЭТИ"
им. В. И. Ульянова (Ленина)
e-mail: mpanteleev@gmail.com

Информационная среда проектирования компетентностных моделей на основе технологий семантического Web

Обсуждается проблема создания информационной среды для проектирования компетентностных моделей распределенными группами экспертов. Анализируются требования к такой среде. Рассмотрен сценарий групповой работы экспертов, компоненты информационного обеспечения, основанные на онтологическом подходе, архитектура информационной среды и перспективы ее развития.

Ключевые слова: проектирование компетентностных моделей, информационная среда, распределенные группы экспертов, онтологический подход

Актуальность

Современный этап развития отечественной системы образования связан с освоением и широким внедрением в практику *компетентностного подхода* [1]. Данный подход положен в основу нового поколения федеральных государственных образовательных стандартов высшего профессионального образования (ФГОС ВПО) и предполагает задание результатов освоения образовательных программ в форме *компетентностной модели* (КМ) выпускника [2]. КМ представляет собой структурированную по группам совокупность компетенций, а под компетенцией понимается способность применять знания, умения, опыт и личностные качества для успешной деятельности в определенной области. С вводом в действие образовательных стандартов третьего поколения в вузах страны развернута широкомасштабная работа по созданию и внедрению соответствующих этим стандартам образовательных программ. Построение КМ, являясь первым этапом разработки образовательных программ, в значительной степени определяет их качество и практическую востребованность.

Процесс проектирования КМ характеризуется следующими основными особенностями:

- является интеллектуально-емкой коллективной экспертной деятельностью, требующей привлечения широкого круга высококвалифицированных специалистов из различных предметных областей;
- в построении КМ, в первую очередь в части профессиональных компетенций, наряду с представителями академического сообщества активное участие должны принимать представители работодателей;
- высокая динамика обновления знаний и требований рынка труда обуславливают достаточно частое обновление КМ, ФГОС ВПО содержит требование ежегодного обновления образовательных программ.

По своей сути задача разработки КМ относится к классу задач управления знаниями, эффективное решение которых в настоящее время невозможно без соответствующих средств информационной поддержки [3]. В связи с этим актуальным является создание информационной среды проектирования компетентностных моделей (ИС ПКМ) на основе перспективных технологий управления знаниями.

Анализ ИС ПКМ

Процесс разработки КМ организуется в соответствии с ее структурой. Согласно принятой в ФГОС ВПО структуризации КМ на верхнем уровне выделено две группы компетенций: общекультурные (ОК) и профессиональные (ПК). ОК являются надпрофессиональными, т. е. инвариантны видам профессиональной деятельности. ПК, напротив, непосредственно связаны с видами, задачами и объектами профессиональной деятельности, они характеризуют профессиональный портрет выпускника по конкретному направлению (профилю) подготовки. Группа ПК, в свою очередь, делится на подгруппы, соответствующие видам профессиональной деятельности выпускника. Например, для технических специальностей она может включать научно-исследовательские, проектно-конструкторские, монтажно-наладочные, эксплуатационные и другие группы компетенций.

Разработка КМ в общем случае выполняется несколькими группами экспертов, формируемыми в соответствии со структурой модели. Так, общекультурные и профессиональные компетенции, как правило, разрабатываются разными группами экспертов. Вместе с тем, близкие по содержанию подгруппы профессиональных компетенций (например, монтажно-наладочные и эксплуатацион-

ные) могут разрабатываться одной рабочей группой (РГ). Каждый эксперт может участвовать в разработке КМ для различных образовательных программ и быть членом любого числа РГ в рамках одной или нескольких КМ.

Анализ особенностей процесса проектирования КМ позволил сформулировать следующие основные требования к ИС ПКМ:

- поддержка работы групп экспертов, включающих представителей работодателей, в распределенном режиме;
- возможность накопления и повторного (многократного) использования знаний;
- использование форматов хранения данных, допускающих автоматизацию их обработки на семантическом уровне.

Требование поддержки распределенного режима работы обусловлено, в частности, необходимостью привлечения к разработке КМ, наряду с представителями академической общественности, представителей широкого круга организаций-работодателей. Как правило, это высококвалифицированные специалисты, испытывающие дефицит времени. Возможность включаться в работу в любое удобное время с любого сетевого компьютера позволит упростить организационные аспекты процесса коллективной разработки КМ, повысить эффективность работы экспертов и, как следствие, качество конечного результата.

Поддержка распределенного режима работы предполагает разработку и реализацию механизмов синхронизации версий КМ, формируемых отдельными пользователями, и согласования их мнений, включая возможность неформального обмена мнениями в процессе обсуждения формулировок конкретных компетенций.

Возможность накопления, структуризации и повторного использования знаний является важнейшей особенностью систем управления знаниями [3]. Процесс проектирования ИС ПКМ предполагает определение состава и структуры знаний, образующих информационное обеспечение системы, методы их структуризации, определение механизмов формирования и поэтапного накопления знаний в процессе разработки КМ.

Накопление банка компетенций создает предпосылки для последующей автоматизации решения ряда важных задач, возникающих в процессе проектирования и реализации образовательных программ (ОП). Примерами таких задач являются:

- группирование компетенций для формирования дисциплин и модулей учебного плана ОП;
- оптимизация учебных планов разных ОП, реализуемых вузом, в части выявления возможностей и степени их унификации на уровне отдельных модулей путем выявления и группирования сходных компетенций;

- уточнение перечня направлений и профилей подготовки с учетом заявленных работодателями компетенций;
- анализ соответствия КМ и ОП, реализуемых вузом, компетентностным моделям и образовательным программам зарубежных университетов в рамках Болонского процесса;
- анализ способов формирования различных типов компетенций и контроля уровня их освоения.

Эффективная автоматизация решения такого рода задач и поддержка интероперабельности ИС ПКМ с другими, основанными на знаниях, системами могут быть обеспечены за счет использования семантических форматов представления знаний, соответствующих открытым стандартам семантического Web [4].

Подход к построению ИС ПКМ

В соответствии с выявленными на этапе анализа требованиями при проектировании ИС ПКМ приняты следующие основные подходы.

Разработка КМ для определенной образовательной программы в рамках ИС ПКМ рассматривается как отдельный проект, доступ к которому (через авторизацию) получают члены сформированной и утвержденной рабочей группы. В рамках одного проекта в общем случае работает несколько рабочих групп, каждая из которых разрабатывает свою группу (или группы) компетенций.

Поддержка работы экспертов в распределенном режиме обеспечивается реализацией Web-доступа к ИС ПКМ. Это позволяет эксперту из любой организации, авторизованному в системе, принять участие в разработке КМ, используя только сетевой компьютер с браузером. Пользователь получает права просмотра и редактирования соответствующих групп компетенций в соответствии с членством и ролью в той или иной РГ.

Предполагается, что каждая РГ имеет координатора, обладающего дополнительными правами по сравнению с другими членами группы. В частности, ему предоставлены следующие возможности:

- формирование на основе авторских версий КМ, предложенных разными членами РГ, исходной интегральной версии для обсуждения;
- моделирование форума по обсуждению компетенций;
- формирование окончательной редакции соответствующей группы компетенций.

Обсуждение формулировок различных компетенций осуществляется посредством форума. Интеграция мнений членов РГ выполняется координатором РГ с учетом неформального обмена мнениями в форуме.



Рис. 1. Состав информационного обеспечения ИС ПКМ

В процессе коллективной разработки КМ в рамках ИС ПКМ возможны различные сценарии взаимодействия экспертов [5]. Базовый сценарий коллективной работы экспертов включает следующие этапы:

1. Координатор рабочей группы объявляет о начале разработки соответствующей группы компетенций с указанием срока подготовки членами РГ своих предложений (авторских версий КМ).

2. Члены рабочей группы к указанному сроку готовят свои предложения и сохраняют их на сервере, после чего они становятся доступны координатору РГ.

3. Координатор интегрирует поступившие от членов РГ предложения, создает объединенную исходную редакцию КМ и публикует ее для обсуждения членами группы. Он также определяет срок, в течение которого обсуждение должно быть завершено.

4. Предложенные формулировки компетенций обсуждаются членами РГ в форме свободной дискуссии. При этом могут комментироваться текущие и предлагаться новые формулировки компетенций. В качестве средства обсуждения используется форум.

5. По истечении установленного срока координатор с учетом высказанных членами рабочей группы мнений формирует итоговую редакцию соответствующей группы компетенций и присваивает ей статус окончательной.

6. Разработанные разными РГ группы компетенций интегрируются в общую КМ для соответствующего направления подготовки.

Для поддержки автоматизации обработки данных на семантическом уровне ИС ПКМ строится на основе технологий семантического Web и онтологического подхода [6]. В соответствии с этим подходом используемые системой знания о предметной области хранятся в форме явно заданных онтологий, что повышает гибкость системы и делает ее открытой.

Информационное обеспечение ИС ПКМ включает три группы компонентов (рис. 1):

- базовая онтология и компоненты представления (хранения) результирующих документов (компетентностных моделей);
- компоненты, поддерживающие организацию групповой работы экспертов;
- компоненты, обеспечивающие информационную поддержку собственно процесса проектирования КМ.

Фрагмент базовой онтологии компетентностных моделей представлен на рис. 2. Использование онтологической модели представления знаний позволяет специфицировать ряд важных ограничений на отношения между сущностями базовой онтологии. Так, совокупность возможных видов профессиональной деятельности ограничена выбранным направлением подготовки. Например, такие виды профессиональной деятельности, как проектно-конструкторская и проектно-технологическая, имеют место в сфере техники и технологий, но не характерны для медицины и гуманитарных направлений подготовки. Аналогично накладываются ограничения на отношения уровня образо-

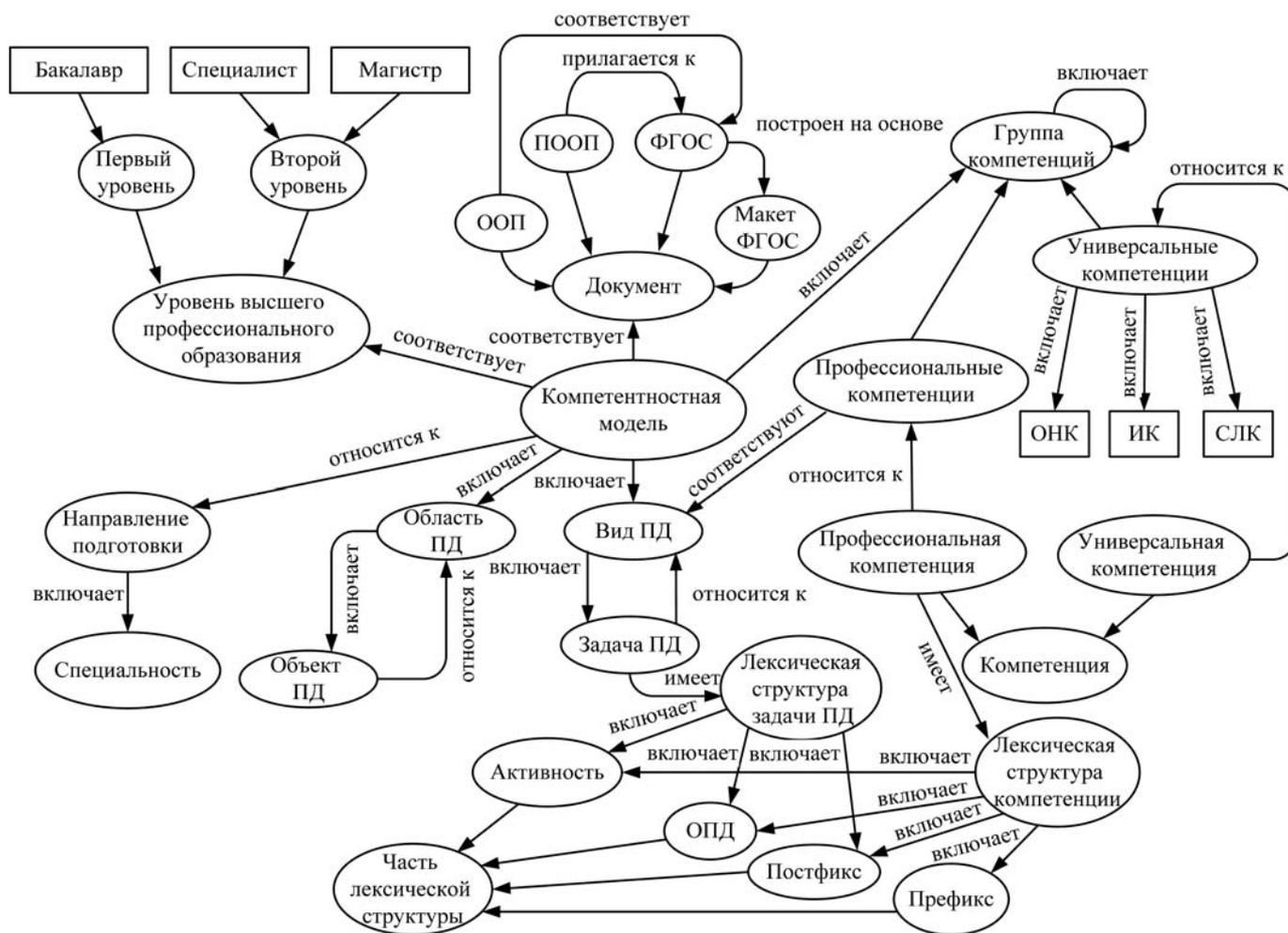


Рис. 2. Фрагмент базовой онтологии компетентностных моделей ИС ПКМ: ООП — основная образовательная программа; ПООП — примерная ООП; ФГОС — федеральный государственный образовательный стандарт; ОНК — общенаучные компетенции; ИК — инструментальные компетенции; СЛК — социально-личностные компетенции; ПД — профессиональная деятельность; ОПД — объект ПД

вания с задачами профессиональной деятельности и связанными с ними компетенциями. Например, научно-исследовательские компетенции, связанные с исследованием и разработкой новых концепций, моделей и т. п. характерны только для магистерского уровня, но не для уровня бакалавров. Явная фиксация такого рода ограничений в онтологии позволяет эффективно использовать ее в качестве компонента информационной среды, поддерживающей разработку компетентностных моделей.

Формирование профессиональных и общекультурных компетенций имеет свои особенности. В ИС ПКМ поддерживаются два способа задания профессиональных компетенций:

- свободный ввод в виде произвольного текста;
- формирование на основе лексического шаблона с использованием структурированных формализованных знаний — управляемых словарей и онтологий.

Лексический шаблон включает четыре компонента и имеет следующую структуру:

<Вводное_слово> <Тип_активности>
<Объект_ПД> [<Расширение>],

где ПД — профессиональная деятельность.

В качестве вводного слова компетенции используются такие термины, как "способность", "готовность", тип активности выражается терминами "анализировать", "проектировать", "разрабатывать" и т. п. Эти термины могут выбираться из соответствующих управляемых словарей, являющихся элементом информационного обеспечения ИС ПКМ.

Объекты профессиональной деятельности могут выбираться из онтологии предметных областей. В последние годы различные сообщества специалистов активно ведут разработку соответствующих формальных онтологий. Значительные успехи достигнуты, в частности, в разработке онтологий

био- и нанотехнологий. В области программной инженерии известна онтология SWEBOOK [7]. В ближайшие годы следует ожидать быстрого роста числа совместно используемых (доступных в сети Интернет) онтологий.

ИС ПКМ поддерживает реестр онтологий с возможностью навигации по нему, выбора и загрузки нужных онтологий с учетом их атрибутов (источник, автор(ы), дата обновления и др.).

Расширение является необязательным элементом шаблона, позволяющим указать, например, инструментарий, с использованием которого должна решаться соответствующая задача. Ниже приведен пример, в котором элементы компетенции, соответствующие элементам шаблона, разделены символом "|":

"Способность | проектировать | информационные процессы и системы | с использованием инновационных инструментальных средств"

Отметим, что лексический шаблон не ограничивает разработчика в формулировках компетенции, позволяя произвольно редактировать сформированный на основе шаблона вариант в соответствующем поле ввода. Вместе с тем, привязка терминов, используемых при формулировке компетенций, к онтологиям и управляемым словарям создает возможность последующей автоматизации обработки накапливаемых компетенций на семантическом уровне. В частности, могут выявляться схожесть компетенций на основе формально определенной синонимии объектов профессиональной деятельности в разных онтологиях, отношений "класс—подкласс", "часть—целое" и т. п.

Вместе с тем, компетенции, сформированные как произвольный текст, могут в дальнейшем использоваться для построения онтологий предметных областей, для которых они в настоящий момент отсутствуют.

При формировании общекультурных компетенций информационная поддержка реализована в виде банка информационных источников, структурированного по следующим атрибутам:

- тип документа (нормативный документ, методические материалы рекомендательного характера, научная публикация);
- авторы и год публикации источника;
- подгруппы компетенций.

Действующий макет ФГОС ВПО не предусматривает деления общекультурных (универсальных) компетенций на подгруппы. Однако в литературе рассмотрены различные подходы к структуризации данной группы компетенций. Согласно одному из наиболее известных, выделено три подгруппы: общенаучные (ОНК), инструментальные (ИК) и социально-личностные (СЛК). Такое деление за-

фиксировано в базовой онтологии ИС ПКМ (рис. 2) и используется при классификации информационных источников.

Архитектура ИС ПКМ

Архитектура ИС ПКМ, разработанной с использованием описанных выше подходов, представлена на рис. 3.

Модуль администрирования поддерживает выполнение через Web-интерфейс функций администрирования ИС ПКМ, в частности:

- создание проекта по разработке КМ для некоторой образовательной программы;
- создание рабочих (экспертных) групп по разработке различных групп компетенций и отдельных пользователей;
- регистрация пользователей и назначение им групп и ролей.

Модуль поддержки функций координатора РГ доступен только пользователям ИС ПКМ, выполняющим роль координатора соответствующей экспертной РГ. Данный модуль поддерживает функции интеграции авторских версий экспертов на этапе формирования исходной общей версии разрабатываемой группы компетенций, редактирования предложенных компетенций, управления форумом в части обсуждения соответствующей группы компетенций, завершения работы над группой компетенций (перевод документа в статус окончательной версии).

Модуль редактирования области и объектов ПД позволяет при необходимости редактировать соответствующие разделы документа. При проектировании образовательных программ эти разделы, как правило, полностью берутся из ФГОС ВПО, однако в ряде случаев может потребоваться его редактирование, в частности, при использовании ИС ПКМ для разработки образовательных стандартов. В процессе редактирования данного раздела могут также использоваться онтологии предметных областей.

Модуль редактирования видов и задач ПД поддерживает функции редактирования видов и задач ПД. При редактировании могут использоваться словари видов и задач ПД и онтологии предметных областей.

Модуль редактирования ПФ-компетенций поддерживает редактирование профессиональных компетенций двумя описанными выше способами: с использованием формализованных знаний и произвольного текста.

Модуль редактирования ОК компетенций обеспечивает создание и редактирование общекультурных компетенций с возможностью информационной поддержки посредством доступа к структуриро-

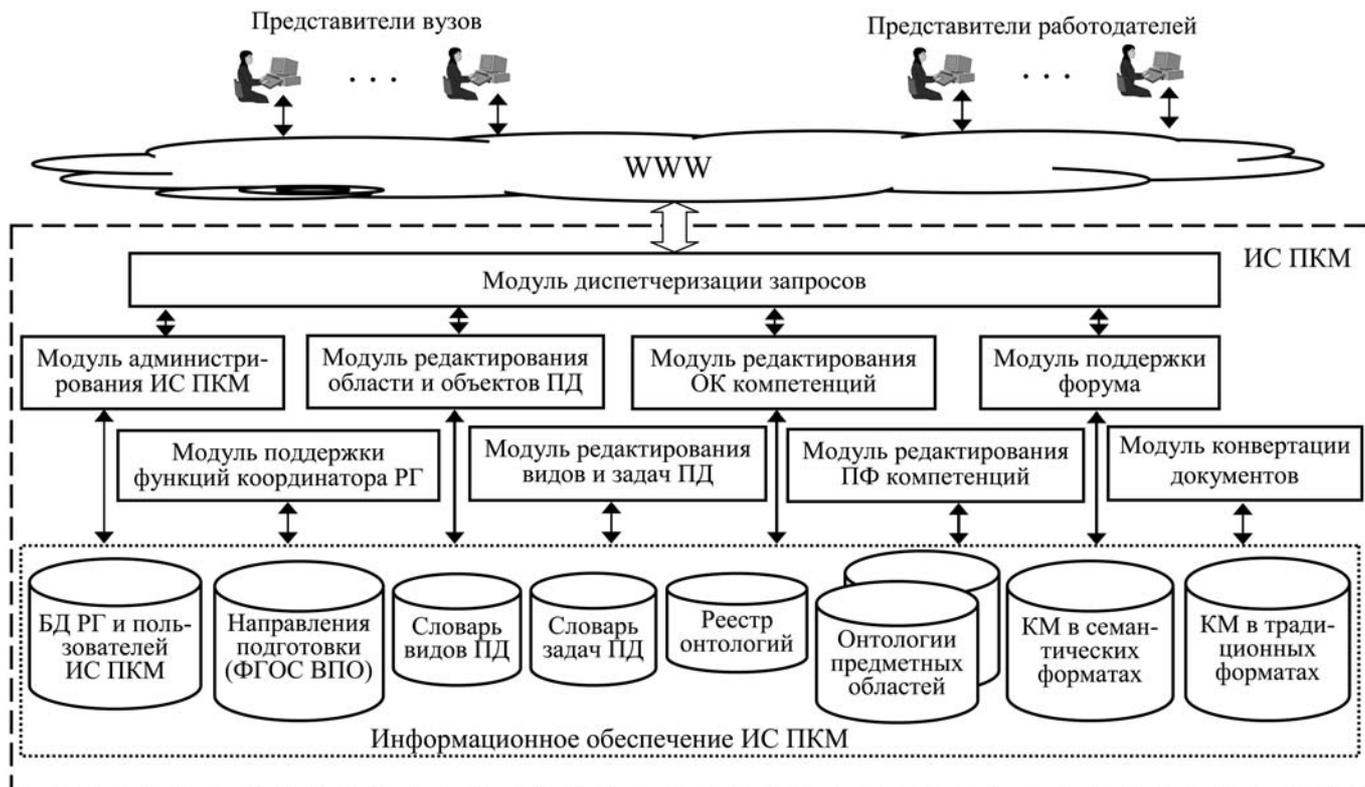


Рис. 3 Архитектура ИС ПКМ

ванному хранилищу информационных источников по данной тематике.

Модуль поддержки форума предоставляет пользователям среду неформального обсуждения формулировок компетенции, предлагаемых различными экспертами. Члены РГ могут, используя форум, комментировать вынесенные на обсуждение компетенции и предлагать свою редакцию. Возможность редактирования общей версии КМ по результатам обсуждения имеет только координатор соответствующей РГ.

Разработанные в ИС ПКМ компетентностные модели хранятся в формате семантического Web (язык RDF). Для преобразования документов в форматы традиционных офисных редакторов (например, MS Word) используется *модуль конвертации документов*.

Заключение

В настоящее время ИС ПКМ развернута на сервере СПбГЭТУ "ЛЭТИ" (<http://competence.eltech.ru/>) и проходит апробацию в процессе разработки компетентностных моделей в рамках проектирования новых образовательных программ в соответствии с ФГОС ВПО.

Дальнейшее развитие ИС ПКМ планируется в направлении реализации механизмов анкетирования экспертов, поддержки проектирования об-

разовательных программ на основе компетентностных моделей, наращивания информационного обеспечения за счет онтологий предметных областей и создания аналитических сервисов для работы с банком компетенций.

Список литературы

1. Байденко В. И. Компетентностный подход к проектированию государственных образовательных стандартов высшего профессионального образования (методологические и методические вопросы): Метод. пособ. М.: Исслед. центр проблем качества подготовки специалистов. 2005. 114 с.
2. Байденко В. И. Выявление состава компетенции выпускников вузов как необходимый этап проектирования ГОС ВПО нового поколения: Метод. пособ. М.: Исслед. центр проблем качества подготовки специалистов, 2006. 54 с.
3. Maier R. Knowledge Management Systems: Information and Communication Technologies for Knowledge Management. 3rd Edition Berlin: Springer, 2007. 720 p.
4. Davies J., Fensel D., Harmelen F. Towards the Semantic Web: Ontology-driven Knowledge Management. John Wiley & Sons, 2005. 328 p.
5. Пузанков Д. В., Пантелеев М. Г., Вехорев М. Н., Лебедев С. В. Сценарии разработки компетентностных моделей группами экспертов в единой информационной среде // Изв. СПбГЭТУ "ЛЭТИ". 2010. № 2. С. 44–51.
6. Пантелеев М. Г., Вехорев М. Н., Лебедев С. В. Онтологический подход к разработке компетентностных моделей // Изв. СПбГЭТУ "ЛЭТИ". 2010. № 1. С. 63–70.
7. Abran A., Cuadrado-Gallego J., Garcia-Barriocanal E., Mendes O., Sanchez-Alonso S., Sicilia M. A. Engineering the ontology for the Swebok: Issues and techniques // In Calero C., Ruiz F. and Piattini M. (Eds.), Ontologies for software engineering and software technology. New York: Springer, 2006. P. 103–122.

ЖУРНАЛ В ЖУРНАЛЕ

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

№ 1

ЯНВАРЬ

2011

Главный редактор:

ГАЛУШКИН А. И.

Редакционная коллегия:

АВЕДЬЯН Э. Д.
БАЗИАН Б. Х.
БЕНЕВОЛЕНСКИЙ С. Б.
БОРИСОВ В. В.
ГОРБАЧЕНКО В. И.
ЖДАНОВ А. А.
ЗЕФИРОВ Н. С.
ЗОЗУЛЯ Ю. И.
КРИЖИЖАНОВСКИЙ Б. В.
КУДРЯВЦЕВ В. Б.
КУЛИК С. Д.
КУРАВСКИЙ Л. С.
РЕДЬКО В. Г.
РУДИНСКИЙ А. В.
СИМОРОВ С. Н.
ФЕДУЛОВ А. С.
ЧЕРВЯКОВ Н. И.

Иностранные члены редколлегии:

БОЯНОВ К.
ВЕЛИЧКОВСКИЙ Б. М.
ГРАБАРЧУК В.
РУТКОВСКИЙ Л.

Редакция:

БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

От главного редактора 56

Галушкин А. И.

Нейрокомпьютеры в решении задач обеспечения
информационной безопасности 58

Аведьян Э. Д., Пантюхин Д. В.

Алгоритм нелинейного преобразования аргументов
в нейронной сети СМАС. 64

Скрибцов П. В., Казанцев П. А., Долгополов А. В.

Виртуальные нейросетевые процессоры — метод
кроссплатформенного ускорения расчета нейросетевых
задач на современных многоядерных процессорах 73

От главного редактора

Журнал "Нейросетевые технологии" посвящается важному направлению современной науки — одному из разделов информационных технологий, связанному с применением в вычислительных системах такого понятия, как нейронные сети. Нейросетевые технологии включают в себя следующие разделы:

- теория нейронных сетей;
- нейроматематика;
- нейроуправление;
- нейрочипы.

Нейронная сеть — это сеть с конечным числом слоев из одиночных элементов — аналогов нейронов с различными типами связей между слоями нейронов. При этом число нейронов в слоях, число слоев и тип связей между ними выбираются необходимыми для обеспечения заданных качества и времени решения задачи.

Нейронная сеть представляет собой высокопараллельную динамическую систему с топологией направленного графа, которая может получать выходную информацию посредством реакции ее состояния на входные воздействия.

Необходимо отметить основные свойства нейронных сетей как логических структур:

- однородные нейронные сети обладают свойством постепенной деградации при выходе из строя отдельных элементов. Это было отмечено еще Ф. Розенблаттом при построении трехслойного перцептрона со случайными связями в первом слое, когда число элементов данного слоя берется избыточным. При этом функция, реализуемая нейронной сетью, как бы распределяется по структуре;
- структура однородной нейронной сети предоставляет возможность массового параллельного синхронного выполнения большого числа операций, состоящих, в свою очередь, из простейших операций сложения, умножения и нелинейного безынерционного преобразования. В структуре нейронной сети отсутствуют сложные и "длинные" иррациональные операции над операндами (деления, извлечения корня и т. д.), характерные для алгоритмов, свойственных однопроцессорным машинам;
- нейронные сети реализуют достаточно гибкое и сложное функциональное преобразование входного пространства состояний в выходное, причем гибкость этого преобразования при необходимости может регулироваться числом слоев нейронной сети и типом связей;
- структура нейронных сетей допускает аналитическое описание преобразования входного пространства в выходное. При наличии этого

свойства структура нейронной сети позволяет также аналитически получать алгоритм настройки нейронных сетей и контролировать его функционирование в процессе решения задачи;

- сложность нейронной сети, используемой для решения конкретной задачи, отображает в себе сложность решаемой задачи;
- в будущем, с использованием аппарата линейных последовательных машин Гилла, структура однородных нейронных сетей позволит подойти к решению проблемы аналитического описания, а следовательно, и синтеза алгоритмов адаптации в многослойных нейронных сетях.

Основные преимущества нейронных сетей как логического базиса алгоритмов решения сложных задач:

- инвариантность методов синтеза нейронной сети к размерности пространства признаков и их размеров;
- соответствие современным и перспективным технологиям;
- отказоустойчивость в смысле монотонного, а не катастрофического изменения качества решения задачи в зависимости от числа вышедших из строя элементов.

Нейроматематика — это раздел вычислительной математики, связанный с решением задач с помощью алгоритмов, представимых в нейросетевом логическом базисе. Цель нейроматематики — создание алгоритмов с высокой степенью параллельности как для формализуемых, так и для трудноформализуемых и неформализуемых задач.

Нейронным (или нейросетевым) алгоритмом будем называть вычислительную процедуру, основная часть которой может быть реализована в виде нейронной сети той или иной структуры.

Нейроуправление — раздел современной теории управления, связанный с применением нейронных сетей для построения так называемых нейроконтроллеров в сложных системах управления. Именно необходимость управления сложными существенно нелинейными многомерными объектами сделало необходимым введение нейроконтроллеров в контур управления.

Нейрокомпьютер — это вычислительная система с архитектурой аппаратного и программного обеспечения, адекватной выполнению алгоритмов, представленных в нейросетевом логическом базисе. Нейронные сети по сути дела являются формальным аппаратом описания основной части алгоритма решения задачи на нейрокомпьютерах.

Принципиальным моментом появления нейрокомпьютеров является отказ от булевского логического базиса на уровне элементов вычислительных машин с переходом на базис пороговой логики, который в простейшем варианте моделирует функ-

ции нервной клетки. При этом изменяется не только элементная база, но и архитектура ЭВМ.

Основой построения многих систем являются специализированные нейрокомпьютеры, построенные на базе нейрочипов — специальных сверхбольших интегральных схем (СБИС), архитектура которых эффективно реализует фрагмент формальной нейронной сети. В настоящее время разработана большая номенклатура нейрочипов:

- цифровые;
- аналогово-цифровые и аналоговые;
- оптоэлектронные и другие.

Разработки трех направлений — алгоритмов решения задач, теории нейронных сетей и нейрокомпьютеров находятся в тесной взаимосвязи:

— с одной стороны, структура нейронной сети для каждой задачи определяется самой задачей; с другой стороны, развитие теории нейронных сетей побуждает применять для решения задач более сложные структуры нейронных сетей;

— с одной стороны, уровень совершенства технических устройств определяет диапазон возможностей для построения нейронных сетей и нейросетевых алгоритмов; с другой стороны, развитие нейроматематики подталкивает теорию сетей, а та в свою очередь направляет развитие технического исполнения.

Сфера применения нейросетевых технологий настолько широка, что наверняка приведенный ниже перечень можно значительно расширить.

1. Вычислительные и информационные системы:

- нейронная память, нейрочипы памяти;
- нейросетевые алгоритмы распределения заданий в многопроцессорных вычислительных системах;
- интеллектуальные сенсоры с применением нейросетевых технологий;
- нейросетевые алгоритмы слияния информации от многих датчиков;
- нейросетевые алгоритмы анализа надежности программного обеспечения;
- интеллектуальные источники питания на базе нейросетевых технологий;
- нейросетевые технологии обработки текстовой информации; поиск информации в сети Интернет; нейроагенты;
- нейроэкспертные системы;
- нейросетевые алгоритмы защиты в системах информации от несанкционированного доступа, включая обнаружение вирусов, атак, и криптографические системы;
- извлечение знаний;
- биометрические системы, в том числе:
 - дактилоскопические системы;
 - выделение и распознавание лиц в видеоконтенте и идентификация личности по форме руки, рисунку радужной оболочки глаза, по подписи, манере работы с клавиатурой;
 - идентификация жестов и позы человека.

2. Нейрокомпьютеры в обработке изображений:

- сегментация изображений;
- скелетонизация изображений;
- выделение движущихся объектов;
- распознавание изображений;
- распознавание линий и углов на изображении;
- распознавание края изображения;
- обеспечение инвариантности к группе преобразований при обработке изображений;
- нейросетевые алгоритмы обработки текстур;
- нейросетевые алгоритмы обработки 3D- и стереоизображений;
- нейрокомпьютеры в виртуальной реальности;
- нейросетевые алгоритмы в базах данных изображений;
- распознавание номеров автомобилей и вагонов;
- распознавание иероглифов;
- распознавание рукописного текста;
- нейросетевые алгоритмы обработки сканируемых изображений;
- обработка аэрокосмических изображений.

3. Нейрокомпьютеры в системах обработки сигналов:

- нейросетевые алгоритмы фильтрации сигналов;
- нейросетевые эквалайзеры;
- нейросетевые алгоритмы кодирования сигналов;
- нейросетевые алгоритмы обработки речевой информации, в том числе: распознавание отдельных слов и слогов, идентификация диктора, распознавание слитной речи, преобразование текста в речевой сигнал;
- нейросетевые алгоритмы обработки радиолокационной информации, в том числе: алгоритмы идентификации целей по характеристикам отраженного радиолокационного сигнала, алгоритмы сопровождения радиолокационных целей, нейросетевые алгоритмы в фазированных антенных решетках;
- нейросетевые алгоритмы обработки гидролокационной информации;
- нейрокомпьютеры в системах связи;
- нейрокомпьютеры в системах виброконтроля;
- нейросетевые алгоритмы обработки сейсмической информации;
- нейрокомпьютеры в музыке.

4. Нейроуправление динамическими системами:

- нейроуправление роботами;
- нейроуправление манипуляторами;
- нейросетевые алгоритмы решения прямой и обратной кинематических задач;
- зрение роботов;
- навигация роботов;
- нейроуправление подводным роботом;
- нейроуправление микророботами;
- нейросетевые системы активного виброуправления;
- нейроуправление сервосистемами;

- нейросетевые алгоритмы учета трения;
 - нейрорегулирование гидравлическими и пневматическими системами;
 - нейрорегулирование температурой;
 - нейрорегулирование движением на транспортных магистралях;
 - нейрорегулирование электроэрозионным станком.
5. Применение нейрокомпьютеров в машиностроении:
- нейрокомпьютеры в авиации;
 - нейрокомпьютеры в автомобильной промышленности;
 - гибкое производство и нейрокомпьютеры;
 - нейрокомпьютеры в горнорудной промышленности;
 - нейрокомпьютеры и диагностика машин и механизмов;
 - управление конструкциями с помощью нейрокомпьютеров;
 - нейрокомпьютеры в космических исследованиях;

- интеллектуальные здания;
- нейрокомпьютеры и исследования материалов;
- нейрокомпьютеры в химической промышленности;
- нефтегазодобывающая и перерабатывающая промышленность и нейрокомпьютеры;
- сварка и нейрокомпьютеры;
- нейрокомпьютеры в энергетике, в том числе прогнозирование нагрузки, нейрорегулирование в энергетике, диагностика энергетических объектов;
- атомные электростанции;
- нейрокомпьютеры в электронной промышленности.

6. Нейрокомпьютеры в медицине.

7. Нейросетевые технологии в исследованиях по программе "Протеом человека". Как отмечено выше, этот список может быть расширен.

Журнал ориентирован на студентов, преподавателей и научных работников в области супервычислений, параллельных алгоритмов.

УДК 004.032.26

А. И. Галушкин, д-р техн. наук, проф., нач. лаб.,
Международный центр по информатике
и электронике (ИнтерЭВМ)
e-mail: neurocomputer@yandex.ru

Нейрокомпьютеры в решении задач обеспечения информационной безопасности

Представлены основные направления применения нейросетевых технологий в системах обеспечения информационной безопасности, в частности при решении следующих задач: обработка текстовых сообщений; обнаружение несанкционированного доступа; обнаружение вирусов; биометрическая идентификация; оценка информационной безопасности.

Ключевые слова: информационная безопасность, нейросетевые технологии, атаки, вирусы, биометрия, криптография

1. О выборе задач обеспечения информационной безопасности для решения в нейросетевом логическом базисе

Задачи обеспечения информационной безопасности можно разделить на две группы: адекватные нейросетевому логическому базису и "общие". Для "общих" задач большой размерности время их ре-

шения на фон-неймановских или транспьютероподобных машинах может превысить допустимое. При достижении необходимого времени решения может превышать стоимость вычислительной системы, решающей задачи. Именно в этих случаях возникает необходимость разработки нейросетевых алгоритмов и нейросетевой аппаратной поддержки их решения (рис. 1, см. третью сторону обложки).

2. Об отечественных работах в области теории нейронных сетей в поддержку разработок нейросетевых алгоритмов решения задач обеспечения информационной безопасности

На рис. 2 представлена общая структура нейронной сети, используемой для решения самых различных математических и прикладных задач. Наряду с несколькими сотнями монографий зарубежных авторов в области теории нейронных сетей, опубликованных за последние 15–20 лет, можно отметить отечественные монографии [1–4],

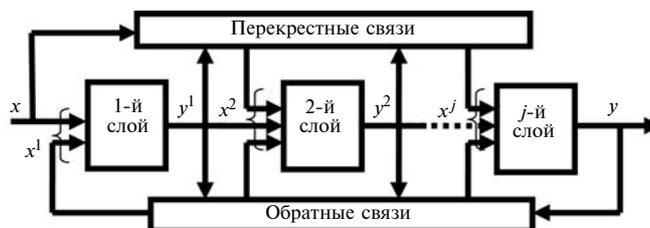


Рис. 2. Общая структура многослойной нейронной сети

опубликованные в серии "Нейрокомпьютеры и их применение". В целом необходимо отметить три базовых режима работы нейронных сетей, используемых в задачах обеспечения информационной безопасности:

- обучение (квалификация учителя полная);
- самообучение (квалификация учителя нулевая);
- обучение с учителем, имеющим конечную квалификацию, в том числе имеющим отрицательную квалификацию (режим вредительства).

3. Нейроматематика — новый раздел вычислительной математики, связанный с разработкой нейросетевых алгоритмов решения задач

Основные результаты работ отечественных авторов в области нейроматематики представлены в работах [5—7]. В работе [5] представлена также библиография зарубежных работ, опубликованных с 1988 по 2001 год по нейросетевым алгоритмам решения математических задач (более 5000 наименований), систематизированная по видам математических задач следующим образом.

1. Общие работы по нейроматематике.
2. Арифметические и логические операции на нейронных сетях.
3. Системы линейных алгебраических уравнений.
4. Матричные операции.
5. Системы нелинейных алгебраических уравнений.
6. Системы линейных неравенств.
7. Вычислительная геометрия.
8. Нахождение минимума и максимума функций.
9. Аппроксимация функций.
10. Экстраполяция функций.
11. Задачи оптимизации.
 - 11.1. Общий раздел.
 - 11.2. Линейное программирование.
 - 11.3. Квадратичное программирование.
 - 11.4. Нелинейное программирование.
 - 11.5. Комбинаторика.
 - 11.6. Задача коммивояжера.
 - 11.7. Задачи составления расписаний.
 - 11.8. Динамическое программирование.
12. Теория графов.
 - 12.1. Общие работы.
 - 12.2. Нахождение путей в графах.
 - 12.3. Циклы в графах.
 - 12.4. Разбиение графов.
 - 12.5. Клики графов.
 - 12.6. Укладка графов.
 - 12.7. Раскладка графов.
 - 12.8. Сети Петри.
13. Генерация случайных чисел.
14. Задачи в конечных полях и кольцах.
15. Доказательство теорем.

16. Сортировка.
17. Дифференциальные игры.
18. Обыкновенные дифференциальные уравнения.
19. Интегральные уравнения.
20. Дифференциальные уравнения в частных производных.
 - 20.1. Общий раздел.
 - 20.2. Генерация решеток.
 - 20.3. Задачи электродинамики.
 - 20.4. Задачи гидродинамики.
 - 20.5. Задачи аэродинамики.
 - 20.6. Уравнение Шредингера.
21. Хаос, аттракторы и нейронные сети.

Нейросетевые парадигмы решения данных задач принципиально отличаются от классических алгоритмов, ориентированных на однопроцессорные и кластерные ЭВМ, и фактически составляют новый раздел вычислительной математики.

4. Прикладные направления применения нейрокомпьютеров, имеющие косвенное отношение к решению задач обеспечения информационной безопасности

Необходимо отметить, что некоторые прикладные области активного применения нейрокомпьютеров имеют косвенное, а иногда и прямое отношение к разработке систем обеспечения информационной безопасности. Эти прикладные области освещаются во многих сотнях зарубежных монографий, а также в некоторых отечественных, отмеченных ниже. В работе [8] представлены нейросетевые алгоритмы решения следующих задач:

- фильтрация изображений;
 - инвариантная обработка;
 - обработка 3D-изображений;
 - космические системы;
 - распознавание номерных знаков;
 - нейрофизиологические исследования систем технического зрения,
- а в работе [9] решение таких задач
- фильтрация сигналов;
 - экстраполяция функций;
 - обработка речи.

Работа [10] является результатом исследований российских ученых в следующих областях:

- космические системы;
- медицина (включая диагностику туберкулезных заболеваний);
- нейроимплантаты;
- клеточные нейронные сети для обработки изображений;
- клеточные нейрончики.

Название работы [11] говорит о ее важности для решения отдельных частных задач обеспечения информационной безопасности.

5. Нейрокомпьютеры — перспективное направление развития сверхвысокопроизводительной вычислительной техники, в том числе для решения сложных задач обеспечения информационной безопасности

На рис. 3 представлена качественная картина сравнения нейрокомпьютеров и вычислительных систем с фон-неймановской архитектурой в координатах производительность—стоимость. Эта тен-

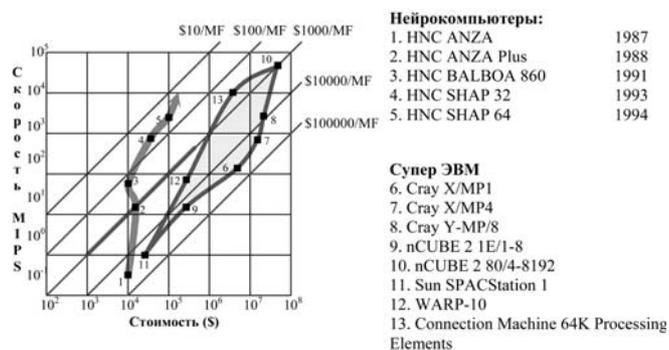


Рис. 3. Сравнение суперкомпьютеров по критерию "производительность—стоимость"

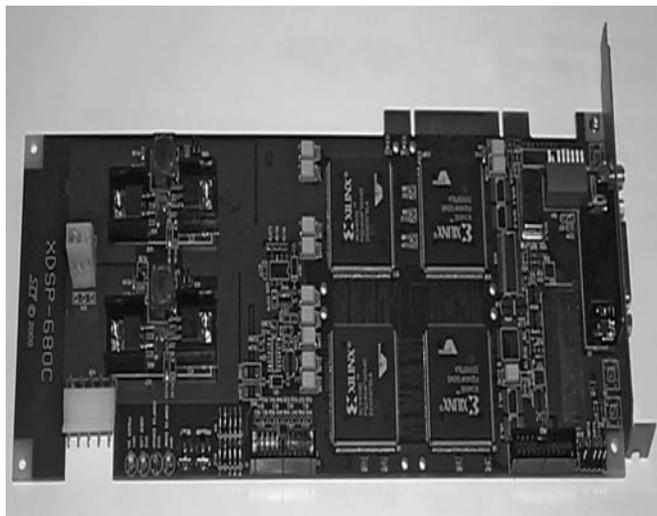


Рис. 5. Нейроплата EZB627 PCI на базе нейрочипов ZISC фирмы IBM

денция сохранилась и на современном уровне развития технологии микроэлектроники и вычислительной техники. В отечественных монографиях [12—15] представлены материалы по отечественным и зарубежным разработкам нейрокомпьютеров, в том числе отечественной разработки нейрокомпьютера на базе нейрочипов с использованием ПЛИС большой интеграции (рис. 4, см. третью сторону обложки). На рис. 5 представлено фото современной нейроплаты зарубежного производства на базе заказных нейрочипов ZISC (Zero Instruction Set Computer) фирмы IBM. На рис. 6 представлена интеллектуальная видеокамера с ZISC нейрочипом для решения задач обработки изображений, в частности для решения задачи идентификации личности по изображениям лица в целях обеспечения информационной безопасности.

6. Прикладные задачи обеспечения информационной безопасности для решения в нейросетевом логическом базисе

Анализ показывает, что наиболее эффективными являются следующие прикладные задачи обеспечения информационной безопасности с применением нейросетевых технологий:

- обработка текстовых сообщений;
- обнаружение несанкционированного доступа;
- обнаружение вирусов;
- биометрические системы;
- нейросетевая криптография;
- радиочастотная идентификация;
- оценка информационной безопасности;

Работы по обеспечению информационной безопасности с применением нейросетевых технологий активизируются во всем мире. Например, корпорация Lockheed Martin выделила лаборатории CyLab (www.cylab.cmu.edu) университета Карнеги — Меллона грант на разработку новых технологий киберзащиты. В лабораториях при финансовой поддержке Пентагона ведутся десятки

проектов по информационной безопасности: создание самоизлечивающихся цифровых систем, функционирующих по принципу биологических объектов; автоматическое преобразование текстов программ для повышения их безопасности; построение теоретических моделей программной уязвимости и т. д.

В двухтомной монографии [16] содержится систематизированное описание разработок и применения нейрокомпьютеров в Китае, в том числе описаны следующие

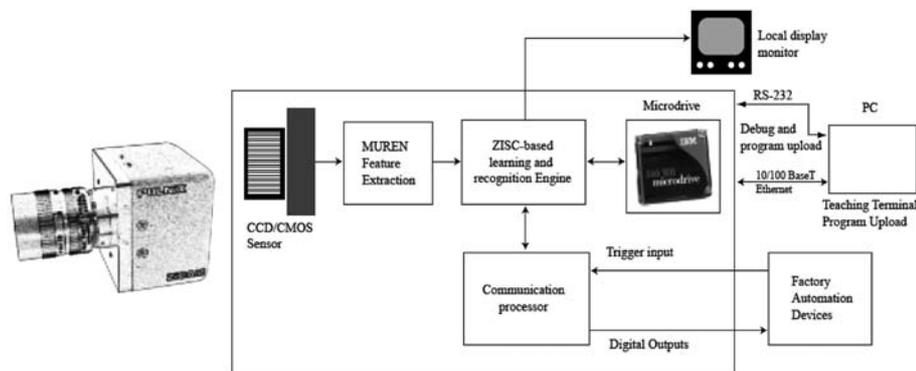


Рис. 6. ZiCAM Zi640 (фирма Pulnix, США) — интеллектуальная видеокамера с ZISC нейрочипом

направления применения нейросетевых технологий:

- обработка текстов;
- нейросетевые алгоритмы защиты в системах информации от несанкционированного доступа;
- биометрические системы:
 - дактилоскопические системы;
 - распознавание лиц по портрету;
 - идентификация личности по форме ладони;
 - идентификация личности по рисунку радужной оболочки глаза;
 - идентификация личности по подписи;
 - идентификация личности по манере работы с клавиатурой.

6.1. Обработка текстовых сообщений в интересах обеспечения информационной безопасности

Обработка текстовых сообщений использует нейрокомпьютеры для решения следующих задач:

1. Кластеризация текстовых сообщений.
2. Анализ количества и качества кластеров.

3. Обнаружение новых кластеров.

Эти режимы используются при объективном наличии в естественном языке различного рода неоднозначностей при морфологическом, синтаксическом и семантическом анализе текстовых сообщений.

Алгоритмы кластеризации на нейронных сетях являются в настоящее время недостаточно изученными, несмотря на более чем 30-летнюю историю их развития.

6.2. Обнаружение несанкционированного доступа

Обнаружение несанкционированного доступа в вычислительные системы — важное направление внедрения нейросетевых технологий. Основными задачами здесь являются:

- создание нейросетевых систем обнаружения несанкционированного доступа;
- создание единой базы данных по компьютерным атакам и преступлениям в целях последующего выявления скрытых закономерностей.

На рис. 7 представлена обобщенная архитектура системы обнаружения атак. Основной модуль, в котором целесообразно применение нейросетевых технологий, использует различные методы выявления атак (рис. 8). Для указанных на данном рисунке методов выявления аномалий на рис. 9, 10 представлено место нейронных сетей в указанных методах выявления атак.



Рис. 7. Обобщенная архитектура системы обнаружения атак

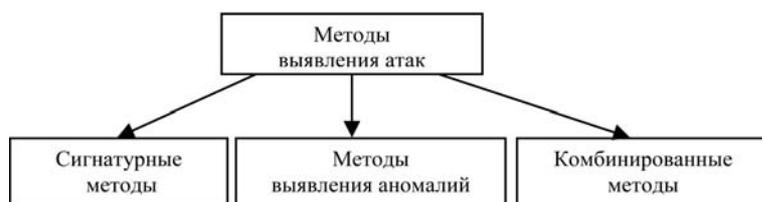


Рис. 8. Классификация методов выявления атак



Рис. 9 Место нейронных сетей в сигнатурных методах выявления атак



Рис. 10. Место нейронных сетей в методах выявления аномалий

Ниже представлены зарубежные фирмы — разработчики нейросетевых алгоритмов систем обнаружения несанкционированного доступа:

- IBM (система FAMS);
- HNC Software Financial Solution (система Falcon);
- Nestor (семейство систем PRISM);
- Проект Neural Networks Intrusion Detection Project (NNID);
- ACTS ACO95, ASPECT (Advanced Security for Personal Communications Technologies);
- IASM (Управления космических и морских систем SPAWAR);
- NEWS (фирма Logicon);
- IDES, NIDES, MIDAS, ISOA, NSM, NADIR, DIDS, GRIDS (центральное управление — Китай);
- Snort (Бразилия).

6.3. Обнаружение вирусов

Обнаружение вирусов — важное направление применения нейрокомпьютеров в решении задач обеспечения информационной безопасности. Основным направлением работ здесь является построение обучаемых систем распознавания и классификации вирусов в антивирусных программах и системах. Так в работе [21] представлена разработанная интеллектуальная антивирусная система для уничтожения вирусов в загрузочном секторе ПЭВМ, основанная на классификации с помощью нейронных сетей. Описана методика ее обучения.

6.4. Нейросетевые технологии в биометрических системах идентификации личности как эффективные средства обеспечения информационной безопасности

Ниже представлен перечень биометрических признаков, используемых при решении различных задач:

- создание биометрического паспорта;
- антитеррористические системы;
- другие системы.

Биометрические признаки классифицируются следующим образом:

Статические признаки

1. Портрет, полученный фотографическим путем или с помощью видеокамер (одной или нескольких).

2. Рисунок сосудов глазного дна.

3. Радужная оболочка глаза.

4. Геометрия руки.

5. Дактилоскопия.

6. Термографическое наблюдение лицевых артерий.

7. Идентификация по венам руки.

Динамические признаки

8. Рукописная подпись и динамика ее произнесения.

9. Клавиатурный почерк.

10. Особенности голоса.

11. Походка, жесты.

Экспериментальные статические признаки

12. Индивидуальные особенности генетического кода личности.

13. Запах тела.

14. Следы пота.

15. Строение уха.

16. Отпечаток ладони.

На рис. 11 (см. третью сторону обложки) представлена классификация биометрических систем в пространстве "стоимость—качество" [22].

6.5. Нейросетевая криптография

Ниже представлены основные направления применения нейросетевых технологий в криптографии:

- генерация секретных ключей;
- системы с открытыми ключами;
- системы цифровой подписи;
- генераторы случайных чисел;
- визуальная криптография;
- генерация случайных текстов.

Для решения данных задач, как правило, применяются специфические нейросетевые структуры. Ниже представлены специфические архитектуры и режимы работы нейронных сетей для решения задач нейрокриптографии:

- нейросетевые сети с обратными связями (рекуррентные нейронные сети);
- хаотические нейронные сети;
- хаотические клеточные нейронные сети для визуальной криптографии;
- нейронные сети в режиме вредительства.

Наряду с зарубежными работами в этой области необходимо отметить некоторые отечественные работы Ставропольского государственного университета (руководитель проф. Червяков Н. И.):

- нейронные сети в системах криптографической защиты информации (Нейрокомпьютер. 2001. № 10);
- нейросетевой генератор криптографических ключей пороговой схемы разделения секрета (Нейрокомпьютер. 2004. № 10);
- нейросетевой блок локализации ошибок криптографического нейропроцессора (Нейрокомпьютер. 2004. № 10).

6.6. Радиочастотная идентификация

Радиочастотная идентификация в системах обеспечения информационной безопасности — важное направление применения нейросетевых технологий — объективная. Необходимость применения нейросетевых технологий в задачах радиоидентификации возникает в случае наличия многих идентификаторов, движущихся относительно друг друга в сложной шумовой обстановке. При этом

нейросетевые алгоритмы работают в режиме так называемого слепого разделения сигналов (*blind separation*), адекватного режиму обучения без учителя (самообучения) нейронных сетей.

6.7. Оценка информационной безопасности

В настоящее время в данной области активно развиваются следующие перспективные направления применения нейросетевых технологий:

- оценка информационной безопасности регионов;
- оценка информационной безопасности вычислительных сетей;
- оценка готовности к информационным войнам;
- выработка стратегий информационной защиты.

При этом выходными сигналами нейросетевой экспертной системы оценки и прогноза информационной безопасности сетей и регионов являются следующие:

- вероятность сохранения конфиденциальности информации;
- вероятность сохранения информации от вирусных искажений;
- вероятность сохранения информации от случайных искажений.

Необходимо отметить дополнительные задачи и вопросы, на которые нужно ответить при создании современных нейросетевых экспертных систем оценки информационной безопасности:

1. Какие характеристики функционирования системы необходимо использовать для обучения?
2. Время обучения и темп подачи информации.
3. Место нейросетевых методов в общей архитектуре системы активного аудита.
4. Вычислительная сложность и эффективная вычислимость.
5. Иерархия нейросетевых методов аудита.
6. Ошибки первого и второго рода. Как оценить качество аудита?
7. Можно ли реализовать выбор решений в зависимости от происшедших событий?
8. Интеграция нейросетевых методов с другими методами аудита.
9. Сравнительный анализ методов с точки зрения выбранных критериев эффективности.
10. Редко происходящие события.
11. Параметры настройки нейросети, их интерпретация на проблемном уровне.
12. Самообучение метода. Что необходимо делать для адаптивности механизмов аудита?

Выводы

1. Обеспечение информационной безопасности — одна из многих задач, в которых развиваются высокие технологии и ставятся сложные задачи.

2. Нейросетевые технологии необходимы для решения именно сложных задач, недоступных классическим методам решения и классическим ЭВМ.

3. Смежными по отношению к нейросетевым технологиям являются новые технологии киберзащиты, а именно:

- самоизлечивающиеся цифровые системы (по принципу функционирования биологических систем);
- автоматическое преобразование текстов программ для повышения их безопасности;
- теоретические модели программной уязвимости.

4. Необходимо отметить также другие смежные научные направления:

- Нейрочипы в многоагентных системах.
- Хаос в нейронных сетях.
- Квантовые нейрокомпьютеры, использование которых вероятно будет эффективным для решения задач обеспечения информационной безопасности.

Список литературы

1. **Галушкин А. И.** Теория нейронных сетей. Кн. 1. (Серия "Нейрокомпьютеры и их применение") М.: ИПРЖР, 2000.
2. **Головко В. А.** Нейронные сети: обучение, организация и применение. Кн. 4. (Серия "Нейрокомпьютеры и их применение"). М.: ИПРЖР, 2001.
3. **Нейронные** сети: история развития теории / под науч. ред. Я. З. Цыпкина, А. И. Галушкина. Кн. 5. (Серия "Нейрокомпьютеры и их применение") М.: ИПРЖР, 2001.
4. **Зозуля Ю. И.** Интеллектуальные нейросистемы. Кн. 12. (Серия "Нейрокомпьютеры и их применение") М.: Радиотехника, 2003.
5. **Нейроматематика** / под науч. ред. А. И. Галушкина. Кн. 6. (Серия "Нейрокомпьютеры и их применение") М.: ИПРЖР, 2002.
6. **Горбаченко В. И.** Нейросетевые алгоритмы решения краевых задач теории поля. Кн. 10. (Серия "Нейрокомпьютеры и их применение") М.: Радиотехника, 2003.
7. **Галушкин А. И.** Нейроматематика (проблемы развития) // Нейрокомпьютер. 2003. № 1.
8. **Нейрокомпьютеры** в системах обработки изображений (под науч. ред. Ю. В. Бутеева и А. И. Галушкина). Кн. 7. (Серия "Нейрокомпьютеры и их применение"). М.: Радиотехника, 2003.
9. **Нейрокомпьютеры** в системах обработки сигналов / под науч. ред. Ю. В. Бутеева и А. И. Галушкина, Кн. 9. (Серия "Нейрокомпьютеры и их применение") М.: 2003.
10. **Нейрокомпьютеры** в решении прикладных задач обработки изображений / Под науч. ред. А. Н. Балухто и А. И. Галушкина. (Серия "Нейрокомпьютеры и их применение"). М.: "Радиотехника". В печати.
11. **Харламов А. А.** Нейрокомпьютеры в обработке речевой и текстовой информации. (Серия "Нейрокомпьютеры и их применение"). М.: Радиотехника. В печати.
12. **Галушкин А. И.** Нейрокомпьютеры. Кн. 3. (Серия "Нейрокомпьютеры и их применение"). М.: ИПРЖР, 2000.
13. **Нейрокомпьютеры** в системах обработки классов. Кн. 15. (Серия "Нейрокомпьютеры и их применение"). М.: Радиотехника, 2003.
14. **Кирсанов Э. Ю.** Нейрокомпьютеры с параллельной архитектурой. Кн. 16. (Серия "Нейрокомпьютеры и их применение"). М.: Радиотехника. 2003.

15. **Рамбиди Н. Г.** Бимолекулярные нейросетевые устройства. Кн. 33. (Серия "Нейрокомпьютеры и их применение". М.: ИПРЖР, 2000.

16. **Кеннеди Дж.** Нейросетевые технологии в диагностике аномальной сетевой активности (cannadyj@scis.nova.edu): Пер. А. В. Лукацкого, Ю. Ю. Цаплева и В. П. Сахарова.

17. **Корнеев В. В., Сажин С. В.** Система контроля за функционированием компьютеров и компьютерных сетей на основе применения нейронных сетей // Сборник трудов Всероссийской конференции "Нейрокомпьютеры и их применение". М.: 1999.

18. **Райан Д., Менг-Джанг Лин, Миккулайнен Р.** Обнаружение атак с помощью нейросетей // Компьютерная хроника. 2000. № 7.

19. **Корнеев В. В., Масолович А. В., Савельева Е. В., Шапшаев А. Е.** Распознавание программных модулей и обнаруже-

ние несанкционированных действий с применением аппарата нейронных сетей // Информационные технологии. 1997. № 10. С. 12—15.

20. **Design of a neural network for recognition and classification of computer viruses / A. Doulmas, K. Mavroudakakis, D. Gritzails, S. Katsikas (Dept. of Math., Aegean Univ., Samos, Greece).**

21. **Присяжнюк С. П., Гриняев С. Н.** Нейросетевая антивирусная система для поиска и уничтожения загрузочных вирусов // 3 Всерос. семинар "Нейроинформатика и ее приложение". Красноярск. 6—8 окт., 1995: Тез. докл. — Красноярск, 1995. С. 21—22.

22. **Иванов А. Н.** Нейросетевые алгоритмы биометрической идентификации личности. Кн. 15. (Серия "Нейрокомпьютеры и их применение") М.: Радиотехника, 2003.

УДК 004.3:518.5

Э. Д. Аведьян, д-р техн. наук, зам. нач. лаб.,

Центр информационных технологий
и систем органов исполнительной власти
(ЦИТиС), г. Москва,
e-mail: avedian@mail.ru

Д. В. Пантюхин, инж.,
Международный центр по информатике
и электронике (ИнтерЭВМ), г. Москва

Алгоритм нелинейного преобразования аргументов в нейронной сети СМАС¹

Приводится краткое описание нейронной сети СМАС, предназначенной для запоминания функций многих переменных. Отмечается, что одним из важнейших элементов этой сети является блок, в котором происходит нелинейное преобразование вектора аргументов функции в вектор номеров ячеек ассоциативной памяти. Дается вывод аналитической формы этого алгоритма, его программная реализация и некоторые результаты компьютерного моделирования, которые могут быть использованы для тестирования алгоритма.

Ключевые слова: нейронная сеть, СМАС, алгоритм, номера ячеек, память, программа реализации

Введение

Спектр применения нейронной сети СМАС (*Cerebellar Model Articulation Controller* — мозжечковая модель суставного регулятора) очень широк. Это и задачи управления динамическими процессами [1—8], задачи моделирования нелинейных систем [9], системы управления роботами-мани-

пуляторами [10—13], системы классификации [14—16], системы раннего предупреждения в банковской сфере [17], система позиционирования в nanoиндустрии [18], задачи управления беспилотным воздушным средством [19], демпфирования колебаний строительного крана [20]. Кроме названных выше приложений, можно указать также задачи подавления и управления вибрацией, построения цифровой модели местности, распознавании образов, задачи управления движением и многие другие. К сожалению, цитированные выше работы — работы зарубежных авторов.

В нашей стране эта парадигма нейронных сетей мало известна. Причин тому несколько. Работы автора нейронной сети СМАС [21, 22] труднодоступны. В нашей стране было мало публикаций по нейронной сети СМАС. Кроме того, в опубликованных относительно давно статьях, связанных с подробным описанием и исследованием СМАС, в журнале "Информационные технологии" [23, 24], и в опубликованной недавно работе, в журнале "Интеллектуальные информационные системы" [25], алгоритм вычисления номеров ячеек ассоциативной памяти, который является одним из важнейших элементов нейронной сети СМАС, приводился в словесной форме, а аналитическое выражение этого алгоритма приводилось без доказательства в силу его громоздкости. Возможно, что отсутствие доказательства также оказалось одним из факторов, не способствовавших широкому распространению нейронной сети СМАС в нашей стране.

Устранению этого пробела служит настоящая работа. В первой части, следуя в основном работе [25], приводится краткое описание нейронной сети СМАС, вводятся обозначения, необходимые для последующего анализа. Вторая часть посвящена выводу аналитической формы этого алгоритма, его программной реализации и некоторым результатам компьютерного моделирования.

¹ Работа выполнена при поддержке ФАНИ, государственный контракт 02.514.12.4003 от 11 июня 2009 г.

1. Структура нейронной сети СМАС

Как и любая нейронная сеть, СМАС предназначена для запоминания и восстановления функций $y_c(v)$ от N -переменных v . Такими функциями могут быть модели нелинейных статических или динамических объектов, модели регуляторов систем управления, разделяющие поверхности в задачах распознавания образов и другие. Наиболее существенным отличием СМАС от других нейронных сетей является следующее.

1. Аргументы запоминаемой и воспроизводимой функции принимают только дискретные значения.

2. Нелинейное преобразование аргументов функции происходит неявно с помощью алгоритма вычисления номеров ячеек ассоциативной памяти, в которых хранятся числа, определяющие значение функции.

Дискретный характер аргументов функции, с одной стороны, придает сети очень важное свойство, которое заключается в том, что каждое новое предъявленное для обучения значение функции либо улучшает свойство сети, либо оставляет его на прежнем уровне. С другой стороны, дискретный характер аргументов функции СМАС приводит к тому, что для решения аналоговых задач, а таких задач большинство, приходится вводить аналого-цифровое преобразование входных сигналов — аргументов функции СМАС, добавляя в систему вынужденные ошибки квантования.

Предположим, что функция N -переменных $y_c(v)$ определена на N -мерном гиперпараллелепипеде $V = \{v^{(1)} = \overline{v_{\min}^{(1)}, v_{\max}^{(1)}}; v^{(2)} = \overline{v_{\min}^{(2)}, v_{\max}^{(2)}}; \dots; v^{(N)} = \overline{v_{\min}^{(N)}, v_{\max}^{(N)}}\}$. Для того чтобы СМАС запомнил функцию $y_c(v)$, сначала задается число уровней квантования по каждой переменной $x_{\max}^{(i)} = \overline{1, N}$, затем каждое ребро гиперпараллелепипеда квантуется с постоянным шагом

$$\Delta^{(i)} = \frac{v_{\max}^{(i)} - v_{\min}^{(i)}}{x_{\max}^{(i)}}, \quad i = \overline{1, N},$$

и каждому элементу квантования каждой компоненты присваиваются целочисленные номера $x^{(i)} = 1, 2, \dots, x_{\max}^{(i)}$, $i = \overline{1, N}$. Эти номера связаны со значениями аргументов $v^{(i)}$, $i = \overline{1, N}$, функции $y_c(v)$, задаваемых в середине каждого интервала, следующим соотношением:

$$v^{(i)} = v_{\min}^{(i)} + \Delta^{(i)}(x^{(i)} - 0,5), \quad x^{(i)} = 1, 2, \dots, x_{\max}^{(i)}, \\ i = \overline{1, N}.$$

В результате масштабирования, смещения и дискретизации переменных $v^{(i)}$, $i = \overline{1, N}$, вместо функции $y_c(v)$ рассматривается функция $y(x)$ от N переменных $x = (x^{(1)}, x^{(2)}, \dots, x^{(N)})^T$, которая определена на целочисленной N -мерной сетке:

$$X = \{x^{(1)} = \overline{1, x_{\max}^{(1)}}; x^{(2)} = \overline{1, x_{\max}^{(2)}}; \dots; x^{(N)} = \overline{1, x_{\max}^{(N)}}\},$$

и которая, естественно, совпадает с исходной функцией в точках дискретизации.

2. Алгоритм нелинейного преобразования аргументов или алгоритм вычисления номеров ячеек ассоциативной памяти

При построении алгоритма вычисления номеров ячеек ассоциативной памяти и алгоритма обучения СМАС ее автор [21, 22] исходил из нейрофизиологического подхода, согласно которому каждый входной сигнал возбуждает определенную область мозжечка, суммарная энергия которой соответствует значению запоминаемой функции. В нейронной сети СМАС предполагается, что каждый входной сигнал (аргумент функции) возбуждает или делает активными ровно ρ^* ячеек памяти, суммарное содержимое которых равно значению запоминаемой функции. Следовательно, даже в случае скалярного аргумента $x = x^{(1)}$ ему соответствует ρ^* -мерный вектор номеров активных ячеек памяти. Параметр ρ^* играет очень важную роль. В литературе он известен как обобщающий параметр (*generalization parameter*). Его значение определяет разрешающую способность, требуемый объем памяти, скорость обучения СМАС и ряд других важных параметров этой парадигмы нейронных сетей.

Одномерный случай. Рассмотрим сначала структуру алгоритма вычисления номеров ячеек ассоциативной памяти для одномерного случая, $N = 1$. Каждому значению скалярного аргумента $x^{(1)} = \overline{1, x_{\max}^{(1)}}$ соответствует ровно ρ^* активных ячеек памяти с номерами

$$x^{(1)}, x^{(1)} + 1, x^{(1)} + 2, \dots, x^{(1)} + \rho^* - 1, \quad (1)$$

так что максимальному значению аргумента $x^{(1)} = x_{\max}^{(1)}$ соответствуют ячейки памяти с номерами $x_{\max}^{(1)}, x_{\max}^{(1)} + 1, x_{\max}^{(1)} + 2, \dots, x_{\max}^{(1)} + \rho^* - 1$. Отсюда следует, что число ячеек памяти, необходимых для хранения функции одной переменной в СМАС, равно

$$M^{(1)} = x_{\max}^{(1)} + \rho^* - 1, \quad (2)$$

что на $\rho^* - 1$ больше числа значений функции, запоминаемых нейронной сетью. Таким образом, в одномерном случае нейронная сеть СМАС, с точки зрения требуемого объема памяти, неэффективна, однако ее эффективность даже в этом случае проявляется в том, что она оказывается способной достаточно точно запомнить гладкую функцию по небольшому числу данных n обучающей последовательности, когда $n \ll M^{(1)}$ [22].

Номера активных ячеек памяти, которые соответствуют значению скалярной переменной $x^{(1)} = \overline{1, x_{\max}^{(1)}}$, можно представить ρ^* -мерным вектором-столбцом $m^{(1)}$, значения компонент которого равны номерам активных ячеек. Так, например, если $\rho^* = 4$, а значение аргумента $x^{(1)} = 2$, то значения компонент вектора-столбца $m^{(1)}$ номеров активных ячеек в соответствии с (1) равны 2, 3, 4 и 5. Последовательность значений компонент $m_k^{(1)} = \overline{1, \rho^*}$ вектора $m^{(1)}$ в одномерном случае кажется несущественной, поскольку их значения указывают на номера ячеек, сумма содержимого которых равна значению запоминаемой функции, а оно, естественно, не зависит от того, в какой последовательности выбираются данные из ячеек. Поэтому вектор $m^{(1)}$ в рассматриваемом случае мог иметь вид $m^{(1)} = (2, 3, 4, 5)^T$ или, например, вид

$$m^{(1)} = (5, 3, 2, 4)^T. \quad (3)$$

В многомерном случае, $N > 1$, структура этой последовательности приобретает принципиальное значение. Она предложена автором СМАС [21, 22] исходя из требования, чтобы близко расположенным друг к другу входным векторам соответствовало бы большое число общих номеров активных ячеек памяти, и, соответственно, далеко расположенным друг от друга входным векторам соответствовало бы меньшее число общих номеров активных ячеек памяти, либо их вовсе не было бы. Это требование придает СМАС хорошие интерполяционные свойства: если близко расположенные входные векторы имеют много общих активных ячеек памяти, то и значения функции в этих точках в некотором смысле должны быть близки друг к другу в силу того, что значение функции в СМАС определяется как сумма содержимого активных ячеек памяти.

Алгоритм вычисления номеров активных ячеек памяти для одномерного случая, $N = 1$, имеет следующий вид.

1. Значения ρ^* компонент $m_k^{(1)}$, $k = \overline{1, \rho^*}$, вектора $m^{(1)}$ вычисляются по схеме (1).

2. Последовательность компонент вектора $m_k^{(1)}$, $k = \overline{1, \rho^*}$, строится по следующему правилу: первым элементом этого вектора является то число из последовательности элементов $m_k^{(1)}$, остаток от деления которого на ρ^* равен 1, на втором месте — число, остаток от деления которого на ρ^* равен 2, и т. д. и, наконец, на последнем месте — число, остаток от деления которого на ρ^* равен 0.

В соответствии с этим правилом вектор $m^{(1)}$ (3) должен иметь следующую структуру:

$$m^{(1)} = (5, 2, 3, 4)^T.$$

Для того чтобы задать выражение для вычисления значений компонент вектора $m^{(1)}$ введем функцию

$$a \overline{\text{mod}} b = \begin{cases} b, & a \text{ mod } b = 0; \\ a \text{ mod } b, & a \text{ mod } b \neq 0, \end{cases} \quad (4)$$

где a, b — целые числа, а функция $a \text{ mod } b$ — остаток от деления a на b . Тогда значения всех ρ^* компонент $m_k^{(1)}$, $k = \overline{1, \rho^*}$, вектора $m^{(1)}$ вычисляются согласно следующему выражению:

$$m_{[(x^{(1)} \overline{\text{mod}} \rho^* + i) \overline{\text{mod}} \rho^*]}^{(1)} = x^{(1)} + i, \quad i = \overline{0, \rho^* - 1}. \quad (5)$$

Для вычисления компонент вектора $m^{(1)}$ согласно выражению (5) достаточно определить номер той компоненты, значение которой равно $x^{(1)}$, т. е. вычислить (5) при $i = 0$. В качестве поясняющего примера в табл. 1 приведены выражения для вектора $m^{(1)}$ размерности $\rho^* = 8$ для ряда значений аргумента $x^{(1)} = 22, 23, 24, 25, 26, 27, 28, 29, 30$.

Таблица 1
Структура номеров активных ячеек памяти СМАС, одномерный случай, $\rho^* = 8$

Значение аргумента $x^{(1)}$	Вектор номеров активных ячеек памяти	Примечание
$x^{(1)} = 22$	$m^{(1)} = (25, 26, 27, 28, 29, 22, 23, 24)^T$	$22 \overline{\text{mod}} 8 = 6$
$x^{(1)} = 23$	$m^{(1)} = (25, 26, 27, 28, 29, 30, 23, 24)^T$	$23 \overline{\text{mod}} 8 = 7$
$x^{(1)} = 24$	$m^{(1)} = (25, 26, 27, 28, 29, 30, 31, 24)^T$	$24 \overline{\text{mod}} 8 = 8$
$x^{(1)} = 25$	$m^{(1)} = (25, 26, 27, 28, 29, 30, 31, 32)^T$	$25 \overline{\text{mod}} 8 = 1$
$x^{(1)} = 26$	$m^{(1)} = (33, 26, 27, 28, 29, 30, 31, 32)^T$	$26 \overline{\text{mod}} 8 = 2$
$x^{(1)} = 27$	$m^{(1)} = (33, 34, 27, 28, 29, 30, 31, 32)^T$	$27 \overline{\text{mod}} 8 = 3$
$x^{(1)} = 28$	$m^{(1)} = (33, 34, 35, 28, 29, 30, 31, 24)^T$	$28 \overline{\text{mod}} 8 = 4$
$x^{(1)} = 29$	$m^{(1)} = (33, 34, 35, 36, 29, 30, 31, 32)^T$	$29 \overline{\text{mod}} 8 = 5$
$x^{(1)} = 30$	$m^{(1)} = (33, 34, 35, 36, 37, 30, 31, 32)^T$	$30 \overline{\text{mod}} 8 = 6$

Из табл. 1 следует, что если разница между двумя различными значениями аргумента $x^{(1)}$ минимальна и равна единице, то таким значениям аргумента соответствуют векторы номеров активных ячеек памяти, у которых совпадают 7 из 8 активных ячеек, если эта разница равна 2, то совпадают 6 активных ячеек и т. д. Если же эта разность больше 7, то общих ячеек у векторов $m^{(1)}$ нет (сравните адреса для $x^{(1)} = 22$ и $x^{(1)} = 30$).

В общем случае, если для двух значений аргумента $x^{(1)}$, равных соответственно $x^{(1)} = c$ и $x^{(1)} = d$, при произвольном целом ρ^* выполняется условие

$$|c - d| = \rho < \rho^*,$$

то число общих ячеек памяти, соответствующих этим значениям аргумента, равно

$$\Delta\rho = \rho^* - \rho,$$

т. е. чем меньше расстояние между двумя значениями аргумента, тем больше общих ячеек памяти, соответствующих этим значениям аргумента.

Многомерный случай. В многомерном случае каждой компоненте $x^{(i)}$, $i = \overline{1, N}$, вектора x так же, как и в одномерном случае, соответствует ρ^* -мерный вектор $m^{(i)}$, $i = \overline{1, N}$, номеров активных ячеек памяти, которые вычисляются, как это было описано выше в одномерном случае, согласно выражениям (4), (5). В результате этих вычислений N -мерному вектору x ставится в соответствие промежуточная матрица \mathbf{M} активных ячеек памяти с элементами $m_k^{(1)}$, $i = \overline{1, N}$, $k = \overline{1, \rho^*}$:

$$x = \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ \vdots \\ x^{(N)} \end{pmatrix} \rightarrow \mathbf{M} = \begin{pmatrix} m_1^{(1)} & m_2^{(1)} & m_3^{(1)} & \dots & m_{\rho^*}^{(1)} \\ m_1^{(2)} & m_2^{(2)} & m_3^{(2)} & \dots & m_{\rho^*}^{(2)} \\ m_1^{(3)} & m_2^{(3)} & m_3^{(3)} & \dots & m_{\rho^*}^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_1^{(N)} & m_2^{(N)} & m_3^{(N)} & \dots & m_{\rho^*}^{(N)} \end{pmatrix}. \quad (6)$$

На следующем шаге матрице \mathbf{M} размера $N \times \rho^*$ ставится в соответствие ρ^* -мерный вектор активных ячеек памяти вектора x . Для этого выполняется последовательное слияние элементов каждого k -го столбца матрицы \mathbf{M} в один k -й элемент вектора номеров активных ячеек памяти \mathbf{m} , так что этот ρ^* -мерный вектор приобретает следующий вид:

$$\mathbf{m} = (m_1^{(1)}, m_1^{(2)}, m_1^{(3)}, \dots, m_1^{(N)}; m_2^{(1)}, m_2^{(2)}, m_2^{(3)}, \dots, m_2^{(N)}; \dots; m_{\rho^*}^{(1)}, m_{\rho^*}^{(2)}, m_{\rho^*}^{(3)}, \dots, m_{\rho^*}^{(N)})^T, \quad (7)$$

каждый элемент которого

$$m_k = m_k^{(1)}, m_k^{(2)}, m_k^{(3)}, \dots, m_k^{(N)}, k = \overline{1, \rho^*}, \quad (8)$$

однозначно определяет номер активной ячейки памяти СМАС. Выражения (8) представляют собой символьную запись номеров активных ячеек памяти. К особенностям чисел, входящих в символическую запись числа m_k , $k = \overline{1, \rho^*}$ (8), относится тот факт, что остаток от деления каждого из них на ρ^* равен k в соответствии с формулой (5).

Алгоритм вычисления чисел m_k по значениям $m_k^{(i)}$, $i = \overline{1, N}$, принимает наиболее простую форму, когда максимально возможные значения компонент вектора x могут быть записаны в следующем виде:

$$x_{\max}^{(i)} = (\mu^{(i)} - 1)\rho^* + 1, i = \overline{1, N}, \quad (9)$$

где $\mu^{(i)}$, $i = \overline{1, N}$ — некоторые целые числа.

Заметим, что в этом случае число ячеек памяти, необходимых для хранения в СМАС функции i -й переменной согласно (2), оказывается равным

$$M^{(i)} = x_{\max}^{(i)} + \rho^* - 1 = (\mu^{(i)} - 1)\rho^* + 1 + \rho^* - 1 = \mu^{(i)}\rho^*, i = \overline{1, N}. \quad (10)$$

Напомним, что значения переменных $M^{(i)}$, $i = \overline{1, N}$, также определяют наибольшие значения, которые принимают переменные $m_k^{(i)}$, $i = \overline{1, N}$, $k = \overline{1, \rho^*}$.

Рассмотрим способ вычисления чисел $m_k = m_k^{(1)} m_k^{(2)} m_k^{(3)} \dots m_k^{(N)}$, $k = \overline{1, \rho^*}$, которые являются номерами активных ячеек памяти, на примере трех трехмерных входных векторов

$$\begin{aligned} \mathbf{x}[1] &= (1 \ 1 \ 1)^T, \\ \mathbf{x}[2] &= (2 \ 3 \ 6)^T, \\ \mathbf{x}[3] &= (5 \ 5 \ 9)^T, \end{aligned} \quad (11)$$

при следующих значениях параметров:

$$\mu^{(1)} = 2, \mu^{(2)} = 2, \mu^{(3)} = 3 \text{ и } \rho = 4. \quad (12)$$

В этом случае согласно (9) и (10) максимальные значения переменных $x^{(i)}$, $i = \overline{1, 3}$, и $M^{(i)}$, $i = \overline{1, 3}$, равны:

$$x_{\max}^{(1)} = 5, x_{\max}^{(2)} = 4, x_{\max}^{(3)} = 9, \quad (13)$$

$$M^{(1)} = 8, M^{(2)} = 8, M^{(3)} = 12. \quad (14)$$

Согласно формулам (7) и (8) входные векторы $\mathbf{x}[1]$, $\mathbf{x}[2]$, $\mathbf{x}[3]$ связаны с символическими номе-

рами векторов активных ячеек $\mathbf{m}[1]$, $\mathbf{m}[2]$, $\mathbf{m}[3]$ следующей зависимостью:

$$\mathbf{x}[1] = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \rightarrow \mathbf{M}[1] = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{bmatrix} \rightarrow \mathbf{m}[1] = \begin{bmatrix} 1, 1, 1 \\ 2, 2, 2 \\ 3, 3, 3 \\ 4, 4, 4 \end{bmatrix};$$

$$\mathbf{x}[2] = \begin{bmatrix} 1 \\ 3 \\ 6 \end{bmatrix} \rightarrow \mathbf{M}[2] = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 3 & 4 \\ 9 & 6 & 7 & 8 \end{bmatrix} \rightarrow \mathbf{m}[2] = \begin{bmatrix} 1, 5, 9 \\ 2, 6, 6 \\ 3, 3, 7 \\ 4, 4, 8 \end{bmatrix};$$

$$\mathbf{x}[3] = \begin{bmatrix} 5 \\ 5 \\ 9 \end{bmatrix} \rightarrow \mathbf{M}[3] = \begin{bmatrix} 5 & 6 & 7 & 8 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} \rightarrow \mathbf{m}[3] = \begin{bmatrix} 5, 5, 9 \\ 6, 6, 10 \\ 7, 7, 11 \\ 8, 8, 12 \end{bmatrix}.$$

Из предыдущих соотношений следует, что трехмерному вектору $\mathbf{x}[1]$ соответствует четырехмерный вектор активных ячеек, имеющих символические номера 1, 1, 1; 2, 2, 2; 3, 3, 3 и 4, 4, 4. Соответственно, вектор $\mathbf{x}[2]$ активизирует ячейки памяти с символическими номерами 1, 5, 9; 2, 6, 6; 3, 3, 7 и 4, 4, 8, а вектор $\mathbf{x}[3]$ активизирует ячейки памяти с символическими номерами 5, 5, 9; 6, 6, 10; 7, 7, 11 и 8, 8, 12. Здесь приведено 12 символических номеров активных ячеек; конечно, таких ячеек много больше хотя бы потому, что значения первых двух компонент символических номеров могут принимать значения от 1 до 8, а третьей — от 1 до 12 согласно (14). Еще раз обратим внимание на тот факт, что в любой символический номер входят только те числа, у которых остатки от деления каждого из них на ρ^* равны.

Пользоваться символическими значениями номеров активных ячеек памяти сложно. Возникает вопрос нахождения алгоритма, с помощью которого можно было бы поставить в соответствие символическим номерам активных ячеек их десятичные значения. С этой целью упорядочим символические номера ячеек по следующему правилу.

На первом месте расположим символический номер 1, 1, 1, которому будет соответствовать десятичный номер 1. Будем трактовать выражения для символической записи номеров активных ячеек как специальную позиционную запись числа $m_k = m_k^{(1)} m_k^{(2)} m_k^{(3)} \dots m_k^{(N)}$, $k = \overline{1, \rho^*}$, старший разряд которого равен $m_k^{(1)}$, а младший разряд — $m_k^{(N)}$. Поэтому следующим вторым символическим номером будет 1, 1, 5, за которым следует третий номер 1, 1, 9. Поскольку возможные значения третьего разряда в рассматриваемом примере равны 1, 5 и 9

(при условии, что значение первого разряда при делении на ρ^* равно 1), то следующим символическим номером будет 1, 5, 1, т. е. в последовательности символических номеров быстрее всего изменяются значения в младших разрядах, и медленнее — в старших. Предыдущие рассуждения позволяют выписать последовательность всех символических номеров, у которых значение первого разряда равно единице (табл. 2).

Поскольку все значения второго и третьего разряда исчерпаны, то следующие 6 символических номеров активных ячеек будут иметь вид, приведенный в табл. 3.

Последняя часть последовательности символических номеров, для которых старший разряд имеет наибольшее значение, равное $M^{(1)} = 8$, приведена в табл. 4, причем десятичные значения символических номеров активных ячеек основаны на

Таблица 2

Начало последовательности символических номеров активных ячеек
 $\mu^{(1)} = 2, \mu^{(2)} = 2, \mu^{(3)} = 3$ и $\rho = 4$

Десятичные номера активных ячеек, $m^{(d)}$	Символические номера активных ячеек, m_1
1	1, 1, 1
2	1, 1, 5
3	1, 1, 9
4	1, 5, 1
5	1, 5, 5
6	1, 5, 9

Таблица 3

Продолжение последовательности символических номеров активных ячеек
 $\mu^{(1)} = 2, \mu^{(2)} = 2, \mu^{(3)} = 3$ и $\rho = 4$

Десятичные номера активных ячеек, $m^{(d)}$	Символические номера активных ячеек, m_2
7	2, 2, 2
8	2, 2, 6
9	2, 2, 10
10	2, 6, 2
11	2, 6, 6
12	2, 6, 10

Таблица 4

Окончание последовательности символических номеров активных ячеек
 $\mu^{(1)} = 2, \mu^{(2)} = 2, \mu^{(3)} = 3$ и $\rho = 4$

Десятичные номера активных ячеек, $m^{(d)}$	Символические номера активных ячеек, m_4
43	8, 4, 4
44	8, 4, 8
45	8, 4, 12
46	8, 8, 4
47	8, 8, 8
48	8, 8, 12

Таблица 5

Промежуточная последовательность символических номеров активных ячеек, значение старшего разряда равно 6, $\mu^{(1)} = 2, \mu^{(2)} = 2, \mu^{(3)} = 3$ и $\rho = 4$

Десятичные номера активных ячеек, $m^{(d)}$	Символические номера активных ячеек, m
	6, 2, 2 6, 2, 6 6, 2, 10
	6, 6, 2 6, 6, 6 6, 6, 10

том простом условии, что общее число активных ячеек равно $6 \times 8 = 48$, где 6 — число активных ячеек в каждой таблице, а 8 — число таблиц.

Определим теперь десятичный номер, например символического номера 6, 6, 6. Этот номер должен быть вписан в выделенную серым цветом ячейку левого столбца табл. 5.

Десятичный номер каждого символического номера активной ячейки определим по следующему правилу: этот номер равен числу номеров в предшествующих таблицах, в данном случае $(6 - 1)6$, плюс число, равное номеру в текущей таблице — *текущему номеру*, в данном случае 5, т. е. десятичный номер символического номера 6, 6, 6 равен $30 + 5 = 35$.

Нетрудно подсчитать, что число десятичных номеров, входящих в каждую таблицу, в силу структуры данных, образующих таблицу, равно $\mu^{(2)}\mu^{(3)} = 6$, и, следовательно, число номеров в предшествующих таблицах, в общем случае при $N = 3$, равно

$$(m_k^{(1)} - 1)\mu^{(2)}\mu^{(3)}, k = \overline{1, \rho^*}. \quad (15)$$

Для определения в общем случае текущего номера обратим внимание на структуру табл. 2—5. Каждая из этих таблиц также состоит из блоков, разделенных линией. В данном случае блоков два, поскольку $\mu^{(2)} = 2$. Число элементов, входящих в каждый блок, равно $\mu^{(3)} = 3$. Для того чтобы вычислить текущий номер, например, номер 5 в любой из приведенных таблиц, надо вычислить число блоков, предшествующих блоку, в котором находится текущий номер, и умножить на число элементов в блоке, равное $\mu^{(3)} = 3$. Общая формула для вычисления числа предшествующих блоков имеет вид $\text{int}[(m_k^{(2)} - 1)/\rho^*]$, $k = \overline{1, \rho^*}$, где $\text{int}[a]$ — целая часть числа a , и, следовательно, число элементов в предшествующих блоках равно

$$\text{int}[(m_k^{(2)} - 1)/\rho^*]\mu^{(3)}, k = \overline{1, \rho^*}. \quad (16)$$

Операция $\text{int}[a]$ взятия целой части числа a обеспечивает правильное вычисление числа предшествующих блоков в (16) независимо от значения k .

В данном случае число предшествующих блоков равно $\text{int}[(6 - 1)/4] = 1$, и поэтому число элементов, предшествующих блоку, в котором находится текущий номер, равно $1 \times \mu^{(3)} = 3$.

Для того чтобы окончательно определить десятичный номер текущего символического номера, необходимо определить десятичный номер текущего символического номера в блоке, в котором он находится, и полученное число сложить с числом номеров (16) в предшествующих блоках текущей таблицы и с числом номеров (15) в предшествующих таблицах. Нетрудно видеть, что десятичный номер в текущем блоке равен

$$\text{Int}[(m_k^{(3)} - 1)/\rho^*] + 1, k = \overline{1, \rho^*}, \quad (17)$$

и, следовательно, равен $\text{int}[(6 - 1)/4] + 1 = 2$. Заметим, что первое слагаемое в выражении (17) соответствует числу десятичных номеров в блоке, предшествующих текущему символическому номеру.

Суммируя (15)—(17), получим алгоритм вычисления десятичных номеров $m^{(d)}$ при $N = 3$ по значениям соответствующих символических номеров:

$$m^{(d)} = (m_k^{(1)} - 1)\mu^{(2)}\mu^{(3)} + \text{int}[(m_k^{(2)} - 1)/\rho^*]\mu^3 + \text{int}[(m_k^{(3)} - 1)/\rho^*] + 1, k = \overline{1, \rho^*}. \quad (18)$$

Формула (18) позволяет вычислить десятичные номера для любых значений символических номеров при $N = 3$, в том числе и для отсутствующих в табл. 5 десятичных номеров:

$$m_2 = 6, 2, 2 \rightarrow m^{(d)} = (6 - 1) \cdot 2 \cdot 3 + \text{int}[(2 - 1)/4] \cdot 3 + \text{int}[(2 - 1)/4] + 1 = 30 + 0 + 0 + 1 = 31;$$

$$m_2 = 6, 2, 6 \rightarrow m^{(d)} = (6 - 1) \cdot 2 \cdot 3 + \text{int}[(2 - 1)/4] \cdot 3 + \text{int}[(6 - 1)/4] + 1 = 30 + 0 + 1 + 1 = 32;$$

$$m_2 = 6, 2, 10 \rightarrow m^{(d)} = (6 - 1) \cdot 2 \cdot 3 + \text{int}[(2 - 1)/4] \cdot 3 + \text{int}[(10 - 1)/4] + 1 = 30 + 0 + 2 + 1 = 33;$$

$$m_2 = 6, 6, 2 \rightarrow m^{(d)} = (6 - 1) \cdot 2 \cdot 3 + \text{int}[(6 - 1)/4] \cdot 3 + \text{int}[(2 - 1)/4] + 1 = 30 + 1 \cdot 3 + 0 + 1 = 34;$$

$$m_2 = 6, 6, 6 \rightarrow m^{(d)} = (6 - 1) \cdot 2 \cdot 3 + \text{int}[(6 - 1)/4] \cdot 3 + \text{int}[(6 - 1)/4] + 1 = 30 + 1 \cdot 3 + 1 + 1 = 35;$$

$$m_2 = 6, 6, 10 \rightarrow m^{(d)} = (6 - 1) \cdot 2 \cdot 3 + \text{int}[(6 - 1)/4] \cdot 3 + \text{int}[(10 - 1)/4] + 1 = 30 + 1 \cdot 3 + 2 + 1 = 36.$$

Найдем теперь алгоритм вычисления десятичных номеров при $N = 4$. Для иллюстрации вывода будем пользоваться теми же данными, что были приняты при $N = 3$, а для четвертой переменной $x^{(4)}$ примем следующее условие: $\mu^{(4)} = 3$, и, следовательно, согласно (9) и (10), $x_{\max}^{(4)} = 9$ и $M^{(4)} = 12$.

Для вывода алгоритма воспользуемся табл. 6, которая является аналогом табл. 5 при $N = 4$. В табл. 6 десятичные номера активных ячеек отсутствуют, они должны быть заполнены с помощью алгоритма вычисления десятичных номеров.

Табл. 6 разбита двойной линией на два основных блока в соответствии со значением $\mu^{(2)} = 2$. Каждый из этих блоков разбит на три блока второго уровня жирными линиями в соответствии со значением $\mu^{(3)} = 3$.

Нетрудно видеть, что число десятичных номеров, входящих в каждую таблицу, в силу структуры данных, образующих таблицу, равно $\mu^{(2)}\mu^{(3)}\mu^{(4)} = 18$. Число десятичных номеров, входящих в каждый блок, также в силу структуры данных, образующих таблицу, равно $\mu^{(3)}\mu^{(4)} = 9$, а каждый блок второго уровня содержит число номеров, равное $\mu^{(4)} = 3$.

Для вычисления текущего номера в табл. 6, например символического номера 6, 6, 6, 6, выделенного серым цветом, учтем, что он находится во втором блоке второго уровня, принадлежащего второму блоку. Поэтому десятичный номер символического номера, по аналогии со случаем $N = 3$, будет равен сумме следующих числовых значений:

- число номеров в предшествующих таблицах

$$(m_k^{(1)} - 1)\mu^{(2)}\mu^{(3)}\mu^{(4)}, k = \overline{1, \rho^*}, \quad (19)$$

где число $(m_k^{(1)} - 1)$ — число предшествующих таблиц;

- число номеров в предшествующих блоках табл. 5, равных произведению числа предшествующих блоков на число элементов в блоке:

$$\text{int}[(m_k^{(2)} - 1)/\rho^*]\mu^{(3)}\mu^{(4)}; \quad (20)$$

- число номеров в предшествующих блоках второго уровня, принадлежащих блоку, в котором находится символический номер:

$$\text{int}[(m_k^{(3)} - 1)/\rho^*]\mu^{(4)}, \quad (21)$$

- номер в текущем блоке второго уровня, равный

$$\text{int}[(m_k^{(4)} - 1)/\rho^*] + 1. \quad (22)$$

Таким образом, алгоритм вычисления десятичных номеров m_k , $k = \overline{1, \rho^*}$ при $N = 4$ по значениям соответствующих символических номеров имеет следующий вид:

$$\begin{aligned} m_k &= (m_k^{(1)} - 1)\mu^{(2)}\mu^{(3)}\mu^{(4)} + \\ &+ \text{int}[(m_k^{(2)} - 1)/\rho^*]\mu^{(3)}\mu^{(4)} + \\ &+ \text{int}[(m_k^{(3)} - 1)/\rho^*]\mu^{(4)} + \text{int}[(m_k^{(4)} - 1)/\rho^*] + 1, \\ &k = \overline{1, \rho^*}. \end{aligned} \quad (23)$$

Общая структура вывода алгоритма и результаты (18) для размерности $N = 3$ и результаты (23) для размерности $N = 4$ позволяют записать алгоритм вычисления десятичных номеров по соответствующим символическим номерам для произвольного значения размерности N входного вектора \mathbf{x} в следующем виде:

$$\begin{aligned} m_k &= (m_k^{(1)} - 1) \prod_{l=2}^N \mu^{(l)} + \text{int}((m_k^{(2)} - 1)/\rho^*) \times \\ &\times \prod_{l=3}^N \mu^{(l)} + \dots + \text{int}((m_k^{(N-1)} - 1)/\rho^*)\mu^{(N)} + \\ &+ \text{int}((m_k^{(N)} - 1)/\rho^*) + 1, k = \overline{1, \rho^*}. \end{aligned} \quad (24)$$

В таком виде алгоритм вычисления десятичных номеров представлен в работах [23, 25].

3. Программная реализация алгоритма вычисления номеров активных ячеек памяти СМАС

Программная реализация алгоритма (24) представлена здесь тремя Object Pascal-подпрограммами: основной подпрограммой **procedure** SmacAddrOutput, которая входному вектору ставит в соответ-

Таблица 6

Промежуточная последовательность символических номеров активных ячеек, значение старшего разряда равно 6, $\mu^{(1)} = 2$, $\mu^{(2)} = 2$, $\mu^{(3)} = 3$ и $\rho = 4$.
Размерность входного вектора $N = 4$

Десятичные номера активных ячеек, $m^{(d)}$	Символические номера активных ячеек, m
	6, 2, 2, 2 6, 2, 2, 6 6, 2, 2, 10
	6, 2, 6, 2 6, 2, 6, 6 6, 2, 6, 10
	6, 2, 10, 2 6, 2, 10, 6 6, 2, 10, 10
	6, 6, 2, 2 6, 6, 2, 6 6, 6, 2, 10
	6, 6, 6, 2 6, 6, 6, 6 6, 6, 6, 10
	6, 6, 10, 2 6, 6, 10, 6 6, 6, 10, 10

вие вектор активных ячеек памяти и вычисляет выход нейронной сети СМАС по соответствующему входному вектору, и двумя вспомогательными подпрограммами **procedure** OneDmAddress и **function** AmodBB, которые использует основная подпрограмма. Кроме того, основная подпрограмма использует не приведенную здесь подпрограмму ScalarProdD для выполнения скалярного произведения двух векторов. Подпрограмма **procedure** OneDmAddress необходима для вычисления адресов активных ячеек памяти по соответствующему значению скалярной компоненты входного вектора по формуле (5), а подпрограмма **function** AmodBB предназначена для вычисления по модулю в соответствии с формулой (4).

Переменные, входящие в подпрограмму CmacAddrOutput, имеют следующий смысл:

RoP — размерность вектора номеров активных ячеек памяти;

InpDimP — размерность N входного вектора x ;

XinputP — входной вектор x ;

XmaxDP — вектор максимальных значений компонент входного вектора;

WP — вектор памяти нейронной сети СМАС;

ActAddrP — вектор активных номеров ячеек памяти;

MJUmultiP, MJU — вспомогательные векторы размерности N ;

YP — выход нейронной сети СМАС, как сумма содержания активных ячеек.

```

procedure CmacAddrOutput(const RoP, InpDimP: Integer;
const XinputP, XmaxDP: VecInpD; WP: Memory;
var MJUmultiP: VecInpD; var ActAddrP: VecRo;
var YP: Real);
var
  i, j: Integer;
  OD: VecRo;
  c, MJU: VecInpD;
  MNRo: MATaddress;
  s: Real;
begin
  for i:= 1 to InpDimP do // Подготовка
    MJU[i]:= ((XmaxDP[i] - 1) div RoP) + 1;
  // вспомогательного вектора
  MJUmultiP [InpDimP]:= 1; // MJUmultiP
  for i:= InpDimP downto 2 do
  // Вычисление вспомогательного вектора
  MJUmultiP[i-1]:= MJU[i]*MJUmultiP[i];
  // вектора MJUmultiP

  for i:= 1 to InpDimP dp // Подготовка
    begin // матрицы MNRo
      OneDmAddress(RoP, XinputP[i], OD); // активных
      for j:= 1 to RoP do // адресов
        MNRo[i, j]:= OD[j];
      end;
  for j:= 1 to RoP do // Цикл
    begin // вычисления
      c[1]:=MNRo[1, j] - 1; // активных
      for i:= 2 to InpDimP do // адресов
        c[i]:= (MNRo[i, j] - 1) div RoP;
      ActAddrP[j]:= ScalarProdD
        (InpDimP, c, MJUmultiP) + 1;
      end;
  s:= 0.0; // Цикл
  for i:= 1 to RoP do // вычисления
    s:= s + WP[ActAddrP[i]]; // выхода
  YP:= s; // СМАС
end;

```

```

Procedure OneDmAddress(const RoP, XP: Integer; var
  ODAR: VecRo);
var
  i, j: Integer;
begin
  for i:= 0 to Ro - 1 do
    begin
      j:= AmodBB(XP + i, RoP);
      ODAR[j]:= XP + i;
    end;
  end;

function AmodBB(a, b: Integer): Integer;
begin
  if a mod b = 0 then
    Result = b
  else
    Result:= a mod b;
  end;

```

4. Тестовые примеры

Ниже приведены результаты пересчета ряда входных векторов в номера ячеек ассоциативной памяти, полученные в результате работы моделирующей нейронную сеть СМАС Object Pascal программы системы визуального программирования Delphi. Здесь приведены значения параметров, определяющих номера ячеек памяти, а именно: размерность входного вектора, максимальные значения компонент входного вектора, значения обобщающего параметра ρ^* , число возможных значений входного вектора, равное $\prod_{i=1}^N x_{\max}^{(i)}$, а также размерность M вектора памяти нейронной сети СМАС

$$\begin{aligned}
 M &= \rho^* \prod_{i=1}^N \frac{x_{\max}^{(i)} \rho^*}{\rho^*} = \rho^* \prod_{i=1}^N \frac{M^{(i)}}{\rho^*} = \\
 &= (\rho^*)^{-N+1} \prod_{i=1}^N (x_{\max}^{(i)} + \rho^* - 1). \quad (25)
 \end{aligned}$$

Вывод выражения для этого числа приведен в работах [23, 25].

1. **Размерность входного вектора** $N = 2$, $X = \{x^{(1)} = \overline{1,257}; x^{(2)} = \overline{1,257}\}$, $\rho^* = 8$, число входов 66049, число ячеек памяти $M = 8712$.

Входные векторы x и соответствующие им векторы активных ячеек памяти m приведены в табл. 7 и 8.

Таблицы иллюстрируют также отмеченный ранее факт, что близко расположенным векторам соответствуют близко расположенные векторы активных ячеек памяти.

В табл. 7 вектор разности векторов 2 и 1 минимален и равен вектору $\Delta x = (0 \ 1)^T$. Следствием близости этих векторов является минимальная разность векторов номеров 2 и 1 активных ячеек памяти m , которая равна вектору $\Delta m = (1 \ 0 \ 0 \ 0 \ 0 \ 0)^T$. Аналогичный эффект имеет место и для векторов 4 и 5.

2. **Размерность входного вектора** $N = 3$, $X = \{x^{(1)} = \overline{1,33}; x^{(2)} = \overline{1,33}; x^{(3)} = \overline{1,33}\}$, $\rho^* = 8$, число входов — 35937, число ячеек памяти — 1000.

Таблица 7

Таблица соответствия ряда входных векторов и векторов номеров активных ячеек памяти, $N = 2$, $\rho^* = 8$

N вектора	Входной вектор x	Вектор номеров активных ячеек памяти m
1	$x = (1 \ 1)^T$	$m = (1 \ 34 \ 67 \ 100 \ 133 \ 166 \ 199 \ 232)^T$
2	$x = (1 \ 2)^T$	$m = (2 \ 34 \ 67 \ 100 \ 133 \ 166 \ 199 \ 232)^T$
3	$x = (39 \ 188)^T$	$m = (1345 \ 1378 \ 1411 \ 1443 \ 1476 \ 1509 \ 1278 \ 1311)^T$
4	$x = (257 \ 256)^T$	$m = (8481 \ 8514 \ 8547 \ 8580 \ 8613 \ 8646 \ 8679 \ 8711)^T$
5	$x = (257 \ 257)^T$	$m = (8481 \ 8514 \ 8547 \ 8580 \ 8613 \ 8646 \ 8679 \ 8712)^T$

Таблица 8

Таблица соответствия ряда входных векторов и векторов номеров активных ячеек памяти, $N = 3$, $\rho^* = 8$

Входной вектор	Вектор номеров активных ячеек памяти
$x = (1 \ 1 \ 1)^T$	$m = (1 \ 26 \ 51 \ 76 \ 101 \ 126 \ 151 \ 176)^T$
$x = (1 \ 1 \ 2)^T$	$m = (2 \ 26 \ 51 \ 76 \ 101 \ 126 \ 151 \ 176)^T$
$x = (15 \ 27 \ 6)^T$	$m = (422 \ 447 \ 467 \ 492 \ 517 \ 541 \ 366 \ 391)^T$
$x = (33 \ 33 \ 32)^T$	$m = (825 \ 850 \ 875 \ 900 \ 925 \ 950 \ 975 \ 999)^T$
$x = (33 \ 33 \ 33)^T$	$m = (825 \ 850 \ 875 \ 900 \ 925 \ 950 \ 975 \ 1000)^T$

В приведенной табл. 8 входные векторы подобраны так, чтобы проиллюстрировать эффект близости входных векторов, аналогичный отмеченному в табл. 7. Кроме того, в обеих таблицах присутствуют векторы x минимальной и максимальной длины, это приводит к тому, что вектор минимальной длины активирует первую ячейку памяти, а вектор максимальной длины — последнюю ячейку памяти.

Заключение

Результаты пересчета входных векторов в векторы номеров активных ячеек памяти, приведенные в табл. 7 и 8, могут быть использованы для тестирования корректности работы компьютерной программы, выполняющей это преобразование. Эта программа, дополненная программой обучения нейронной сети СМАС, описанная, в частности, в работах [22, 23, 25], позволит успешно выполнить программную реализацию нейронной сети СМАС и осуществить практическую реализацию этой парадигмы нейронных сетей.

Список литературы

1. **Ersu E., Tolle H.** A new concept for learning control inspired by brain theory // Proc. Ninth IFAC World Congress, Budapest. 1984. V. 2. P. 1039—1044.
2. **Tolle H., Militzer J., Ersu E.** Zur Leistungsfähigkeit lokal vernetzter assoziativer Speicher und ihren Einsatzmöglichkeiten in lernenden Regelungen // Messen, Steuern, Regeln. 1989. V. 3. P. 98—105.
3. **Tolle H., Ersu E.** Neurocontrol. Learning control systems inspired by neural architectures and human problem solving strategies. Springer-Verlag, 1992. 211 p.
4. **Brown M., Harris Ch.** Neurofuzzy adaptive modelling and control. Prentice Hall, 1994. 508 p.
5. **Miller W. T., Glanz F. H., Kraft L. G.** CMAC: An associative neural network alternative to backpropagation // Proceedings of the IEEE. 1990. V. 78, N 10. P. 1561—1567.
6. **Kraft L. G., Campagna D. P.** A comparison between CMAC neural network control and two traditional adaptive control systems // IEEE Control System Magazine. Special issue on neural networks in control systems. 1990. V. 10, N 3. P. 36—43.
7. **Shirashi H., Ipri S. L., Cho D.-I. D.** CMAC neural network controller for fuel-injection systems // IEEE Transactions on Control Systems Technology. Special Issue on Automotive Control. 1995. V. 3, N 1. P. 32—38.
8. **Chen F.-Ch., Chang Ch.-H.** Practical stability issues in CMAC neural network control systems // IEEE Transactions on Control Systems Technology. 1996. V. 4, N 1. P. 86—91.
9. **Yu W., Rodriguez F. O., Moreno-Armendariz M. A.** Hierarchical Fuzzy CMAC for Nonlinear Systems Modelling. IEEE Transaction on Fuzzy Systems. V. 16, N 5. October 2008. P. 1302—1314.
10. **Majors M., Stori J., Cho D.-I.** Neural network control of automotive fuel-injection systems // IEEE Control Systems. 1994. V. 14, N 3. P. 31—36.
11. **Kuno H., Takayama K.** Learning control of robotic manipulators based on neurological model CMAC // Preprints of the 11th JFAC World Congress, Tallinn. 1990. V. 9. P. 268—273.
12. **Lin Yi., Song Shin-Min.** Learning hybrid position/force control of a quadruped walking machine using a CMAC neural network // Journal Robot. System (USA). 1997. V.14, No. 6. P. 483; Tallinn. 1990. V. 9. P. 268—273.
13. **Handelman D. A.** Adaptive collaborative control of highly redundant robot. Unmanned Systems Technology X. // Proceeding of SPIE. V. 6962, 69620Y. 2008. P. 1—10.
14. **Li B., Elliman D.** Neuro-fuzzy classification by a CMAC // Proc. of International Conference on Computational Intelligence and Multimedia Applications 1998. ICCIMA 1998. Gippsland, Vic., Australia, 9—11 Feb. 1998. P. 282—287.
15. **Horvath G.** Kernal CMAC: an Efficient Neural Network for Classification and Regression // Acta Polytechnica Hungarica. 2006. V. 3. P. 5—20.
16. **Lin C.-J. Lee J.-H., Lee C.-Y.** A novel hybrid learning algorithm for parametric fuzzy CMAC networks and its classification applications // Expert Systems with Applications: An International Journal. Vol. 35. Is. 4. Nov. 2008. P. 1711—1720.
17. **Fu J.-C., Shi J., Nguyen M. N.** An Early Warning System Based on Fuzzy CMAC // Machine Learning and Cybernetics, 2007 International Conference. 19—22 Aug. 2007. V. 1. P. 156—159.
18. **Fan K.-C., Lai Z.-F.** An intelligent Nano-positioning Control System Driven by an Ultrasonic Motor // International Journal of Precision Engineering and Manufacturing. 2008. V. 9. N 3. P. 40—45.
19. **Harmon F. G., Frank A. A., Joshi S. S.** The Control of a Parallel Hybrid-Electric Propulsion System for a Small Unmanned Aerial Vehicle using a CMAC Neural Network // Neural Networks. 2005. V. 18. P. 772—780.
20. **Rodriguez F. O., Yu W., Moreno-Armendariz M. A.** Anti-swing control with hierarchical fuzzy CMAC compensation for an overhead crane // Proc. 22nd IEEE International Symposium on Intelligent Control Part of IEEE Multi-conference on Systems and Control. Singapore, 1—3 October 2007.
21. **Albus J. S.** A new approach to manipulator control: the cerebellar model articulation controller // ASME Trans., J. Dynamic Systems, Measurement and Control. 1975. V. 97. N 3. P. 220—227.
22. **Albus J. S.** Data storage in the cerebellar model articulation controller (CMAC) // ASME Trans., J. Dynamic Systems, Measurement and Control. 1975. V. 97, N 3. P. 228—233.
23. **Аведьян Э. Д.** Ассоциативная нейронная сеть СМАС. Ч. I. Структура, объем памяти, обучение и базисные функции // Информационные технологии. 1997. № 5. С. 6—14.
24. **Аведьян Э. Д.** Ассоциативная нейронная сеть СМАС. Ч. II. Процессы обучения, ускоренное обучение, влияние помех, устранение влияния помех в двухслойной сети // Информационные технологии. 1997. № 6. С. 16—27.
25. **Аведьян Э. Д.** Дискретная ассоциативная нейронная сеть СМАС для построения непрерывных моделей // Интеллектуальные информационные системы. 2008. № 2.

П. В. Скрибцов, канд. техн. наук, ген. директор,
П. А. Казанцев, канд. техн. наук,
программист-математик,
А. В. Долгополов, программист-математик,
ООО "ПАВЛИН Техно"
e-mail: info@pawlin.ru

Виртуальные нейросетевые процессоры — метод кроссплатформенного ускорения расчета нейросетевых задач на современных многоядерных процессорах¹

Рассматриваются виртуальные нейросетевые процессоры, которые с использованием различных аппаратно-программных платформ позволяют эффективно ускорять нейросетевые задачи: распознавание лиц, походки, анализ временных рядов, улучшение качества изображений, поиск различных объектов на фото- и видеозаписях, классификация мимических выражений лица.

Ключевые слова: нейросетевые процессоры, расчет нейросетевых задач, методы ускорения расчета нейросетевых задач

Введение

На данный момент нейросетевые технологии считаются максимально эффективными для решения широкого круга задач: распознавание образов, прогнозирование, управление, решение различных математических задач. Сложность применения нейронных сетей в режиме реального времени состоит в сравнительно низкой производительности при реализации их стандартными средствами C++. Нейронные сети хорошо распараллеливаются [1], и в настоящее время для ускорения нейросетевых вычислений могут использоваться различные технологии: Intel MKL [2], CUDA [3], ACML [4]. Что делать, если необходимо иметь работающее приложение при отсутствии той или иной технологии? В простейшем случае для решения данной проблемы применяют связки if/else, #ifdef, которые приводят к снижению гибкости системы и трудностям переноса приложения с одной платформы на другую.

¹ Работа выполнена в рамках работ по государственному контракту ГК-02.514.11.4127_24.11.09

В статье рассматриваются виртуальные нейросетевые процессоры, которые с использованием различных аппаратно-программных платформ позволяют эффективно ускорять нейросетевые задачи: распознавание лиц, походки, анализ временных рядов, улучшение качества изображений, поиск различных объектов на фото и видео, классификация мимических выражений лица.

Технологии распараллеливания MKL, CUDA

Современные графические процессорные устройства (GPU) являются мощными специализированными вычислительными устройствами, которые могут быть использованы для ускорения решения целого ряда вычислительно сложных задач. Помимо высокой производительности GPU обладают рядом других преимуществ: низким энергопотреблением, низкой стоимостью, высокой доступностью. Достаточно простым и удобным инструментом для разработки приложений, использующих вычислительные мощности GPU, является технология Nvidia CUDA [3].

Библиотека Intel[®] Math Kernel Library (Intel[®] MKL) [2] обеспечивает выполнение высокооптимизированных многопоточных математических операций для научных, инженерных и финансовых приложений, требующих наибольшей производительности. В нее включены различные математические функции, методы решения разреженных матриц, методы быстрого преобразования Фурье, алгоритмы векторной математики и многое другое. Она содержит средства оптимизации производительности для микроархитектур Intel нового поколения. Набором оптимизированных функций для ускорения векторно-матричных операций обладает также библиотека Intel OpenCV [5], которая, хотя и уступает MKL по эффективности, зато является абсолютно бесплатной.

Для создания переносимого кода требуется наличие эффективных систем высокоуровневого программирования. Одной из таких систем можно считать виртуальный нейросетевой процессор.

Виртуальный нейросетевой процессор

Виртуальный нейросетевой процессор спроектирован для задач с большим объемом данных и поддерживает в данный момент следующие программно-аппаратные платформы: OpenCV, Intel MKL, CUDA.

Процессор поддерживает многослойные нейронные сети прямого распространения [1, 6] с различными функциями активации. В настоящий

момент поддерживаются следующие функции активации:

- $f(x) = \frac{1}{1 + e^{-x}}$;
- $f(x) = \frac{2}{1 + e^{-x}} - 1$;
- $f(x) = \frac{1}{1 + e^{-x}} - 0,5$;
- $f(x) = 1,7159 \tanh\left(\frac{2}{3}x\right)$;
- $f(x) = x$.

Отличительной чертой данного процессора является простой интерфейс. Процессор может работать в двух режимах: режиме пакетного расчета выходов для большого числа примеров и режиме обучения нейронной сети на обучающей выборке. На данный момент для всех программно-аппаратных платформ поддерживается два алгоритма обучения:

- QUICKPROP [7];
- RPROP[8, 9].

Для OpenCv и MKL дополнительно поддерживаются алгоритмы:

- LM(Levenberg Marquardt) [6];
- TEL(Target Error Learning) [10].

Для некоторых платформ ускорения важно, чтобы данные хранились на устройстве, поэтому в интерфейс процессора введены функции загрузки/выгрузки сети на процессор, загрузки обучающей последовательности, загрузки входов для расчета. В дальнейшем расчет ведется уже с данными, которые хранятся на устройстве, что позволяет сократить время пересылки между центральным процессором и устройством, на котором проводится ускорение.

Для смены программно-аппаратной платформы достаточно изменить текстовый конфигурационный файл.

Методы ускорения нейросетевых задач на многоядерных процессорах

Ускорение расчета выходов нейронной сети для большого числа примеров. Виртуальный нейросетевой процессор оптимизирован на работу одной нейронной сети с большим количеством данных. Расчет одного слоя для нескольких примеров можно представить в матричном виде. Пусть дана матрица входов для расчета $X_{n \times m}$:

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix};$$

матрица весовых коэффициентов слоя $W_{k \times m}$:

$$\begin{bmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \dots & \dots & \dots & \dots \\ w_{k1} & w_{k2} & \dots & w_{km} \end{bmatrix};$$

смещения слоя $B_{1 \times k}$:

$$[b_{11} \ b_{12} \ \dots \ b_{1k}],$$

где n — число примеров для расчета; m — число входов слоя; k — число нейронов (выходов) слоя.

Необходимо вычислить матрицу выходов слоя $Y_{n \times k}$:

$$\begin{bmatrix} y_{11} & y_{12} & \dots & y_{1k} \\ y_{21} & y_{22} & \dots & y_{2k} \\ \dots & \dots & \dots & \dots \\ y_{n1} & y_{n2} & \dots & y_{nk} \end{bmatrix}.$$

Выходы слоя рассчитываются в два этапа согласно следующим формулам:

$$O = XW^T;$$

$$y_{ij} = f(o_{ij} + b_j),$$

где $f(\cdot)$ — заданная функция активации; o_{ij} — выход j -го нейрона текущего слоя для i -го примера; b_j — смещение для j -го нейрона текущего слоя.

Умножение матриц хорошо ускоряется на многоядерных процессорах с помощью технологий CUDA и MKL. Активирование слоя так же легко распараллеливается по числу примеров и по числу нейронов в слое.

Умножение матриц CUDA. Эффективная реализация умножения матриц использует shared-память (память, разделяемая между нитями одного блока).

Пусть необходимо найти матрицу $C = AB$. Матрица C разбивается на подматрицы C_{sub} размерностью $block_size \times block_size$. Один блок нитей вычисляет одну подматрицу C_{sub} , а каждая нить в блоке вычисляет один элемент этой матрицы (рис. 1). C_{sub} равна произведению двух подматриц: подматрицы A_{sub} ($block_size \times A.width$), которая имеет такое же число строк, как и подматрица C_{sub} , и подматрицы B_{sub} размерности ($B.height \times block_size$), которая имеет такое же число столбцов, что и подматрица C_{sub} . Эти две подматрицы при необходимости разделяются на множество квадратных подматриц размерности $block_size$, и подматрица C_{sub} вычисляется как сумма произведений этих квадратных матриц. Чтобы получить эти произведения, осуществляется загрузка двух соответ-

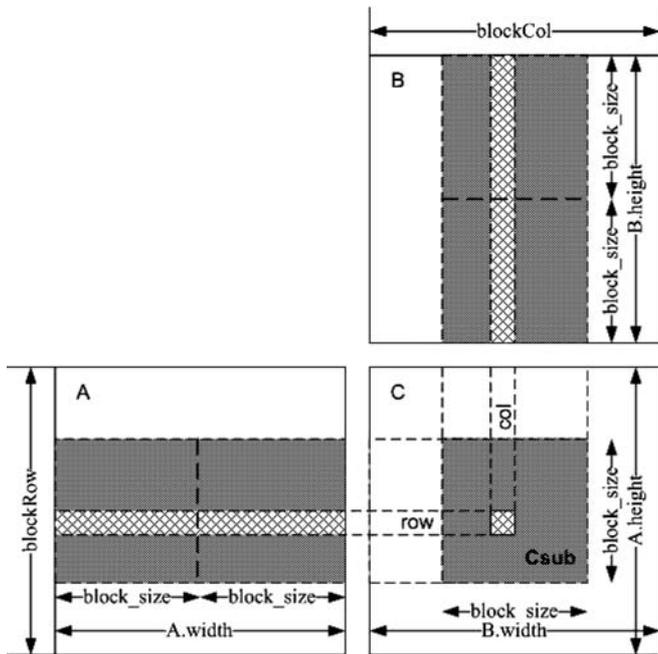


Рис. 1. Умножение матриц с использованием разделяемой памяти

вующих квадратных матриц из глобальной памяти в разделяемую память (одна нить загружает один элемент матрицы), затем каждая нить вычисляет один элемент умножения. Каждая нить аккумулирует результаты вычислений в регистрах, а после выполнения один раз записывает результат в глобальную память.

При такой организации вычислений используются преимущества быстрой разделяемой памяти, а также экономится пропускная способность глобальной памяти, поскольку чтение из глобальной памяти выполняется один раз.

Размерность подматрицы оптимально равна 16×16 . Для избежания лишних ветвлений в ядре необходимо, чтобы размерности перемножаемых матриц были кратны размеру блока.

При вычислении матрицы $C = AB^T$ матрица B будет загружаться построчно, как матрица A . Если размерность матрицы в shared-памяти кратна 16, то каждый ее столбец размещается в одном банке. Поэтому перебор нитью по очереди элементов из каждого столбца приводит к конфликту банков. Для того чтобы этого избежать, используется простой прием — матрица в разделяемой памяти дополняется столбцом. За счет этого расположение столбцов в памяти сдвигается, и каждый столбец равномерно распределяется по всем банкам памяти.

Для эффективного умножения матриц может использоваться библиотека CUBLAS.

Ускорение обучения нейронных сетей. При обучении нейронных сетей, как правило, используется

алгоритм обратного распространения ошибки. Алгоритм обратного распространения ошибки — это способ ускорения расчета градиента, который используется другими методами обучения (RPROP, QUICKPROP).

В случае пакетного обучения, когда используется суммарный градиент по всем примерам, алгоритм может быть выражен в матричном виде.

1. Для последнего слоя сети вычисляется матрица D с элементами

$$d_{ij} = (o_{ij} - t_{ij})f'(o_{ij}),$$

где $i = 1 \dots N$, $j = 1 \dots K$; N — число примеров обучающей выборки; K — число выходов сети; o_{ij} — выход сети; t_{ij} — обучающий выход.

2. Для скрытых слоев сети матрица D рассчитывается согласно формулам (1) и (2):

$$D^i = D^{i+1}(W^{i+1})^T, \quad (1)$$

где D^i — матрица размерности $(N \times n_i)$; n_i — число нейронов i -го слоя, D^{i+1} — матрица размерности $(N \times n_{i+1})$; n_{i+1} — число нейронов $(i+1)$ -го слоя; W^{i+1} — матрица весовых коэффициентов между текущим слоем i и следующим слоем $i+1$. После расчета матрица D^i умножается на производные выходов i -го слоя:

$$d_{ij} = d_{ij}f'(o_{ij}). \quad (2)$$

3. Суммарный градиент весов рассчитывается согласно формуле

$$\text{grad } W^{i+1} = (D^{i+1})^T O^i,$$

где $\text{grad } W^{i+1}$ — матрица градиентов весов размерности $(n_{i+1} \times n_i)$; O^i — матрица выходов слоя размерности $(N \times n_i)$.

4. Суммарный градиент смещений для i -го слоя вычисляется по формуле

$$\text{grad } b_j = \sum_{k=1}^N d_{kj},$$

где d_{kj} — элемент матрицы D^i .

Ускорение расчета среднеквадратичной ошибки. Среднеквадратичная ошибка нейронной сети для выборки находится по формуле

$$E_{msr} = \frac{1}{2} \sqrt{\sum_{i=1}^N \sum_{j=1}^M (t_{ij} - o_{ij})^2},$$

где t_{ij} — элемент обучающей выборки; o_{ij} — выходы сети; N — число примеров в обучающей выборке, M — число выходов нейронной сети.

Для ускорения с помощью CUDA расчет можно разбить на два этапа:

1) расчет ошибок для каждого примера

$$\sum_{j=1}^N (t_i - o_{ij})^2$$

— распараллеливается по примерам

(одна нить — один пример);

2) расчет суммарной ошибки по примерам — данный расчет хорошо ускоряется на CUDA с помощью алгоритма редукции.

Чтобы избежать лишних операций чтения/записи в глобальную память, можно эти этапы объединить в один.

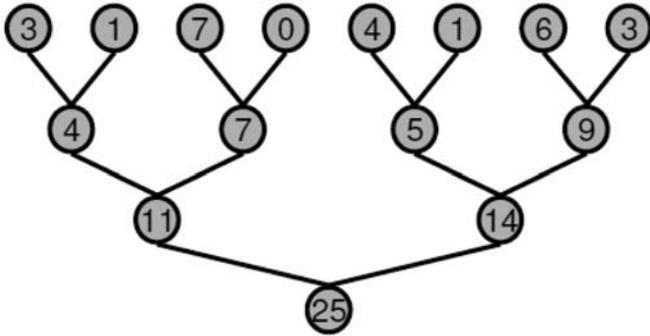


Рис. 2. Иерархическое суммирование элементов внутри блока

Алгоритм редукции CUDA. Пусть имеется массив a_0, a_1, \dots, a_{N-1} и некоторая бинарная операция или функция (сложение, вычитание, максимум, минимум). Редукцией массива a_0, a_1, \dots, a_{N-1} относительно заданной операции (в данном случае сложения) называется выражение

$$A = (a_0 + a_1) + \dots + a_{N-1}.$$

Алгоритм редукции имеет низкую арифметическую интенсивность, поэтому необходимо максимизировать пропускную способность памяти [11].

Каждому блоку нитей соответствует часть массива (массив делится поровну между всеми блоками). Блок нитей вычисляет сумму всех элементов своей части и записывает результирующее значение в выходной массив. При этом каждой нити соответствует один элемент массива, сначала каждая нить загружает свой элемент в shared-память, а затем иерархически суммирует, как показано на рис. 2.

После нахождения частичных сумм массива необходимо выполнить сложение этих сумм.

Алгоритм редукции реализован в библиотеках CUDPP [12], Thrust [13].

Сравнение производительности программно-аппаратных платформ

Были проведены эксперименты с использованием виртуального нейросетевого процессора. Тестирование выполнялось для сети конфигурации 100—16—12—1 на обучающей последовательности с числом примеров 50 000. Сравнялось время выполнения одной итерации обучения для алгоритма RPROP. Результаты тестирования для различных конфигураций оборудования представлены на рис. 3. За наиболее медленную точку для сравнения берется виртуальный нейросетевой процессор, реализованный на технологии Intel OpenCV (хотя такой процессор все равно в несколько раз быстрее "наивной" C++ реализации нейросетевых алгоритмов, даже скомпилированных оптимизирующим компилятором с применением SSE).

Коэффициент ускорения CUDA относительно MKL и OpenCV для одной итерации RPROP представлены на рис. 4.

По графикам видно, что технологии CUDA и MKL позволяют ощутимо ускорять нейросетевые вычисления (в среднем на порядок).

Применение нейросетевых процессоров

Виртуальные нейросетевые процессоры были применены в приложении PWNBinary Trainer [14]. PWNBinaryTrainer — простое консольное GPGPU приложение, разработанное как альтернатива обучению нейронных сетей в дорогих коммерческих пакетах. Процесс обучения в настоящее время занимает минуты вместо часов и с сопоставимой точ-

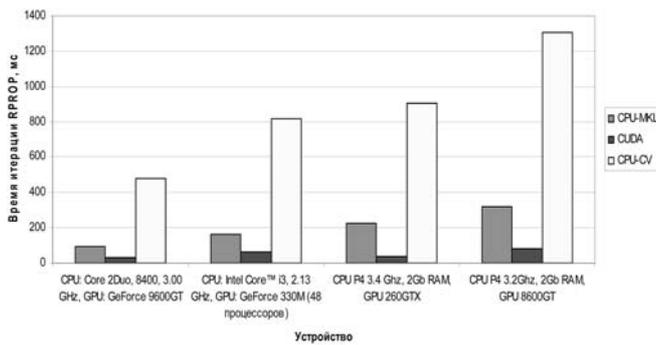


Рис. 3. Время выполнения одной итерации RPROP на различных конфигурациях оборудования

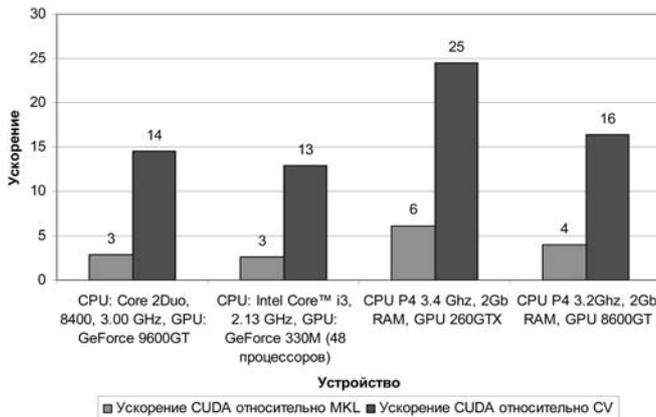


Рис. 4. Коэффициент ускорения CUDA относительно других технологий (OpenCV, обозначена как "CV") на различных конфигурациях оборудования

ностью. На данный момент приложение ориентировано на бинарные задачи классификации (объект/не объект). Ускорение обучения сети может достигать 30^x по сравнению с пакетом STATISTICA 7.

Выводы

Компании Intel и NVIDIA, находящиеся в своеобразной "гонке вооружений", соревнуются в достижении большей производительности вычислений, что порождает актуальность реализации нейросетевых вычислений на различных платформах при сохранении одинакового интерфейса к доступу данной функциональности в целях обеспечения взаимозаменяемости аппаратных ускорителей. Применение технологии виртуальных нейросетевых процессоров позволяет абстрагироваться от определенной программно-аппаратной платформы, сконцентрироваться на прикладных аспектах применения нейронных сетей и вместе с тем дает ощутимое ускорение нейросетевых вычислений (как обучения, так и расчета выходного сигнала для больших объемов данных). В настоящее время ведутся коммерческие разработки с использованием виртуальных процессоров для нейросетевого распознавания объектов на фото- и видеоизображениях (детекция лиц и ключевых точек лица на изображениях), распознавания человека по походке, анализа данных лазерного сканирования местности, предсказания спроса и многих других.

Список литературы

1. **Галушкин А. И.** Теория нейронных сетей. Кн. 1: учеб. пособие для вузов. М.: ИПРЖР, 2000. 416 с.
2. **Highly optimized math library** [Электронный ресурс]. URL: <http://software.intel.com/en-us/intel-mkl/> (13.09.2010).
3. **Что такое CUDA?** [Электронный ресурс]. URL: http://www.nvidia.ru/object/what_is_cuda_new_ru.html (13.09.2010).
4. **AMD Core Math Library (ACML)** [Электронный ресурс]. URL: <http://developer.amd.com/cpu/Libraries/acml/Pages/default.aspx> (13.09.2010).
5. **OpenCV** [Электронный ресурс]. URL: <http://ru.wikipedia.org/wiki/OpenCV> (13.09.2010).
6. **Хайкин С.** Нейронные сети полный курс. 2-е изд. М.: Издат. дом "Вильямс", 2006.
7. **Wilamowski B. M., Iplikci S., Kaynak O., Efe V. O.** An Algorithm for Fast Convergence in Training Neural Networks // International Joint Conference on Neural Networks. 2001. Vol. 3. P. 1778—1782.
8. **Riedmiller M., Braun H.** A direct adaptive method for faster backpropagation learning: The RPROP algorithm. San Francisco, 1993.
9. **Fahlman S. E.** An empirical study of learning speed in back-propagation networks. Technical report, CMU-CS-88-162, Carnegie-Mellon University, 1988.
10. **Скрибцов П. В.** Настройка коэффициентов нейронной сети с использованием функционала целевой ошибки // Нейрокомпьютеры: разработка, применение. 2004. № 4.
11. **Harris M.** NVIDIA CUDA SDK, Optimizing Parallel Reduction in CUDA. URL: http://developer.download.nvidia.com/compute/cuda/1_1/Website/projects/reduction/doc/reduction.pdf (25.10.2010).
12. **CUDPP** [Электронный ресурс]. URL: <http://ggpu.org/developer/cudpp> (13.09.2010).
13. **What is Thrust?** [Электронный ресурс]. URL: <http://code.google.com/p/thrust/> (13.09.2010).
14. **PWNBinaryTrainer 1.0** [Электронный ресурс]. URL: <http://www.pawlin.com/content/view/19/8/> (13.09.2010).

IV международный форум информационных технологий "ITFORUM 2020/Информационное общество 2.0"

26–28 апреля 2011 года

ОРГАНИЗАТОРАМИ ФОРУМА ЯВЛЯЮТСЯ:

Правительство Нижегородской области, www.itforum2020.ru
Всероссийское ЗАО "Нижегородская ярмарка", www.yarmarka.ru

МИССИЯ ФОРУМА

Форум направлен на демонстрацию растущего научно-технического потенциала России в области ИКТ, заинтересованность и степень вовлеченности региона в федеральные программы и проекты, выявление главных проблем по их реализации, содействие развитию ИКТ в нашей стране путем увеличения объема обмена знаниями и опытом между зарубежными странами и Россией и внутри страны между различными регионами и ИТ-компаниями, предоставление возможности непосредственного знакомства с последними технологиями и их разработчиками.

КОНГРЕСС:

- Качество жизни граждан в условиях развития бизнеса в информационном обществе.
- Электронное государство и эффективность государственного управления.
- Российский рынок информационных и телекоммуникационных технологий.
- Базовая инфраструктура информационного общества.
- Безопасность в информационном обществе.
- Цифровой контент и культурное наследие.

Информационные технологии и системы 19-я специализированная выставка

1. Специальные тематические экспозиции
2. Информационные технологии
3. Телекоммуникации
4. Почтовая связь

КОНТАКТЫ:

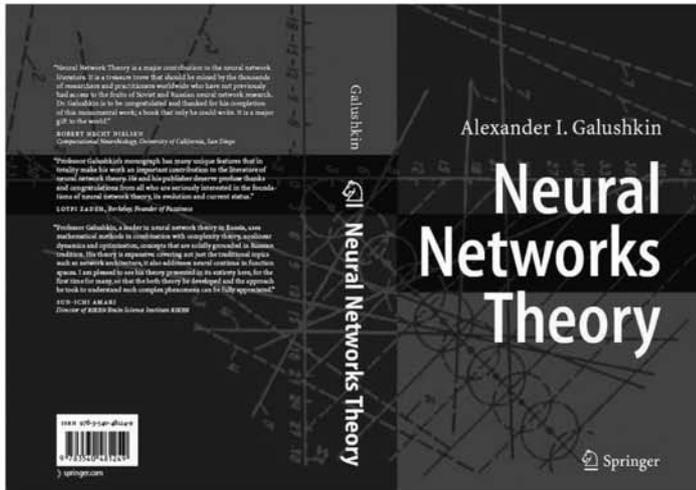
Конгресс:

Стешин Александр Андреевич
Телефон — +007 (831) 433-66-26
Факс — +007 (831) 437-33-57
e-mail: steshin@inform.kreml.nnov.ru

Выставки:

Ветрова Мария Борисовна
Телефон — +007 (831) 277-55-97
Факс — +007 (831) 277-53-71
e-mail: mary@yarmarka.ru

ПРЕДСТАВЛЯЕМ КНИГУ



Теория нейронных сетей, развитие которой в значительной степени определяет уровень решения сложных научно-технических задач, связанных с развитием высоких технологий в самых различных отраслях промышленности, народном хозяйстве и военной технике, является важным разделом современных научных исследований.

В августе 2007 г. в издательстве «Springer» вышла из печати на английском языке монография доктора технических наук, профессора А.И. Галушкина «Neural Networks Theory», которая содержит результаты многолетних исследований автора в области теории нейронных сетей – логической основы построения принципиально новых, по сравнению с классическими, вычислительных систем – нейрокомпьютеров.

В 2010 г. в издательстве "Горячая Линия – Телеком" опубликована монография А.И. Галушкина "Нейронные сети: основы теории", которая является переводом монографии, выпущенной издательством "Springer".

Монография является одной из немногих, если не единственной, монографией российского ученого, в которой представлены предисловия трех известных ученых с мировым именем:

- Роберта Хехт-Нильсена – ведущего разработчика нейрокомпьютеров в США;
- Лотфи Заде – автора концепции размытой логики;
- Шун-иши Амари – директора Института исследований мозга в Японии.

В частности, Роберт Хехт-Нильсен отмечает следующее: "Эта книга представляет собой долгожданный панорамный обзор советской и российской нейросетевой традиции. Книга является кладом важных идей и значительных результатов, которые не доступны более нигде в английском варианте. Автор, доктор А.И. Галушкин, является ведущим российским экспертом в области нейронных сетей и был ведущим советским и российским разработчиком нейронных сетей с 1970 г. В этот период доктор А.И. Галушкин имел доступ ко всем важным западным публикациям по нейросетевой тематике. Поэтому ценность этой книги удваивается, так как она написана не просто экспертом, но и человеком, который знает и ссылается на интеллектуальные достижения западной школы. Монография "Нейронные сети: основы теории" является наиболее значимым вкладом в литературу по нейросетевой тематике. Этот найденный вклад должен быть использован тысячами исследователей и практиков по всему миру, у которых до сих пор не было возможности воспользоваться плодами советских и российских исследований в области нейронных сетей. Доктора Галушкина следует поздравить и поблагодарить за написание этой монументальной работы – книги, которую мог написать только он. Это по-настоящему дар всему миру".

Лотфи Заде пишет: "Монография профессора А.И. Галушкина имеет множество уникальных свойств, которые в общей сложности делают его работу важным вкладом в литературу по теории нейронных сетей. Он и его издатель заслуживают щедрых благодарностей и поздравлений от всех, кто всерьез имеет интерес к созданию, развитию и текущему положению дел теории нейронных сетей".

А профессор Амари отмечает: "Профессор А.И. Галушкин, ведущий специалист по теории нейронных сетей в России, использует математические методы в комбинации с теорией сложности, нелинейной динамикой и оптимизацией, а также другими концепциями, крепко укоренившимися в российской научной школе. Его теория очень обширна: она охватывает не только традиционные аспекты, такие как архитектура сети, но также рассматривает континуальные нейронные сети в пространствах функций. Я с большим удовольствием воспринял выход книги, в которой эта теория описана во всей своей полноте. Огромная ценность самой теории и используемого автором метода описания такого сложного явления, как нейросетевая система, не может вызывать никаких сомнений".

От редакции необходимо отметить, что изданная "Springer" монография является естественным продолжением большого числа работ, опубликованных автором за сорок лет работы в этой области.

Результаты работ автора в области теории нейронных сетей стали основой для развития нейроматематики – нового раздела вычислительной математики, связанного с решением сложных математических задач в нейросетевом логическом базисе, а также основой для развития нейроуправления – нового раздела теории управления, ориентированного на сложные нелинейные, многомерные объекты управления с переменными параметрами и структурой.



CONTENTS

Talalay M. S., Trushin K. V. Review of Regular VLSI Design Approaches 2

Design rules for small process node become more aggressive and more complex nowadays. The usage of strongly predefined structures for layout design is the way to cope with the complexity of manual cells design within the limits of design rules. This paper describes modern design methodologies within the limits of regular layout. It lays special emphasis on one of the design for manufacturability techniques — Regular Fabrics.

Keywords: regular layout, regular fabrics, design for manufacturability, VPGA (via pattern gate array)

Bibilo P. N., Loginova I. P., Romanov V. I., Cheremisina L. D. The system CLTT for Computer-Aided Design of Topology of Functional Blocks of Custom Digital VLSI. 8

The system CLTT of computer-aided design of functional blocks of custom very large-scale CMOS integrated circuits is described. The proposed design flow allows, having the initial VHDL description of functional blocks, to get lay-out with compact placement of macrocells as well as data and power wires in accordance with technological rules.

Keywords: very-large-scale integration circuit, macrocells of regular structure, combinational logic, programmable logic arrays, VHDL, placement, routing, topology description

Evgenov G. B. Synergetic Methodology for Integration of Applied Knowledge 15

There is described a new information technology for development of intelligent applied systems based on different knowledge: geometric, mathematic, knowledge for hard and soft computing. This technology provides a dramatic time reduction of systems creation and implementation.

Keywords: informational technology, artificial intelligence, hybrid systems, programming theory, a clear knowledge, fuzzy knowledge, the knowledge module, geometric knowledge, mathematical knowledge, the neural network model, linguistic variable, fuzzy knowledge base

Kolesnikova S. I. Feature of Usage of Linear Convolution of Criteria in Method of Paired Comparisons 24

Problem of incorrect estimation of alternative set in Saaty's method of analytical hierarchies for regularities discovery and decision-making support is considered. Modified procedure for estimation of significance of alternatives is suggested. The properties of this procedure are proofed and the illustrative examples are given.

Keywords: method of analytical hierarchies, alternative-weight coefficients, intelligence system, test pattern recognition, quality of decision-making

Komarov E. G. The Determination of a Standard Object Model and Rating Points Based on it 30

The paper presents a model of the standard object developed as a set of fuzzy numbers and linguistic values of the scale used for evaluating the characteristics of objects. A model of the rating estimation based on the standard object is developed in this paper. The model extends the information obtained with the help of traditional ratings.

Keywords: uncertainty, standard objects, rating points

Mandrikova O. V., Solovyev I. S. Analysis of the Earth's Magnetic Field Variations on the Basis of Wavelet-Technology 34

The wavelet-technology is proposed for studying features of geomagnetic signals. This technology allows one to extraction geomagnetic disturbances in the automatic mode, to create a solar quiet daily variation and to calculate the geomagnetic activity index K.

Keywords: wavelet-transform, the Earth's magnetic field, geomagnetic activity index K

Gerasimov Yu. Yu., Sokolov A. P., Katarov V. K. Elaboration of a Logistic System for Industrial Wood Harvesting and Transportation 39

The topic of this article is the use of new informational technologies and mathematical methods for optimal roads construction and planning of forest roads net. General set of the task, using models, methods and facilities are described in the article.

Keywords: wood harvesting, planning of forest road net, forest road construction, spatial models, mathematical models, optimization, geographical informational systems

Dzhashitov V. E., Pankratov V. M., Golikov A. V. Multimedia Information-computer Technologies in Scientific Educational Lectures and the Express-Examination of Knowledge on the Exact Sciences 44

The innovative approach to creation of scientific educational lectures on the exact sciences with the subsequent the express-examination of knowledge is offered. Methodical aspects and practical experience of creation and application of multimedia computer technologies are considered at realization of the new approach. Lectures are presented in the form of packages of presentation slides in PowerPoint and the built-in programs

of dynamic visualization. Computer modeling and dynamic three-dimensional visualization of principles of action, the equations of motion, physical processes and fields of examined measuring instruments, kinematics and dynamics of different mechanisms and phenomena of mechanical motion is envisioned. The original software for dynamic visualization is developed.

Keywords: information-computer technologies in education, multimedia lectures on the exact sciences, devices, inertial sensors, mechanisms, physical fields and processes, dynamic visualization

Puzankov D. V., Panteleyev M. G., Zhukova N. A., Semenov V. S. Information System for Designing of Competence Models Based on the Semantic Web Technologies 49

The problem of creation of information system for designing of competence models by distributed expert groups is discussed. The requirements to such system are analyzed. The scenario for collaborative work of experts, components of dataware based on the ontological approach, architecture of information system and its perspectives are considered.

Keywords: designing of competence models, information system, distributed expert groups, ontological approach

Galushkin A. I. Neurocomputers is Solving the Problems of Information Security 58

The main directions of the neural network technologies application in the systems of informational security providing are presented in the article, particularly by the following tasks solution:

- text messages processing;
- detection of illegal access;
- viruses detection;
- biometric identification;
- estimation of informational security.

Keywords: informational security, neural network technologies, attacks, viruses, biometrics, cryptography

Avedyan E. D., Pantiukhin D. V. Nonlinear Transformation Algorithm in the CMAC Neural Network 64

The short description of the neural network CMAC intended for storing of functions of many variables is given. It is noticed that one of the major elements of this network is the block in which there is a nonlinear transformation of a vector of arguments of function to a vector of numbers of associative memory cells. Deduction of the analytical form of this algorithm its program realization and some results of computer modeling which can be used for algorithm testing are given.

Keywords: neural network CMAC, algorithm, cells number, memory, program realization

Skritsov P. V., Kazantsev P. A., Dolgoplov A. V. Virtual Neural Network Processors — Cross-Platform Approach to Accelerate Neural Network Processing Using Multi-Core Computers 73

The article introduces the virtual neural network processors, which using different hardware and software platforms can effectively accelerate the neural network tasks: face recognition, gait recognition, time series analysis, image enhancement, search for various objects in the photo and video, the classification of mimic facial expressions.

Keywords: neural networks, machine learning, neural networks training acceleration, parallel computing, multi-core processors, CUDA

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е. В. Конова*.

Корректор *Т.В. Пчелкина*.

Сдано в набор 09.11.2010. Подписано в печать 20.12.2010. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.

Усл. печ. л. 9,8. Уч.-изд. л. 11,03. Заказ 16. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Отпечатано в ООО "Подольская Периодика"

142110, Московская обл., г. Подольск, ул. Кирова, 15