

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

1(137)
2008

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

НЕЙРОСЕТИ

- Чернецова Е. А. Метод автоматизированной обработки изображений, представленных в виде массива величин интенсивности пикселей, с помощью нейронных сетей 2
- Корсунов Н. И., Юдин А. А. Решение задач теплопроводности с использованием нейросетевых компьютерных технологий 6

ОПТИМИЗАЦИЯ

- Карпенко А. П., Федорук В. Г. Обзор программных систем многокритериальной оптимизации. Отечественные системы 15
- Гагарин А. В. Интеллектуальный алгоритм оптимизации параметров ресурсоемких моделей 23
- Петунин А. А., Мухачева Э. А., Филиппова А. С. Метод прямоугольной аппроксимации для решения задач нерегулярного фигурного раскроя — упаковки 28

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Аксарин М. В., Тарасюк М. В. Модель скрытого канала реляционной БД с многоуровневым доступом 32
- Шниперов А. Н. Синтез и анализ высокоскоростных симметричных криптосистем на основе управляемых операций 36

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

- Мандриков Е. А., Кулев В. А., Шалыто А. А. Применение генетических алгоритмов для создания управляющих автоматов в задаче о "Флибах" 42
- Долгов А. И., Короченцев Д. А. Сложение возможных и вероятностных нечетких треугольных чисел 46

КОМПЬЮТЕРНАЯ ГРАФИКА

- Милов А. Н. Два подхода к организации масштабируемых графических архитектур стандарта OpenCL на базе цифровых процессоров обработки сигналов 51
- Орлов А. А., Канунова Е. Е. Цифровая обработка текста на изображениях рукописей как линейчатых объектов 57
- Мельман С. В. Визуализация объемов в задачах анализа физических полей синоптических объектов 62

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Манцивода А. В., Малых А. А. Достижения в Интернете и будущее информационной среды российского образования 67
- Серков Л. А. Синергетическое моделирование образовательных процессов 74
- Черняховская Л. Р., Герасимова И. Б., Нугаева К. Р. Поддержка принятия решения по управлению качеством образовательного процесса на основе хранения знаний университета 79
- Contents 88
- Приложение. Авдошин С. М., Белкин С. А. Фабрики приложений

Главный редактор
НОРЕНКОВ И. П.

Зам. гл. редактора
ФИЛИМОНОВ Н. Б.

Редакционная
коллегия:

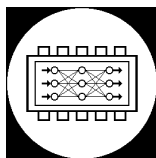
АВДОШИН С. М.
АНТОНОВ Б. И.
БАТИЩЕВ Д. И.
БАРСКИЙ А. Б.
БОЖКО А. Н.
ВАСЕНИН В. А.
ГАЛУШКИН А. И.
ГЛОРИОЗОВ Е. Л.
ГОРБАТОВ В. А.
ДОМРАЧЕВ В. Г.
ЗАЛЕЩАНСКИЙ Б. Д.
ЗАРУБИН В. С.
ИВАННИКОВ А. Д.
ИСАЕНКО Р. О.
КОЛИН К. К.
КУЛАГИН В. П.
КУРЕЙЧИК В. М.
ЛЬВОВИЧ Я. Е.
МАЛЬЦЕВ П. П.
МЕДВЕДЕВ Н. В.
МИХАЙЛОВ Б.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

Редакция:

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

Аннотации статей размещены на сайте журнала по адресу <http://www.informika.ru/text/magaz/it/> или <http://novtex.ru/IT>

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



УДК 551.46.06.504.054.001.5

Е. А. Чернецова, канд. техн. наук,
Российский государственный
гидрометеорологический университет,
Санкт-Петербург E-mail: chernetsova@list.ru

Метод автоматизированной обработки изображений, представленных в виде массива величин интенсивности пикселей, с помощью нейронных сетей

Обсуждается метод обработки изображений, представленных в виде массива интенсивностей пикселей, с помощью нейронных сетей (НС), в котором в качестве входных параметров НС используются физические и геометрические характеристики объекта, а выходной параметр представляет собой вероятность принадлежности темного объекта к заданному классу изображений.

Введение

Проблема автоматизации обработки изображений особенно актуальна при решении задач дистанционного мониторинга, когда требуется высокоскоростная обработка больших объемов данных в месте установки оборудования с выдачей человеку-оператору только координат местонахождения объектов наблюдения.

Главной проблемой обработки изображений, представленных в виде массива интенсивностей пикселей, является то, что пространственные черты изображенных объектов выражены неявно. Выявление пространственных черт объектов заключается в обработке массива пикселей методами, которые позволяют либо выявить контуры объектов, либо определить характеристики областей изображения с одинаковой интенсивностью пикселей. Методы выявления контуров объектов используются в задачах фильтрации, сжатия изображений и включают в себя описание изображения в терминах коэффициентов дискретного косинусообразного преобразования [1], преобразования Фурье [2], вейвлет-преобразования [3]. Если же решается задача классификации изображенных объектов, то необходимо разделение изобра-

жения на области, имеющие одинаковые характеристики. Сегментация изображения в этом случае основана на иерархической кластеризации пикселей, когда один пиксель сравнивается со своими пространственными "соседями" и приписывается к тому или иному сегменту по критерию однородности или на основе статистической метрики подобия [4, 5].

После завершения сегментации изображения должна быть проведена классификация изображенных объектов. Алгоритмы классификации базируются в основном на байесовских или статистических решениях. Недостатком этих методов является сложность выработки классификационных правил из-за вовлеченности в этот процесс многих нелинейных и плохо изученных факторов. Эти сложности можно преодолеть при использовании алгоритмов нейронных сетей (НС). Нейронные сети, в отличие от статистических классификаторов, не требуют точно определенного отношения между входным и выходным векторами, так как формируют свои собственные отношения вход—выход из набора данных, используя построение границ решений [6].

Теоретический анализ

Основным элементом для построения НС служит искусственный нейрон [7]. Он традиционно представляется линейной суммой с N входами (каждому их входов ставится в соответствие некоторый весовой коэффициент w_i) и одним выходом, соединенным с нелинейным элементом, реализующим активационную функцию нейрона $F(x)$.

При выборе структуры нейронной сети для решения задачи классификации необходимо учитывать следующие аспекты:

- способность сети к обучению, т. е. возможность научить систему распознавать требуемое число объектов. Чем больше в сети слоев и нейронов, тем выше ее способности и одновременно с этим потребности в аппаратных ресурсах;
- быстродействие, которое достигается уменьшением сложности сети, так как чем меньше нужно аппаратных ресурсов, тем быстрее осуществляется работа НС.

Чтобы удовлетворить эти взаимопротиворечивые требования, необходимо решить задачу оптимизации структуры НС. Для решения этой проблемы можно воспользоваться теоремой Хегт—

Нильсена [8], которая доказывает представимость функции многих переменных общего вида с помощью двухслойной НС прямого распространения с ограниченными функциями активации сигмоидного вида:

$$F(x) = 1/(1 + \exp(-x)). \quad (1)$$

Сигмоидальные функции являются монотонно возрастающими и имеют отличные от нуля производные на всей области определения. Эти характеристики обеспечивают правильное функционирование и обучение сети.

Двухслойная НС прямого распространения, имеющая N входов и K выходов, содержит скрытый слой, состоящий из M нейронов, и выходной, состоящий из K нейронов. Результатом работы НС является нелинейное преобразование N -мерного входного вектора X в K -мерный выходной вектор Y :

$$Y = F((X \cdot W) \cdot V), \quad (2)$$

где W — матрица весовых коэффициентов внутреннего слоя НС размерностью $[N \times M]$; V — матрица весовых коэффициентов выходного слоя НС размерностью $[M \times K]$; F — активационная функция.

Выражение (2) представляет собой матричную форму записи системы из K нелинейных уравнений. Путем подбора значений переменных W и V можно добиться примерного равенства значения Y_i одному из наперед заданных значений Y_j , поставленных в соответствие одному или нескольким входным векторам X_j . Этот процесс называется *обучением сети*. Итерации обучения НС повторяются до тех пор, пока для всех эталонных образов не будут достигнуты требуемые значения сигналов на выходах НС с ошибкой, меньшей некоторой заданной ошибки обучения δ . Таким образом, вероятность правильной классификации зависит от заданного значения δ и от числа нейронов в каждом слое НС.

Для уменьшения сложности сети в качестве входного массива данных предлагается вводить не значения интенсивности пикселей, как в [11], а предварительно полученные значения характеристик сегментов изображения.

Методика проведения сегментации изображения и вычисления характеристик полученных областей

Сегментацию изображения, представляющую собой двумерный массив значений интенсивности пикселей в терминах "оттенки серого", предлагается проводить по критерию однородности пикселей, так как вероятностные распределения значений интенсивности пикселей неизвестны, а любой другой выбор статистической метрики, как это проведено, например, в [5], исходит, опять же, из области значений интенсивности. Таким обра-

зом, на изображении присутствует объект (объекты), более темный, чем фон. Для нахождения границ темных объектов выбираются самые темные пиксели в качестве точки старта, затем наращивается область вокруг них, и определяются границы пятна, соответствующие значениям оттенка серого больше некоторого порога [9].

После проведения сегментации изображения вычисляются следующие характеристики полученных объектов.

- **Размер A** области, на которой наблюдается объект (в км²).
- **Периметр P** — длина в (км) границ объекта.
- **Сложность**, определяемая как

$$C = \frac{P}{2\sqrt{\pi A}}. \quad (3)$$

Этот параметр обычно принимает малые числовые значения для районов с простой геометрией и большие значения для сложных геометрических районов.

- **Протяженность S** . Этот параметр получают с помощью метода главных компонент [10] векторов, составляющими которых являются координаты пикселей, принадлежащих объекту. Если λ_1 и λ_2 являются двумя собственными числами, связанными с вычисляемой матрицей ковариации, и $\lambda_1 > \lambda_2$, то значение протяженности

$$S = \frac{100\lambda_2}{\lambda_1 + \lambda_2}. \quad (4)$$

Поясним алгоритм вычисления протяженности S темного объекта. Для этого рассмотрим изображение пятна на поверхности в терминах координат пикселей (рис. 1).

Координаты пикселей, принадлежащих пятну, представляют собой двумерный вектор a :

$$a = [2,2; 3,3; 4,3; 4,4; 5,4; 5,5; 6,5; 6,6; 7,6; 7,7].$$

		Координата x									
Координата y	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	1,10	2
	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9	2,10	3
	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9	3,10	4
	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9	4,10	5
	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9	5,10	6
	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9	6,10	7
	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9	7,10	8
	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9	8,10	9
	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9	9,10	10

Рис. 1. Протяженное и тонкое пятно на поверхности в терминах координат пикселей

Координата y	Координата x								
	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	2
	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	3
	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	4
	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	5
	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	6
	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	7
	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	8
	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	9
	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	10

Рис. 2. Темный объект, по форме стремящийся к окружности в терминах пикселей

Рассчитываем матрицу ковариации b вектора a :

$$b = \text{cov}(a).$$

Рассчитываем вектор c собственных чисел матрицы ковариации b :

$$c = \text{eig}(b).$$

Получаем следующие значения собственных чисел:

$$c = \begin{cases} 0,1298 & \text{— это значение собственного числа } \lambda_2; \\ 5,1369 & \text{— это значение собственного числа } \lambda_1. \end{cases}$$

Рассчитываем протяженность по формуле (4):

$$S = 100 \cdot 0,1298 / (0,1298 + 5,1369) = 2,4645.$$

На рис. 2 изображен темный объект, который по форме больше стремится к окружности.

Координаты пикселей, принадлежащих пятну, представляют собой двумерный вектор $a1$:

$$a1 = [2,2; 2,3; 2,4; 2,5; 3,2; 3,3; 3,4; 3,5; 4,2; 4,3; 4,4; 4,5; 4,6; 4,7; 4,8; 5,2; 5,3; 5,4; 5,5; 5,6; 5,7; 5,8; 6,4; 6,5; 6,6; 6,7; 6,8; 7,6; 7,7; 7,8].$$

Вычисляем ковариационную матрицу двумерного вектора $a1$:

$$b1 = \text{cov}(a1).$$

Вычисляем собственные числа ковариационной матрицы $b1$:

$$c1 = \begin{cases} 1,3269 & \text{— это значение собственного числа } \lambda_2 \\ 4,8950 & \text{— это значение собственного числа } \lambda_1. \end{cases}$$

Рассчитываем протяженность второго темного объекта на плоскости:

$$S1 = 100 \cdot 1,3269 / (1,3269 + 4,895) = 21,3263.$$

Сравнивая $S = 2,4645$ и $S1 = 21,3263$, видим, что числовое значение S мало для протяженных и тонких объектов и велико для объектов, чья форма близка к окружности.

- **Стандартное отклонение для объекта** — стандартное отклонение (в дБ) значений интенсивности пикселей, принадлежащих темному объекту.
- **Стандартное отклонение для фона** — стандартное отклонение (в дБ) значений интенсивно-

сти пикселей, принадлежащих району, окружающему темный объект.

- **Максимальная контрастность** — различие (в дБ) между средним значением интенсивности пикселей фона и самым малым значением интенсивности пикселей вне темного объекта.
- **Средняя контрастность** — различие (в дБ) между средним значением интенсивности пикселей фона и средним значением интенсивности пикселей темного объекта.
- **Максимальный градиент** — максимальный граничный градиент (в дБ) "фон—объект".
- **Средний градиент** — средний граничный градиент (в дБ) "фон—объект".
- **Стандартное отклонение градиента** — стандартное отклонение (в дБ) значений граничного градиента.

Полученные значения параметров объекта используются в качестве исходного вектора для НС, выходным значением которой является вероятность принадлежности анализируемого объекта к тому или иному классу.

Экспериментальная часть

Предлагаемый метод автоматизированной обработки изображений был применен для решения задачи обнаружения нефтяных пятен на радиолокационном изображении морской поверхности. Известно, что присутствие нефтяной пленки на поверхности моря уменьшает малые волны из-за возрастающей вязкости верхнего слоя и заметно уменьшает энергию обратного рассеивания сигнала, поэтому на радиолокационном изображении возникают темные области. Однако на изображении, полученном с помощью радиолокатора, могут возникнуть также затемнения, вызванные наличием локальных ветров низкой скорости над морской поверхностью или присутствием натуральных морских сликов.

Пример обрабатываемых радиолокационных изображений приведен на рис. 3. Автоматизиро-

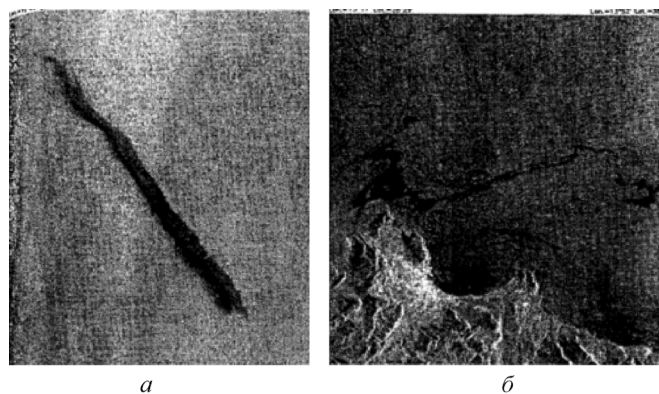


Рис. 3. Изображение нефтяного пятна (а) и природного слика (б)

ванная обработка радиолокационного изображения начинается с определения границ темного объекта. Анализируется его общая поверхность и строится гистограмма, в которой по оси абсцисс откладывается уровень интенсивности серого цвета на изображении, а на оси координат — число пикселей. Типичная форма такой гистограммы приведена на рис. 4. Гистограмма изображения нефтяного пятна имеет два пика, меньший из которых сосредоточен в районе среднего значения рассеяния темного объекта, больший — в районе среднего значения фона. Локальный минимум между пиками используется для фрагментации изображения.

Для радиолокационных изображений, классифицированных средствами наземного контроля, были рассчитаны граничные значения всех 11 параметров для нефтяных пятен и ветровых сликов, которые в дальнейшем использовались на этапе обучения НС. Установлено, что нефтяные пятна имеют меньшую сложность и более тонкую форму, чем ветровые слики. Средние значения гради-

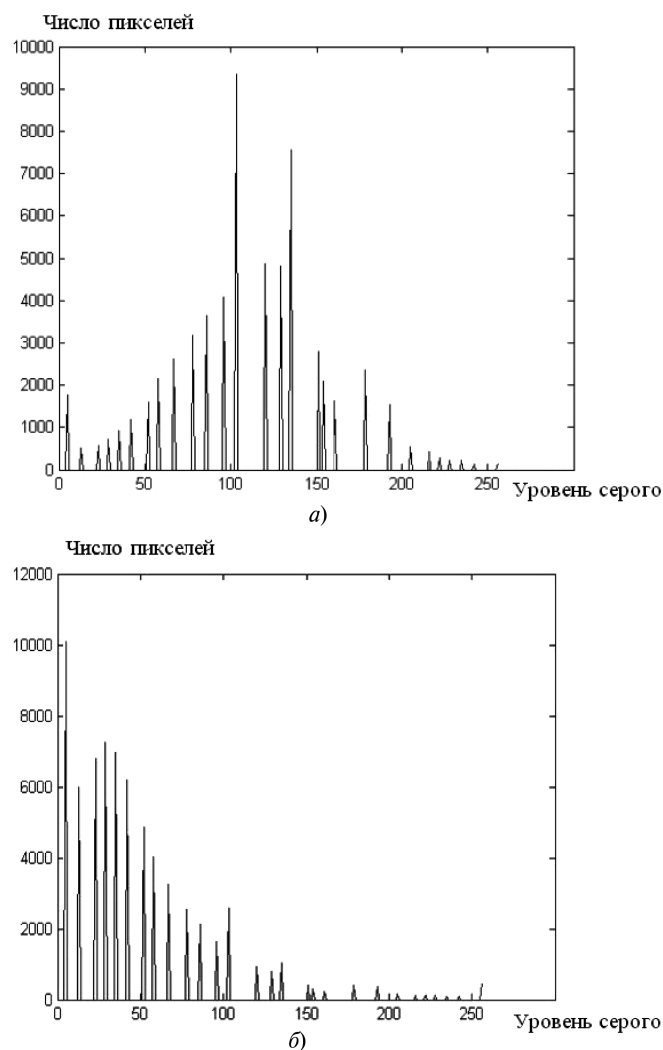


Рис. 4. Гистограмма нефтяного пятна (а) и природного слика (б)

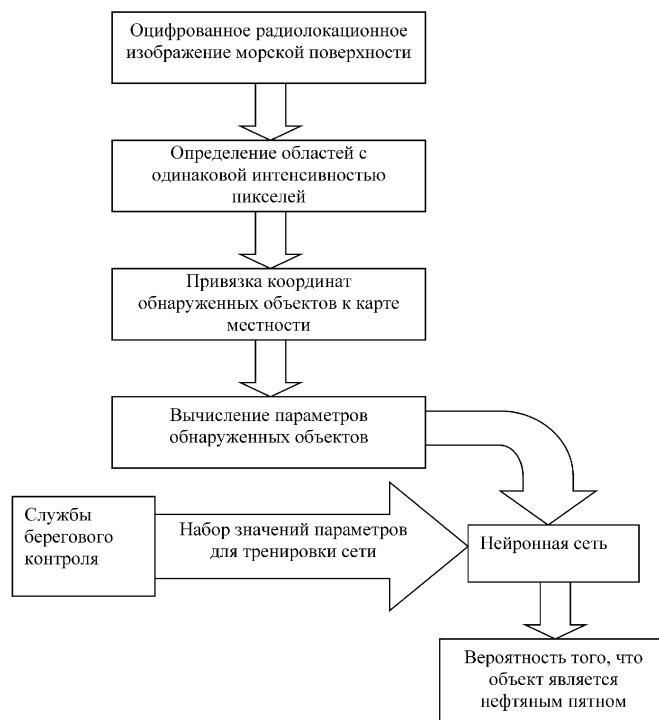


Рис. 5. Схема проведения эксперимента

ента вдоль границ у нефтяного пятна выше, чем у ветровых сликов, которые обычно протяженнее, чем нефтяные пятна.

Тренировка НС проводилась с использованием алгоритма обратного распространения, который применяет технику поиска градиента и итерационно подбирает весовые коэффициенты в сети для минимизации функции ошибки по среднему квадрату между желаемым и действительным значением выхода. Процедура итерации прекращалась, когда не наблюдалось значительных изменений в значении общей ошибки. Таким образом, обучение сети сводилось к решению задачи оптимизации функционала ошибки градиентным методом.

Схема проведения эксперимента изображена на рис. 5.

Результаты

Экспериментально установлено, что структура нейронной сети 11-8-8-1 с двумя слоями весов и сигмоидальными функциями активации показала лучшие характеристики как по точности классификации нефтяных пятен и ветровых сликов на радиолокационном изображении морской поверхности, так и по времени обучения (около 15 000 обучающих циклов).

Для обнаружения нефтяных пятен достаточным является разрешение 100×100 м.

Для реализации автоматизированной обработки изображений был создан программный

продукт средствами языка технических вычислений MatLab, который может быть применен в системах дистанционного мониторинга для получения информации в реальном времени о состоянии загрязнения акваторий портов и прибрежных зон.

Список литературы

1. **Chang H. S. and Kang K.** A compressed domain scheme for classifying block edge patterns // IEEE Trans. Image Process. Feb. 2005. Vol. 14. N 2. P. 145–151.
2. **Moody A., Johnson D. M.** Land-surface phonologies from AVHRR using the discrete Fourier transform // Remote Sens. Environ. 2001. Vol. 75. P. 305–323.
3. **Xu Y., Weaver J. B., Healy D. M.** Wavelet transform domain filters: A spatially selective noise filtration technique // IEEE Trans. Image Process. Nov. 1994. Vol. 3. N 6. P. 747–757.

4. **Chang Y.-L., Li X.** Adaptive image region growing // IEEE Trans. Image Process. Nov. 1994. Vol. 1.3. N 6. P. 868–872.
5. **Lee S., Crawford M. M.** Unsupervised Multistage Image Classification using hierarchical Clustering with a Bayesian Similarity Measure // IEEE Trans. Image Process. March 2005. Vol. 14. N 3. P. 313–321.
6. **Оссовский С.** Нейронные сети для обработки информации: Пер. с польск. М.: Финансы и статистика, 2004. 496 с.
7. **Уоссерман Ф.** Нейрокомпьютерная техника: теория и практика: Пер. с англ. М.: Мир, 1992.
8. **Hecht-Nielsen R.** Theory of the Back Propagation Neural Network // Int. Joint Conf. on Neural Networks, Sheraton Washington Hotel, Washington D. C. June 18–22, 1989. Vol. 1. P. 593–606.
9. **Hu Q., Hou Z., Nowinski W. L.** Supervised Range-Constrained Thresholding // IEEE Trans. Image Process. Jan. 2006. Vol. 15. N 1. P. 228–240.
10. **Глинский В. В., Ионин В. Г.** Статистический анализ. М.: Инфра-М, 2002. 323 с.
11. **Нейрокомпьютеры** в системах обработки изображений / Под ред. А. И. Галушкина. М.: Радиотехника, 2003. 192 с.

УДК 519.632

Н. И. Корсунов, д-р техн. наук, проф.,
А. А. Юдин,
Белгородский государственный
технологический университет

Решение задач теплопроводности с использованием нейросетевых компьютерных технологий

Рассматривается возможность получения приближенного решения дифференциального уравнения теплопроводности с помощью решетчатых нейронных сетей на основе реализации метода конечных разностей.

Введение

При рассмотрении процессов распространения теплоты, фильтрации газа и жидкости в пористой среде (например, фильтрации нефти и газа в подземных песчаниках) приходится сталкиваться с решением задач теплопроводности. Уравнение теплопроводности является параболическим уравнением, поэтому для его решения можно использовать методы решения дифференциальных уравнений в частных производных с переменными коэффициентами.

Получение точного аналитического решения дифференциальных уравнений в частных производных в общем случае невозможно или, по крайней мере, сильно затруднено. По этой причине используются методы приближенного численного решения подобных задач, к которым относятся метод конечных разностей и метод конечных элементов.

Формулировка задач

Плоская задача теплопроводности формулируется следующим образом. Дано тело произвольной формы, размеры которого вдоль оси Z являются пренебрежимо малыми по сравнению с другими линейными размерами (рис. 1). В таком случае температуру вдоль оси Z можно считать не изменяющейся.

Материал, из которого состоит тело, является однородным и изотропным. Обозначим его коэффициент теплоемкости c , коэффициент теплопроводности λ и плотность ρ .

Задана некоторая функция $W(x, y)$, определяющая внутренние источники теплоты. Необходимо найти конечное равновесное распределение температур внутри заданного тела – некоторую неизвестную непрерывную функцию $U = U(x, y)$, которая внутри заданного тела удовлетворяла бы граничным условиям на границе тела, задаваемым в одной из следующих форм:

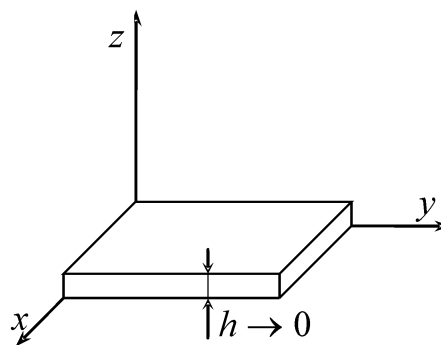


Рис. 1. Тело, соответствующее формулировке задачи

- фиксированная температура границы тела:

$$U|_{\Gamma} = f(x, y); \quad (1)$$

- фиксированный тепловой поток на границе тела:

$$\frac{\partial U}{\partial n}\bigg|_{\Gamma} = f(x, y); \quad (2)$$

- теплообмен тела с окружающей средой постоянной температуры

$$\frac{\partial U}{\partial n}\bigg|_{\Gamma} + k(U(x, y) - \theta(x, y)) = 0, \quad (3)$$

где $\frac{\partial U}{\partial n}$ — производная по направлению нормали n ;

$\theta(x, y)$ — температура окружающей среды в данной точке плоскости.

Дифференциальное уравнение теплопроводности для двумерного температурного поля T имеет следующий вид (при условии, что $\lambda = \text{const}$ — тело однородно):

$$\begin{aligned} \text{ср} \frac{\partial T(x, y, \tau)}{\partial \tau} = \lambda \frac{\partial^2 T(x, y, \tau)}{\partial x^2} + \\ + \lambda \frac{\partial^2 T(x, y, \tau)}{\partial y^2} + W(x, y), \end{aligned} \quad (*)$$

где τ — координата по времени; x, y — координаты на плоскости.

Данное уравнение может быть решено аналитически только для ограниченного набора тел: одномерного бесконечного, полубесконечного или конечного стержня, круга, кольца или прямоугольника в двумерном пространстве, параллелепипеда, цилиндра или сферы в трехмерном пространстве и некоторых других. В связи с этим применяются приближенные численные методы решения дифференциальных уравнений.

Как правило, для получения решения, приближенного к точному, используются вариации так называемых сеточных методов или метода конечных элементов, основывающихся на выделении набора узловых точек, разбиении с их помощью исходного тела на набор простых элементов и приближении искомой функции более простой, в частности, кусочно-линейной, а ее производных — конечно-разностными выражениями. В конечном итоге решение такой приближенной задачи сводится к решению системы линейных уравнений.

В настоящее время значительного выигрыша в скорости при решении подобных задач можно добиться за счет использования параллельных вычислений.

Существующие алгоритмы распараллеливания требуют длительной предварительной подготовки перед началом работы.

Нейронные сети в самой своей структуре содержат распараллеливание и не требуют предварительной подготовки перед началом работы. Для работы нейронной сети не нужно создавать специальные алгоритмы, достаточно задать структуру сети, а за счет эффективного распараллеливания наблюдается выигрыш по скорости. По этой причине наиболее целесообразно в подобных задачах использовать нейронные сети.

Поставленная задача: исследовать возможность приближенного решения задачи с использованием решетчатых нейронных сетей. Проанализировать условия, при которых сеть выбранного вида будет устойчива. Рассмотреть вопрос точности приближения искомой функции нейронной сетью и скорости сходимости.

Исходные данные: дана решетчатая нейронная сеть. В отличие от сети Хопфилда, строится по минимальной погрешности. Сеть относится к классу конструируемых, веса находятся при конструировании сети. Линейная функция активации выбирается априори.

Требуется: определить архитектуру сети — число ячеек, их связи, число слоев решетки, весовые коэффициенты.

Выбор структуры сети

Далее будем применять следующие обозначения:

N — число нейронов в нейронной сети;

K — число узлов в сетке;

w_{ij} — вес связи, идущей от i -го нейрона к j -му;

i, j, k — узлы (нейроны) сети;

Ω — двумерная область размерностью $N \times N$ с равномерным шагом по обеим координатам.

Рассмотрим полносвязную нейронную сеть с линейными активационными функциями нейронов с единичным коэффициентом, для которой выполняются следующие условия:

1. $w_{ij} = w_{ji}$ — сеть симметрична. (4)

2. Существует один или более нейронов с константной функцией активации, определяющие граничные условия задачи. В частности, узлу, в котором действует граничное условие вида (1), соответствует нейрон с константным выходом $f = f(x, y)$; вида (2) — "обычный" нейрон с дополнительным входом единичного смещения весом $f = f(x, y)$; и наконец, вида (3) — два нейрона с суммарным выходом $f = u(x, y) + k(u(x, y) - \theta(x, y)) = (k + 1)u - k\theta$, где u — выход первого нейрона, θ — выход второго нейрона.

3. Существует как минимум один нейрон, соответствующий краевому условию вида (2) или (3), или, в противном случае, выполняется равенство

$$\sum_i w_{0i} = 0. \quad (5)$$

Физический смысл этого условия состоит в том, что суммарный тепловой поток через поверхность тела без внутренних источников теплоты в равновесном состоянии будет равен нулю.

Вес связи w_{ij} имеет следующий смысл, в соответствии с терминологией Хопфилда:

- при $i < j$ — связь прямая;
- при $i \geq j$ — связь обратная.

Каждый нейрон (за исключением дополнительных нейронов, использованных для представления граничных условий вида (3)) соответствует одному углу сетки разбиения тела. Выходное значение нейрона соответствует значению искомой функции $U = U(x, y)$ в данном узле. Предполагаем, что между узлами функция $U(x, y)$ может быть приближена линейной.

Разобьем временной интервал на m отрезков длиной τ , где τ — шаг дискретизации по времени.

Для решения используем многослойную ячеистую нейронную сеть. Структуру такой сети можно описать следующим образом.

- Сеть разделена на m слоев, каждый из которых содержит n -мерную прямоугольную решетку нейронов, имеющих линейные функции активации $f(x) = x$. Каждый нейрон сети соответствует узлу дискретизации в области Ω ; i -й слой соответствует моменту времени $t_i = i\tau$. Каждый нейрон имеет $2n$ связей с непосредственными соседями внутри слоя, обратную связь и две связи с нейронами с тем же номером в предыдущем и последующем слоях (за исключением нейронов первого и последнего слоя, у которых одна из этих связей будет отсутствовать). Пример двухслойной нейронной сети при $n = 2$ приведен на рис. 2.

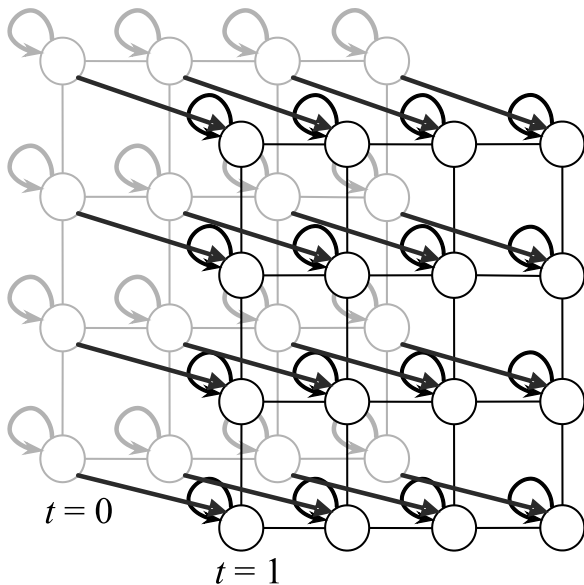


Рис. 2. Двухслойная нейронная сеть при $n = 2$

- Обозначим номер нейрона в слое t с n -мерным вектором координат X внутри слоя как $L(X, t)$.
- Выходное значение нейрона с номером I обозначим как f_i .
- Назовем вектор X *предшествующим* вектору Y по k -й координате, если $x_i = y_i \forall i \neq k, x_k = y_k - h$, и *последующим*, если $x_i = y_i \forall i \neq k, x_k = y_k + h$. Обозначим номер нейрона, предшествующего i -му по j -й координате как $P(i, j)$; следующего за i -м — как $S(i, j)$; множество номеров нейронов, соседних с нейроном i , — как $B(i)$.
- Обозначим значения функции $k(r, t)$ как k_i .
- Введем вспомогательную величину

$$\bar{k}_{L(r, t)} = \frac{\tau}{h^2} \sum_{i=1}^n \left(\frac{2k_1}{k_1 + k_{L(r, t)}} + \frac{2k_2}{k_2 + k_{L(r, t)}} \right);$$

$$k_1 = k_{L(P(r, i), t)}, k_2 = k_{L(S(r, i), t)}; \quad (6)$$

заметим, что при $k = \text{const}$ $\bar{k} = \frac{\tau}{h^2} 2n$.

Для сохранения устойчивости сети шаги дискретизации τ и h , согласно [1], должны удовлетворять неравенству

$$\tau \leq \frac{1}{2a} h^2, \text{ где } a — \text{коэффициент в уравнении}$$

$$\text{теплопроводности } \frac{\partial f}{\partial t} = a^2 \frac{\partial^2 f}{\partial x^2}.$$

- Определим вес обратных связей w_{ii} следующим образом:

$$w_{ii} = b - \frac{1-b}{k_i \bar{k}_i}, \quad (7)$$

где b — "настроечный" коэффициент сети, задаваемый пользователем. Значение b должно лежать в диапазоне $(0, 1)$. Позднее будет показано, что значение этого коэффициента влияет на скорость сходимости сети, но не на значения выходных сигналов нейронов в стабильном состоянии.

- Вес связей внутри слоя w_{ij} определим как

$$w_{ij} = \begin{cases} \frac{1-b}{\bar{k}_i} \cdot \frac{2k_i}{k_j + k_i} \cdot \frac{\tau}{h^2}, & i \in B(j); \\ 0, & i \notin B(j). \end{cases} \quad (8)$$

- Вес связей между слоями

$$\begin{cases} w_{L(X, i), L(Y, i+1)} = 0, & X \neq Y; \\ w_{L(X, i), L(X, i+1)} = \frac{1-b}{k_{L(X, i)} \bar{k}_{L(X, i)}}, \end{cases} \quad (9)$$

где X, Y — векторы с n координатами, описывающие положение нейрона внутри слоя; k — значения коэффициента, зависящего от свойств области (например, от коэффициента теплопроводности среды или диффузии) в точке Ω , соответствующей узлу дискретизации с координатами X .

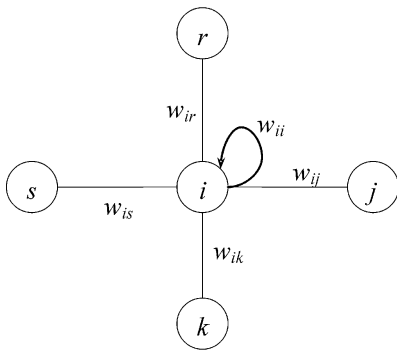


Рис. 3. Пример узла сетки с четырьмя соседними нейронами

Заметим, что в отличие от связей внутри слоя межслойные связи являются однонаправленными.

- Активационные функции нейронов — линейные: $f(x) = x$.
- Зададим граничные условия вида $f(r, t) = \mu(r, t)$ следующим образом: активационные функции нейронов, соответствующих узлам, в которых задано граничное условие, являются константными:

$$f_i(x) = \mu(r_i, t_i) = \mu_i = \text{const.} \quad (10)$$

- Зададим граничные условия вида $f'_i(r, t) = \mu'_i(r, t) = d_i$ следующим образом: активационные функции нейронов, соответствующих узлам, в которых задано граничное условие, имеют вид

$$f_i(x) = K \left(x + v_i \frac{1}{2n} \right);$$

$$w_{ii} = \frac{b}{K}, \quad (11)$$

где $K = \frac{2n}{|B(i)|}$ — коэффициент, зависящий от числа соседних узлов.

- Граничные условия вида $f'(r, t) = \alpha[u(r, t) - \theta(t)]$, где $\theta(t)$ — температура окружающей среды, зададим следующим образом: активационные функции нейронов, соответствующих узлам, в которых задано граничное условие, имеют вид

$$f_i(x) = x + \alpha[x - \theta(t)]. \quad (12)$$

Для большей наглядности приведем на рис. 3 отдельный узел нейронной сети с номером i , а также соседние с ним нейроны r, s, j, k . Через w обозначены веса связей между нейронами в пределах одного временного слоя.

Алгоритм работы нейронной сети

Шаг 1. Начало.

Шаг 2. Инициализация всех нейронов каждого слоя сети. Вычисление их весов обратной связи.

Шаг 3. Задание непосредственных соседей для каждого нейрона сети в пределах одного слоя. Вычисление весов связей каждого нейрона с соседними нейронами.

Шаг 4. Задание в качестве соседа каждого нейрона во всех слоях, кроме первого, нейрона с тем же номером в предыдущем слое. Вычисление связей между слоями.

Шаг 5. Пока выходные значения всех нейронов не перейдут в стабильное состояние (разница между новым и старым выходным сигналами больше заданной точности ϵ), выполнять шаг 6.

Шаг 6. Для каждого слоя t сети выполнять шаги 7, 8.

Шаг 7. Вычисление входного состояния для каждого нейрона i слоя t сети:

$$U_{\text{вх } i, t} = U_{\text{вых } i(t)} w_{i, i_t} + \sum_{j=1}^N (U_{\text{вых } j_t} w_{i, j_t}) + U_{\text{вых } i(t-1)} w_{i(t), i(t-1)},$$

причем цикл осуществляет обход непосредственных соседей нейрона i , для первого слоя последнее слагаемое равно 0.

Шаг 8. Определение выходного сигнала каждого нейрона слоя $U_{\text{вых } i(t)} = f(U_{\text{вх } i(t)})$.

Шаг 9. Конец.

В алгоритме на шаге 5 термин стабильности заимствован у Хопфилда и характеризует сходимость к решению задачи за определенное число шагов.

Математическое обоснование

Гипотеза: мы можем получить приближенное решение поставленной задачи как стабильное состояние нейронной сети описанного вида.

Предположение: решение произвольной одномерной задачи описанного вида может быть приближено с помощью нейронной сети предложенного вида.

Теплопроводность возникает при наличии разности температур, вызванной какими-либо внешними причинами. При этом согласно второму закону теплопроводности происходит передача тепла из областей с более высокой температурой в области с более низкой температурой.

С течением времени температура в каждой точке тела принимает определенное установившееся значение. Это подтверждается как математически решением задачи теплопроводности, так и на практике.

Выходное значение нейрона соответствует значению искомой функции в узле сетки.

Область разбита на элементарные объемы и каждому из них соответствует один нейрон.

Выходное значение нейрона будем считать совпадающим с текущим значением потенциальной энергии.

В замкнутой одномерной системе зададим такт работы сети вычисленное число изменения энергии между двумя потенциалами:

$$\Delta q_1 = -\Delta q_2 = (f_1^{t+1} - f_1^t) = (1-b)f_2^t - bf_1^t - f_1^t = (1-b)(f_2^t - f_1^t). \quad (13)$$

Если рассмотреть произвольное число соседних нейронов, то

$$\Delta q_j = (1-b) \sum_{i \neq j} (F_i^t - F_j^t). \quad (14)$$

Полная энергия представляет собой сумму выходных значений всех нейронов. Если на вход нейрона i подается величина T_i , а выход сети Q_i , то устанавливается соответствие нейронной сети физическому смыслу, определяемому законом сохранения энергии. Так как активационная функция нейрона принята линейной с единичным коэффициентом, то сумма весов выходных связей каждого нейрона также равна единице.

Вследствие этого на каждом шаге будет выполняться равенство

$$\sum_{i=1}^n \sum_{j=1}^n Q_i^{t+1} w_{ij} = \sum_{i=1}^n T_i^t. \quad (15)$$

Воспользовавшись тем, что

$$\sum_{i=1}^n Q_i^{t+1} w_{ij} = T_j^{t+1},$$

и перегруппировав слагаемые в (15), получим

$$\begin{aligned} \sum_{i=1}^n T_i^{t+1} &= \sum_{j=1}^n \sum_{i=1}^n Q_i^{t+1} w_{ij} = \\ &= \sum_{i=1}^n \sum_{j=1}^n Q_i^{t+1} w_{ij} = \sum_{i=1}^n T_i^t. \end{aligned}$$

Таким образом, в ячеистой нейронной сети, соответствующей замкнутой системе, суммарное значение выходных величин остается постоянным, что соответствует физическому смыслу модели, определяемому выражением (14) в стабильном состоянии.

Поскольку для сетки в каждом i -м узле дискретизации в отделимости

$$f(x) = |f_{i-1} - 2f_i + f_{i+1}| < \varepsilon \quad \forall i \in (1, N-1) \quad (16)$$

и в ячеистой нейронной сети в стабильном состоянии

$$f_i^{t+1} = w_{ii} f_i^t + w_{(i-1)i} f_{i-1}^t + w_{(i+1)i} f_{i+1}^t;$$

$$(w_{ii} - 1) f_i^t + w_{(i-1)i} f_{i-1}^t + w_{(i+1)i} f_{i+1}^t = 0,$$

справедливо соотношение

$$2f_i^t = \frac{w_{(i-1)i}}{w_{ii}-1} f_{i-1}^t + \frac{w_{(i+1)i}}{w_{ii}-1} f_{i+1}^t. \quad (17)$$

При выборе весов w_{ii} , $w_{(i-1)i}$, $w_{(i+1)i}$ таким образом, чтобы выполнялось равенство $\frac{w_{(i-1)i}}{w_{ii}-1} =$

$\frac{w_{(i+1)i}}{w_{ii}-1} = 1$, выражение (17) обращается в (16).

Заметим, что стабильное состояние сети удовлетворяет конечно-разностному представлению дифференциального уравнения

$$\begin{aligned} \frac{f(x, t) - f(x, t + \tau)}{\tau} = \\ = k_x \left(\frac{f(x-h, t) - 2f(x, t) + f(x+h, t)}{h^2} \right) \end{aligned} \quad (18)$$

для $k = \text{const}$ или для $k = k(r, t)$:

$$\begin{aligned} \frac{f(x, t) - f(x, t + \tau)}{\tau} = \frac{1}{h} \left(a(x+h, t) \times \right. \\ \left. \times \frac{f(x+h, t) - f(x, t)}{h} - a(x, t) \frac{f(x, t) - f(x-h, t)}{h} \right), \end{aligned} \quad (19)$$

где $a(x, t)$ — разностный коэффициент теплопроводности, вычисляемый по формуле

$$a(x, t) = \frac{2k(x+h, t)k(x, t)}{k(x+h, t) + k(x, t)}, \quad (20)$$

вследствие того, что равенство

$$\begin{aligned} f_{L(X, t)} &= w_{L(X, t), L(X, t)} \times \\ &\times f_{L(X, t)} + \sum_{i=1}^n (w_{L(P(X, i), t), L(X, t)} f_{L(P(X, i), t)} + \\ &+ w_{L(S(X, i), t), L(X, t)} f_{L(S(X, i), t)} + \\ &+ w_{L(X, t-1), L(X, t)} f_{L(X, t-1)}), \end{aligned} \quad (21)$$

описывающее стабильное состояние нейронной сети, может быть преобразовано к виду, эквивалентному разностному представлению:

$$\begin{aligned} f_{L(X, t)} &= \left(b - \frac{1-b}{k\bar{k}} \right) f_{L(X, t)} + \\ &+ \frac{1-b}{\bar{k}} \left(\frac{\tau}{h^2} \right) \sum_{i=1}^n \left(\frac{2k_{L1}}{k_{L(X, t)} + k_{L1}} f_{L1} + \right. \\ &+ \left. \frac{2k_{L2}}{k_{L(X, t)} + k_{L2}} f_{L2} \right) + \frac{1-b}{k\bar{k}} f_{L(X, t-1)}; \\ f_{L(X, t)} - f_{L(X, t-1)} &= -k\bar{k} f_{L(X, t)} + \\ &+ k_{L(X, t)} \left(\frac{\tau}{h^2} \right) \sum_{i=1}^n \left(\frac{2k_{L1}}{k_{L(X, t)} + k_{L1}} f_{L1} + \right. \end{aligned}$$

$$\begin{aligned}
& + \frac{2k_{L2}}{k_{L(X,t)} + k_{L2}} f_{L2}) f_{L(X,t)} - f_{L(X,t-1)} = \\
& = k_{L(X,t)} \left(\frac{\tau}{h^2} \right) \sum_{i=1}^n \left(\frac{2k_{L1}}{k_{L(X,t)} + k_{L1}} (f_{L1} - f_{L(X,t)}) - \right. \\
& \quad \left. - \frac{2k_{L2}}{k_{L(X,t)} + k_{L2}} (f_{L(X,t)} - f_{L2}) \right).
\end{aligned}$$

Такое конечно-разностное представление имеет точность второго порядка по h и первого — по τ . Отметим, что такое представление не единственно; существует целый ряд разностных представлений, на которых основаны различные расчетные схемы метода конечных разностей. Выбор именно этой формы разностного представления обусловлен тем, что при его использовании ячеистая нейронная сеть имеет минимальное число связей, составляющих прямоугольную решетку, в отличие от большинства других, требующих ввода дополнительных "диагональных" связей. Более простая структура связей позволяет упростить выражения, используемые в доказательствах выдвигаемых утверждений, сохраняя возможность в дальнейшем обобщить полученные результаты на ячеистые нейронные сети с более сложной структурой.

Лемма 1. Стабильное состояние нейронной сети предложенного вида, содержащей N нейронов, соответствует решению системы линейных уравнений порядка N .

Доказательство. Проведем доказательство по индукции.

Рассмотрим нейронную сеть, состоящую из двух нейронов A, B (соответствует системе из двух элементарных объемов), источника "единичного смещения" O и связей между ними. Тогда в ней будет присутствовать шесть связей с весами $w_{AA}, w_{AB}, w_{OA}, w_{BB}, w_{BA}, w_{OB}$. В этом случае выходные сигналы в стабильном состоянии определяются следующими выражениями:

$$\begin{cases} X_A = X_A w_{AA} + X_B w_{BA} + 1 \cdot w_{OA}; \\ X_B = X_A w_{AB} + X_B w_{BB} + 1 \cdot w_{OB} \end{cases}$$

или, после проведения к более привычному виду:

$$\begin{cases} X_A(w_{AA} - 1) + X_B w_{BA} + w_{OA} = 0; \\ X_A w_{AB} + X_B(w_{BB} - 1) + w_{OB} = 0. \end{cases}$$

Очевидно, мы получили систему двух линейных уравнений с двумя неизвестными, решением которой являются выходные сигналы описанной сети в стабильном состоянии.

Второй шаг индукции: пусть некоторой нейронной сети, для которой доказано утверждение леммы 1, соответствует система уравнений

$$AX = B. \quad (22)$$

При добавлении в систему еще одного нейрона получаем, что

$$\begin{bmatrix} A & w \\ w & x \end{bmatrix} \begin{bmatrix} X \\ x \end{bmatrix} = \begin{bmatrix} B \\ b \end{bmatrix}. \quad (23)$$

Таким образом, при добавлении нейрона мы вновь получаем нейронную сеть, для которой справедливо доказываемое утверждение.

Лемма 2. "Изолированная" (т. е. не содержащая нейронов, задающих граничные условия) нейронная сеть предложенного вида стабильна.

Нестабильность нейронной сети может быть вызвана одной из двух причин:

- существование в сети циклов с положительными обратными связями. Эта проблема характерна для нейронных сетей, использующих неограниченные активационные функции нейронов;
- "осцилляция" сети, т. е. ее колебание между несколькими состояниями. Эта проблема, как правило, характерна для синхронных нейронных сетей, в которых все нейроны изменяют свое выходное значение одновременно.

В силу условия (4), наложенного на веса связей сети, в сети не может существовать связей с весом, большим единицы. Учитывая также то, что активационные функции нейронов являются линейными с коэффициентом 1, мы получаем, что суммарный выходной сигнал нейронов не будет превышать суммарный входной сигнал. Таким образом, мы можем гарантировать отсутствие положительных обратных связей в сети.

Рассмотрим разность между минимальным и максимальным выходными значениями нейронов на некотором шаге. Выходной сигнал

$$X^{(t+1)} = WX^{(t)} + B.$$

Учитывая условия (4) и (5), получаем, что

$$\sum_{i=1}^N w_{ij} = 1 \quad \forall i \in (1, K).$$

Тогда, считая, что $\min X_i = X_l$, получаем, что новое минимальное выходное значение нейрона будет удовлетворять следующему выражению:

$$\min_i X_i = \min_i (w_{li} X_l + \sum_{j \neq l} (w_{jl} X_j)).$$

В свою очередь, очевидно, что $X_i \geq X_l$, а следовательно,

$$\begin{aligned} \min_i X_i & \geq \min_i (w_{li} X_l + \sum_{j \neq l} (w_{jl} X_j)) = \\ & = \min_i (\sum w_{jl}) X_l = X_l. \end{aligned}$$

Пользуясь аналогичным рассуждением, получаем, что

$$\begin{aligned} \max_i X_i & \leq \max_i (w_{li} X_l + \sum_{j \neq l} (w_{jl} X_j)) = \\ & = \max_i (\sum w_{jl}) X_l = X_l. \end{aligned}$$

Лемма 3. В произвольной нейронной сети предложенного вида суммарное значение выходных сигналов нейронной сети ограничено.

Рассмотрим все возможные варианты.

1. Нейронная сеть, соответствующая замкнутой системе, — суммарное значение постоянно:

$$\begin{aligned}\sum_{j=1}^n F_j^{(t+1)} &= \sum_{j=1}^n \sum_{i=1}^n F_j^{(t)} w_{ij} = \\ &= \sum_{i=1}^n \sum_{j=1}^n F_j^{(t)} w_{ij} = \sum_{i=1}^n F_i^{(t)} = \text{const.}\end{aligned}$$

2. Нейронная сеть, соответствующая системе с граничными условиями второго рода:

$$\begin{aligned}\sum_{j=1}^n F_j^{(t+1)} &= \sum_{j=1}^n \left(\sum_{i=1}^n (F_i^{(t)} w_{ij}) + O_j \right) = \\ &= \sum_{i=1}^n \sum_{j=1}^n F_i^{(t)} w_{ij} + \sum O_j,\end{aligned}$$

где O_j — выходное значение j -го нейрона.

Учитывая условие (6), получаем, что

$$\sum_{i=1}^n \sum_{j=1}^n F_i^{(t)} w_{ij} + \sum O_j = \sum_{i=1}^n \sum_{j=1}^n F_i^{(t)} w_{ij} = \text{const.}$$

3. Нейронная сеть, соответствующая системе с граничными условиями первого рода:

$$\begin{aligned}\sum_{i=1}^k F_i^{(t+1)} &= \sum_{i=1}^k \left(\left(\sum_{j=1}^k F_j^{(t)} w_{ij} \right) + \right. \\ &+ \left. \left(\sum_{j=k+1}^n F_j^{(t)} w_{ij} \right) \right) = \sum_{i=1}^k \sum_{j=1}^k F_j^{(t)} w_{ij} + C = \\ &= \sum_{j=1}^k \left(\sum_{i=1}^k w_{ij} \right) F_j^{(t)} + C.\end{aligned}$$

Обозначим $\sum_{i=1}^k w_{ij}$ как α_j , заметим, что, так как $k < n$, то $\forall i: \alpha_i < 1$. Тогда равновесие будет достигаться при выполнении условия

$$\sum (1 - \alpha_i) F_i^{(t)} = C.$$

Лемма 4. Произвольная нейронная сеть предложенного вида стабильна.

Заметим, что нейроны с константным выходом не могут усиливать сигнал и, значит, не могут быть причиной появления положительных обратных связей. Минимальное выходное значение нейрона X_i можно записать в следующей форме:

$$\begin{aligned}\min_i X_i &= \min_i (w_{li} X_l + \sum_{j \neq l} (w_{jl} X_j) + b_i) \geq \\ &\geq \min_i (w_{li} X_l + \sum_{j \neq l} (w_{jl} X_l) + b_i) \geq \min_l (\sum w_{ji}) X_l + \\ &+ \min_i b_i = X_l + \min_i b_i.\end{aligned}$$

Докажем соответствие данной системы известным физическим законам.

Закон теплопроводности Фурье:

$$q = -\lambda \text{grad } T,$$

где q — поток теплоты, проходящий в единицу времени через единицу поверхности; λ — коэффициент теплопроводности; T — температура.

Рассмотрим простейший случай замкнутой одномерной системы, содержащей два элементарных объема. За один "такт" работы сети вычисленное количество передаваемой между элементарными объемами энергии составит

$$q = \frac{\frac{E_B}{2} - \frac{E_A}{2}}{\Delta x} = k \frac{\Delta E}{\Delta x} \approx -k \frac{\delta E}{\delta x}.$$

Энергия моделируемой системы может быть вычислена как сумма входных значений всех нейронов модели. Заметим, что, так как мы приняли активационную функцию нейрона линейной с единичным коэффициентом и сумма весов выходных связей каждого нейрона равна 1, то на каждом шаге будет выполняться равенство

$$\sum_{i=1}^n \sum_{j=1}^n O_i^{(t+1)} w_{ij} = \sum_{i=1}^n I_i^{(t)},$$

где $I_i^{(t)}$ — входное значение нейрона в момент времени t ; $O_i^{(t)}$ — выходное значение нейрона в момент времени t .

Так как

$$\sum_{i=1}^n O_i^{(t+1)} w_{ij} = I_j^{(t+1)},$$

то, перегруппировав слагаемые, получаем:

$$\begin{aligned}\sum_{i=1}^n I_j^{(t+1)} &= \sum_{j=1}^n \sum_{i=1}^n O_i^{(t+1)} w_{ij} = \\ &= \sum_{i=1}^n \sum_{j=1}^n O_i^{(t+1)} w_{ij} = \sum_{i=1}^n I_i^{(t)}.\end{aligned}$$

Таким образом, в решетчатой нейронной сети, соответствующей замкнутой системе, суммарное значение выходных величин нейронов будет оставаться постоянным, что соответствует физическому смыслу модели.

В системе, не изолированной от внешнего мира, очевидно, тепловой поток через ее границу будет ненулевым, и тепловая энергия системы будет увеличиваться или уменьшаться.

Утверждение 1. Выходные сигналы нейронов в стабильном состоянии нейронной сети не зависят от величины b .

Доказательство. Используем выражение для равновесного состояния нейрона.

$$f_{L(X, t)} = \left(b - \frac{1-b}{k\bar{k}}\right)f_{L(X, t)} + \frac{1-b}{\bar{k}}\left(\frac{\tau}{h^2}\right) \sum_{i=1}^n (f_{L1} + f_{L2}) + \frac{1-b}{k\bar{k}}f_{L(X, t-1)},$$

где $L1 = L(P(X, i), t)$; $L2 = L(S(X, i), t)$.

Преобразуем его:

$$(1-b)f_{L(X, t)} = -\frac{1-b}{2kn(\tau/h^2)}f_{L(X, t)} + \frac{1-b}{2n(\tau/h^2)}\left(\frac{\tau}{h^2}\right) \sum_{i=1}^n (f_{L1} + f_{L2}) + \frac{1-b}{2kn(\tau/h^2)}f_{L(X, t-1)}.$$

Так как по определению $b \neq 1$, можно разделить левую и правую части равенства на $1-b$:

$$f_{L(X, t)} = -\frac{1}{2kn(\tau/h^2)}f_{L(X, t)} + \frac{1}{2n} \sum_{i=1}^n (f_{L1} + f_{L2}) + \frac{1}{2kn(\tau/h^2)}f_{L(X, t-1)}.$$

Истинность утверждения очевидна из того, что в преобразованном выражении величина b не участвует.

Рассмотрим, каким образом величина "настроенного" коэффициента b влияет на скорость сходимости нейронной сети.

Новое выходное состояние нейронов сети вычисляется следующим образом:

$$f_{L(X, t)} = \left(b - \frac{1-b}{k\bar{k}}\right)f_{L(X, t)} + \frac{1-b}{\bar{k}}\left(\frac{\tau}{h^2}\right) \sum_{i=1}^n (f_{L1} + f_{L2}) + \frac{1-b}{k\bar{k}}f_{L(X, t-1)}.$$

При уменьшении значения коэффициента b выходное состояние i -го нейрона сети изменяется быстрее за счет того, что значение множителя $\frac{1-b}{\bar{k}}$ при $b \rightarrow 0$ возрастает, и при $b = 0$ множитель $\frac{1-b}{\bar{k}} = \frac{1}{\bar{k}}$. Очевидно, что при этом каждый соседний нейрон сильнее влияет на изменение входного состояния текущего нейрона.

Исходя из вышесказанного можно сделать вывод, что при уменьшении значения коэффициента b в области $[0, 1)$ увеличивается скорость сходимости сети, и наоборот.

При значении коэффициента $b = 1$, новое входное значение каждого нейрона сети (кроме

нейронов, соответствующих граничным условиям) будет иметь вид:

$$f_{L(X, t)} = \left(1 - \frac{1-b}{k\bar{k}}\right)f_{L(X, t)} + \frac{1-b}{\bar{k}}\left(\frac{\tau}{h^2}\right) \times \sum_{i=1}^n (f_{L1} + f_{L2}) + \frac{1-b}{k\bar{k}}f_{L(X, t-1)} = f_{L(X, t)}.$$

Получили, что при таком значении коэффициента входное состояние нейронов остается неизменным, и работа сети нарушается.

Утверждение 2. Стабильное состояние нейронной сети удовлетворяет краевым условиям вида $f(r, t) = \mu(r, t)$, $r \in \Gamma$.

Доказательство. Очевидно, что в силу (10) выходные значения нейронов, в которых определено граничное условие, постоянны и всегда ему удовлетворяют.

Утверждение 3. Стабильное состояние нейронной сети удовлетворяет краевым условиям вида $f'(r, t) = \mu(r, t)$, $r \in \Gamma$.

Доказательство. Рассмотрим первый слой сети ($f_{L(X, t-1)} = 0$). В стабильном состоянии для нейронов, соответствующих узлам сети, в которых задано граничное условие, выполняется следующее равенство:

$$f_{L(X, t)} = K\left(\frac{b}{\bar{K}}f_{L(X, t)} + \frac{1-b}{\bar{k}_i}\left(\frac{\tau}{h^2}\right) \times \sum_{i=1}^n \left(\frac{2k_{L1}}{k_{L(X, t)} + k_{L1}}f_{L1} + \frac{2k_{L2}}{k_{L(X, t)} + k_{L2}}f_{L2}\right) + \frac{(1-b)\mu_{L(X, t)}}{2n}\right);$$

$$f_{L(X, t)} = bf_{L(X, t)} + \frac{(1-b)K}{\bar{k}_i}\left(\frac{\tau}{h^2}\right) \times \sum_{i=1}^n \left(\frac{2k_{L1}}{k_{L(X, t)} + k_{L1}}f_{L1} + \frac{2k_{L2}}{k_{L(X, t)} + k_{L2}}f_{L2}\right) + \frac{K\mu_{L(X, t)}}{2n};$$

$$(1-b)f_{L(X, t)} = \frac{(1-b)K}{\bar{k}_i}\left(\frac{\tau}{h^2}\right) \times \sum_{i=1}^n \left(\frac{2k_{L1}}{k_{L(X, t)} + k_{L1}}f_{L1} + \frac{2k_{L2}}{k_{L(X, t)} + k_{L2}}f_{L2}\right) + \frac{(1-b)K\mu_{L(X, t)}}{2n};$$

$$f_{L(X, t)} = \frac{K}{\bar{k}_i}\left(\frac{\tau}{h^2}\right) \sum_{i=1}^n \left(\frac{2k_{L1}}{k_{L(X, t)} + k_{L1}}f_{L1} + \frac{2k_{L2}}{k_{L(X, t)} + k_{L2}}f_{L2}\right) + \frac{K\mu_{L(X, t)}}{2n}.$$

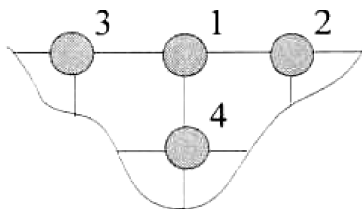


Рис. 4. Нейронная сеть на границе области

Примем $k = \text{const}$, т. е. $\bar{k} = 2n \frac{\tau}{h^2}$. Поскольку $K = \frac{2n}{|B(i)|}$, получим:

$$f_{L(X, t)} = \frac{2n}{|B(i)| 2n(\tau/h^2)} \left(\frac{\tau}{h^2} \right) \times \\ \times \sum_{i=1}^n (f_{L1} + f_{L2}) + \frac{2n \mu_{L(X, t)}}{|B(i)| 2n};$$

$$f_{L(X, t)} = \frac{1}{|B(i)|} \sum_{i=1}^n (f_{L1} + f_{L2}) + \frac{\mu_{L(X, t)}}{|B(i)|};$$

$$f_{L(X, t)} = \frac{1}{|B(i)|} \left(\sum_{i=1}^n (f_{L1} + f_{L2}) + \mu_{L(X, t)} \right).$$

Рассмотрим двумерный случай, приведенный на рис. 4.

Для 1-го нейрона будет выполняться равенство

$$f_1 = \frac{f_2 + f_3 + f_4 + \mu_1}{3}.$$

Тогда, учитывая, что

$$\left. \frac{\partial u}{\partial n} \right|_{x_i} \approx (f_1 - f_2) + (f_1 - f_3) + (f_1 - f_4),$$

после тривиальной подстановки получаем:

$$\left. \frac{\partial u}{\partial n} \right|_{x_i} = \mu_i.$$

Таким образом, для выбранного узла будет выполняться второе краевое условие.

Вычислительный эксперимент

Цель эксперимента: проверить правильность решения уравнений нейронной сетью.

Методика проведения эксперимента: выберем простое уравнение, аналитическое решение которого известно. Определим параметры ячеистой нейронной сети и покажем, дает ли нейронная сеть заданную точность в зависимости от числа узлов и коэффициента b .

Для иллюстрации рассмотренной методики используем уравнение распространения теплоты в составном стержне с постоянным коэффициентом теплопроводности:

$$\frac{\partial f}{\partial t} = k^2 \frac{\partial^2 f}{\partial x^2};$$

$$\begin{cases} f(0, t) = 10; \\ f(l, t) = 25; \\ f(x, 0) = 5. \end{cases}$$

Будем считать, что стержень состоит из частей длины $0, l = 1$ с постоянным коэффициентом теплопроводности $k = 3$.

Аналитическое решение уравнения имеет вид

$$f(x, t) = \sum_{n=0}^{\infty} \left(\frac{40(-1)^n - 10}{\pi n} \times \right. \\ \left. \times \sin\left(\frac{\pi n x}{10}\right) e^{-9t \frac{\pi^2 n^2}{100}} \right) + 1,5x + 10.$$

Результат решения задачи с применением нейронных сетей приведен на рис. 5. Решение, полученное с помощью нейронной сети, соответствующей равномерной сетке дискретизации и состоящей из $121 = 11 \cdot 11$ узлов, отклоняется от аналитического на значения порядка 10^{-6} , сравнимые со значением, используемым в критерии завершения работы сети.

При многократном запуске нейронной сети с различными значениями коэффициента b установлено, что с уменьшением коэффициента b увеличивается скорость сходимости сети, и наоборот.

Кроме того, с уменьшением шага дискретизации (и, соответственно, с увеличением числа узлов) число шагов, необходимых для перехода сети в стабильное состояние, изменяется не пропорционально. В таблице приведены числа узлов дискретизации и соответствующие им числа шагов, за которые сеть пришла в стабильное состояние при $b = 0,5$ и 10 слоях с $\tau = 0,003$.

В результате проведения эксперимента получили, что гипотеза о сходимости, сформулирован-

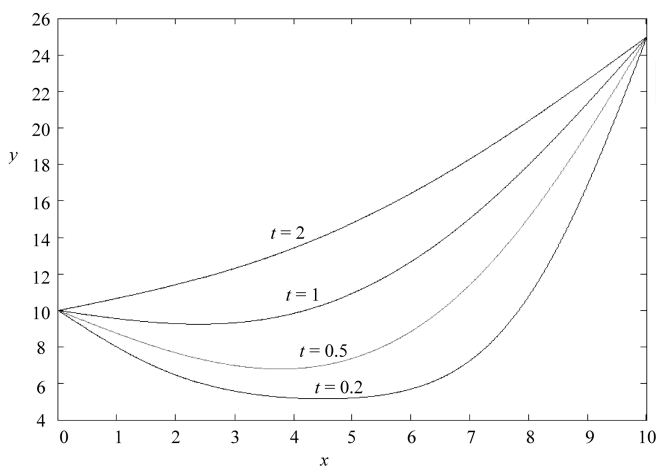


Рис. 5. Общий вид решения примера

Зависимость скорости перехода сети в стабильное состояние от числа узлов дискретизации

Число узлов дискретизации	121	484	1089	1936	3025	5929	12 100
Число шагов	15	70	136	195	229	240	243

ная в математическом обосновании, подтвердилась на практике.

При сравнении с методом сеток нейронная сеть показала в среднем 5 % выигрыша в скорости нахождения решения, и это при осуществлении вычислений на одиночной ЭВМ.

Таким образом, в результате эксперимента подтвердилась гипотеза о преимуществе в скорости нахождения решения.

Заключение

Предложенная нейронная сеть позволяет получать приближенное численное решение задач теплопроводности, представленных параболическими дифференциальными уравнениями в частных производных. Кроме того, данная нейронная сеть не требует обучения. Точность решения с помощью нейронной сети сравнима с решениями широко распространенных численных методов, однако подход, основанный на использовании клеточных нейронных сетей, позволяет более эффективно использовать параллельные вычисления. Это достигается в основном благодаря не-

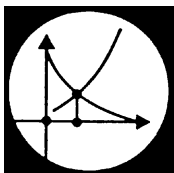
большому радиусу взаимодействия между нейронами в сети.

Кроме того, в результате проведения эксперимента была доказана гипотеза о сходимости и эквивалентности стабильного состояния сети решению задачи.

Скорость решения сетью задачи не уступает известным методам, вследствие чего можно говорить о разумности использования ячеистых нейронных сетей при решении подобных задач.

Список литературы

1. Тихонов А. Н., Самарский А. А. Уравнения математической физики. М.: Изд-во МГУ, 1977.
2. Арфкен Г. Математические методы в физике. М.: Атомиздат, 1969. 712 с.
3. Lee H., Kang I. Neural algorithms for solving differential equations // Journal of Computational Physics. 1990. V. 91. P. 110–117.
4. Wang L., Mendel J. M. Structured trainable networks for matrix algebra // IEEE Int. Joint Conference on Neural Networks. 1990. V. 2. P. 125–128.
5. Gupta A., Karyptis G., Kumar V. Highly Scalable Parallel Algorithms for Sparse Matrix Factorizations // IEEE Transactions on Parallel and Distributed Systems. 1995. V. 8. N 5. P. 502–520.
6. Lagaris I. E., Likas A., Fotiadis D. Artificial Neural Networks for Solving Ordinary and Partial Differential Equations // IEEE Transaction on Neural Networks. 1998.
7. Fortuna L., Arena P., Balya D., Zarandy A. Cellular Neural Networks: A Paradigm for Nonlinear Spatio-Temporal Processing // Circuits and Systems Magazine, IEEE. 2001. V. 1. N 4. P. 6–21.
8. Rekeczky C., Szatmari I., Foldes P. and Roska T. Analogic Cellular PDE Machines // Analogical and Neural Computing Systems Laboratory. Computer and Automation Research Institute. Hungarian Academy of Sciences, Budapest, Hungary. Tech. Report, 2004.



ОПТИМИЗАЦИЯ И МОДЕЛИРОВАНИЕ

УДК 519.6:004.4

Введение

А. П. Карпенко, д-р физ.- мат. наук, проф.,
E-mail: karpenko@ng.ru

В. Г. Федорук, канд. техн. наук, доц.,
МГТУ им. Н. Э. Баумана,
E-mail: fedoruk@comcor.ru

Обзор программных систем многокритериальной оптимизации.

Отечественные системы

Работа представляет собой обзор наиболее известных и развитых отечественных программных систем многокритериальной оптимизации.

В условиях определенности задачу принятия решений можно сформулировать следующим образом. Имеется компактное множество альтернатив D_X . Выбор каждой из альтернатив $X^i \in D_X$ вызывает наступление некоторого исхода $Y^i \in D_Y$, где D_Y — множество всех возможных исходов. Имеется также некоторый механизм, позволяющий лицу, принимающему решения (ЛПР) или группе лиц, принимающей решения (ГПР), оценить качество исхода. Ставится задача найти наилучшую или наилучшие альтернативы.

В условиях неопределенности альтернативе $X^i \in D_X$ может соответствовать не единственный исход, а некоторая совокупность исходов, отражающая "природные" неопределенности или неопределенности типа "активного партнера".

Поскольку при выборе наилучшей альтернативы приходится учитывать различные противоречивые требования, всякая достаточно сложная задача принятия решения является многоцелевой.

Задача многоцелевого принятия решений может быть сформулирована в разных терминах: с использованием функций выбора, в терминах бинарных отношений, на основе "критериального языка" [1]. В обзоре используется самый простой, развитый и наиболее популярный критериальный язык, в терминах которого указанная задача называется задачей *многокритериального принятия решений* (MCDM — *Multi-Criteria Decision Making*).

В задаче *однокритериального принятия решений* в условиях определенности полагается, что на множестве D_X задана критериальная функция $f(X)$, обладающая следующим свойством: если альтернатива X^i предпочтительнее альтернативы X^j , то значение $f(X^i)$ меньше (для определенности) значения $f(X^j)$. В результате выбор наилучшей альтернативы сводится к отысканию альтернативы с наименьшим значением критериальной функции. Функция $f(X)$ в этом случае называется критерием (критерием качества, критерием оптимальности, целевой функцией, функцией предпочтения, функцией полезности и т. п.). Вообще говоря, критерий оптимальности может быть как количественным (принимать числовые значения), так и качественным (лингвистическим) — принимать значения типа "много", "мало" и т. п. Типичными для технических задач являются количественные критерии.

В задаче *многокритериального принятия решений* предполагается заданной вектор-функция $F(X) = (f_1(X), f_2(X), \dots)$ и ЛПР желательнее найти альтернативу, которая минимизировала бы (для определенности) все компоненты этой вектор-функции. Компоненты $f_1(X), f_2(X), \dots$ вектор-функции $F(X)$ называются частными критериями оптимальности.

В зависимости от мощности множества D_X выделяют два класса задач многокритериального принятия решений [2]. Если мощность множества D_X велика (например, множество является континуальным), то задача называется задачей многокритериальной оптимизации (МСО — *Multi-Criteria Optimization*). В англоязычной литературе часто также используют (как синоним) термин МОО (*multi-objective optimization*). Если число альтернатив во множестве D_X невелико, то эта задача называется задачей многокритериального анализа (МСА — *Multi-Criteria Analysis*).

Далее для компактности изложения вместо термина "система многокритериальной оптимизации" будем использовать термин "МКО-система".

Методы решения МКО-задач и задач многокритериального анализа существенно различны. МКО-системы часто включают в себя подсистемы многокритериального анализа.

Противоречивый характер целей ЛПР обуславливает, как правило, противоречивость частных критериев — свои минимальные значения частные критерии принимают в различных точках множества D_X (напомним, что множество D_X полагается компактным). В результате, ЛПР, принимая альтернативу $X^i \in D_X$ в качестве решения МКО-задачи, всегда должно идти на компромисс — допуская ухудшение значений одних частных критериев во имя улучшения значений других критериев. Вследствие этого обстоятельства деятельность ЛПР при решении МКО-задачи является плохо формализуемой и требует привлечения его предыдущего опыта и интуиции.

В настоящее время при оптимизации сложных технических систем (например, во многих задачах САПР) возникают МКО-задачи, имеющие следующие особенности [3]:

- высокая размерность вектора альтернатив X (порядка 100);
- сложная структура множества допустимых альтернатив D_X , обуславливаемая большим числом (порядка 100) и нелинейностью ограничивающих функций, формирующих это множество;
- высокая размерность критериальной вектор-функции $F(X)$ (порядка 100);
- высокая сложность математических моделей оптимизируемых технических систем (дифференциальные уравнения в частных производных и пр.), приводящая к высокой вычислительной сложности частных критериев (порядка 100 ч процессорного времени на современных однопроцессорных ЭВМ);
- сложная топология частных критериев оптимальности (овражность, многоэкстремальность и пр.).

Плохая формализуемость и указанные выше особенности современных МКО-задач делают необходимым использование для их решения программных систем многокритериальной оптимизации (МКО-систем), которые и являются предметом данного обзора.

Для рассматриваемых в обзоре МКО-систем представлена фактическая информация о системе (наименование, место разработки, авторы и пр.). Основное внимание уделено функциям системы. При описании методов решения МКО-задач излагаются только их основные черты, детали, возможно весьма существенные, опускаются.

В данном обзоре представлены отечественные МКО-системы. Зарубежные МКО-системы станут предметом самостоятельного обзора.

1. Постановка МКО-задачи

В условиях определенности МКО-задачу запишем в следующем виде:

$$\min_{X \in D_X} F(X) = F(X^*), \quad (1)$$

где $X = (x_1, x_2, \dots, x_n)$ — вектор переменных, принадлежащий непустому замкнутому (компактному) множеству допустимых значений $D_X \subset R^n$; $F(X) = (f_1(X), f_2(X), \dots, f_m(X))^T$ — вектор критериев оптимальности, осуществляющий отображение множества D_X в непустое критериальное множество $D_Y = \{Y | Y = F(X), X \in D_X\} \subset R^m$; X^* — искомое решение задачи; R^n, R^m — арифметические векторные пространства размерности $n \geq 1$ и $m > 1$, соответственно; t — символ транспонирования. Запись (1) понимается лишь в том смысле, что ЛПР желательнее уменьшить значения частных критериев оптимальности $f_1(X), f_2(X), \dots, f_m(X)$.

Множество D_X обычно задается с помощью k ограничивающих функций $g_i(X) \geq 0, i \in [1, k]$:

$$D_X = \{X | g_i(X) \geq 0, i \in [1, k]\}. \quad (2)$$

Заметим, что в постановке МКО-задачи чаще всего не выделяют в явном виде ограничения типа равенств, полагая, что в случае их наличия, они с помощью ограничений вида $g_i(X) \geq 0, g_{i+1}(X) \leq 0$ сведены к ограничениям (2).

Будем писать $Y^1 \leq Y^2$ и говорить, что вектор $Y^1 \in D_Y$ доминирует по Парето вектор $Y^2 \in D_Y$, если $Y^1 \neq Y^2$ и $y_i^1 \leq y_i^2, i \in [1, m]$. Множеством Парето D_Y^* постав-

ленной МКО-задачи называется совокупность векторов $Y^* \in D_Y$, среди которых нет доминируемых:

$$D_Y^* = \{Y^* \in D_Y \mid \{Y' \in D_Y \mid Y' \leq Y^*\} = O\}. \quad (3)$$

Множество векторов $X \in D_X$, которые порождают множество Парето, будем обозначать D_X^* и называть эффективным по Парето множеством:

$$D_X^* = \{X^* \in D_X \mid F(X^*) \in D_Y^*\}. \quad (4)$$

Неформально эффективное по Парето множество — это такая совокупность решений X^* МКО-задачи, на которой ни один из частных критериев оптимальности не может быть уменьшен без увеличения значения одного или нескольких других частных критериев.

Исключительное место, которое множество Парето и эффективное по Парето множество занимают в теории многокритериальной оптимизации, обусловлено тем обстоятельством, что согласно принципу Эджворта—Парето [4], при "разумном" поведении ЛПР выбор решения следует проводить на множестве Парето. Можно сказать, что в случае наличия у ЛПР только той информации, которая дана в постановке МКО-задачи, решением этой задачи является множество Парето. Для получения одного "наилучшего" решения МКО-задачи требуется дополнительная информация о предпочтениях ЛПР.

Многие методы решения МКО-задачи основаны на гипотезе существования единственной, неизвестной ЛПР скалярной функции его предпочтений $u(X)$, $X \in D_X$, такой, что ее минимум достигается в искомой точке $X^* = \arg \min_{X \in D_X} u(X)$. При этом полагается, что функция $u(X)$ "согласована" с вектор-функцией $F(X)$ в том смысле, что $X^* \in D_X^*$.

В условиях неопределенности критериальная вектор-функция F и, может быть, ограничивающие функции $g_i(X)$, $i \in [1, k]$ зависят, кроме вектора переменных X , также от векторного фактора неопределенности $Z \in D_Z$, значения компонентов которого неизвестны в момент принятия решения. Здесь D_Z — множество допустимых значений фактора неопределенности.

2. Методы решения МКО-задачи

Методы решения МКО-задачи чрезвычайно разнообразны (что является, в конечном счете, следствием отмеченной выше плохой формализуемости этой задачи). Существует несколько способов классификации этих методов (см. например, [5—7]). В обзоре мы используем классификацию, основанную на содержании и форме использования дополнительной информации о предпочтениях ЛПР [6], в соответствии с которой выделяются следующие классы методов решения МКО-задачи:

- методы зондирования;
- априорные методы;
- апостериорные методы;
- адаптивные методы.

Методы каждого из этих классов имеют свои достоинства и ни один из них не свободен от недостатков, не позволяющих признать его универсальным.

Общей идеей решения МКО-задачи является сужение множества D_X вплоть до одной или немногих точек. Заметим, что построение эффективного по Парето множества D_X^* можно интерпретировать как часть этого пути.

Методы зондирования не предполагают формализации в той или иной форме дополнительной информации

о предпочтениях ЛПР. Схема решения МКО-задачи в этом случае может быть, например, следующей.

Шаг 1. Покрытие множества D_X некоторой сеткой $\{X^i \in D_X, i \in [1, N]\}$, число узлов N в которой определяется требуемой точностью решения.

Шаг 2. Вычисление в узлах этой сетки значений всех частных критериев оптимальности $Y^i = (f_1(X^i), f_2(X^i), \dots, f_m(X^i)) = F(X^i)$, $i \in [1, N]$.

Шаг 3. Неформальный анализ ЛПР полученных результатов и принятие решения.

Основной проблемой рассмотренной схемы является возможность получения нерепрезентативных аппроксимаций множеств D_X^* , D_Y^* — неравномерных распределений точек X^i , $F(X^i)$ во множествах D_X^* , D_Y^* , соответственно.

Поскольку, как отмечалось во введении, размерности векторов X , Y могут быть высокими, а множества D_X , D_Y , D_X^* , D_Y^* могут иметь сложную топологию, важнейшей задачей в методе зондирования, которую призвана решать МКО-система, является предоставление ЛПР полученных результатов в удобной форме.

При малой мощности множества $\{X^i \in D_X, i \in [1, N]\}$ методы зондирования можно отнести к методам многокритериального анализа.

Априорные методы предполагают формализацию дополнительной информации о предпочтениях ЛПР до начала решения задачи, априори. Чаще всего эта информация формализуется в форме, позволяющей свести многокритериальную задачу к однокритериальной.

Наиболее известным априорным методом является *метод скалярной свертки*. В этом случае скалярный критерий образуется, например, как взвешенная сумма частных критериев (аддитивная свертка):

$$\Phi(X) = \sum_{i=1}^m \lambda_i f_i(X), \quad \sum_{i=1}^m \lambda_i = 1, \lambda_i > 0, \quad (5)$$

где λ_i — априори назначаемые ЛПР весовые коэффициенты частных критериев оптимальности. Чтобы избавиться от проблемы различных единиц измерения частных критериев, часто используют их различные нормализации.

В реальной вычислительной практике априорные методы в "чистом" виде используются не часто — на их основе обычно строятся апостериорные и адаптивные методы.

Отметим следующий широко известный факт. Функция (5) достигает минимума в точке, принадлежащей эффективному по Парето множеству, т. е. если $\Phi(X^*) = \min_{X \in D_X} \Phi(X)$, то $X^* \in D_X^*$. Обратное утверждение

верно только для выпуклого множества D_Y^* . Таким образом, для выпуклого множества Парето отыскание точки X^* эквивалентно отысканию соответствующих весовых коэффициентов.

Апостериорные методы предполагают внесение ЛПР в МКО-систему дополнительной информации о своих предпочтениях в ходе решения задачи, апостериори. При этом обычно дополнительная информация, как и в априорных методах, в конечном счете формализуется в виде некоторого скалярного критерия оптимальности.

В качестве примера апостериорного метода приведем схему *метода последовательных уступок* [1].

Шаг 1. Ранжирование ЛПР частных критериев оптимальности $f_1(X)$, $f_2(X)$, ..., $f_m(X)$ по важности.

Шаг 2. Отыскание МКО-системой на множестве D_X минимума f_1^* первого по важности критерия $f_1(X)$ (для простоты записи принято, что первым по важности является критерий $f_1(X)$, вторым — критерий $f_2(X)$ и т. д.).

Шаг 3. Анализ ЛПР полученного значения критерия f_1^* и назначение максимально допустимой уступки по этому критерию Δ_1 .

Шаг 4. Отыскание МКО-системой на множестве D_X минимума f_2^* второго по важности критерия $f_2(X)$ с критериальным ограничением $f_1(X) \leq f_1^* + \Delta_1$.

Шаг 5. Анализ ЛПР полученного значения критерия f_2^* и назначение максимально допустимой уступки по этому критерию Δ_2 .

Шаг 6. И т. д. для третьего и последующих по важности критериев.

Адаптивные методы состоят из совокупности итераций, каждая из которых включает в себя фазу анализа, выполняемого ЛПР, и фазу расчетов, выполняемых МКО-системой. По характеру информации, получаемой МКО-системой от ЛПР на фазе анализа, можно выделить три класса адаптивных методов [2, 5]:

- методы, в которых ЛПР непосредственно назначает весовые коэффициенты частных критериев оптимальности;
- методы, в которых ЛПР накладывает ограничения на значения частных критериев оптимальности;
- методы, в которых ЛПР выполняет только сравнение (в терминах "лучше", "хуже", "одинаково") предлагаемых МКО-системой альтернатив.

В последнем случае общая схема решения МКО-задачи может быть, например, следующей.

Шаг 1. ЛПР оценивает предлагаемые МКО-системой решения в терминах "лучше", "хуже", "одинаково" и предоставляет их системе.

Шаг 2. МКО-система организует на этой основе каким-либо методом скалярной оптимизации один шаг поиска минимума неявно заданной функции $u(X)$ предпочтений ЛПР.

Шаг 3. ЛПР оценивает полученные результаты и т. д.

Рассмотренные классы методов решения МКО-задачи в значительной мере переплетаются друг с другом, так что не всегда используемые методы удается однозначно отнести к тому или иному классу.

В случае наличия "природной" неопределенности, рассмотренные методы обычно дополняют преобразованием вектора критериев $F(X, Z)$ к виду, не содержащему фактор неопределенности $Z \in D_Z$, путем использования принципа гарантированного результата. В случае неопределенности типа "активный партнер" используют подходы теории игр.

Если решение принимает не ЛПР, а ГПР, то чаще всего также используют методы, построенные на основе рассмотренных классов методов.

3. Система MOVI

Система MOVI (*Multicriteria Optimization and Vector Identification*) разработана в Институте машиноведения им. А. А. Благонравова РАН (Москва) и в Naval Postgraduate School (Монтерей, шт. Калифорния, США) под руководством проф. Р. Б. Статникова. Система предназначена для корректной постановки и решения задач многокритериальной оптимизации и векторной идентификации. Система позволяет использовать модели MatLab/Simulink; имеется возможность подключения внешних математических моделей. Методологической основой системы является *метод исследования пространства параметров*, относящийся к классу методов зондирования [3]. Система в настоящее время поддерживается, детальное описание системы и ее демонстрационная версия доступны по адресу <http://www.psi-movi.com>.

Требования к среде. Процессор — Intel PII/300 или выше; оперативная память — 64 Мбайт. Система функционирует под управлением операционных систем MS Windows 98, Windows NT 4.0 с Service Pack 3 или выше, Windows 2000, Windows XP. В качестве системы управления базой данных используется СУБД Interbase 6.0 или выше.

Особенности постановки задачи. Рассматривается МКО-задача в условиях определенности, решения принимаются ЛПР. Полагается, что множество допустимых значений D_X непрерывно и

$$D_X \subset \Pi = \{X \mid d_i \leq x_i \leq u_i, i \in [1, n],$$

где d_i, u_i — некоторые вещественные константы; Π — гиперпараллелепипед. Частные критерии $f_1(X), f_2(X), \dots, f_m(X)$ могут быть только количественными, оптимальными значениями могут быть как их минимальные, так и максимальные значения. На частные критерии может быть наложено l критериальных ограничений вида

$$f_{ij}(X) \leq f_{ij}^+, i_j \in [1, m], j \in [1, l], l \leq m, \quad (6)$$

где f_{ij}^+ — некоторые вещественные константы.

Основные функции системы:

- ввод исходных данных;
- генерация во множестве Π независимых случайных точек $X^1, X^2, \dots, X^N$;
- проверка для этих точек выполнимости ограничений, формирующих множество D_X и исключение точек, не прошедших проверку;
- вычисление во всех оставшихся точках значений частных критериев $f_1(X), f_2(X), \dots, f_m(X)$ и формирование на этой основе так называемой таблицы испытаний;
- проверка выполнимости критериальных ограничений (6) и исключение из множества D_X и таблицы испытаний точек, не прошедших проверку;
- построение множества Парето и соответствующей таблицы;
- анализ полученных результатов;
- определение наилучшего решения.

Генерация во множестве Π независимых случайных точек $X^1, X^2, \dots, X^N$ выполняется на основе ЛПГ-последовательностей [3], но могут быть использованы и другие генераторы случайных чисел. В "университетской" версии системы число переменных n ограничено 8. В коммерческой версии это число определяется только длиной периода генератора случайных чисел и при использовании ЛПГ-последовательностей может достигать 51, а при использовании других генераторов — многих сотен.

Построение множества Парето осуществляется путем удаления из усеченного множества D_X тех векторов, которые могут быть улучшены одновременно по всем критериям.

Система предоставляет ЛПР возможность исследования зависимостей критериев от переменных, а также критериев от критериев, как в виде таблиц, так и в виде графиков и гистограмм. Все таблицы и результаты можно экспортировать в MS Excel и получить их твердые копии.

Если решение МКО-задачи на одном компьютере требует неприемлемо большого времени, то ее можно решать одновременно на нескольких компьютерах. При этом на компьютере с номером j следует создать свою задачу со своим набором независимых случайных точек $X_j^1, X_j^2, \dots, X_j^{N_j}$ либо разбить параллелепипед Π на соответствующее число компьютеров подобластей и в каждой из подобластей решить свою МКО-задачу. В обоих

случаях система позволяет выполнить "объединение задач" — построить объединенное множество Парето.

Пример решенной задачи. В работе [3] приведено несколько примеров использования метода исследования пространства параметров для решения задач оптимального проектирования и идентификации технических систем. Например, приводится задача оптимального проектирования резонансной виброплощадки, математическая модель которой имеет 10 переменных, пять критериев оптимальности, а множество D_X определяется семью ограничивающими функциями.

4. Система PFV

Система PFV (Pareto Front Viewer) разработана в Вычислительном центре РАН под руководством проф. А. В. Лотова [8]. В качестве среды разработки использована программа MS EXEL. Система реализует *метод достижимых целей*, предложенный А. В. Лотовым в 1973 г. и относящийся к классу методов зондирования. Основной идеей метода является выбор ЛПР решения на основе графического анализа множества D_Y , которое называется множеством достижимых целей, и прежде всего, его границы — множества Парето D_Y^* . Метод достижимых целей реализован в ряде других программных систем с функциями, близкими к рассмотренным ниже. Описание этих систем, а также их демонстрационные версии доступны по адресу: <http://www.ccas.ru/mmes/mmeda/mcdm.htm>.

Особенности постановки задачи. Рассматривается МКО-задача в условиях определенности, решения принимаются ЛПР, множество переменных D_X полагается непрерывным, частные критерии $f_1(X), f_2(X), \dots, f_m(X)$ могут быть только количественными, оптимальными могут быть как их минимальные, так и максимальные значения. Полагается, что на множестве D_X определена конечная мера μ_X и известен способ получения на этом множестве достаточно большого числа независимых случайных точек, согласованных с мерой μ_X (например, равномерно распределенных на множестве D_X).

Требования к среде. Система функционирует под управлением операционной системы MS Windows 98/NT/2000/XP.

Основные функции системы:

- задание модели задачи;
- задание критериев и параметров вычислений;
- построение базы покрытия;
- визуализация покрытия и выбор предпочтительного варианта.

Множество D_Y покрывается окрестностями совокупности N независимых случайных точек, которая называется базой покрытия. Под *покрытием* множества D_Y понимается совокупность шаров с центрами в точках базы покрытия. Известна вероятностная оценка для минимального числа N , как функции максимально допустимого радиуса шаров, требуемых точности и надежности покрытия. Эта оценка позволяет построить покрытие, аппроксимирующее множество D_Y с заданной точностью и надежностью. В процессе построения покрытия система предъявляет ЛПР график зависимости полноты покрытия от радиуса покрытия, что позволяет ему оперативно принять решение о завершении построения покрытия.

Визуализация осуществляется системой в виде заданных ЛПР двумерных сечений построенного покрытия. Значения третьего критерия задаются на этих сечениях цветом. Любая из точек сечения может быть указана

компьютерной "мышью", после чего система предъявит ЛПР соответствующее решение во множествах D_X, D_Y .

Пример решенной задачи. Имеется информация об использовании метода достижимых целей для решения целого ряда реальных МКО-задач, прежде всего, в области экономики. Отметим задачу об оптимизации непрерывной разливки стали, содержащую пять частных критериев и более 300 переменных [9].

5. Система "Quick Choice"

"Quick Choice" — система многокритериального выбора вариантов, разработана в Санкт-Петербургском государственном политехническом университете под руководством проф. И. Г. Черноруцкого [1]. Состояние системы в настоящее время авторам обзора неизвестно.

Особенности постановки задачи. Рассматривается МКО-задача в условиях определенности, решения принимаются ЛПР, множество переменных D_X полагается дискретным. Частные критерии оптимальности $f_1(X), f_2(X), \dots, f_m(X)$ могут быть как количественными, так и лингвистическими. В первом случае оптимальными могут быть как минимальные, так и максимальные значения частных критериев.

Требования к среде. Система требует минимум 8 Мбайт оперативной памяти, процессор — не ниже Intel 486 и функционирует под управлением операционной системы MS Windows 2000/XP.

Основные функции системы:

- диалоговый ввод информации о задаче;
- ранжирование частных критериев оптимальности по важности на основе метода t -упорядочения;
- проверка корректности информации об относительной важности критериев;
- формирование t -оптимального множества;
- сужение t -оптимального множества методом ограничений до одной альтернативы;
- импорт данных из базы данных dBase или Paradox, а также из текстового файла;
- создание отчетов.

Метод t -упорядочения относится к классу адаптивных методов и является развитием известного метода, предложенного В. В. Подиновским [10]. В методе t -упорядочения частные критерии $f_i(X), f_j(X)$ называются равноценными, если для ЛПР всякие векторы

$$(f_1(X), \dots, f_i(X), \dots, f_j(X), \dots, f_m(X)), \\ (f_1(X), \dots, f_i(X) + \delta, \dots, f_j(X) - \delta, \dots, f_m(X)), X \in D_X,$$

одинаковы по предпочтительности для любого δ . Аналогично критерий $f_i(X)$ полагается более важным, чем критерий $f_j(X)$, если вектор $(f_1(X), \dots, f_i(X), \dots, f_j(X), \dots, f_m(X))$ более предпочтителен для ЛПР, чем вектор $(f_1(X), \dots, f_i(X) + \delta, \dots, f_j(X) - \delta, \dots, f_m(X))$ для любого δ . Отношения равноценности и предпочтительности позволяют сформировать t -отношение предпочтения более сильное, чем отношение предпочтения Парето, что дает возможность сузить множество эффективных по Парето точек D_X^* до

t -оптимального множества $D_T \subset D_Y^*$.

Сужение множества D_T осуществляется с помощью адаптивного *метода ограничений*, схема которого имеет следующий вид.

Шаг 1. Для каждого из частных критериев оптимальности ЛПР назначает верхние допустимые значения $f_i^+(X)$.

Шаг 2. Система строит множество D_C — сужение множества D_T :

$$D_C = \{X \in D_T \mid f_i(X) \leq f_i^+, i \in [1, m]\}.$$

Шаг 3. Если множество D_C — пусто, то система предлагает ЛПР ослабить ограничение на один или несколько критериев и возвращается к шагу 2.

Шаг 4. Система выбирает из множества D_C вектор X^* , который обеспечивает на этом множестве минимум аддитивной свертки, и предъявляет его ЛПР.

Шаг 5. Если вектор X^* удовлетворяет ЛПР, вычисления заканчиваются, иначе система предлагает ЛПР задать новые верхние допустимые значения $f_i^+(X)$ и возвращается к шагу 2.

В качестве весовых множителей в аддитивной свертке система позволяет использовать случайные коэффициенты, а также коэффициенты, найденные на основе принципа гарантированного результата.

Данные для анализа предоставляются ЛПР в числовом виде, а также в виде диаграмм.

Пример решенной задачи. В качестве примера рассмотрена задача выбора модели автомобиля по девяти критериям из 277 альтернатив.

6. Система "ВЫБОР-12М"

Система автоматизированного выбора "ВЫБОР-12М" разработана в Московском энергетическом институте под руководством проф. Ю. В. Кандырина [11]. В качестве среды разработки использована среда визуального программирования Delphi, обеспечена совместимость с MS Excel 5.0/95. Описание системы и ее демонстрационная версия доступны по адресу <http://www.pilab.ru>.

Особенности постановки задачи. Рассматривается МКО-задача в условиях определенности, решения принимаются ЛПР, множество переменных D_X полагается дискретным. На компоненты векторов переменных $X^1, X^2, \dots$ могут быть наложены двусторонние или односторонние ограничения вида $d_j \leq x_j \leq u_j, x_j \geq d_j, x_j \leq u_j$, где $d_j, u_j, j \in [1, n]$ — минимально и максимально допустимые значения компоненты x_j вектора X , соответственно.

Требования к среде. Система требует минимум 8 Мбайт оперативной памяти, процессор — не ниже Intel Pentium-100 и функционирует под управлением операционной системы MS Windows 9x.

Основные функции системы:

- ввод исходной информации о задаче;
- сужение исходного множества переменных;
- создание отчетов.

Сужение исходного множества переменных может быть выполнено в системе итерационно следующими методами:

1) путем учета ограничений на компоненты векторов переменных (в результате получается множество D_X);

2) с помощью последовательного решения совокупности однокритериальных задач для заданных ЛПР частных критериев оптимальности, например: на множестве D_X выполняем оптимизацию критерия $f_1(X)$ — получаем множество D_X^1 ; на множестве D_X^1 выполняем оптимизацию критерия $f_2(X)$ — получаем множество D_X^2 ; и т. д. до получения множества достаточно малой мощности;

3) на основе построения системой эффективного по Парето множества D_X^* для двух любых заданных ЛПР критериев оптимальности;

4) путем решения МКО-задачи адаптивным методом последовательных уступок;

5) выделением одного из частных критериев в качестве "главного" критерия, перевод остальных критериев в разряд ограничений и решение полученной однокритериальной задачи;

6) с помощью решения МКО-задачи априорным методом скалярной свертки. Система поддерживает два вида свертки — аддитивную (5) и мультипликативную:

$$\Phi(X) = \prod_{i=1}^m f_i^{\lambda_i}(X), \lambda_i > 0, \quad (7)$$

где λ_i — априори назначаемые ЛПР весовые коэффициенты частных критериев оптимальности.

Данные для анализа предоставляются ЛПР в числовом виде, а также в виде 2D диаграмм.

Пример решенной задачи. В качестве примера рассмотрена задача выбора транзистора по двум критериям из 13 альтернатив, имеющих размерность, равную 12.

7. Прочие MCDM-системы

"IOSO NM 1.0" — программная система многокритериальной оптимизации, разработана компанией "IOSO Technology Center" под руководством проф. И. Н. Егорова. Свое название система получила от метода скалярной оптимизации на основе самоорганизации (*Indirect Optimization based on Self-Organization*) [12], который относится к классу методов аппроксимации [13]. Система является частью семейства программных средств IOSO, предназначенных для решения (в том числе на мультипроцессорных вычислительных системах) широкого класса задач однокритериальной и многокритериальной оптимизации. Авторы системы указывают на возможность простой интеграции ее с известными CAE-системами (NAS-TRAN, ANSYS, CFD, FineDesign, Fluent и т. д.).

На критериальную вектор-функцию F и ограничивающие функции $g_i(X), i \in [1, k]$ могут быть наложены аддитивные или мультипликативные помехи различной интенсивности. Максимальное число переменных и ограничивающих функций в системе равно 100, число частных критериев оптимальности — 10.

Система функционирует под управлением операционной системы MS Windows NT/2000/XP и реализует метод зондирования.

Имеется опыт использования системы "IOSO NM 1.0" для оптимизации авиационных газотурбинных двигателей, автомобильных двигателей и пр.

Описание системы доступно по адресу <http://www.iootech.com>.

Система "ДИСО". "ДИСО" — диалоговая система оптимизации разработана в Вычислительном центре РАН [14]. Основным назначением системы является решение задач многомерной локальной и глобальной оптимизации. Система содержит также подсистему многокритериальной оптимизации. Существенной особенностью системы является возможность изменять постановку задачи в процессе ее решения, например, добавлять новые ограничивающие функции, переводить частные критерии оптимальности в разряд ограничений. Задача описывается на специальном внутреннем языке программирования системы DIFALG, подобном языку ALGOL-60. Вычисление значений частных критериев оптимальности и ограничивающих функций выполняется с использованием интерпретатора этого языка.

Система реализует несколько методов свертывания векторного критерия оптимальности в скалярный, а также метод зондирования, который базируется на двух методах построения дискретной аппроксимации эффективного по Парето множества.

1. Метод на основе построения векторов, принадлежащих ε -эффективному по Парето множеству:

$$D_\varepsilon^* = \{X \in D_X \mid Y(X) \leq Y(X^*) + \varepsilon, X^* \in D_X^*\}.$$

2. Метод на основе построения ε -сети эффективно по Парето множества W_ε^* , которое определяется следующим образом: для любого $X^1 \in D_X^*$ существует такой вектор $X^2 \in W_\varepsilon^*$, что $Y(X^2) \leq Y(X^1) + \varepsilon$; не существует двух векторов $X^1 \in W_\varepsilon^*$, $X^2 \in W_\varepsilon^*$ таких, что $Y(X^1) \leq Y(X^2)$.

В обоих случаях $\varepsilon \in R^m$ — вектор положительных констант — точность аппроксимации.

Современное состояние системы "ДИСО" авторам обзора неизвестно.

"СИМОП" — программная система многоэкстремальной оптимизации разработана в Нижегородском государственном университете им. Н. И. Лобачевского [15, 16] [<http://www.software.unn.ac.ru>]. Система ориентирована на использование в составе систем автоматизированного проектирования, но возможно и автономное использование системы. Одной из отличительных особенностей системы является возможность изменения требований к конструируемому объекту в процессе поиска рационального варианта (при этом информационная база системы обеспечивает возможность использования всех прошлых оценок частных критериев оптимальности).

Система реализует метод свертывания векторного критерия оптимальности в скалярный. В результате с помощью метода штрафных функций МКО-задача сводится к однокритериальной задаче безусловной глобальной оптимизации, которая, в свою очередь, сводится к одномерной задаче скалярной оптимизации. Последняя задача решается информационно-статистическими алгоритмами глобального поиска [17].

В настоящее время "СИМОП" функционирует только в качестве учебно-исследовательской системы, однако на ее основе разрабатывается система "Absolut Expert" (см. ниже).

"Absolut Expert" — программная система параллельных вычислений в задачах выбора глобально-оптимальных решений. Система разрабатывается в Нижегородском государственном университете им. Н. И. Лобачевского под руководством проф. В. П. Гергеля [<http://www.itlab.unn.ru>].

Система предназначена для решения сложных МКО-задач, отличающихся, прежде всего, высокой вычислительной сложностью частных критериев оптимальности. Множество переменных D_X может быть невыпуклым и многосвязным. Путем редукции размерности на основе разверток типа кривой Пеано МКО-задача сводится к одномерной задаче скалярной оптимизации, которая решается параллельно информационно-статистическими алгоритмами глобального поиска (так же, как в системе "СИМОП").

Система ориентирована на использование параллельных вычислительных систем с распределенной памятью и коммуникационной библиотеки MPI. Предполагается также разработка версий системы для вычислительных систем с общей памятью (на основе технологии OpenMP) и гетерогенных вычислительных систем с SMP-узлами [18]. Система строится по принципу открытой архитектуры, позволяющей просто добавлять новую функциональность.

"Универсальный механизм" — программный комплекс разработан в Брянском государственном техническом университете под руководством проф. Д. Ю. Погорелова. Комплекс представляет собой специализированное многофункциональное приложение, предназначенное для решения задач кинематики и динамики механических систем. "Универсальный механизм" (УМ) интегрирован с программной системой MatLab/Simulink — позволяет связывать построенную механическую модель с моделями подсистем различной природы, описанными в MatLab/Simulink. Возможен импорт МКЭ-моделей из программных систем ANSYS, NASTRAN, ABAQUS, I_DEAS и пр. Поддерживаются распределенные вычисления на вычислительных кластерах.

Одним из модулей комплекса является *модуль оптимизации*, который и рассматривается далее.

Модуль поддерживает решение МКО-задачи методом зондирования (строится только дискретная аппроксимация множества D_Y), а также широко известным методом анализа иерархий. Решения принимаются ЛПР.

Метод анализа иерархий Т. Саати относится к классу априорных методов [1, 5] и имеет следующую схему.

Шаг 1. ЛПР формирует $(m \times m)$ -матрицу **S** попарных сравнений каждого из частных критериев оптимальности с каждым.

Шаг 2. Система находит собственный вектор матрицы **S**, соответствующий максимальному собственному числу этой матрицы.

Шаг 3. Нормированные компоненты этого вектора используются в качестве весовых множителей λ_i , $i \in [1, m]$, в аддитивной свертке (5).

Шаг 4. Система находит решение однокритериальной задачи оптимизации (5), которое принимается за искомое решение X^* .

Описание системы, а также ее демонстрационная версия доступны по адресу <http://www.umlub.ru>.

Заключение

Развитая МКО-система должна, очевидно, удовлетворять общепринятым требованиям к современному программному обеспечению (надежность, переносимость, высокое быстродействие, безопасность — защита от потери информации, масштабируемость — расширение силами пользователя, удобный многопользовательский интерфейс, web-доступ и пр.). Кроме того, к современным МКО-системам должен быть предъявлен ряд специфических требований. Назовем основные из них.

Развитая МКО-система является сложным программным продуктом, разработка которого требует значительных денежных и прочих ресурсов. Поэтому целесообразна ориентация системы на достаточно широкий класс МКО-задач. В частности, поскольку большинство реальных сложных МКО-задач содержит в том или ином виде неопределенность, математическая модель задачи должна предусматривать возможность задания как "природных" неопределенностей, так и неопределенностей типа "активный партнер", а методы, реализованные в системе, должны поддерживать решение задач в условиях неопределенностей указанных типов — требования T11, T12, соответственно.

Как отмечено во введении, современные МКО-задачи имеют высокую вычислительную сложность. Поэтому развитая МКО-система должна поддерживать параллельные вычисления как на вычислительных системах с общей памятью, так и на распределенных вычислительных системах — требования T21, T22.

Классификация рассмотренных МКО-систем

Наименование системы	Требования						
	T1	T2	T3	T4	T5	T6	T7
MOVI	—	T22	—	—	—	—	—
PFV	—	—	—	—	—	—	—
"Quik Choise"	—	—	—	—	—	—	—
"Выбор-12М"	—	—	—	+/-	—	—	—
"IOSO NM 1.0"	T11	T22	—	—	—	+	+
"ДИСО"	—	—	—	—	+	—	—
"СИМОП"	—	—	—	—	—	—	—
"Absolut Expert" *)	—	+	—	?	?	?	?
УМ **)	—	T22	—	+/-	+	+	+/-

*) Проектируемая система
**) Речь идет о модуле "Оптимизация"

МКО-система может использоваться в качестве элемента системы поддержки принятия весьма ответственных решений. Такие решения, как правило, принимаются коллективно, группой лиц. Из этого следует, что современная развитая МКО-система должна поддерживать принятие решений как ЛПР, так и ГПР — требование Т3.

Как отмечалось выше, каждый из методов многокритериальной оптимизации имеет свои преимущества и недостатки, и, соответственно, является более или менее эффективным для решения каждой конкретной задачи. Поэтому развитая МКО-система должна, на наш взгляд, поддерживать широкий набор различных методов многокритериальной оптимизации — требование Т4.

Большинство методов многокритериальной оптимизации сводится или включает в себя решение задачи однокритериальной, вообще говоря, многоэкстремальной оптимизации. Эффективность таких методов в значительной мере определяется эффективностью используемых методов решения однокритериальной задачи. Поскольку универсальных методов решения задач однокритериальной многоэкстремальной оптимизации не существует, МКО-система должна предоставлять пользователю несколько высокоэффективных методов однокритериальной глобальной оптимизации — требование Т15.

Основной областью применения сложных МКО-систем являются системы автоматизированного проектирования. Поэтому развитая современная МКО-система должна предусматривать импорт моделей задачи из широко используемых CAD, CAM, CAE-систем [19] — требование Т6.

По той же причине МКО-система должна иметь возможность экспорта результатов в указанные системы — требование Т7.

Классификация рассмотренных МКО-систем, с точки зрения удовлетворения перечисленным требованиям, приведена в таблице (знак вопроса означает отсутствие соответствующей информации).

Таблица показывает, что среди современных отечественных МКО-систем отсутствуют системы, удовлетворяющие всем сформулированным требованиям. В наибольшей степени этим требованиям удовлетворяет система "IOSO NM 1.0", а также проектируемая система "Absolut Expert". Многими чертами развитой МКО-системы обладает система УМ, которая, однако, не является универсальной системой (интегрирована с программным комплексом "Универсальный механизм"). Отметим, что ни одна из рассмотренных систем не поддерживает групповое принятие решений (требование Т3).

В настоящее время отчетливой тенденцией является использование в системах поддержки принятия решений (СППР) интеллектуального компонента. Развитые МКО-системы также должны иметь такой компонент для накопления информации о предпочтениях ЛПР, обработки этой информации и использовании для автоматизированного принятия решений. Ни одна из рассмотренных систем не имеет интеллектуального компонента.

Заметим в завершение, что известно большое число отечественных СППР, например диалоговая система "MPRIORITY 1.0" [<http://tomakechoice.com>], экспертная система многокритериального выбора, оценки и принятия решения "ЭКСПЕРТ-2001" [<http://www.anitel.ru/expert.htm>], СППР "Выбор" [<http://freesoft.ru>] и др., которые выходят за рамки данного обзора.

Список литературы

1. Черноруцкий И. Г. Методы принятия решений. СПб.: БХВ-Петербург, 2005.
2. Растринин Л. А., Эйдук Я. Ю. Адаптивные методы многокритериальной оптимизации // Автоматика и телемеханика. 1985. № 1. С. 12—15.
3. Соболев И. М., Статников Р. Б. Выбор оптимальных параметров в задачах со многими критериями. М.: Дрофа, 2006.
4. Ногин В. Д. Принятие решений в многокритериальной среде: количественный подход. М.: ФИЗМАТЛИТ, 2002.
5. Ларичев О. И. Теория и методы принятия решений. М.: Университетская книга, Логос, 2006.
6. Лотов А. В. Введение в экономико-математическое моделирование. М.: Наука, 1984.
7. Машунин Ю. К. Методы и модели векторной оптимизации. М.: Наука, 1985.
8. Березкин В. Е., Каменев Г. К., Лотов А. В. Реализация метода достижимых целей для нелинейных моделей в MS EXCEL. М.: Изд. Вычислительного центра РАН. 2000.
9. Miettinen K., Mäkelä M. M., Mäkelä T. Optimal Control of Continuous Casting by Nondifferentiable Multiobjective Optimization // Computational Optimization and Applications. 1998. Vol. 11. P. 177—194.
10. Подиновский В. В., Ногин В. Д. Парето-оптимальные решения многокритериальных задач. М.: Наука, 1982.
11. Кандырин Ю. В. Автоматизированный многокритериальный выбор альтернатив в инженерном проектировании. М.: Изд. МЭИ. 1993.
12. Egorov I. N. Indirect Optimization Method on the Basis of Self-Organization, Curtin University of Technology, Optimization Techniques and Applications (ICOTA'98). Perth, Australia. 1998. Vol. 2. P. 683—691.
13. Box G. E. P., Draper N. R. Empirical Model Building and Response Surfaces. New York: John Wiley & Sons, 1987.
14. Evtushenko Y., Mazourik V., Ratkin V. Multicriteria optimization in the DISO system, Lecture Notes in Control and Information Sciences // System Modelling and Optimization. 1988. Vol. 113. P. 231—240.
15. Гергель В. П., Стронгин Р. Г. Система многоэкстремальной оптимизации // Пакеты прикладных программ. Программное обеспечение оптимизационных проблем. 1987. С. 39—50.
16. Gergel V. P. A software system for multiextremal optimization // European Journal of Operation Research. 1993. V. 65. N 3. P. 305—313.
17. Батищев Д. И., Львович Я. Е., Фролов В. Н. Оптимизация в САПР: Учебник. Воронеж: Изд-во Воронежского государственного университета, 1997.
18. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: БХВ-Петербург, 2004.
19. Норенков И. П. Основы автоматизированного проектирования. М.: Изд-во МГТУ им. Н. Э. Баумана, 2006.
20. <http://www.psi-movi.com>
21. <http://www.ccas.ru/mmes/mmeda/mcdm.htm>
22. <http://www.pilab.ru>
23. <http://www.itlab.unn.ru>
24. <http://www.iosotech.com>
25. <http://www.software.unn.ac.ru>
26. <http://www.unlab.ru>
27. <http://tomakechoice.com>
28. <http://www.anitel.ru/expert.htm>
29. <http://freesoft.ru>

А. В. Гагарин,

Уфимский государственный авиационный
технический университет

Интеллектуальный алгоритм оптимизации параметров ресурсоемких моделей

Рассматривается задача идентификации параметров цифровых моделей объектов исследования, моделирование которых требует значительных вычислительных ресурсов. Для решения этой задачи разработан гибридный генетический алгоритм (ГА) с использованием радиальной базисной нейронной сети для аппроксимации целевой функции. Дается подробное описание алгоритма и его сравнение с существующими аналогами. Тестирование предложенного алгоритма проводится на задаче определения относительных фазовых проницаемостей (ОФП) воды и нефти для гидродинамической модели нефтяного резервуара. Приводятся математическая модель параметризации ОФП и результаты эксперимента, показывающие значительное превосходство гибридного ГА над обычным ГА по скорости оптимизации.

Введение

Современные технологические процессы во многом опираются на результаты, получаемые с помощью полномасштабной цифровой модели объекта исследования. Основным критерием качества такой модели обычно считается совпадение расчетных и наблюдаемых значений основных контролируемых параметров объекта за моделируемый период времени. В данной работе рассматривается задача построения адекватной гидродинамической модели фильтрации углеводородных флюидов, использующейся в современных методах оценки запасов углеводородов и планирования оптимального сценария их добычи. Среди наиболее важных контролируемых параметров такой модели можно отметить дебиты жидкостей по скважинам, забойные давления и т. п.

Всей совокупности исходных геофизических данных заведомо недостаточно для однозначного построения модели, что вызывает необходимость решения плохо обусловленной обратной задачи для определения неизвестных геологических, гидродинамических и других свойств резервуара (обозначаемых в дальнейшем $x_1, x_2, \dots, x_n$ или X) на основе известной истории разработки. Процесс поиска решений этой задачи получил название *history matching* (или *адаптация модели*). Адаптацию можно рассматривать как задачу оптимизации, если ввести некоторую оценку качества модели, которая выступает в роли целевой функции для оптимизации. Такая постановка задачи характерна не только для нефтяной отрасли, но и для многих других инженерных приложений, где требуется определить неизвестные параметры системы по известным выходным сигналам. Качество модели оценивается по значению невязки между расчетными и

измеренными значениями интересующих величин. Широко используемая форма целевой функции (невязки) для задач адаптации может быть записана следующим образом:

$$Q = F(y_i^{\text{calc}}(x_1, x_2, \dots, x_n), y_i^{\text{obs}}) = \sum_i w_i (y_i^{\text{calc}}(x_1, x_2, \dots, x_n) - y_i^{\text{obs}})^2, \quad (1)$$

где y_i^{calc} — расчетное значение интересующей величины; y_i^{obs} — ее измеренное значение; w_i — весовой коэффициент.

Суммирование в общем случае проводится по всем измеряемым данным и временным шагам. Веса w_i учитывают используемые нормализации и другие коэффициенты, позволяющие, например, подчеркнуть важность какого-либо параметра в общей сумме. Целевая функция как функция переменных $x_1, x_2, \dots, x_n$ зачастую имеет несколько локальных минимумов и область локализации глобального минимума неизвестна. Это ограничивает применимость методов локальной оптимизации (например, градиентных) только поздней стадией адаптации. Область, содержащая потенциальное решение, должна быть определена алгоритмом с возможностями глобальной оптимизации. В работе [1] для решения задачи адаптации использовалась комбинация градиентного и туннельного методов, последний из которых предназначен для поиска глобального оптимума. В работе [2] для определения петрофизических свойств породы в модели применялись генетические алгоритмы. Работа [3] посвящена использованию эволюционных стратегий для поиска оптимальных параметров модели.

В данной работе для решения задачи адаптации предлагается гибридный генетический алгоритм (ГА), направленный на снижение ресурсоемкости поиска решения. В последнее время ГА широко применяются для решения сложных задач оптимизации. К их преимуществам перед традиционными методами локального поиска можно отнести отсутствие ограничений на гладкость оптимизируемой функции и ее производных, способность выходить из локальных минимумов за счет использования популяции решений, отсутствие необходимости в априорной информации о предметной области [4–6]. Развитие ГА с целью ускорения эволюционного поиска в настоящее время идет по многим направлениям. К классическим можно отнести усовершенствования основных операторов селекции, скрещивания и мутации [4, 7, 8], распараллеливание [9], разработку самонастраивающихся ГА [10]. Одним из наиболее перспективных направлений является создание гибридных схем, когда ГА работает в паре с другим алгоритмом оптимизации, например градиентным [11]. Такая комбинация позволяет использовать преимущества обоих методов одновременно. Популяционная природа ГА делает поиск устойчивым, в то время как дополнительный метод оптимизации может его существенно ускорить, давая наилучшее приближение оптимального решения.

Предложенный гибридный алгоритм использует нейронную сеть (НС) для аппроксимации целевой функции в целях получения прогноза оптимального решения. НС выступает в роли идентификатора неизвестной системы и динамически обучается в процессе оптимизации. Отличительными особенностями предложенного алгорит-

ма являются универсальность, возможность скачкообразного улучшения решения нейросетевым контуром, полное использование информации, появляющейся в процессе оптимизации, для обучения НС. Предложенный гибрид ГА + НС целесообразно применять в задачах с относительно высокой ресурсоемкостью вычисления значения целевой функции. Адаптация модели резервуара является примером такой задачи, так как оценка невязки по формуле (1) включает в себя проведение численного моделирования резервуара за расчетный период. Применение дополнительного нейросетевого контура позволяет сократить количество расчетов целевой функции, необходимых для нахождения решения заданной точности, по сравнению со стандартным ГА. Это дает возможность значительно снизить ресурсоемкость (увеличить скорость) поиска решения сложных оптимизационных задач.

Разработанный алгоритм универсален и может использоваться для решения оптимизационных задач в любых инженерных приложениях. Он предоставляет возможность сократить число итераций, необходимых для достижения заданной точности решения, или найти лучшее решение по сравнению со стандартным ГА при ограничении по времени. Эффективность предложенного алгоритма проверялась на реальной задаче определения относительных фазовых проницаемостей для тестовой модели резервуара.

Гибридный ГА с применением нейронной сети (ГА + НС)

Для осуществления поиска с использованием генетического алгоритма необходимо вычислять значения целевой функции для каждого сгенерированного решения в текущей популяции. Схема применения стандартного ГА для поиска экстремума целевой функции представлена на рис. 1.

Для эффективной реализации эволюционного поиска ГА необходимо генерировать достаточное число новых решений на каждой итерации. Исходя из этого, классическую схему оптимизации целесообразно использовать в случаях, когда значение целевой функции рассчитывается сравнительно быстро. При этом высокой скорости оптимизации можно добиться, ограничившись лишь заложенными в ГА механизмами эволюционного поиска. В тех же случаях, когда стоимость вычисления целевой функции возрастает, становится целесообразным усложнение алгоритма поиска в целях сокращения числа итераций, необходимых для нахождения

решения заданной точности. Как уже отмечалось, для этого весьма эффективно могут применяться гибридные ГА. Перспективной гибридной схемой является комбинация ГА с аппроксиматором целевой функции на основе нейронной сети. В частности, с ее помощью можно решать описанную задачу оптимизации в условиях неповторяемого эксперимента. Он характеризуется наличием априорно заданного множества приближенных решений, т. е. наборов параметров с соответствующими значениями целевой функции $\{X^i, Q^i\}_r$. Неповторимость эксперимента означает невозможность вычисления значений этой функции для новых решений, генерируемых ГА.

Существует ряд работ, посвященных этому вопросу [12–14], в которых описывается схожая идея применения НС типа "многослойный персептрон" совместно с ГА. Известное множество приближенных решений (обычно это данные наблюдения за объектом исследования) используется в качестве обучающей выборки. Перед началом оптимизации проводится обучение НС. На ее вход подаются векторы оптимизируемых параметров, а на выходе сеть должна аппроксимировать значения целевой функции. Далее запускается ГА, для которого значения целевой функции вычисляются нейронной сетью. Найденное решение представляет собой прогноз оптимального решения (рис. 2).

Проблемой метода является необходимость в проверке качества найденного решения, так как нейросетевая аппроксимация может плохо описывать целевую функцию, соприкасаясь с ней только в точках из обучающей выборки. На качество аппроксимации влияет выбранная структура НС, а также комплекс связанных с обучением проблем, таких как репрезентативность обучающей выборки, число обучающих примеров, коррелированность их между собой и т. п. В связи с этим авторы [12] рекомендуют использование ансамбля НС для повышения достоверности прогноза.

Проверка оптимальности решения обычно требует повторения эксперимента и может быть реализована с помощью численного моделирования, наблюдения за откликом реального объекта или другими зависящими от задачи способами. Если найденное решение неудовлетворительно, то весь цикл, начиная с создания и обучения НС, может быть повторен. Проверка дает возможность пополнять известное множество приближенных решений. Недостатком такой схемы является отсутствие целенаправленной стратегии повторения циклов алгоритма (рис. 3).

В данной работе предлагается подход к решению задачи оптимизации общего вида в случае повторяемого эксперимента, когда есть возможность вычисления значений целевой функции в процессе оптимизации. В инженерных приложениях это обычно означает возможность математического моделирования объекта исследования. В зависимости от сложности модели, расчетного периода и других факторов время вычисления одного значения целевой функции, даже на современных системах распределенных вычислений (кластерах), может быть велико. Например, время гидродинамического моделирования нефтяного месторождения может исчисляться сутками. Применение ГА в чистом виде для адаптации подобных моделей может оказаться ресурсоемким подходом вследствие большого числа генерируемых новых решений. Поэтому необходимо стремиться умень-

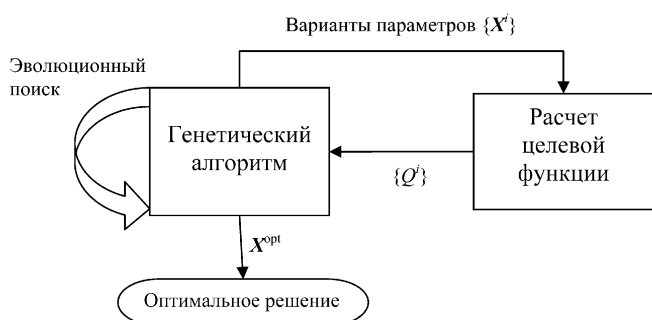


Рис. 1. Классическая схема использования ГА для поиска оптимума функции

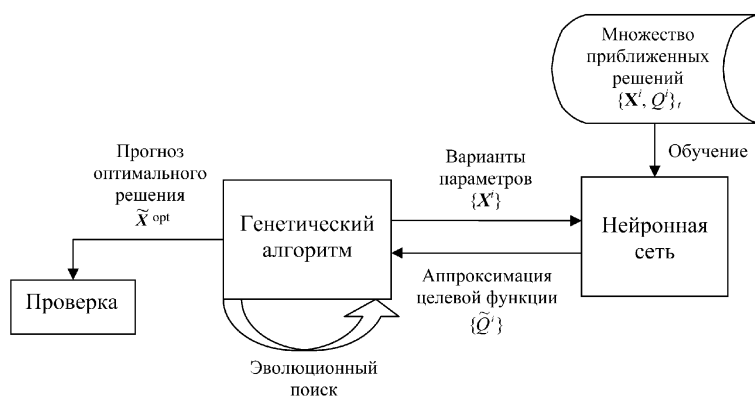


Рис. 2. Гибридная схема ГА + НС для неповторяемого эксперимента

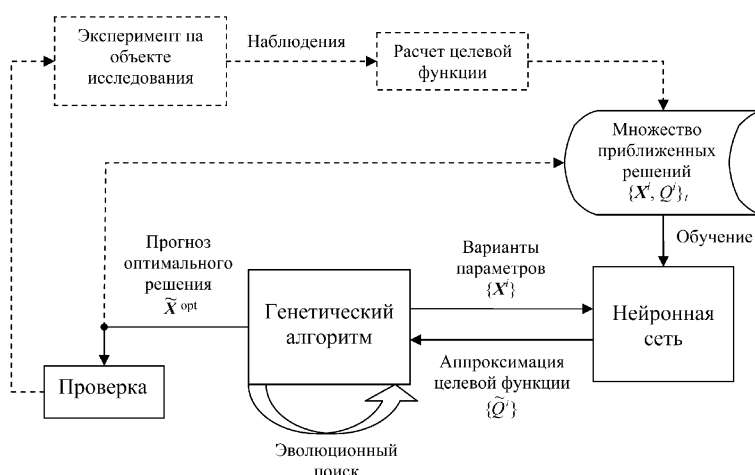


Рис. 3. Цикл взаимодействия гибридного ГА и объекта исследования с проверкой оптимальности решения

шить их число при условии достижения той же или лучшей точности результата.

С учетом этого критерия в данной работе разработана усовершенствованная гибридная нейросетевая схема ГА. В ее основу положен классический подход к оптимизации (см. рис. 1) с использованием основного ГА, для которого рассчитываются реальные значения целевых функций. Он манипулирует текущей популяцией решений и обеспечивает устойчивый поиск на основе эволюционных механизмов. На каждой итерации основного ГА задействуется дополнительный нейросетевой контур, целью которого является получение прогноза оптимального решения. Схема работы этого контура соответствует описанной для случая неповторяемого эксперимента (см. рис. 2). Она включает в себя обучение НС и поиск предположительно оптимального решения вспомогательным ГА, для которого целевые функции вычисляет НС. Обучающая выборка для НС формируется из множества всех приближенных решений, просчитанных к данному моменту времени. Таким образом, основной ГА задает общую стратегию оптимизации, пополнения обучающей выборки и обучения НС на каждой итерации, которая отсутствовала в схеме на рис. 3. Она дает возможность постоянно улучшать качество нейросетевой аппроксимации.

Найденный нейросетевым контуром вектор параметров добавляется в текущую популяцию основного ГА. В случае качественного прогноза добавленное решение

окажется лучшим среди всех остальных особей и будет чаще всех выбрано селекцией для создания потомков. Таким образом, эволюционный поиск будет направлен в область лучших решений. Если же прогноз окажется неудачным, основной ГА продолжит поиск на основе своих механизмов, сохранив текущих особей-лидеров. Разработанная гибридная схема изображена на рис. 4. Блок-схема предлагаемого алгоритма представлена на рис. 5.

В качестве инструмента аппроксимации в данной работе применялась радиально-базисная нейронная сеть (РБНС) [15, 16]. Выбор структуры такой сети заключается в подборе числа нейронов в единственном скрытом слое и позиций центров соответствующих радиально-базисных функций. Получить хорошее начальное приближение этих позиций позволяет подход на основе кластеризации входных данных, используемый в данной работе. Веса линейных нейронов выходного слоя РБНС можно определить, решив задачу наименьших квадратов для достижения минимума ошибки обучения. Однако такой способ обычно не обеспечивает хорошей аппроксимации на новых данных. Поэтому в данной работе веса обоих слоев РБНС уточняются с помощью алгоритма обратного распространения ошибки. К преимуществам РБНС можно отнести упрощенную процедуру выбора структуры, а также гладкость получаемых после обучения аппроксимаций. Как недостаток можно рассматривать локальность этих аппроксимаций [15].

Алгоритм обучения РБНС заключался в начальном нахождении центров каждой радиальной функции с помощью быстрой кластеризации по k средним (k -means), после чего веса обоих слоев корректировались по алгоритму обратного распространения ошибки. Использовалась модифицированная версия алгоритма k -means, всегда сходящаяся к квазиоптимальным положениям центров [17]. Во второй фазе обучения также использовалась модификация стандартного алгоритма обратного распространения, известная под названием RPROP [18].

В свете разработанного алгоритма ГА + НС необходимо отметить работу [19], где рассматривалась задача минимизации целевой функции с применением аналогичной гибридной схемы. Отличие заключалось в способе использования и типе НС (применялся многослойный персептрон). Если в данной работе НС применяется для решения прямой задачи (идентификации системы), то в работе [19] она обучалась решать обратную задачу, т. е. из значения целевой функции получать соответствующий вектор ее аргументов. В связи с этим схема работы нейросетевого контура также отличалась. На каждой итерации главного ГА из лучшего на данный момент значения целевой функции вычиталась небольшая величина. Полученное значение подавалось на вход обученной НС, которая на выходе давала соответствующий этому значению прогноз вектора параметров. Этот прогноз после проверки заменял худшую особь в популяции главного ГА.

Такая схема лишена необходимости использования вспомогательного ГА в нейросетевом контуре (НС сразу выдает предположительно лучшее решение). Однако она

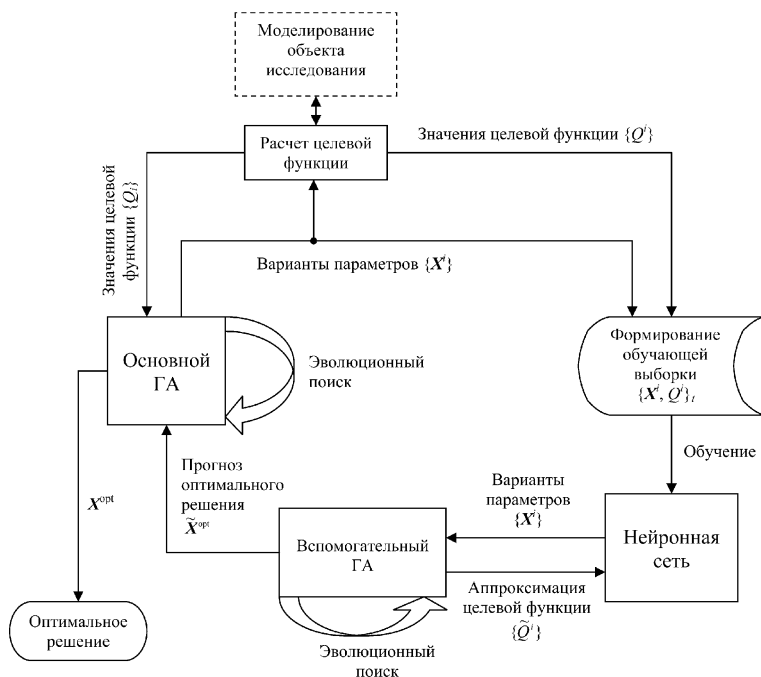


Рис. 4. Предложенная гибридная схема ГА + НС для повторяемого эксперимента

имеет очевидный недостаток, заключающийся в некорректности обратной задачи, решаемой НС. Одному и тому же значению целевой функции может соответствовать несколько (или даже бесконечно много) различных значений ее аргументов. В частности, это может привести к тому, что ошибка обучения не опустится ниже не-

которого значения вследствие противоречащих друг другу изменений весов. Однако несмотря на это авторам удалось добиться существенного увеличения скорости поиска ГА с бинарным кодированием на тестовых функциях. Практическую значимость полученных результатов снижает небольшая размерность протестированных функций — две переменные для функции Растригина, три — для функции Розенброка.

Гибридная схема ГА и НС, предложенная в данной работе, лишена потенциальных недостатков работы [19]. Она не зависит от наличия обучающей выборки для НС (наблюдений за объектом исследования) и может быть применена для решения любых задач оптимизации. Эволюционный поиск гибридного ГА может быть значительно ускорен правильным прогнозом нейросетевого контура, в худшем случае эффективность поиска остается на уровне стандартного ГА. Однако необходимо учитывать, что предложенный алгоритм целесообразно использовать только в случаях, когда ресурсоемкость вычисления значения целевой функции значительно превышает ресурсоемкость одного цикла работы нейросетевого контура.

Задача адаптации относительных фазовых проницаемостей модели резервуара

Одними из наиболее важных параметров, которые необходимо знать при построении гидродинамической модели, являются относительные фазовые проницаемости (ОФП) пористой среды для воды и нефти. Относительная проницаемость указывает на способность нефти и воды одновременно течь в пористой среде и определяется как отношение эффективной проницаемости фазы к абсолютной (когда порода заполнена только одной фазой). При правильном определении значений этих параметров получают характеристику комплексного влияния на проницаемость порового пространства таких свойств как смачиваемость, распределение флюидов и насыщенность. Выбор кривых ОФП также оказывает решающее влияние на прогнозирование хода процесса заводнения.

Для представления ОФП использовалась следующая математическая модель*. Пусть ОФП воды и нефти задаются выражениями

$$K_{rw}(S_w) = M_{rw} F \left(\frac{S_w - S_{wmin}}{S_{orw} - S_{wmin}}, \alpha_{w,1}, \alpha_{w,2}, \beta_w \right);$$

$$K_{ro}(S_w) = M_{ro} F \left(1 - \frac{S_w - S_{cr}}{S_{orw} - S_{cr}}, \alpha_{o,1}, \alpha_{o,2}, \beta_o \right);$$

$$M_{rw} = K_{rw}(S_{orw}), M_{ro} = K_{ro}(S_{wmin});$$

$$S_{wmin} \leq S_{cr} < S_{orw}$$

где S_w — водонасыщенность, $S_w \in [0,1]$; $K_{rw}(S_w)$ — ОФП воды; $K_{ro}(S_w)$ — ОФП нефти; S_{wmin} — точка минималь-

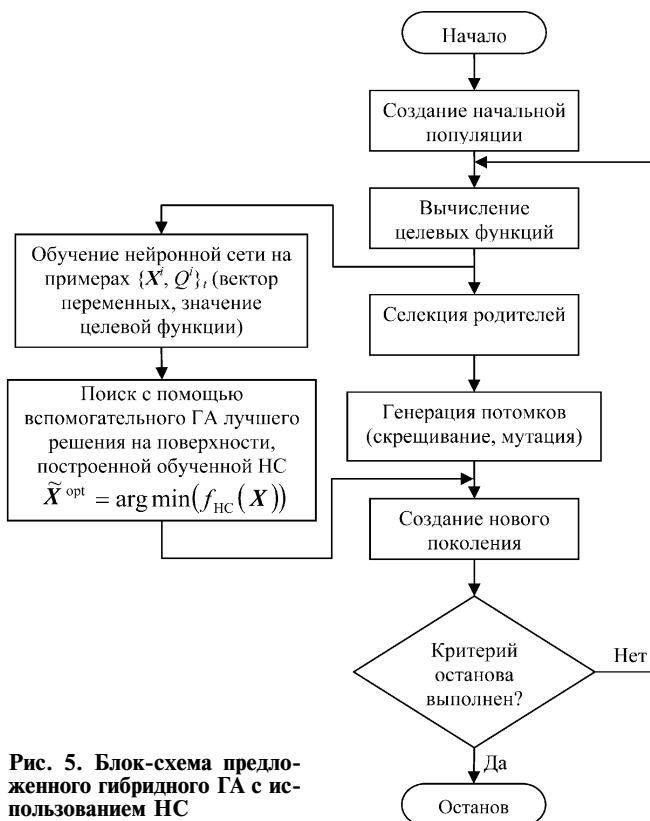


Рис. 5. Блок-схема предложенного гибридного ГА с использованием НС

* Эта модель ОФП предложена А. В. Безруковым (ООО "РН-УфаниПИНефть") и реализована в препроцессоре "Магма", входящем в состав пакета гидродинамического моделирования "BOS"

ной водонасыщенности; S_{cr} — точка начальной водонасыщенности (связанной воды); S_{orw} — точка остаточной нефтенасыщенности; M_{ro} — ОФП нефти при минимальной водонасыщенности; M_{rw} — ОФП воды в точке остаточной нефтенасыщенности; $\alpha_{p,i}$, $\beta_p \in [0, \infty]$, $p = \{w - \text{water}, o - \text{oil}\}$, $i = 1, 2$ — параметры, задающие форму кривых ОФП; $F(x; \alpha_1, \alpha_2, \beta) = x^\beta x^{\alpha_1} + (1 - x^\beta)x^{\alpha_2}$. Величина x^β здесь играет роль весовой функции, определяющей влияние слагаемых x^{α_1} и x^{α_2} на результат. Фактически, веса при этих слагаемых определяются значением β .

Форма кривых ОФП воды и нефти задается соответственно показателями степеней $\alpha_{w,1}$, $\alpha_{w,2}$, β_w , $\alpha_{o,1}$, $\alpha_{o,2}$, β_o , а также масштабирующими множителями M_{ro} и M_{rw} . Все эти параметры можно было варьировать в процессе адаптации. Для каждого из них предварительно задавались некоторые значения по умолчанию, которые подставлялись, если соответствующий параметр не адаптировался. Точки $S_{w \min}$, S_{cr} , S_{orw} задавались экспертно и не изменялись. Во время адаптации для каждой хромосомы по перечисленным восьми параметрам рассчитывались ОФП и проводилось моделирование резервуара.

Тестирование гибридной схемы ГА + НС

Предложенный алгоритм тестировался на задаче определения ОФП воды и нефти для тестовой модели резервуара, включающей 98 скважин. На основе экспертных ОФП были сгенерированы данные по дебитам жидкостей для каждой скважины, которые были приняты в качестве эталонных и использовались для сравнения с расчетными значениями в процессе адаптации. Моделирование проводилось в среде препостпроцессора "Магма" с использованием симулятора UfaSolver, входящих в состав программного комплекса гидродинамического моделирования "BOS" (разработка ООО "РН-УфаниПИНефть").

В эксперименте сравнивалась скорость минимизации целевой функции с помощью обычного ГА и ГА + НС. Вся модель соответствовала одному региону ОФП, т. е. ОФП были везде одинаковыми и адаптировались восемь параметров для их описания. Результаты расчетов усреднялись по пяти запускам адаптации. На каждом запуске

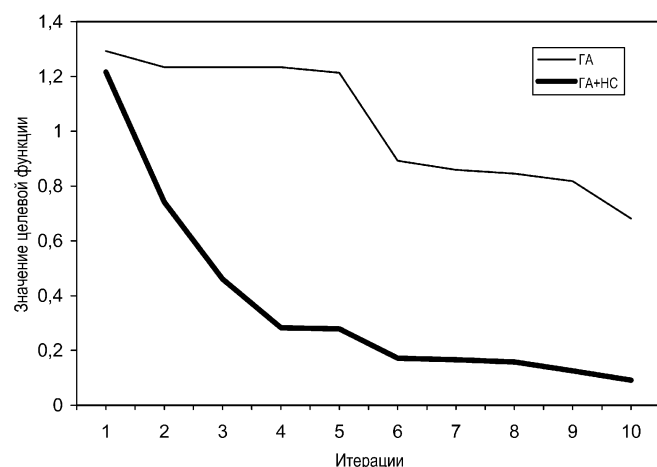


Рис. 6. Сравнение производительности ГА и ГА + НС

просчитывалось 10 поколений ГА. Результаты тестирования представлены на рис. 6.

Основные параметры ГА были установлены такими: 10 особей в популяции, элитная стратегия отбора с сохранением одной лучшей особи; вероятность скрещивания 0,7; вероятность мутации 0,1; оператор скрещивания — SBX; оператор мутации — неравномерная мутация Михалевича с параметром 4,5. В популяцию подмешивалась одна хромосома, полученная с помощью нейросетевого контура.

Из графиков видно, что применение предложенного гибридного алгоритма позволило значительно увеличить точность найденного решения. На третьей итерации ГА + НС минимизировал целевую функцию до значения, которое не было достигнуто стандартным ГА за 10 итераций. В течение первых 4–6 итераций он смог быстро улучшить решение, благодаря нейросетевым прогнозам и механизмам эволюционного поиска. Далее, судя по динамике оптимизации, ГА + НС оказался в области притяжения некоторого, возможно глобального, оптимума.

Заключение

В работе были проведены эксперименты по проверке эффективности разработанного генетического алгоритма с использованием нейронной сети на задаче определения кривых ОФП. По результатам экспериментов гибридный ГА за три итерации достиг точности решения, которая не была найдена стандартным ГА за 10 итераций. Предложенный алгоритм подтвердил свою эффективность и предоставляет возможность сокращения количества расчетов модели резервуара по сравнению с обычным ГА, что существенно снижает ресурсоемкость задачи адаптации.

В настоящее время ведутся работы по дальнейшему совершенствованию гибридного ГА + НС. В первую очередь, они направлены на разработку улучшенных алгоритмов обучения РБНС, позволяющих получать более качественную нейросетевую аппроксимацию целевой функции.

Список литературы

1. Gomez S., Gosselin O., Barker J. W. Gradient-Based History-Matching With a Global Optimization Method // Paper SPE 56756 presented at the 1999 SPE Annual Technical Conference and Exhibition, Houston, 3–6 October.
2. Soleng H. H. Oil Reservoir Production Forecasting with Uncertainty Estimation using Genetic Algorithms // Proc. Congress of Evolutionary Computing, Washington, DC 1999.
3. Schulze-Riegert R. W., Axmann J. K., Haase O., Rian D. T., You Y.-L. Evolutionary Algorithms Applied to History Matching of Complex Reservoirs // SPE Reservoir Evaluation & Engineering, 2002.
4. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы / Под ред. В. М. Курейчика. 2-е изд., испр. и доп. М.: Физматлит, 2006. 320 с.
5. Вороновский Г. К., Махотило К. В., Петрашев С. Н., Сергеев С. А. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. Харьков: Основа, 1997. 112 с.
6. Whitley D. A genetic algorithm tutorial // Statistics and Computing. 1994. Vol. 4. P. 65–85.
7. Herrera F., Lozano M., Verdegay J. Tackling real-coded genetic algorithms: operators and tools for the behavior analysis // Artificial Intelligence Review. 1998. Vol. 12. N 4.
8. Herrera F., Lozano M., Sanchez A. Hybrid crossover operators for real-coded genetic algorithms: an experimental study // Soft Computing. 2005. Vol. 9. N 4.

9. Whitley D., Rana S., Heckendorn R. B. The island model genetic algorithm: on separability, population size and convergence // Journal of computing and information technology. 1999. Vol. 7(1). P. 33—47.
10. Eiben A. E., Hinterding R., Michalewicz Z. Parameter control in evolutionary algorithms // IEEE Trans. Evolutionary Computation. 1999. N 3(2). P. 124—141.
11. Тенев В. А., Паклин Н. Б. Гибридный генетический алгоритм с дополнительным обучением лидера // Интеллектуальные системы в производстве. 2003. № 2. Ижевск: Изд-во ИжГТУ, 2003. С. 181—206.
12. Ling Wang. A hybrid genetic algorithm-neural network strategy for simulation optimization // Applied Mathematics and Computation. 2005. 170.
13. Jan-Tai Kuo, Ying-Yi Wang, Wu-Seng Lung. A hybrid neural-genetic algorithm for reservoir water quality management // Water research. 2006. 40.

14. Xiaodong Huo, Zhongsheng Xie. A novel channel selection method for CANDU refueling based on the BPANN and GA techniques // Annals of nuclear energy. 2005. 32.
15. Хайкин С. Нейронные сети: полный курс. 2-е изд. М.: Издательский дом "Вильямс", 2006. 1104 с.
16. Poggio T., Girosi F. A theory of networks for approximation and learning // Proc. of IEEE. 1990. Vol. 78. P. 1481—1497.
17. Chinrungrueng C., Sequin C. H. Optimal adaptive k-means algorithm with dynamic adjustment of learning rate // IEEE Transactions on Neural Networks. 1995. 6. P. 157—169.
18. Riedmiller M., Braun H. A direct adaptive method for faster backpropagation learning: the RPROP algorithm // Proc. of the IEEE International Conference on Neural Networks. San Francisco, CA. 1993. P. 586—591.
19. Javadi A. A., Farmani R., Tan T. P. A hybrid intelligent genetic algorithm // Advanced Engineering Informatics. 2005. 19.

УДК 004.3

А. А. Петунин, канд. техн. наук,
Уральская государственная архитектурная
художественная академия, г. Екатеринбург,
Э. А. Мухачева, д-р техн. наук, проф.,
А. С. Филиппова, канд. физ.-мат. наук,
Уфимский государственный авиационный
технический университет

Метод прямоугольной аппроксимации для решения задач нерегулярного фигурного раскроя—упаковки

В качестве основной рассматривается задача размещения геометрических объектов произвольной формы. Для ее решения часто применяется аппроксимация фигур прямоугольниками. Предлагается способ выделения класса задач, для решения которых данный подход оказывается целесообразным.

Введение

Под задачами раскроя—упаковки (Cutting & Packing, C & P) понимается широкий класс проблем, допускающих различное прикладное толкование. Общим для этого класса является наличие двух групп объектов. Между элементами этих групп устанавливается и оценивается соответствие. Впервые качественная типология в области раскроя—упаковки проведена в 1991 г. немецким ученым Н. Dykhoff [1]. Она принята в мировой практике и используется при изучении моделей и методов решения задач раскроя—упаковки. Разнообразие моделей определяется, прежде всего, фактором геометрии. Различаются задачи линейного (одномерного), прямоугольного (двумерного) и параллелепипедного (трехмерного) раскроя—упаковки. Среди этих задач выделяются гильо-

тинный раскрой и упаковка. Особо выделены проблемы нестинга — размещения деталей сложной геометрической формы в заданных областях. Для них на первый план выступают информационные проблемы задания фигур, учета и обеспечения их неперекрывания, кодировки и др. Задачи C & P являются типичными представителями NP-трудных проблем и для их решения (включая проблемы нестинга) применяются общие подходы: точные методы, простые эвристики и метаэвристики. Ввиду неполиномиальной сложности точных алгоритмов авторами многих работ уделяется значительное внимание приближенным методам и эвристикам.

Причина растущего интереса к задачам раскроя—упаковки состоит в их принадлежности к NP-трудным проблемам, а также в широкой применимости результатов в различных предметных областях.

Систематическое изучение проблемы раскроя—упаковки заложено в книге ленинградских ученых Л. В. Канторовича и В. А. Залгаллера в 1951 г. [2] и развито в их более поздних работах [3], а также в статьях Р. Gilmore & R. Gomory [4]. Они впервые применили к решению C & P метод линейного программирования с генерацией столбцов на каждом шаге процесса. G. Scheithauer & J. Terno использовали полученный непрерывный оптимум для построения целочисленного решения [5]. С этой целью после округления снизу непрерывного решения формируется малая остаточная задача. Последняя решается с помощью эвристик до достижения нижней границы, или процесс останавливается на лучшем из полученных решений. В 1981 г. S. Baum & Jr. Trotter отметили, что в большинстве случаев разрыв между оптимальными значениями непрерывной релаксации и целочисленной задачей не превышает 1, т. е. округленное решение оптимально [6].

Различаются задачи прямоугольной упаковки и гильотинного раскроя. В том и другом случаях требуется разделить большой прямоугольник на малые таким образом, чтобы стороны прямоугольников были параллельны сторонам большого прямоугольника, и выбранная функция цели достигала минимума. Задачи упаковки и гильотинного раскроя различаются технологией разделения. Гильотинность предполагает возможность только

сквозных резов, параллельных кромкам раскраиваемого материала. Задачи гильотинного раскроя, которые берут свое начало от задачи одномерного раскроя, рассматривались как обобщение последней в первых работах Л. В. Канторовича & В. А. Залгаллера и P. Gilmore & R. Gomory. Методы, алгоритмы и технологии, разработанные для задач одномерного раскроя, адаптированы на случай гильотинного (двух- и трехмерного) раскроя. Подробно эта задача в условиях массового и серийного производства описана в работах Э. А. Мухачевой [7] в России и J. Terno, R. Lindeman & G. Scheithauer [5] в Германии. Метод склейки И. В. Романовского, разработанный вначале для задач линейного раскроя, также был обобщен на гильотинный случай [8].

Большинство существующих точных методов решения задачи прямоугольного негильотинного раскроя сводятся к перебору всего множества допустимых решений. Методы улучшенного перебора объединены для этих задач под названием "метода ветвей и границ". В 1986 г. Ю. Г. Стоян и С. В. Яковлев описали общую схему метода и привели основные алгоритмы [9]. Для получения нижней оценки они предложили использовать релаксацию задачи, переходя от множества допустимых решений к некоторому его подмножеству. Показана возможность использования этого алгоритма для задачи прямоугольной упаковки в полосу и сделан ряд предложений, позволяющих применять алгоритм более эффективно, например, введен способ ранжирования элементов.

Несмотря на многообразие различных подходов к созданию точных алгоритмов размерность решаемых задач редко превосходит 20. Это связано с тем, что пока не найдено способа определения эффективной нижней границы, и для доказательства оптимальности приходится перебирать все возможные варианты. Значительный вклад в теорию и практику моделирования размещения геометрических объектов произвольной формы внесен Л. Б. Беляковой (Россия) и научной школой Ю. Г. Стояна (Украина, Институт машиностроения АН). Введенные на базе работ В. Л. Рвачева и позднее Ю. Г. Стояна R- и Ф-функции как средства математического и компьютерного моделирования внесли существенный вклад в решение проблемы формализации 2D- и 3D-задач C & P и разработки методов их решения [10]. Впервые появились работы по применению элементов интервального анализа в геометрическом проектировании [11].

К многопроходным эвристикам относится метод последовательного уточнения оценок (*Sequentative Value Correction*, SVC), берущий начало от идеи объективно-обусловленных оценок Л. В. Канторовича [12]. Метод SVC реализуется по модифицированной схеме FFD с процедурами приоритета и повторения. Упорядочивание элементов основано на экономическом смысле двойственных переменных в линейном программировании. Метод можно назвать общим для решения задач C & P. Он применяется в случаях линейного и гильотинного раскроя, двух- и трехмерной упаковки, а также и для решения задач нестинга [13]. Кроме того, метод использовался германскими авторами при разработке гибридных алгоритмов [14].

Простые эвристики широко применяются для решения задач нестинга. Долгое время это был единственный способ построения допустимых карт раскроя. Широкое распространение получили методы аппроксимации и декомпозиции, сводящие проблему к последовательному

решению нескольких простых задач. Например, для случая объектов, близких к прямоугольным, используется следующая схема:

- нахождение описывающего объект прямоугольника минимальной площади;
- упаковка прямоугольников в заданную область;
- уплотнение исходных объектов после "отброса" прямоугольных оболочек.

Этот метод достаточно простой в части определения условий взаимного перекрытия. Он давно известен и практикуется до сих пор. В общем случае при решении задач нестинга на первый план выступают проблемы представления исходной информации, движения объектов и проверки условий их непересечения. В современных работах используются две группы способов представления информации: аналитический и дискретный. Среди первых наибольшее применение получил метод R-функций В. П. Рвачева. Дискретными являются радиосографический и линейный способы аппроксимации [13]. Перспективными оказались способы дискретно-логического и цепного кодирования. Специальные эвристические методы используют пообъектное размещение, представляющее собой модификацию метода "первый подходящий". При этом произвольные сдвиги, повороты и проверки взаимного перекрытия реализуются на основе годографа — понятия, введенного в 70-е годы Ю. Г. Стояном.

Для получения приближений к локальным минимумам Ю. Г. Стояном разработан метод последовательно-одиночного размещения. Он представляет собой модификацию метода Гаусса—Зайделя. Размещаемые объекты жестко упорядочены. Первый объект размещается так, чтобы минимизировать функцию цели. Аналогично размещается второй объект относительно свободной области и неподвижного состояния первого объекта. Метод применяется многими авторами в постсоветском пространстве и за рубежом. Перспективным направлением является его применение в комбинации с методом асимптотического перебора локальных экстремумов и в составе метаэвристик в качестве декодера.

В целом задачи нестинга занимают особое место в проблеме C & P. Разработкой и исследованием этих задач занимаются несколько коллективов: Харьковская школа Ю. Г. Стояна; Институт алгоритмов и научных исследований в Германии, R. Heckman & T. Lengauer; W. Milencovic в США; K. Dousland & B. Dousland в Великобритании; J. Blazewich P. Hawryluk & K. Walkowiak, Польша; M. Gomes & J. Olivera, Португалия; H. Yanessa, Бразилия. В России задачами нестинга заняты М. А. Верхотуров, В. В. Мартынов, В. Д. Фроловский, А. А. Петунин, Ю. Г. Павлов и другие авторы.

О применении алгоритмов прямоугольного раскроя—упаковки для решения задач нерегулярного фигурного раскроя

Как уже отмечалось выше, один из известных подходов к решению оптимизационных задач раскроя материала для случая двумерных объектов заключается в аппроксимации объектов прямоугольниками с последующим использованием методов прямоугольного раскроя—упаковки. Практика показывает, что разработанное в настоящее время алгоритмическое и программное обеспечение расчетов прямоугольного раскроя—упаковки позволяет решать большинство реальных задач более эффективно, нежели при "ручных" расчетах [15, 16].

Вместе с тем, для решения многих практических задач нерегулярного фигурного раскроя все еще требуется применение интерактивных методов проектирования, поскольку существующие автоматические алгоритмы зачастую не могут обеспечить необходимый пользователю коэффициент использования материала (КИМ). В качестве аппроксимирующей объект фигуры естественно выбирать прямоугольник минимальной площади. Если по условиям задачи поворот объектов в области упаковки (или на раскраиваемом материале) запрещен, то задача построения прямоугольника минимальной площади совпадает с задачей построения единственного описанного (габаритного) прямоугольника и является достаточно простой независимо от того, какая форма представления геометрической информации об объекте используется — векторная или растровая.

Если на ориентацию объектов никаких ограничений не накладывается, то искомым прямоугольником ищется среди всего множества описанных прямоугольников. Известен следующий простой алгоритм построения прямоугольника минимальной площади:

- строится выпуклая оболочка многоугольника;
- строится конечное множество описанных прямоугольников, таких что одна из сторон прямоугольника содержит одну из сторон выпуклой оболочки многоугольника;
- среди данного конечного множества ищется прямоугольник минимальной площади.

Если границы объекта состоят не только из отрезков прямых, выбор искомого прямоугольника можно осуществлять также на дискретном множестве габаритных прямоугольников, построенных для объекта при его пошаговом повороте на некоторый угол. Однако в данном случае решение задачи аппроксимации будет являться, в общем случае, приближенным. Вместе с тем, вычислительные характеристики современных компьютеров позволяют решать эту задачу с любой необходимой для практики точностью и за время, незаметное для пользователя.

Обоснование аппроксимации прямоугольниками объектов сложной формы

Очевидно, что прием аппроксимации непрямоугольного объекта прямоугольником целесообразно применять только в том случае, если отношение их площадей близко к единице. В противном случае более эффективными становятся алгоритмы раскроя—упаковки, разрабатываемые для случая двумерных объектов сложной формы. На вопрос: "насколько близко к единице должно быть отношение площадей объектов к площади содержащего их прямоугольника?" ответ может дать только вычислительный эксперимент. Для такого эксперимента были выбраны два алгоритма, предназначенные для решения задач раскроя—упаковки прямоугольных объектов и объектов, границы которых состоят из отрезков прямых и дуг окружностей. Оба эти алгоритма входят в состав библиотеки Nest Class Library (NCL) [17] и ориентированы на условия единичного производства, когда геометрические параметры размещаемых объектов, в общем случае, могут быть различными.

Для описания эксперимента введем два понятия:

- Коэффициентом прямоугольности K_p двумерного объекта назовем величину, равную отношению площади

заготовки к площади прямоугольника минимальной площади, содержащего данный объект:

$$K_p = \frac{S}{P},$$

где S — площадь объекта, P — площадь содержащего объект прямоугольника.

- Коэффициентом прямоугольности задания на раскрой назовем величину, равную средней величине коэффициентов прямоугольности отдельных объектов в задании:

$$K_{\text{прям}} = \frac{\sum_{i=1}^n K_i p}{n},$$

где n — число объектов в задании на раскрой.

Было сгенерировано 1000 заданий на раскрой.

Раскраиваемый материал имел прямоугольную форму (1000 × 2000 мм).

В эксперименте использовались 100 прямоугольных объектов (объекты были сгенерированы автоматически, габаритные размеры объектов находились в пределах 300 мм), а также 100 объектов, границы которых состояли из отрезков прямых и дуг окружностей, с теми же ограничениями на габаритные размеры. В качестве фигурных объектов использовались реальные машиностроительные заготовки.

Задания с различными коэффициентами прямоугольности $K_{\text{прям}}$ генерировались автоматически. Для генерации предварительно было выбрано 10 значений коэффициентов прямоугольности (от 0,1 до 1), и задания генерировались таким образом, чтобы их параметры совпадали с выбранными с точностью ±0,02. Например, генерировались 100 заданий с коэффициентом прямоугольности из [0,48; 0,52]. Для определения значения КИМ, соответствующего параметру коэффициента прямоугольности 0,5, принимался средний результат КИМ для всех 100 заданий.

Результаты данного эксперимента приведены на рис. 1.

Как видно на графике, алгоритм прямоугольного раскроя—упаковки становится эффективнее алгоритма нерегулярного фигурного раскроя при значениях коэффициента прямоугольности больше 0,85. Следует, однако, отметить, что данное значение является усреднен-

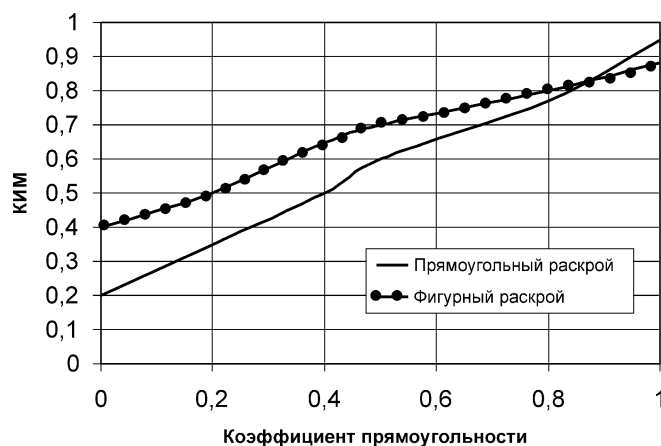


Рис. 1. Зависимость КИМ от коэффициента прямоугольности задания на раскрой

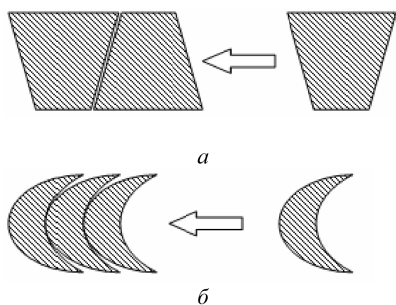


Рис. 2. Способы формирования *bit*-ов: *a* — с поворотом деталей; *б* — без поворота деталей

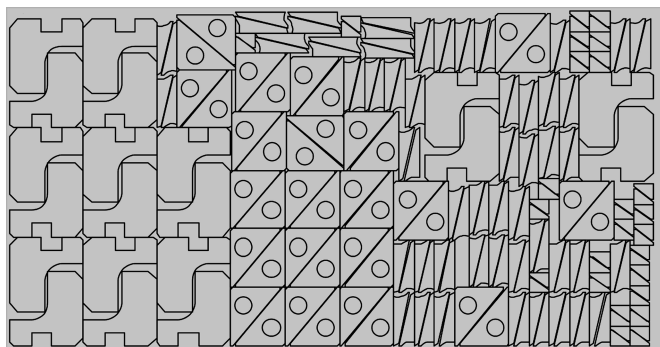


Рис. 3. Пример использования алгоритма прямоугольного раскроя—упаковки для решения задачи нерегулярного фигурного раскроя

ным, и в реальных задачах раскроя при значениях коэффициента прямоугольности $[0,80; 0,87]$ рекомендуется применять одновременно алгоритмы как прямоугольного, так и фигурного раскроя.

При решении задач нерегулярного фигурного раскроя коэффициент прямоугольности задания может быть значительно повышен за счет широко используемого на практике приема: объединения фигурных объектов (деталей) в блоки или куски (*bit*). Под *bit* в этом случае понимается набор деталей, положения которых зафиксированы относительно друг друга. При размещении такой *bit* ведет себя как одна деталь, т. е. все преобразования по перемещению/вращению выполняются одновременно со всеми деталями, входящими в блок. *Bit*-ы в основном состоят из однотипных деталей, но могут содержать и детали различной конфигурации. Способы создания блоков деталей во многом схожи с методами проектирования раскройных карт.

На рис. 2 показаны два способа построения *bit*-ов для однотипных деталей. В первом случае используется поворот детали на 180° , во втором формирование *bit*-а происходит без поворота детали. После завершения формирования *bit*-ов прямоугольники минимальной площади естественно строить для каждого из них.

Ясно, что прием формирования блоков вместе с повышением коэффициента прямоугольности задания также расширяет и класс заданий на раскрой, для которых целесообразно применять оптимизационные алгоритмы прямоугольного раскроя—упаковки. На рис. 3 приведен результат раскроя, полученный методом прямоугольного раскроя—упаковки с использованием *bit*-ов деталей.

Заключение

- Предложена аппроксимация задачи нестинга прямоугольным раскроем—упаковкой. Проведен эксперимент, результаты которого выделяют класс задач, для которых прямоугольная аппроксимация целесообразна.
- Прямоугольная аппроксимация полезна для приближенного вычисления подетальных норм расхода материала. При этом класс задач можно значительно расширить в рамках заданных границ.
- Следующим этапом является аппроксимация сложных объектов прямоугольно-ориентированными многоугольниками. Этой проблеме будет посвящена другая статья.

Список литературы

1. Dykhoff H. A typology of cutting and packing problems // European Journal of Operational research. 1990. Vol. 44. P. 145—159.
2. Канторович Л. В., Залгаллер В. А. Расчет рационального раскроя материалов. Л.: Лениздат, 1951.
3. Канторович Л. В., Залгаллер В. А. Рациональный раскрой промышленных материалов. Новосибирск: Наука, СО. 1971. 299 с.
4. Gilmore P. C. and Gomory R. E. A Linear Programming Approach to the Cutting-stock Problem // Operations Research. 1961. 9. P. 849—859.
5. Terno J., Lindeman R., Scheithauer G. Zuschnittprobleme und ihre praktische Lösung. Leipzig. 1987.
6. Baum S., Trotter Jr. L. Integer rounding for polymatroid and branching optimization problems // SIAM J. Alg. Disc. Meth. 1981. 2(4). P. 416—425.
7. Мухачева Э. А. Рациональный раскрой промышленных материалов. Применение в АСУ. М.: Машиностроение. 1984. 176 с.
8. Романовский И. В. Алгоритмы решения экстремальных задач. М.: Наука, 1977. 88 с.
9. Стоян Ю. Г., Яковлев С. В. Математические модели и оптимизационные методы геометрического проектирования. Киев: Наукова думка, 1986. 268 с.
10. Стоян Ю. Г., Шайтхауер Г. и др. Класс поверхностей О-уровня Ф-функций точечных множеств с границей — окружность или многоугольник // Проблемы машиностроения. 2000. Т. 3. № 12. С. 117—123.
11. Романова Т. Е. Интервальное пространство $I_s^n R$ // Докл. НАН. Украина. 2000. № 9. С. 36—41.
12. Мухачева Э. А., Мухачева А. С., Белов Г. Н. Метод последовательного уточнения оценок: алгоритм и численный эксперимент для задачи одномерного раскроя // Информационные технологии. 2000. № 2. С. 11—17.
13. Верхотуров М. А. Задача нерегулярного раскроя плоских геометрических объектов: моделирование и расчет рационального раскроя // Информационные технологии. 2000. № 5. С. 37—42.
14. Belov G., Scheithauer G. A cutting plane algorithm for one-dimensional cutting stock problem with multiple stock lengths // European Journal of Operational Research. 2002. 141. P. 274—294.
15. Мухачева Э. А., Валеева А. Ф., Картак В. М., Мухачева А. С. Методы локального поиска оптимума в задачах ортогонального раскроя и упаковки: аналитический обзор и перспективы развития // Информационные технологии. Приложение. 2004. № 5. 32 с.
16. Филиппова А. С. Моделирование эволюционных алгоритмов решения задач прямоугольной упаковки на базе технологии блочных структур // Информационные технологии. Приложение. 2006. № 6. 32 с.
17. Петунин А. А., Куреннов Д. В., Полезов А. В. Об одном подходе к решению задач раскроя—упаковки // Вестник УГТУ—УПИ. Конструирование и технология изготовления машин: Сб. научных тр. Ч. 2. Екатеринбург: ГОУ ВПО УГТУ—УПИ. 2005. № 18(70). С. 212—216.



УДК 004.322.067

М. В. Аксарин,

ОАО "Институт сетевых технологий",
М. В. Тарасюк, канд. техн. наук, доц.,
Санкт-Петербургский государственный
институт информационных технологий,
механики и оптики

Модель скрытого канала реляционной БД с многоуровневым доступом

Рассматриваются вопросы защиты БД от уязвимостей ссылочной целостности между отношениями разных классов секретности. Для оценки вероятностно-временных характеристик скрытых каналов и степени их опасности предлагается использовать модели систем массового обслуживания.

Введение

Общеизвестно, что требования по обеспечению безопасности информации могут ограничивать эффективность использования информационных технологий (ИТ), в том числе, производительность, удобство эксплуатации, модернизационную пригодность и т. д. За примерами можно обратиться в область разработки автоматизированных и информационных систем для государственного управления, управления и контроля экологически опасными объектами и т. д. Возможные ограничения эффективности зачастую диктуются формальными свойствами моделей "многоуровневой безопасности", которые обязательны для данного класса систем. В общих чертах исходные требования к реализации моделей безопасности задаются в руководящих документах, выпускаемых уполномоченными правительственными организациями¹.

Многоуровневая безопасность теоретически гарантирует конфиденциальность информации в условиях любого рода несанкционированных воздействий, в том числе, при использовании нарушителем недеklarированных возможностей кода, скрыто внедренных в приложение или его программное окружение. При этом защита обеспечивается не только от угроз нарушения конфиденциальности за счет несанкционированного досту-

па (НСД), но также и от скрытых каналов (СК) — при использовании на разных уровнях доступа общих информационных (скрытые каналы по памяти) и системных (скрытые каналы по времени) ресурсов [1].

В общем случае под скрытыми каналами часто понимают каналы доступа к информации, которые не включаются в модель безопасности и не фигурируют в описании политики безопасности системы. Практическое значение имеют скрытые каналы многоуровневых систем, в которых реализуется мандатная политика безопасности — определяются уровни безопасности и формальные правила взаимодействия процессов каждого из уровней, исключающие прямую утечку информации. Скрытые каналы, как правило, невозможно полностью устранить — возможно, лишь уменьшить их пропускную способность до заданной (условно) безопасной нормы (аналогично защите объектов вычислительной техники от утечки информации по техническим каналам), либо в некоторых случаях обнаружить факт активизации скрытого канала [2].

Далее рассматриваются скрытые каналы применительно к реляционным базам данных (БД) — наиболее сложной ИТ с точки зрения выполнения требований многоуровневой безопасности. В действительности, известные реализации многоуровневых СУБД, строго реализующие мандатные модели и так называемую "многозначность" данных, в том числе классические, доверительные модели [3], а также сегментная модель [4] исключают возникновение скрытых каналов. Все эти модели основаны на разделении многоуровневой БД на одноуровневые разделы, с которыми в изолированном режиме могут работать пользователи каждого уровня доступа. В результате значительно увеличивается информационная избыточность БД (пропорционально числу уровней доступа) и создаются объективные условия для нарушения целостности данных (одни и те же данные необходимо повторно вводить в систему для каждого уровня доступа). В случаях же отказа от полной поддержки многозначности, возникают скрытые каналы, вызванные наличием связей между записями БД на разных уровнях доступа.

Базовая модель скрытого канала

Рассмотрим проблему скрытых каналов (СК) в многоуровневых БД на примере отношения "Проект", которое представляет секретные и несекретные проекты некоторого учреждения (рис. 1). Решетка секретности отношения включает два уровня: несекретный ("lo") и секретный ("hi"). Отношение содержит идентификатор проекта "пр_ид", название проекта и его описание.

Идентификатор проекта "пр_ид" не может служить первичным ключом многоуровневого отношения "Проект", так как в этом случае требование уникальности первичного ключа приведет к образованию скрытого канала. Допустим, пользователь LEVEL_HI уровня "hi"

¹ В США это документы из так называемой "Радужной серии", а в Российской Федерации — руководящие документы Государственной технической комиссии (Федерального агентства по техническому и экспортному контролю).

Проект	
пр_ид	{lo, hi}
название	{lo, hi}
описание	{lo, hi}

Рис. 1. Многоуровневое отношение "Проект"

создает секретную запись с идентификатором "N". Допустим также, что пользователь LEVEL_LO уровня "lo", не зная об этой записи, попытается создать несекретную запись с таким же идентификатором "N". Такой запрос приводит к нарушению уникальности первичных ключей и не может быть исполнен, отказ же от его выполнения позволит пользователю LEVEL_LO сделать вывод о наличии идентификатора "N" на более секретных уровнях. Это классический случай СК на основе целостности первичных ключей. Блокирование таких СК в многоуровневых моделях данных достигается за счет введения истинных первичных ключей, которые, в дополнение к обычным ключам, включают в себя метку секретности. К возникновению СК приводят также ограничения ссылочной целостности многоуровневых отношений. СК на основе ссылочной целостности возникают при создании связи между записями на разных уровнях секретности. Подробная классификация и описание скрытых каналов в БД рассмотрена в работе [5]. В табл. 1 приведены условия возникновения скрытых каналов в БД с гранулярностью доступа уровня поля (метки устанавливаются на отдельные записи) и выше (метки устанавливаются на записи, атрибуты или отношения целиком) при разных правилах сохранения ссылочной целостности (RESTRICT, CASCADE, SET NULL). СК возникают при каскадировании (CASCADE) удаляемых записей с гранулярностью уровня поля, а также при запрете (RESTRICT) на удаление и модификацию записей, на которые имеются ссылки.

Проблема скрытых каналов по-разному решается многоуровневыми моделями данных. Как правило, ими вводятся дополнительные ограничения целостности, иногда довольно сложные (такие, как многозначная зависимость в модели SeaView), а также предлагается специальная семантика данных (например, в доверительных моделях [3]). Ни одна коммерческая СУБД не реализует в полном объеме какую-либо из многоуровневых моделей вследствие их концептуальной сложности. Среди многоуровневых систем наибольшее распространение получили частные решения, которые не обладают внутренней непротиворечивостью, но компенсируют это на уровне прикладного ПО (например, модель LDV) [6]).

На практике большинство многоуровневых СУБД допускают режимы работы с незаблокированными СК.

Скрытые каналы ссылочной целостности

	DELETE		UPDATE	
	Метки поля	Метки записи	Метки поля	Метки записи
RESTRICT	СК	СК	СК	СК
CASCADE	СК	Безопасно	Безопасно	Безопасно
SET NULL	Безопасно	Безопасно	Безопасно	Безопасно

Для таких систем актуальна разработка методик для численной оценки пропускной способности СК, базирующихся на механизмах целостности первичных ключей и ссылочной целостности.

Базовую модель СК легко построить на примере приведенного многоуровневого отношения "Проект", или его сегментного аналога, состоящего из одноуровневых отношений (рис. 2).

Для простоты рассматривается решетка из двух уровней секретности {lo, hi}. Согласно многоуровневой безопасности, пользователь LEVEL_HI имеет полный доступ к секретным записям отношения (одноуровневое отношение "Проект2") и доступ на чтение к несекретным записям (одноуровневое отношение "Проект1"). Пользователь LEVEL_LO не имеет доступа к секретным записям (отношение "Проект2"), но имеет полный доступ к несекретным записям ("Проект1"). Для передачи двоичного символа пользователь LEVEL_HI создает и удаляет запись с условным идентификатором "N" на секретном уровне. Пользователь LEVEL_LO пытается создать запись с таким же идентификатором "N", определяя, таким образом, факт наличия условной записи на секретном уровне "hi". Успешное создание пользователем LEVEL_LO условной записи будет обозначать ее отсутствие на уровне "hi", отказ в ее создании будет сигнализировать о наличии условной записи на уровне "hi". Образованный СК позволяет передавать один двоичный символ на каждый цикл передачи. Аналогично строится СК на основе механизмов ссылочной целостности связанных многоуровневых отношений. При этом параллельное использование двух, трех, и более записейкратно увеличивает пропускную способность СК.

Агрегатная модель скрытого канала

Простейший СК в многоуровневой БД основывается на возможности модулятора (в секретном разделе) и демодулятора (в несекретном разделе) синхронизировать используемые значения записей многоуровневого отношения "Проект" для кодирования символов. В качестве способа противодействия такому СК, не требующего привлечения специальных многоуровневых моделей, может являться автоматическая генерация идентификаторов записей (поле "пр_ид" в многоуровневом отношении "Проект"). Для получения ключей можно воспользоваться любым из методов генерации случайных (псевдослучайных) чисел. Для СК на основе ссылочной целостности можно также ввести запрет на удаление и/или повторное использование ключей с высоких уровней секретности, заменив его, например, обнулением устаревших записей. В сегментной модели это достигается сохранением в ключевом отношении "пр_ид" как актуальных, так и не используемых более идентификаторов.

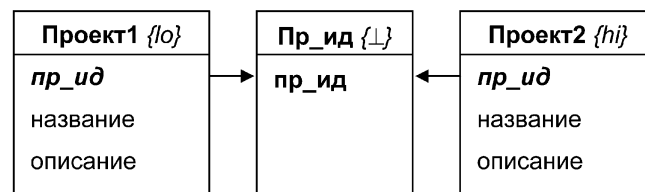


Рис. 2. Сегментное мета-отношение "Проект"

Эти меры позволяют существенно снизить пропускную способность СК, который сохраняется лишь за счет возможности косвенного или явного определения количества записей (идентификаторов) в многоуровневом (либо сегментированном) отношении.

Данный СК образуется агрегированием секретных записей (уровень "hi"). Пользователь LEVEL_HI по-прежнему обладает правом чтения данных с уровня "lo", но не обладает правами их модификации, а пользователь LEVEL_LO не имеет доступа к уровню "hi". Для передачи данных по такому СК пользователь LEVEL_HI добавляет или удаляет записи уровня "hi", а пользователь LEVEL_LO пытается определить общее число записей путем явного наблюдения (например, если ключевое отношение "Пр_ид" находится в несекретном сегменте), либо по косвенным признакам (оценивая время генерации псевдослучайных идентификаторов или текущее значение из последовательности БД). За основу алфавита модуляции выбирается уже не наличие/отсутствие конкретной секретной записи в многоуровневом отношении, а общее число секретных записей.

Оценка пропускной способности агрегатного канала

Как было показано, параллельное использование нескольких простых СК с пропорциональным увеличением пропускной способности в рассмотренных случаях оказывается невозможным. Для повышения пропускной способности можно предложить расширенную схему модуляции, когда каждому информационному символу соответствует определенное число записей в многоуровневом отношении. Такая схема позволит создать скрытый канал, пригодный для передачи информации в условиях маскирующих фоновых запросов (создания и удаления записей отношения, исходящих от легальных процессов или порождаемых специальными маскировочными запросами), а также оценить пропускную способность такого канала.

Пропускную способность для алфавита символов $\{i\}$ можно оценить исходя из вычисления априорных вероятностей наблюдения в многоуровневом отношении "Проект" заданного числа записей $\{P(i)\}$. Предполагается, что программа нарушителя монополюно работает с этим отношением. Для этого он может заблокировать других пользователей или использовать скрытый канал, когда отсутствуют запросы других пользователей. Определение $\{P(i)\}$ должно осуществляться таким образом, чтобы состояние БД соответствовало действительному поведению (модели) пользователей в системе. В противном случае скрытый канал можно обнаружить по наличию статистических аномалий запросов обращения к БД.

Для простоты будем предполагать, что модель запросов создания и удаления записей подчинена экспоненциальному распределению с параметрами A (интенсивность создания) и B (интенсивность удаления) записей. Предположение справедливо, если число источников запросов достаточно велико, и они не коррелированы между собой. Кроме того, будем предполагать, что среднее число записей в отношении "Проект" с течением времени не изменяется, т. е. интенсивность создания одних записей не может превышать интенсивности удаления других записей, и в начальный момент в БД

имеется некоторое число записей, не участвующих в циклах передачи.

Для определения $\{P(i)\}$ проще всего применить модель системы массового обслуживания М/М/1 с бесконечной очередью, что соответствует бесконечному алфавиту дискретных символов $I = \{i: i = 1, 2, \dots\}$ с вероятностями реализации $P(i) = (1 - \rho)\rho^i$, где $\rho = A/B \in [0; 1]$ — коэффициент модуляции символов. Энтропия рассматриваемого алфавита $H(I)$, может быть вычислена по следующей формуле:

$$H(I) = - \sum_{i=0}^{\infty} (1 - \rho)\rho^i \log[(1 - \rho)\rho^i] = \\ = - \frac{(\rho \log[\rho] + (1 - \rho) \log[1 - \rho])}{1 - \rho}. \quad (1)$$

Очевидно, что $H(I) \rightarrow \infty$, если $\rho \rightarrow 1$. При этом происходит неограниченный рост среднего квадратического отклонения числа записей и "размазывание" вероятности по все большим и большим символам алфавита I .

Соотношение (1) оценивает энтропию, а следовательно, и среднее количество информации в символах за неограниченное время. Для того, чтобы оценить пропускную способность нужно найти среднее количество информации, передаваемое по скрытому каналу в расчете на один запрос, которое будем называть удельной информативностью запроса R . При сохранении средней длительности интервалов между запросами общая пропускная способность пропорциональна суммарной информативности всех запросов по всем отношениям БД, на записи которых ссылаются записи отношений более высокого уровня секретности.

Для определения R нужно задать длительность интервала модуляции, в пределах которого выполняется переход отношения "Проект 1" из произвольного в конкрет-

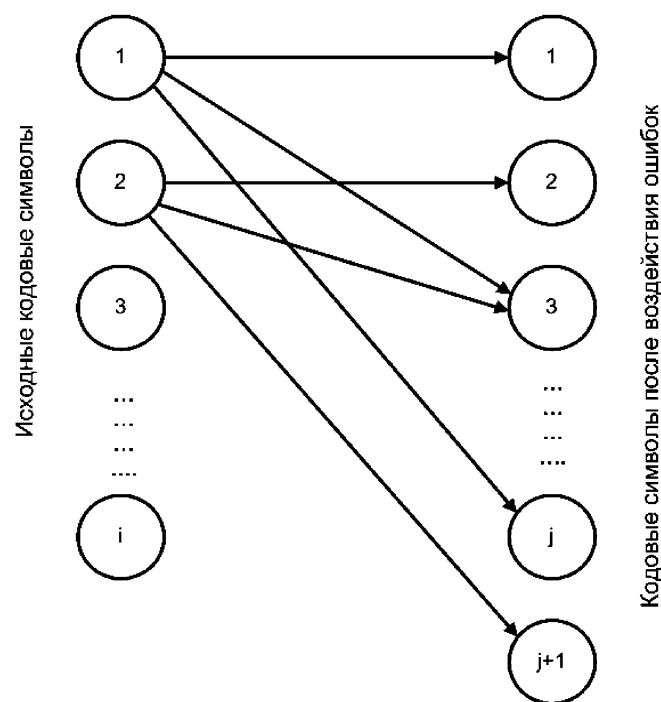


Рис. 3. Модель ошибок

но заданное состояние (определяемое значением передаваемого символа и схемой уровней модуляции). Очевидно, что данное время пропорционально среднему числу записей в "Проект 1", которое для модели (М/М/1) может быть вычислено следующим образом:

$$T = \frac{\tau}{(1 - \rho)},$$

где τ — суммарная длительность операций манипулирования записями "Проект 1" на секретном уровне и оценки их состояния на несекретном уровне. Теперь можно оценить информативность каждого запроса следующим образом:

$$R = -\rho \log[\rho] - (1 - \rho) \log[1 - \rho]. \quad (2)$$

Анализ выражения (2) позволяет сделать следующие выводы. При $\rho \rightarrow 1$ информативность запроса $R \rightarrow 0$. Несмотря на неограниченный рост эффективной мощности алфавита, время, требуемое для перевода "Проект 1" в произвольное состояние, растет быстрее, чем среднее количество информации в алфавите. При $\rho \rightarrow 0,5$ информативность запроса и пропускная способность принимают максимальное значение $R = 1$ бит/запрос. Общая пропускная способность всех скрытых каналов в произвольной БД оказывается пропорциональной числу отношений, участвующих в ограничениях ссылочной целостности многоуровневой БД.

Передача данных в условиях маскирующих запросов

Рассмотренная схема модуляции не дает выигрыша пропускной способности, однако на ее основе могут быть построены алгоритмы, позволяющие передавать данные одновременно с фоновыми запросами. Фоновые запросы обладают маскирующим эффектом и могут формироваться легальными пользователями, либо специальными приложениями, создающими ложные записи, которые приводят к появлению ошибок демодуляции символов.

Ошибка — это переход состояния отношения, характеризующегося наличием в нем i записей, в какое-либо другое состояние, соответствующее наличию в нем $(i + j)$ записей. Такой канал оказывается асимметричным, так как маскирующие запросы могут лишь увеличить число записей. Модель ошибок — состояния переходов скрытого канала — показана на рис. 3.

Модулирующие запросы будем характеризовать экспоненциальным законом распределения (с интенсивностью A_1 запросов создания и с интенсивностью B_1 запросов удаления записей). Аналогичным образом, маскирующие запросы будем описывать экспоненциальным законом создания (с интенсивностью A_2) и удаления (с интенсивностью B_2) маскирующих (ложных) записей. Соответственно, априорные вероятности ошибок веса j будут иметь вид: $P(j) = (1 - \rho_2)\rho_2^j$, где ρ_2 — коэффициент маскировки скрытого канала.

Для оценки информативности запросов с учетом ошибок $R_{\text{ош}}$ нужно

вычислить надежность канала по Шеннону. Для рассматриваемого случая можно записать: $R_{\text{ош}} = R(I) - R(I/I + J)$, где $R(I)$ — исходная информативность алфавита I на множестве исходных уровней модуляции; $R(I/I + J)$ — информативность запросов при принятии предположения о наблюдаемых на выходе демодулятора суммарных уровнях модуляции с учетом маскирующих записей (формально, алфавит I эквивалентен алфавиту $I + J$, поскольку множества $\{i\}$ и $\{j\}$ являются счетными).

Найти информативность запросов в виде аналитического выражения, аналогично случаю отсутствия маскирующих запросов, не представляется возможным. Для оценки информативности запросов при заданных значениях коэффициентов модуляции (ρ_1) и маскировки (ρ_2) необходимо воспользоваться численным анализом. Для ряда значений параметров ρ_1 и ρ_2 на рис. 4 приведены примеры функции информативности запросов.

Анализ семейства графиков на рис. 4 показывает, что оптимальные значения коэффициента модуляции, обеспечивающие максимум информативности запросов, зависят от коэффициента маскировки, что объясняется асимметричностью скрытого канала.

При увеличении коэффициента маскировки максимум информативности запросов сдвигается в сторону больших значений ρ_1 . Численный анализ показывает, что зависимость $\rho_1 = f(\rho_2)$ носит линейный характер с относительно небольшим коэффициентом пропорциональности (рис. 5). Приращение информативности запросов за счет выбора оптимального коэффициента модуляции не превышает 20...25 % от средней информативности запроса.

Заключение

В рамках противодействия скрытым каналам утечки информации в приложениях БД, не удовлетворяющих требованиям многоуровневой безопасности, необходимо исходить из осведомленности нарушителя об особенностях поведения (модель пользователя) штатных пользователей.

Для успешного противодействия скрытым каналам в БД необходимо предложить методики описания динамических характеристик бизнес-процессов и обосновать подходы к выбору норм эффективности защиты данных в зависимости от объективных потребностей системы.

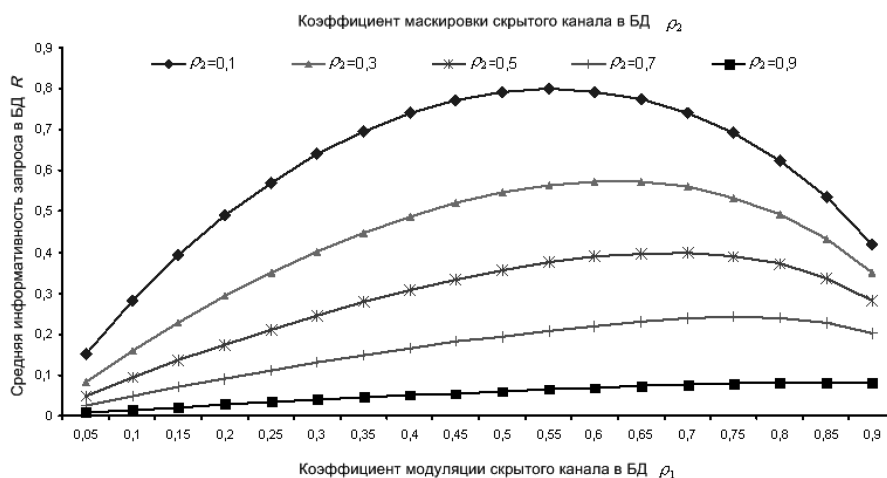


Рис. 4. Функция информативности запросов

Необходимо иметь возможность задания статистических характеристик запросов, порождаемых бизнес-процессами, указания интервалов (и относительного уровня) повышения и снижения интенсивности запросов и учета других факторов, прямо влияющих на функцию информативности запросов. Точность указания данных характеристик будет определять вероятность ошибок первого (когда скрытый канал окажется не полностью заблокированным и возможна утечка) и второго рода (когда число маскирующих записей БД будет превышать разумные пределы, вследствие чего вырастут накладные расходы на обеспечение безопасности).

Для обоснования необходимого уровня защищенности информации удобно воспользоваться аналогией между скрытыми и техническими каналами и считать допустимым такой уровень угрозы, при котором количество информации, получаемое по скрытому каналу, соизмеримо с количеством информации, получаемым по техническому каналу. Базируясь на известных нормах эффективности защиты, можно определить область пространства точек (ρ_1 , ρ_2), в пределах

которой система будет считаться защищенной относительно выбранной нормы.

Список литературы

1. National Computer Security Center. NCSC-TG-030, Guide to Understanding Covert Channel Analysis of Trusted System. USA P. 57-198-1991.
2. ГОСТ ИСО/МЭК 15408—2002. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть III. Требования доверия.
3. Smith K. P., Winslett M. S. Entity Modeling in the MLS Relational Model // Proc. of 18th Very Large Data Base Conference, VLDB, Vancouver, Canada, 1992.
4. Аксарин М. В., Быков Я. А., Тарасюк М. В. Проектирование приложений реляционных БД с учетом требований конфиденциальности и целостности данных // Безопасность информационных технологий. 2005. № 3.
5. NCSC. Technical report-005. National Computer Security Center. NCSC-TR-005. Technical report-005. Vol. 2, 3. DoD, Washington, USA, 1996.
6. Haigh J. T., O'Brien R. C., Thomsen D. J. The LDV Secure Relational DBMS Model. Database Security IV (Jajodia S. and Landwehr C. E., eds.), North-Holland, 1991.

УДК 004.49.056.57

А. Н. Шниперов,
Политехнический институт ФГОУ ВПО
"Сибирский федеральный университет"

Синтез и анализ высокоскоростных симметричных криптосистем на основе управляемых операций

Представлено описание одного из подходов к разработке высокоскоростных симметричных блочных шифров с использованием нового криптографического примитива — управляемых операций. Рассматривается новый симметричный блочный шифр, обладающий высокими показателями как скорости работы, так и криптографической стойкости к распространенным видам атак. Представлены некоторые способы повышения стойкости шифра к современным видам атак.

Введение

Стремительное развитие новых информационных технологий, которые в настоящее время применяются практически во всех сферах человеческой деятельности, вызвало массу проблем, связанных с защитой информации. В настоящее время криптографические методы нашли широкое применение не только для защиты информации от несанкционированного доступа, но и в качестве многих новых электронных информационных технологий — электронного документооборота, электронных денег, электронных образовательных ресурсов

и др. При этом к алгоритмам шифрования предъявляются жесткие технологические требования, которые продиктованы их использованием в различных электронных устройствах (телекоммуникационных системах, ЭВМ, компьютерных сетях, интеллектуальных электронных карточках и др.). Наиболее актуальными требованиями, предъявляемыми к криптографическим средствам, являются скоростные качества шифров, стойкость к криптоанализу, а также простота реализации.

Ужесточение требований по стойкости обусловлено тем, что разностороннее использование криптографии связано с более широкими возможностями для атакующего следовать особенностям конкретных условий, в которых функционирует криптосистема (например, имеются возможности: первая — осуществить внешнее воздействие на устройство шифрования с целью вызвать случайные аппаратные сбои, вторая — выполнить замер потребляемой мощности, третья — определить время вычислений и т. п.) [1].

Возрастание требований по скорости работы связано с необходимостью сохранения высокой производительности автоматизированных систем после интеграции в них механизмов защиты.

Простота аппаратной или программной реализации необходима для снижения стоимости средств шифрования, что напрямую способствует их массовому внедрению и более широким возможностям интеграции с портативной аппаратурой.

Характерной особенностью современных криптосистем с аппаратной или двойственной (программной и аппаратной) ориентацией является использование алгоритмов преобразования данных с предвычислениями, которые вносят существенные ограничения по быстродействию и зачастую требуют значительных вычислительных затрат, особенно при частой смене ключей. Этот недостаток характерен для подавляющего большинства

современных симметричных криптосистем. В связи с этим крайне важным становится уменьшение сложности предвычислений при сохранении высокой криптостойкости преобразования. В этом плане наиболее удачным решением является полный отказ от предварительного преобразования секретного ключа путем замены этой процедуры операциями преобразования подключей в зависимости от преобразуемых данных, которые выполняются одновременно с операциями преобразования данных.

Использование управляемых подстановочных и перестановочных операций, зависящих от входного открытого текста, позволяет создавать различные алгоритмы шифрования с нелинейной структурой, что, в свою очередь, обеспечивает превосходную стойкость к линейному и дифференциальному криптоанализу.

Использование управляемых операций

Одной из особенностей разработки блочных шифров на основе управляемых операций является то, что при выполнении одной операции используются все биты преобразуемого подблока данных, причем характер их использования зависит от того, к какому подблоку данных они относятся, а именно — к управляющему или преобразуемому. С этим моментом связана специфика проектирования блочных шифров при использовании такого типа примитивов. В настоящее время нами хорошо апробирована технология применения перестановок, зависящих от преобразуемых данных, в качестве основного примитива в ряде блочных шифров.

Для таких шифров структура раундового преобразования может быть рассмотрена как усовершенствованная криптосхема Фейстеля [1], где вводится преобразование правого подблока данных с помощью блока управляемых перестановок (БУП) типа $\pi_{32/80}$ (управляемая перестановка 32 битов входного блока данных в зависимости от 80 битов управляющего вектора V), выполняемого одновременно с вычислением раундовой функции по левому подблоку. В этом случае мы имеем преобразование

$$L \leftarrow L;$$

$$R \leftarrow F(L) \oplus \pi_{32/80}^{(V)}(R),$$

где " $\leftarrow$ " определяет операцию присваивания; R и L — левый и правый подблоки данных; F — раундовая функция; V — управляющий вектор, формируемый в зависимости от левого подблока.

Достоинством модернизированной криптосхемы является усиление параллелизма шифрующих преобразований. Для сохранения возможности расшифрования блока шифротекста с использованием того же алгоритма, который осуществляет зашифрование (свойство универсальности), после наложения значения раундовой функции на преобразованный правый подблок выполняется обратная управляемая перестановка $\pi_{32/80}^{-1}$. Обе

упомянутые операции $\pi_{32/80}$, $\pi_{32/80}^{-1}$ выполняются в зависимости от левого подблока данных, что определяет его участие в каждой операции, выполняемой в раунде шифрования. Несмотря на активное использование левого подблока данных при раундовом преобразовании сам он не подвергается преобразованию. С этим связана проблема формирования различных значений управ-

ляющих векторов для операций $\pi_{32/80}$ и $\pi_{32/80}^{-1}$. Эта проблема состоит в том, что в случае равенства управляющих векторов выполнение переменных двух операций $\pi_{32/80}$ и $\pi_{32/80}^{-1}$ сводится к выполнению одной операции $\pi_{32/80}^{-1}$, осуществляемой над вычисленным значением раундовой функции. В криптосистеме SPECTR-H64 [1] данная проблема решается тем, что при формировании управляющих векторов для $\pi_{32/80}$ и $\pi_{32/80}^{-1}$ дополнительно к L используются различные подключи. Полный раунд преобразования имеет следующий вид:

$$L \leftarrow L;$$

$$R \leftarrow F(L) \oplus \pi_{32/80}^{(V)}(R);$$

$$R \leftarrow (\pi^{-1})_{32/80}^{(V)}(R).$$

Рассматриваемая криптосхема представляет общий интерес при проектировании шифров на основе переменных операций, в связи с чем необходимо найти, если имеются, недостатки и рассмотреть возможность их устранения. Определенным недостатком криптосистемы SPECTR-H64 является использование сравнительно большого числа подключей — шести 32-битовых подключей, что требует применения процедур расширения ключа или секретных ключей сравнительно большого размера, в частности, в SPECTR-H64 используется 256-битовый секретный ключ. Таким образом, при построении шифров на основе взаимно обратных управляемых перестановок разработчик сталкивается со специфической проблемой реализации эффективного механизма формирования различных управляющих векторов, соответствующих взаимно обратным БУП. При этом следует сохранить свойства универсальности и высокого параллелизма преобразований [2].

Для решения данной проблемы был предложен способ, который не требует использования дополнительных подключей, а заключается в применении наряду с управляемыми также и фиксированных перестановок, выполняемых над левым и/или правым подблоком данных. При этом сохранение свойства универсальности алгоритма шифрования достигается тем, что для такого преобразования используются две одинаковые раундовые функции и дополнительно следующие элементы:

- перестановочные инволюции (над левым и/или правым подблоком);
- переключаемые фиксированные перестановки (над левым и/или правым подблоком);
- преобразование левого подблока путем наложения на него подключа.

Описание симметричной криптосистемы

Рассмотрим один из вариантов реализации криптосхемы в соответствии с данным способом решения проблемы формирования управляющих векторов для выполнения взаимно обратных переменных операций. Применение переключаемой операции над правым подблоком показано на рис. 1. Процедура $Crypt^{(e)}$ обозначает раундовое преобразование в целом.

Данный механизм основан на том факте, что преобразование в правой ветви представляет собой суперпозицию $\pi_{n/m}^{(V)} \Pi^{(e)}(\pi_{n/m}^{-1})^{(V)}$, которая фактически представ-

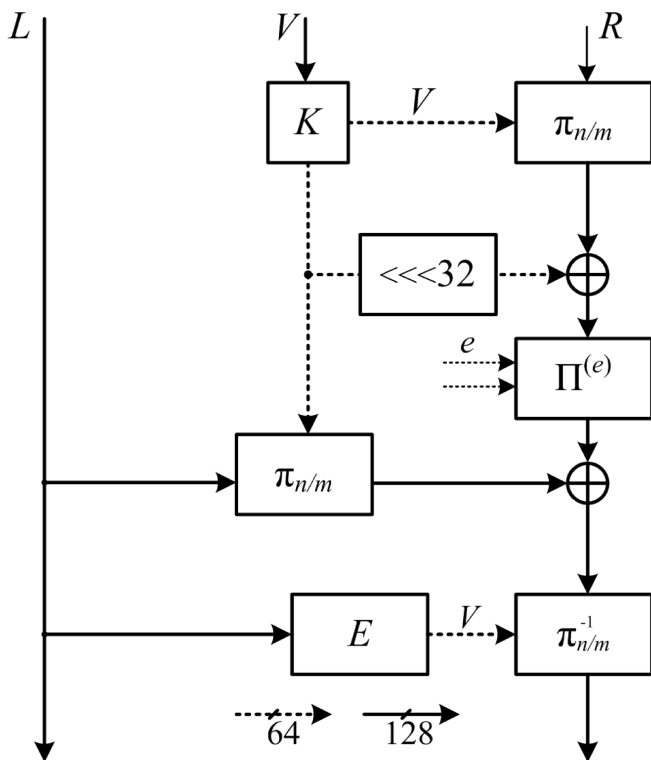


Рис. 1. Общий вид процедуры $Crypt^{(e)}$ с переключаемой операцией

ляет собой операционный блок, управляемый вектором V и выполняющий переменные перестановки, обладающие цикловой структурой перестановки $\Pi^{(e)}$.

Очевидно, что в данной суперпозиции для реализации только тех перестановок, которые имеют цикловую структуру перестановки $\Pi^{(e)}$, естественным требованием является условие равенства управляющих векторов, соответствующих прямому $\pi_{n/m}$ и обратному $\pi_{n/m}^{-1}$ блокам.

В общем случае для разных значений левого подблока (разных значений управляющего подблока данных L) реализуются различные модификации перестановок $\pi_{n/m}$ и соответственно $\pi_{n/m}^{-1}$. Процедура K обеспечивает расщепление входного 128-битного управляющего вектора V на подключи размерностью 64 бит. Достоинством такого варианта построения криптосхемы является отсутствие затратных, с вычислительной точки зрения, алгоритмов формирования рабочего ключа. Возможность осуществления зашифрования и расшифрования с помощью одного и того же алгоритма обеспечивается тем, что для обращаемой фиксированной перестановки выполняется условие $\Pi^{(e=1)} = (\Pi^{(e=0)})^{-1}$. Как при зашифровании, так и при расшифровании указанная суперпозиция реализует перестановки с заданной цикловой структурой, например одноцикловые перестановки.

В частном случае одноциклового перестановки $\Pi^{(e)}$ механизм оптимизации заключается в том, что бит из j -го разряда на входе блока $\pi_{n/m}$ попадает с примерно одинаковой вероятностью во все выходные разряды блока $\pi_{n/m}^{-1}$, кроме j -го, в который он не попадает ни при каком значении управляющего вектора. Аналогичное свойство обеспечивает любая перестановка, не содержащая циклов длины 1. Для обеспечения примерно равномерного влияния каждого входного бита операционного

блока $\pi_{n/m}$ на значение всех выходных битов блока $\pi_{n/m}^{-1}$ можно использовать обращаемую перестановку, содержащую только один цикл длины 1.

Обращаемая перестановка, например, может быть реализована с помощью однослойного БУП, содержащего 32 элементарных переключателя, на управляющий вход которых подается один и тот же бит e . Структура перестановки $\Pi^{(e)}$, реализующей операцию циклического сдвига вправо ($e = 1$) или влево ($e = 0$), показана на рис. 2.

Данный вариант решения проблемы формирования управляющих векторов может быть применен не только для управляемых перестановочных операций, но и для пар взаимно обратных переменных операций других типов, построенных на основе управляемых подстановочно-перестановочных сетей.

Программная реализация симметричной криптосистемы

Для построения скоростных программных шифров нами была использована технология использования псевдослучайных выборок из ключа шифрования [2]. Криптосхемы с подобной недетерминированной выборкой подключей характеризуются тем, что элементы криптографического ключа, используемые при преобразовании входного блока данных, выбираются в зависимости от структуры преобразуемого блока и от структуры ключа шифрования. Для шифров, основанных на подобной криптосхеме, расписание выборки подключей нельзя считать предопределенным, что является существенным фактором повышения стойкости шифрования. За счет этого число раундов шифрования может быть уменьшено и тем самым повышена скорость шифрования. Шифры с выборкой подключей в зависимости от преобразуемых данных можно назвать псевдослучайными по характеру выборки подключей.

Выборка подключей в зависимости от преобразуемых данных в самом общем случае есть просто некоторая формальная операция отображения (подстановки), выполняемая по таблице, которая задается ключом шифрования. Ключ шифрования представляет собой совокупность пронумерованных подключей. Первый блок задает номер выбираемого подключа, а последний — значение, получаемое в результате операции отображения.

В программной реализации разработанной нами криптосистемы *CryptoStar* базовый принцип рассеивания и перемешивания дополняется следующими механизмами:

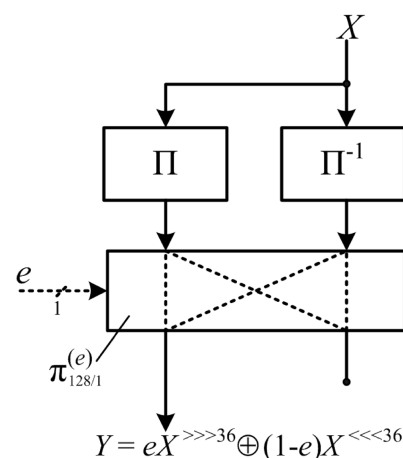


Рис. 2. Структура обращаемой одноцикловой перестановки

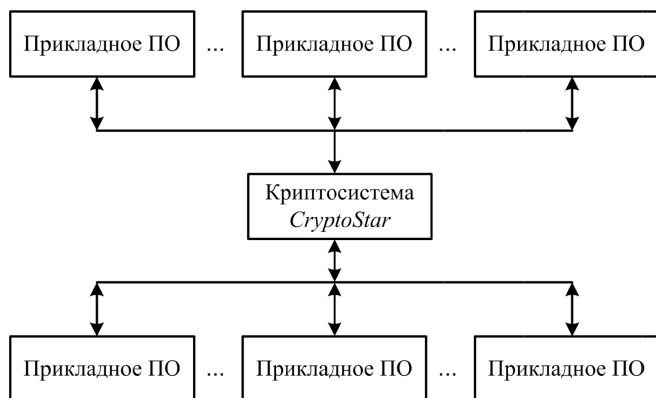


Рис. 3. Функциональная схема работы криптосистемы *CryptoStar* в межпрограммном режиме

- зависимостью архитектуры π-блоков (БУП) от секретного ключа;
- увеличением размера π-блоков;
- увеличением размера блока преобразуемых данных до 512 байт;
- зависимостью алгоритма шифрования от секретного ключа (путем настройки операций преобразования по секретному ключу).
- зависимостью операций преобразования от преобразуемых данных (т. е. управлением выбором модификации текущей операции от входного сообщения);
- зависимостью выборки подключей от преобразуемых данных.

Блочный шифр *CryptoStar* был нами программно реализован в виде автономного СОМ-объекта с использованием объектно-ориентированного языка программирования MS Visual C++ и технологии ATL. Данное программное решение позволило осуществлять интеграцию криптосистемы *CryptoStar* практически в любое разрабатываемое программное обеспечение для платформы Win32, независимо от среды разработки, в целях осуществления информационной защиты потоков данных. Функциональная схема работы криптосистемы *CryptoStar* в межпрограммном режиме представлена на рис. 3, а в сетевом режиме — на рис. 4.

Результаты исследования программного прототипа криптосистемы *CryptoStar*

Для экспериментальной проверки скоростных характеристик криптосистемы *CryptoStar* был разработан программно-аппаратный стенд, основными элементами которого являются:

- IBM PC совместимый персональный компьютер (с центральным

процессором Intel Celeron 500 МГц, 128 Мбайт ОЗУ, 20 Гбайт жестким диском и графическим адаптером);

- операционная система Windows NT 4.0 Service Pack 6a;
- программное обеспечение замера производительности программной криптосистемы *CryptoStar*;
- программное обеспечение для статистического анализа шифротекстов.

Данный экспериментальный стенд позволяет измерять скорость шифрования криптосистемы *CryptoStar* с точностью 1 Кбайт, а также анализировать частотное распределение байтов выходных шифрограмм. На основании последних можно судить о качественных характеристиках перемешивания и рассеивания битов в алгоритме шифрования, а значит, и степени его стойкости.

Для измерения производительности криптосистемы *CryptoStar* использовалось программное обеспечение, разработанное нами специально для проведения эксперимента. Функциональная схема измерения производительности криптосистемы представлена на рис. 5.

Суть эксперимента по измерению производительности криптосистемы сводится к следующему: испытательное программное обеспечение последовательно генерирует несколько блоков открытого текста различного размера (100 Кбайт, 500 Кбайт, 1 Мбайт, 50 Мбайт), далее, используя внутренние функции библиотеки *CryptoStar*,

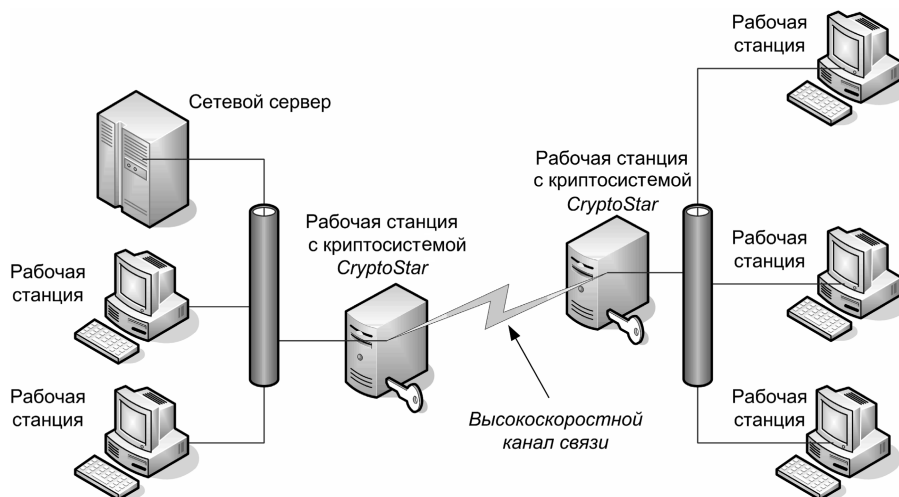


Рис. 4. Функциональная схема работы криптосистемы *CryptoStar* в сетевом режиме

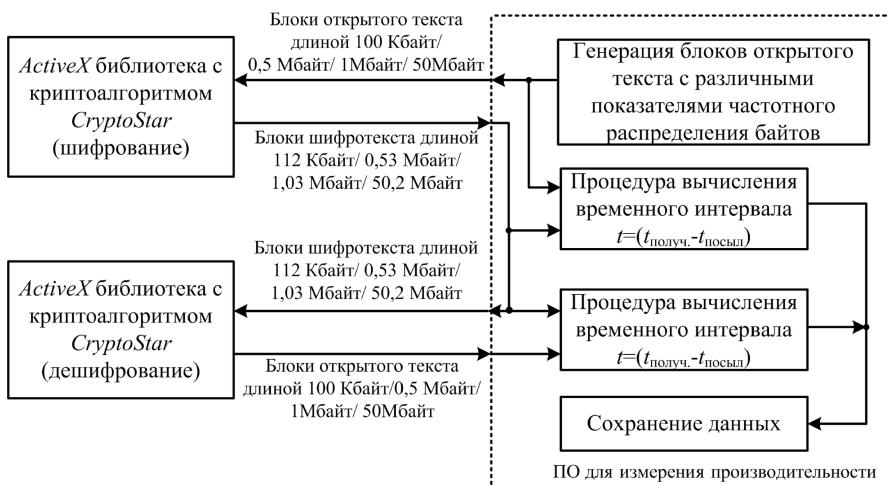


Рис. 5. Функциональная схема замера производительности

зашифровывает их, измеряя время шифрования каждого из них. Затем, используя полученные временные интервалы, находят скорости шифрования каждого из этих блоков. Далее скорости усредняются и выводятся на экран монитора в виде гистограммы. Для измерения скоростей расшифровывания используются полученные блоки шифротекста.

Результаты эксперимента по измерению скоростных характеристик представлены на рис. 6, а (шифрование) и 6, б (расшифровывание) (см. третью сторону обложки).

Частотный анализ криптосистемы сводится к следующему: испытательное программное обеспечение последовательно генерирует несколько блоков открытого текста фиксированного размера (100 Кбайт). Каждый блок представляет собой либо упорядоченную последовательность, состоящую из 1, 10, 100 и 1000 символов, либо псевдослучайную с размерностью всего блока. Далее, на основе внутренних функций библиотеки *CryptoStar* происходит их зашифровывание. Полученные шифротексты обрабатываются процедурой частотного анализа и, как результат, выводится гистограмма распределения байтов в зашифрованном блоке. На рис. 8 представлены графики частотного распределения байтов в выходном шифротексте, если он был получен путем шифрования блока открытого текста, состоящего из одного символа (0x7F и 0x3D соответственно).

Для частотного анализа выходных шифров криптосистемы *CryptoStar* использовалось программное обеспечение (ПО), разработанное специально для проведения эксперимента. Функциональная схема частотного анализа представлена на рис. 7.

На рис. 9 представлен график частотного распределения байтов в выходном шифротексте, если он был получен путем шифрования блока открытого текста, представляющего собой выдержку из литературного журнала.

Из рис. 8 и 9 очевидно, что криптосистема обладает значительным эффектом рассеивания, нарушающим взаимосвязь статистических характеристик входного и выходного текста, т. е. маскировку статистических свойств исходного сообщения. Также легко заметить, что криптосистема обладает эффективным свойством "лавинного" эффекта, когда даже один байт входного текста существенно влияет на все последующие. Оба этих критерия в целом могут свидетельствовать о криптостойкости алгоритма к дифференциальному и линейному криптоанализу.

Таким образом, в ходе эксперимента было установлено, что криптосистема *CryptoStar* является высокоскоростной (скорость шифрования в большинстве случаев

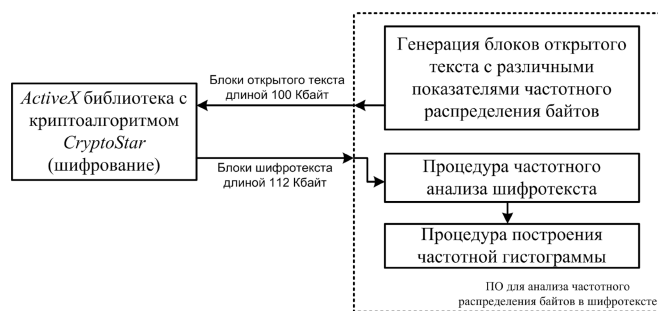


Рис. 7. Функциональная схема частотного анализа криптосистемы *CryptoStar*

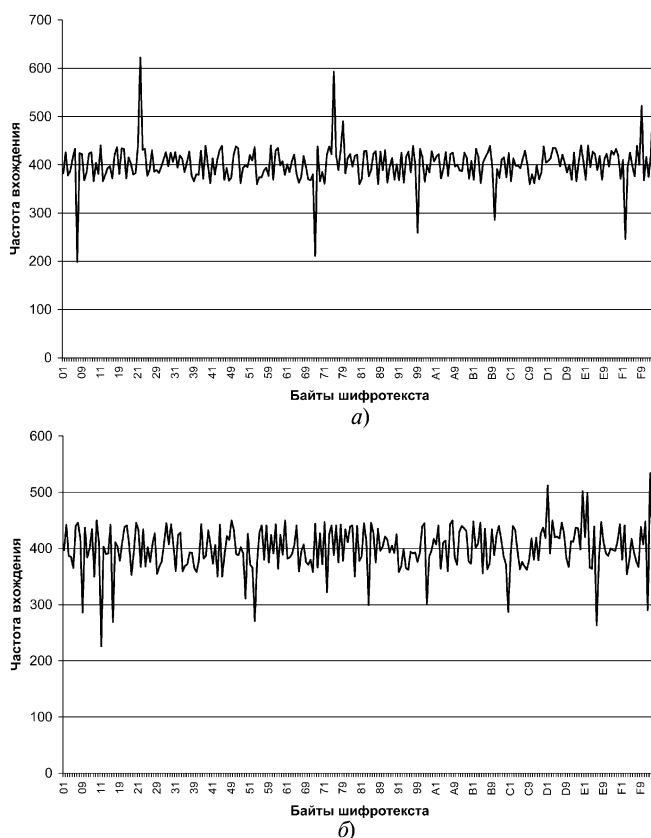


Рис. 8. Частотное распределение байтов, входящих в шифротекст для открытого текста, состоящего из символа 0x7F (а) и символа 0x3D (б)

значительно выше, чем у современных аналогов, таких как AES, DES, BlowFish и т. д.). Также следует отметить, что криптосистема обладает хорошими показателями стойкости к классическим видам криптоанализа, что обусловлено высокими показателями рассеивания и перемешивания.

Устойчивость криптосистемы *CryptoStar* к современным видам атак

Поскольку почти все осуществленные на практике удачные атаки на криптосистемы используют слабости в реализации и размещении механизмов криптоалгоритма, эти слабости часто позволяют взломщикам полностью обойти защиту или существенно ослабить ее теоретическую стойкость. Такие атаки основаны на корреляции между значениями физических параметров, измеряемых в разные моменты времени вычислений (потребление энергии, время вычислений, электромагнитное излучение и т. п.), и внутренним состоянием вычислительного устройства, имеющим отношение к секретному ключу [4].

Как уже было отмечено выше, разработанная криптосистема *CryptoStar* может быть реализована как в программной, так и в аппаратной реализации.

В случае *программного решения криптосхемы* некоторые конкретные разновидности атак на побочный канал (SCA), как, например, измерение потребляемой мощности или излучаемого электромагнитного излучения (ЭМИ), не представляют собой угрозы, поскольку программный продукт — это далеко не единственный запу-

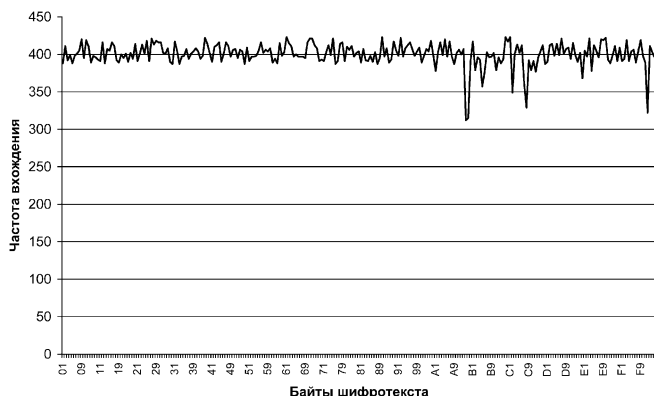


Рис. 9. Частотное распределение байтов, входящих в шифротекст (для открытого текста, состоящего из произвольных символов)

щенный процесс операционной системы. Следовательно, потребляемая мощность и уровень ЭМИ будут постоянно колебаться с трудно восстанавливаемой закономерностью.

Вследствие того, что криптосистема *CryptoStar* (как в программном, так и в аппаратном решении) представляет собой блочный шифр на основе управляемых операций, он становится уязвимым к атаке по времени, которая основана на измерении времени, необходимого модулю шифрования на выполнение операции шифрования [4]. В дальнейшем предполагается осуществить добавление временных задержек в те функциональные узлы криптосистемы, которые имеют самые скоростные характеристики.

В случае *аппаратной реализации криптосистемы* одним из видов нападений, представляющих криптоаналитику дополнительные возможности, является атака на основе генерации ошибок шифратора, обусловливаемых влиянием внешнего воздействия. Рассмотрим случай, в котором криптоаналитику предоставляется дополнительная возможность оказывать на шифратор с установленным в нем ключом некоторое внешнее воздействие с целью вызвать сбои в работе аппаратуры и полученные выходные данные сопоставлять с выходными данными, полученными без сбоев. Будем считать, что экспериментально подобрано такое внешнее воздействие (например, нагревание, высокочастотное или ионизирующее излучение) и его интенсивность, при которой в среднем генерируется одна ошибка за полное время шифрования одного блока. Положим, что вероятность генерации ошибки пропорциональна времени облучения соответствующих регистров, находящихся в благоприятном для возникновения ошибки состоянии, и числу двоичных разрядов, в пределах которых ожидается ошибка. Тогда приближенная оценка вероятности генерации ожидаемой ошибки P в одном опыте, т. е. при шифровании одного блока, определяется выражением

$$P = \frac{p_d}{RZ},$$

где p_d — вероятность генерации ошибки в области данных за время выполнения одного блока данных; R — число раундов преобразования; Z — число элементарных преобразований в одном шаге. Очевидно, что вероят-

ность P главным образом определяется вероятностью p_d , которая, в свою очередь, будет определяться самим крипточипом и его устойчивостью к внешним воздействиям. В любом случае данный тип атаки представляет собой угрозу для аппаратно-ориентированной криптосистемы *CryptoStar*. Для предотвращения такой атаки рекомендуется использовать микропроцессоры (микроконтроллеры, программируемые логические интегральные схемы и т. п.) в специальном, защищенном исполнении, а также прогонять алгоритм шифрования дважды и выдавать результат только при полном совпадении шифрограмм.

Также следует отметить, что аппаратно-ориентированная криптосистема *CryptoStar* может быть уязвимой к атакам по энергопотреблению и мощности. Для повышения стойкости к таким видам атак рекомендуется добавить неиспользуемые (с точки зрения алгоритма шифрования) регистры и/или логические вентили, на которых выполняются бесполезные операции [4].

Выводы

Использование криптографических преобразований, зависящих от преобразуемых данных, имеет большое теоретическое и практическое значение. Теоретическое значение состоит в обосновании нового класса криптографических примитивов и возможности расширения и совершенствования общих принципов построения итеративных схем блочных алгоритмов. Теоретически важным является также то обстоятельство, что математические свойства сложных, на первый взгляд, новых криптографических примитивов, связанных с целой системой битовых преобразований, достаточно просто и эффективно определяются аналитически [3].

Практическое значение концепции управляемых преобразований состоит в возможности создания на ее основе скоростных шифров, максимально адаптированных к аппаратной реализации в виде заказных и программируемых логических интегральных схем (крипточипов). В ряде случаев предлагаемые блочные шифры допускают и скоростную программную реализацию (криптосистема *CryptoStar*).

Высокая эффективность аппаратной реализации блочных шифров на базе управляемых преобразований и разнообразие задач, при решении которых используются операции битовых перестановок, делают актуальным вопрос о внедрении новой команды управляемой перестановки в систему стандартных инструкций универсального процессора. В криптографическом плане появление такой команды приведет к скачкообразному повышению производительности современных программных шифров и хэш-функций, удовлетворяющих высоким требованиям криптостойкости.

Список литературы

1. Молдовян А. А., Молдовян Н. А., Гуц Н. Д., Изотов Б. В. Криптография: скоростные шифры. СПб.: БХВ-Петербург, 2002.
2. Молдовян Н. А., Молдовян А. А., Еремеев М. А. Криптография: от примитивов к синтезу алгоритмов. СПб.: БХВ-Петербург, 2004.
3. Смарт Н. Мир программирования: криптография. М.: Техносфера, 2005.
4. Жуков А. Е. Криптоанализ по побочным каналам // Матер. междунар. конф. РусКрипто-2006.



УДК 004.021

Е. А. Мандриков, В. А. Кулев,
А. А. Шальто, д-р техн. наук, проф.,
Санкт-Петербургский государственный
университет информационных технологий
механики и оптики

Применение генетических алгоритмов для создания управляющих автоматов в задаче о "Флибах"

В работах [1, 2] рассмотрена задача о "Флибах", суть которой состоит в том, чтобы построить флиб — конечный детерминированный автомат (в дальнейшем автомат), который будет предсказывать битовое значение некоторого параметра окружающей среды на основе ранее полученных значений.

Эта задача может быть достаточно просто решена эвристически, если число состояний автомата не ограничивать. В противном же случае, ее эвристически не решить. Поэтому в указанных работах рассматривалась автоматическая генерация автоматов с помощью генетических алгоритмов.

В работе [2] предложен генетический алгоритм, который сравнительно редко строит автоматы, позволяющие предсказывать значение указанного параметра со 100 %-ной точностью. При этом вопрос о числе состояний автоматов специально не рассматривался, но оно значительно превышало необходимое для получения 100 %-ной точности.

Цель настоящей работы — найти компромисс между числом состояний автомата и точностью предсказаний.

1. Постановка задачи

Сформулируем задачу о "Флибах" более подробно. Эту задачу можно рассматривать как моделирование простейшего живого существа, которое способно предсказывать изменения параметра среды, обладающего *периодичностью*. При этом живое существо (флиб) удобно моделировать конечным автоматом с действиями на переходах — автоматом Мили. На вход флиба подается битовое значение параметра окружающей среды. Флиб формирует значение выходной переменной — возможное значение рассматриваемого параметра в следующий момент времени.

Задача состоит в повышении точности (по сравнению с результатами [1, 2]) предсказания значения, которое будет иметь параметр среды в следующий момент времени.

2. Описание окружающей среды

В качестве входных воздействий для флиба (как и в [1, 2]) используется периодическая последовательность значений битов, названная битовой маской, которая задает среду. Для обеспечения периодичности маска в программе зациклена.

3. О возможных решениях

Сформулируем условия, при которых задача может быть решена эвристически. Таким решением будет автомат, число состояний в котором равно длине битовой маски. Продемонстрируем это на примере. Пусть задана битовая маска "1111010010", длина которой равна 10. На рис. 1 приведен граф переходов с линейной структурой для автомата, который решает поставленную задачу со 100 %-ной точностью предсказаний при числе состояний 10.

В этом автомате дуги помечены символами битовой маски, а выходные воздействия — левый циклический сдвиг этой маски. Петли в каждом из состояний умалчиваются.

Этот автомат обладает 100 %-ной точностью предсказаний в том смысле, что если на него подать битовую маску несколько раз, то он ее столько же раз и воспроизведет.

Этот автомат имеет сравнительно большое число состояний, так как строится с помощью универсального алгоритма, который не учитывает свойств заданной битовой маски. Число состояний может быть сокращено, возможно, за счет потери точности предсказания, если эту специфику учитывать. Как будет показано в разд. 5, предложенный в следующем разделе генетический алгоритм позволит находить компромисс между точностью предсказаний и числом состояний автомата.

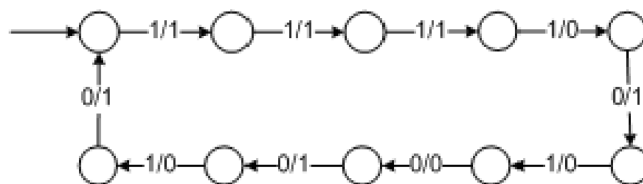


Рис. 1. Граф переходов эвристически построенного автомата

4. Описание генетического алгоритма

Представление автоматов. В данной работе, так же как и в работе [1], автоматы представляются в виде битовых строк фиксированной длины. При этом каждому автомату соответствует строка, состоящая из блоков. Каждому блоку соответствует состояние автомата и переходы из этого состояния. Каждый переход характеризуется парой чисел — номером состояния, в которое автомат переходит, и значением выходного воздействия.

Общая схема. Генетический алгоритм в общем случае состоит из пяти этапов: формирование начальной популяции; тестирование популяции (приспособленность особей); скрещивание (кроссовер); мутация; формирование следующего поколения, и двух операторов (скрещивания двух особей и мутации одной).

Однако для каждой задачи с целью учета ее специфики генетический алгоритм должен быть адаптирован. Также для каждой задачи необходимо задать оценочную функцию, которая называется *fitness-функцией*. Она предназначена для оценки приспособленности особи.

Опишем, как реализуются этапы, операторы и строится *fitness-функция* в данном случае.

Формирование начальной популяции. Начальная популяция заполняется особями — автоматами, число которых равно размеру популяции.

Тестирование популяции. После формирования поколения необходимо оценить приспособленность каждого автомата. Для этого каждый из построенных автоматов устанавливается в начальное состояние, а затем на его вход многократно подаются значения параметра среды, заданные битовой маской. Состояния, в которые при этом переходит автомат, будем называть использованными (существуют также состояния, которые при этих входных воздействиях не могут быть достигнуты).

Лучшим предсказателем является тот автомат, который наиболее точно предсказывает входную последовательность. Если в указанных выше работах в качестве оценочной функции предлагалось использовать число правильно угаданных значений, то в настоящей работе в качестве такой функции предлагается использовать следующее соотношение:

$$Fitness = \begin{cases} \text{если } P = N, \text{ то } fitness = P * N + L; \\ \text{если } P <> N, \text{ то } fitness = P * N + N - L. \end{cases}$$

Здесь N — длина битовой маски; P — число верно предсказанных значений; L — число использованных состояний автомата.

Выбор данного соотношения основан на том факте, что автомат с числом использованных состояний, равных длине битовой маски, имеет простую структуру, описанную в разд. 3. Построив такой автомат, можно пытаться минимизировать число использованных состояний, не уменьшая при этом числа верно предсказанных значений.

Скрещивание. Для каждой операции скрещивания выбирается пара особей из предыдущей популяции. Особи отбираются с помощью "рулетки" [3] и вероятностью, пропорциональной значению функции приспособленности. Эта операция повторяется число раз, которое пропорционально размеру предыдущей популяции. Коэффициент, определяющий это число, будет являться одним из параметров генетического алгоритма на рассматриваемом этапе.

Мутация. Для каждой операции мутации выбирается случайная особь из предыдущей популяции. Как и при скрещивании, особь выбирается с помощью "рулетки", а число мутаций зависит от размера предыдущей популяции. Отношение числа мутаций к размеру популяции также будет являться параметром генетического алгоритма на рассматриваемом этапе.

Формирование следующего поколения. В результате скрещиваний и мутаций строится расширенная промежуточная популяция, состоящая из предыдущей популяции и результатов скрещиваний и мутаций. Из нее выбираются особи для формирования новой популяции такого же размера, как и предыдущая популяция. Выбор осуществляется следующим образом:

- в новую популяцию переносятся лучшие особи из промежуточной популяции, их число является параметром алгоритма на рассматриваемом этапе;
- из оставшихся особей промежуточной популяции с помощью "рулетки" выбираются особи для переноса в новую популяцию. Перенос продолжается до тех пор, пока размер новой популяции не достигнет размера предыдущей.

В итоге формируется новая популяция, которая готова к тестированию и следующей итерации генетического алгоритма.

Таблица 1

Результаты эксперимента 1

Алгоритм	Минимальный результат	Усредненный результат	Максимальный результат
№ 1	71	80,87	89
№ 2	84	92,10	100
№ 3	89	95,07	100
Предложенный алгоритм	95	99,47	100

Таблица 2

Результаты эксперимента 2

Алгоритм	Минимальный результат	Усредненный результат	Максимальный результат
№ 1	66	72,43	80
№ 2	82	89,63	96
№ 3	85	90,90	97
Предложенный алгоритм	96	97,53	100

Оператор скрещивания. Для скрещивания двух особей используется одноточечный оператор скрещивания двух битовых строк.

Оператор мутации. Оператор мутации для каждого состояния автомата с вероятностью, являющейся параметром алгоритма, случайным образом изменяет действия на переходах и номера состояний, в которые автомат переходит.

Заметим, что после выполнения оператора скрещивания или мутации, может получиться автомат с недостижимыми состояниями, но в отличие от работы [2] вводить операцию восстановления связей между состояниями не потребовалось.

5. Эксперименты

Ниже приводятся результаты экспериментов, основанных на применении трех алгоритмов, изложенных в работах [1, 2], и предлагаемого алгоритма.

Все эксперименты проводились при 100 особях в поколении. Число воздействий среды на флиб — 100. Число поколений — 400. В таблицах с результатами экспериментов (табл. 1, 2) приводятся минимальная, средняя и максимальная точности предсказаний построенных автоматов за 30 запусков. При этом для всех трех ранее предложенных алгоритмов авторами настоящей работы были повторены оба эксперимента по программе, опубликованной в работе [2].

Эксперимент 1. Битовая маска из 19 символов: 1111010010111101001. Результаты эксперимента приведены в табл. 1.

Результаты запусков алгоритмов представлены также графически (рис. 2, см. третью сторону обложки).

Из рассмотрения рис. 2 следует, что предлагаемый алгоритм значительно чаще строит предсказатели, обладающие 100 %-ной точностью — в 86 % случаев, против 7 % у прототипа (наиболее близкого аналога).

Выше вопрос о числе состояний предсказателей не рассматривался. Переходя к этому вопросу применительно к предлагаемому алгоритму, приведем сначала графики значений предлагаемой *fitness-функции*, так как она зависит от числа используемых состояний (рис. 3, см. третью сторону обложки).

Из рассмотрения графиков следует, что уже после 60 итераций алгоритм обеспечивает приближение *fitness-функции* к насыщению, что означает получение 100 %-ного предсказателя и минимизацию числа состояний по сравнению с прототипом. Проиллюстрируем графически высказанные утверждения.

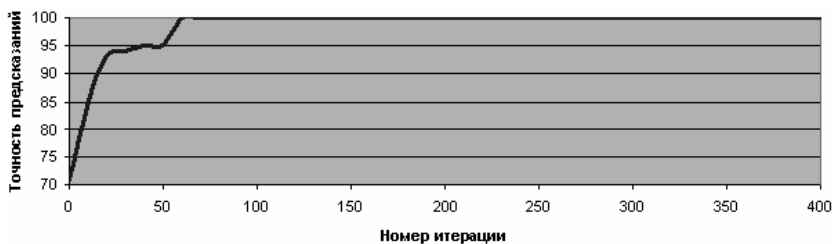


Рис. 4. Зависимость точности предсказаний от номера итерации

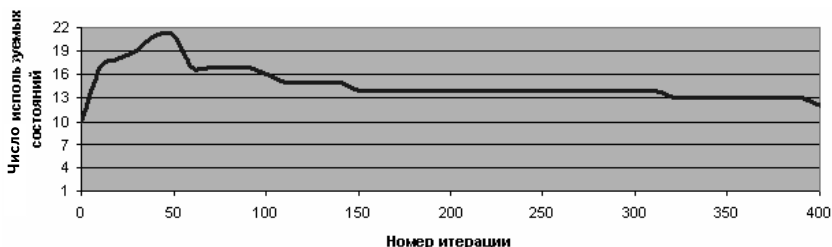


Рис. 5. Зависимость числа используемых состояний от номера итерации

На рис. 4 показано, что при 60 итерациях был получен автомат со 100 %-ной точностью предсказаний.

На рис. 5 показано, что для получения 100 %-ного предсказателя число используемых состояний росло, а после получения такого предсказателя выбранная *fitness-функция* обеспечила уменьшение числа используемых состояний. При этом отметим, что несмотря на то, что битовая маска содержит 19 символов, в ходе эксперимента строятся 100 %-ные предсказатели с числом состояний, превышающим это значение (21 состояние), а в конце эксперимента — с 12 состояниями.

На рис. 6 приведен граф переходов 100 %-ного предсказателя с 12 состояниями, построенный в эксперименте 1. В этом графе неиспользуемые состояния и переходы в них, естественно, не отображены. При этом отметим, что в приведенном графе некоторые из переходов возможно и не используются.

Эксперимент 2. Битовая маска из 31 символа: 1010111101100011110111110011001. Результаты эксперимента приведены в табл. 2.

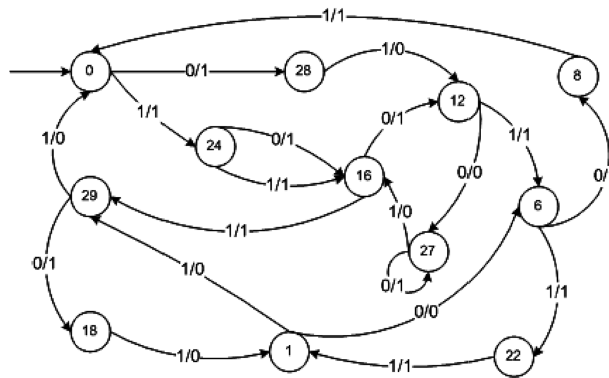


Рис. 6. Наилучший построенный автомат для эксперимента 1

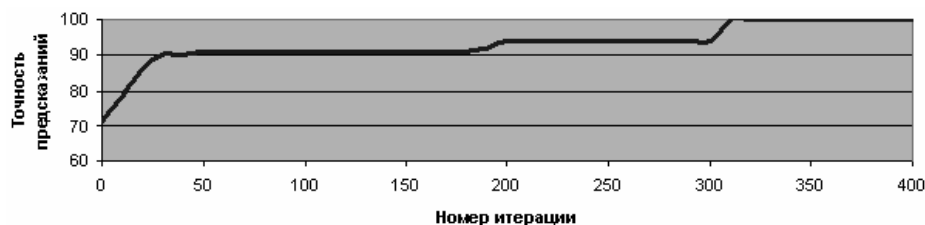


Рис. 9. Зависимость точности предсказаний от номера итерации

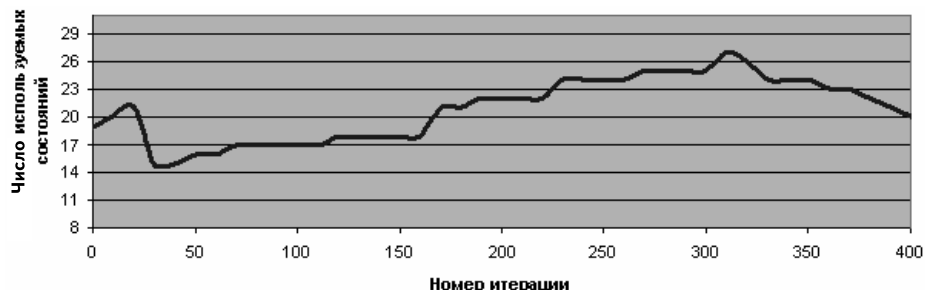


Рис. 10. Зависимость числа используемых состояний от номера итерации

Результаты запуска алгоритмов представлены также графически (рис. 7, см. третью сторону обложки).

Из рассмотрения рис. 7 следует, что в прототипе для рассматриваемой битовой строки вообще не удавалось построить 100 %-ный предсказатель, в то время как предлагаемый алгоритм это позволяет сделать в 17 % случаев.

Переходя к вопросу о числе состояний автомата в этом эксперименте, вновь рассмотрим графики значений *fitness-функций*, представленные на рис. 8 (см. третью сторону обложки).

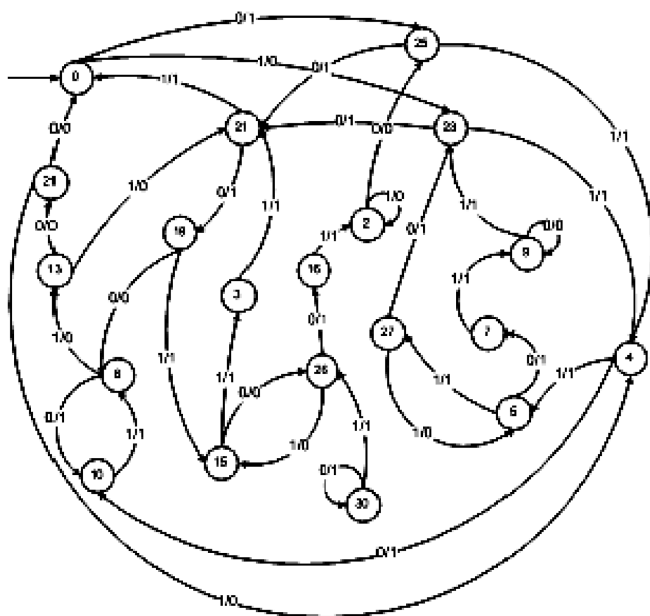


Рис. 11. Наилучший построенный автомат для эксперимента 2

На рис. 9 показано, что в районе 310-й итерации был получен автомат со 100 %-ной точностью предсказаний.

Рис. 10 иллюстрирует тот факт, что для получения 100 %-ного предсказателя число используемых состояний росло, а после получения такого предсказателя выбранная *fitness-функция* обеспечила уменьшение числа используемых состояний. При этом отметим, что в ходе данного эксперимента число состояний автомата не превышало длины битовой маски (31), а в конце эксперимента был построен 100 %-ный предсказатель с 20 состояниями.

На рис. 11 приведен граф переходов 100 %-ного предсказателя с 20 состояниями, построенный в эксперименте 2. В этом графе имеет место та же ситуация для неиспользуемых состояний и переходов, что и в графе на рис. 6.

Из рассмотренных примеров следует, что предложенный генетический алгоритм значительно эффективнее прототипа, как по частоте построения 100 %-ных предсказателей, так и по числу их состояний.

Таким образом, показано, что если автомат имеет число состояний меньше, чем длина битовой маски, то предложенный генетический алгоритм позволяет учесть ее специфику.

Заключение

Вычислительные эксперименты проводились на компьютере с процессором *Core 2 Duo* с тактовой частотой 1830 МГц. Время вычислений при этом в среднем составляло 40 с.

Отметим еще раз, что эффективность предлагаемого подхода базируется на новизне *fitness-функции*, которая в отличие от прототипа учитывает не только число верно предсказанных значений, но и число состояний автомата и длину битовой маски.

Список литературы

1. Воронин О., Дюдни А. Дарвинизм в программировании // Мой компьютер. 2004. № 35. <http://www.myscomp.kiev.ua/text/7458>
2. Лобанов П. Г., Шальто А. А. Использование генетических алгоритмов для автоматического построения конечных автоматов в задаче о "Флибах" / Материалы 1-й Российской мультиконференции по проблемам управления. Сборник докладов 4-й Всероссийской научной конференции "Управление и информационные технологии" (УИТ-2006). СПбГЭТУ "ЛЭТИ". 2006. С. 144-149. <http://is.ifmo.ru/works/flib/>
3. Николенко С. И. Лекция по генетическим алгоритмам. <http://logic.pdmi.ras.ru/~sergey/teaching/ml/notes-04-genetic.pdf>

А. И. Долгов, д-р техн. наук, Д. А. Короченцев,
Ростовский военный институт ракетных войск
им. Главного маршала артиллерии
Неделина М. И., г. Ростов-на-Дону

Сложение возможностей и вероятностных нечетких треугольных чисел

Введено понятие нечетких чисел с возможностью и вероятностной мерами принадлежности и предлагается способ их интегрального описания. Осуществлено качественное сравнение результатов сложения возможностей и вероятностных треугольных чисел.

В существующих публикациях, посвященных выполнению операций над нечеткими числами, не содержатся оценки погрешностей представления степени размытости получаемых результатов. Исследования, посвященные получению таких оценок, являются весьма актуальными, так как определяют применимость нечетких чисел в реальных системах принятия решений.

В данной публикации предлагается способ интегрального описания возможностей и вероятностных нечетких чисел, различающихся функциями, характеризующими соответственно возможность и вероятностную меры принадлежности, которые определяют степень размытости числовых значений.

В приводимой далее постановке решается задача качественной сравнительной оценки функций принадлежности для результатов сложения возможностей и вероятностных чисел.

Описание возможностей и вероятностных чисел

Возможностное число $\tilde{A}_\mu$ в наиболее общем виде может быть представлено посредством следующего интегрального описания:

$$\tilde{A}_\mu = \left(\bigcup_{A_{1,1}}^{A_{1,2}} \mu_1^A(x) dx \right) \cup \dots \cup \left(\bigcup_{A_{i,1}}^{A_{i,2}} \mu_i^A(x) dx \right) \cup \dots \cup \left(\bigcup_{A_{n,1}}^{A_{n,2}} \mu_n^A(x) dx \right). \quad (1)$$

В этой записи i -й член ($i = 1, \dots, n$) символизирует i -ю составную часть возможностного числа, задаваемую на i -м непрерывном интервале множества действительных чисел, включающем

допустимые значения $A_{i,1} \leq x \leq A_{i,2}$, с явным указанием для рассматриваемых значений x из указанного интервала значения функции принадлежности $0 \leq \mu_i^A(x) \leq 1$, при этом $\cup$ символизирует объединение значений функции принадлежности, соответствующих бесконечно малым значениям dx , а знак $\cup$ обозначает объединение составных частей. Совокупность действительных чисел, принадлежащих всем указанным интервалам $A_{i,1} \leq x \leq A_{i,2}$, ($i = 1, \dots, n$), будем называть областью определения возможностного числа $\tilde{A}_\mu$.

Обозначим μ_i^A максимальное значение $\mu_i^A(x)$ на i -м интервале, т. е. $\mu_i^A = \max \mu_i^A(x)$ при $A_{i,1} \leq x \leq A_{i,2}$.

Запись (1) является наиболее общей в том смысле, что символизирует самые различные частные случаи возможностей чисел:

- непрерывному [1] нечеткому числу $\tilde{A}$ соответствуют $A_{i,2} = A_{i+1,1}$ для всех значений $i = 1, \dots, (n-1)$, в то время как случай $A_{i,2} \neq A_{i+1,1}$ хотя бы для одного i соответствует нечеткому числу $\tilde{A}$, не относящемуся к непрерывным;
- нормальному нечеткому числу [2, 3] соответствует $\mu_i^A \leq 1$, где $i = 1$, при этом хотя бы для одного i имеет место $\mu_i^A = 1$;
- для выпуклого [1] нечеткого числа в случае любых значений $x, y, z \in [A_{1,1}, A_{n,2}]$, таких что $x \leq y \leq z$, выполняется следующее соотношение:

$$\mu_i^A(y) \geq \min(\mu_i^A(x), \mu_i^A(z));$$

- для симметричного [1] нечеткого числа существует $x_c = \frac{A_{1,1} + A_{n,2}}{2}$, такое что $\forall \Delta x: (x_c \pm \Delta x) \in [A_{1,1}, A_{n,2}]$ соблюдается равенство $\mu^A(x_c + \Delta x) = \mu^A(x_c - \Delta x)$.

Вероятностное число $\tilde{A}_p$, по аналогии с вышеизложенным, можно записать в следующем общем виде:

$$\tilde{A}_p = \left(\bigcup_{A_{1,1}}^{A_{1,2}} p_1^A(x) dx \right) \cup \dots \cup \left(\bigcup_{A_{i,1}}^{A_{i,2}} p_i^A(x) dx \right) \cup \dots \cup \left(\bigcup_{A_{n,1}}^{A_{n,2}} p_n^A(x) dx \right). \quad (2)$$

В этой записи, как и в (1), i -й член также символизирует i -ю составную часть вероятностного числа, задаваемую на i -м непрерывном интервале множества действительных чисел, включающем допустимые значения $A_{i,1} \leq x \leq A_{i,2}$, с явным указанием плотности вероятностей $0 \leq p_i^A(x) \leq 1$ для рассматриваемых значений x из указанного интервала.

Записи (2) можно поставить в соответствие интегральное выражение

$$\int_{A_{1,1}}^{A_{1,2}} p_1^A(x) dx + \dots + \int_{A_{i,1}}^{A_{i,2}} p_i^A(x) dx + \dots + \int_{A_{n,1}}^{A_{n,2}} p_n^A(x) dx = S_1^{p^A} + \dots + S_i^{p^A} + \dots + S_n^{p^A}.$$

В данном выражении i -е слагаемое соответствует i -му слагаемому в (2) и характеризует площадь $S_i^{p^A}$ соответствующей фигуры в двумерном пространстве $(x, p(x))$, причем будем полагать, что с учетом аксиоматики теории вероятностей сумма интегралов должна быть приравнена 1, т. е. $S_1^{p^A} + \dots + S_i^{p^A} + \dots + S_n^{p^A} = 1$.

Подводя итог сказанному, следует отметить, что возможность и вероятностные числа имеют много общего, а именно:

- в обоих случаях рассматривается некоторое (так называемое универсальное) множество допустимых значений x действительных чисел;
- и в том и в другом случае определяется некоторая мера принадлежности статистического характера, значение которой приписывается каждому из значений x ;
- вероятностное число (2) при $\mu_i^A(x) = p_i^A(x)$ и $S_1^{p^A} + \dots + S_i^{p^A} + \dots + S_n^{p^A} = 1$ представляет собой частный случай возможностного числа (1), при этом вероятностное число, как и возможностное, может быть непрерывным, выпуклым, симметричным.

Среди непрерывных выпуклых нормальных симметричных возможностных и вероятностных нечетких чисел простейшими являются треугольные числа.

Возможностное и вероятностное сложение треугольных чисел

Под возможностным и вероятностным сложением понимается сложение чисел с возможностными и вероятностными мерами принадлежности соответственно.

Как известно, для возможностных чисел широко используется алгебра (в виде совокупности правил сложения, вычитания, умножения и деления), построенная на основе общеизвестного принципа обобщения [2, 5]. Алгебра вероятностных чисел реализуема путем классических преобразований вероятностных показателей [4].

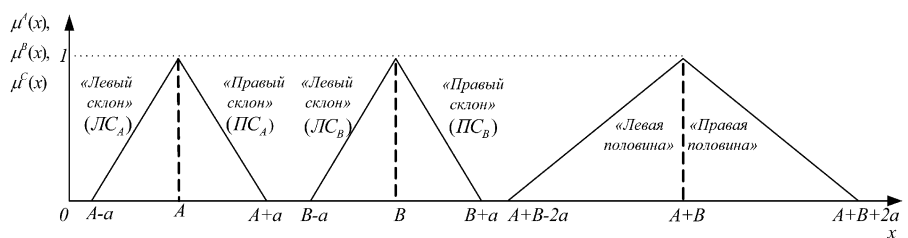


Рис. 1. Операнды, представленные возможностными числами

Будем считать, что слагаемые, представленные в виде возможностных операндов $\tilde{A}_\mu$ и $\tilde{B}_\mu$ и вероятностных операндов $\tilde{A}_p$ и $\tilde{B}_p$, а также возможностная сумма $\tilde{C}_\mu = \tilde{A}_\mu + \tilde{B}_\mu$, являются треугольными числами [4] и описываются соотношениями

$$\begin{aligned} \tilde{A}_\mu &= \left(\bigcup_{A-a}^A \frac{a-A+x}{a} dx \right) \cup \left(\bigcup_A^{A+a} \frac{a+A-x}{a} dx \right); \\ \tilde{B}_\mu &= \left(\bigcup_{B-a}^B \frac{a-B+x}{a} dx \right) \cup \left(\bigcup_B^{B+a} \frac{a+B-x}{a} dx \right); \\ \tilde{C}_\mu &= \left(\bigcup_{C-2a}^C \frac{2a-C+x}{2a} dx \right) \cup \left(\bigcup_C^{C+2a} \frac{2a+C-x}{2a} dx \right); \\ \tilde{A}_p &= \left(\bigcup_{A-a}^A \frac{a-A+x}{a^2} dx \right) \cup \left(\bigcup_A^{A+a} \frac{a+A-x}{a^2} dx \right); \\ \tilde{B}_p &= \left(\bigcup_{B-a}^B \frac{a-B+x}{a^2} dx \right) \cup \left(\bigcup_B^{B+a} \frac{a+B-x}{a^2} dx \right). \end{aligned}$$

Рис. 1 иллюстрирует графически представление слагаемых $\tilde{A}_\mu$, $\tilde{B}_\mu$ и возможностной суммы $\tilde{C}_\mu = \tilde{A}_\mu + \tilde{B}_\mu$, полученной на основе принципа обобщения. На рисунке ЛС и ПС — левый и правый склоны соответственно.

Рис. 2 иллюстрирует представление вероятностных слагаемых $\tilde{A}_p$, $\tilde{B}_p$ и гипотетической вероятностной суммы $\tilde{C}_p = \tilde{A}_p + \tilde{B}_p$.

Рассчитаем значения вероятностной функции принадлежности суммы исходя из вероятностных функций принадлежности слагаемых теоретико-вероятностными методами.

В дальнейшем удобно выполнять операции над искусственно вводимыми для использования переменными x' , y' , z' , представляющими собой значения x , приведенные для каждого операнда $\tilde{A}_p$, $\tilde{B}_p$ и $\tilde{C}_p$ к началу отчета относительно 0, при этом соответственно

$$\begin{aligned} x' &= x - (A - a); \\ y' &= x - (B - a); \\ z' &= x - (A + B - 2a), \end{aligned} \quad (4)$$

а впоследствии вернуться к выражениям, записанным через исходные переменные.

Перейдем от вероятностных слагаемых $\tilde{A}_p$ и $\tilde{B}_p$ к вероятностным слагаемым $\tilde{A}'_p$, $\tilde{B}'_p$, записанным через переменные x' и y' , в результате получим следующие выражения:

$$\tilde{A}'_p = \left(\bigcup_0^a \frac{x'}{a} dx' \right) \cup \left(\bigcup_a^{2a} \frac{2a-x'}{a} dx' \right); \quad (5)$$

$$\tilde{B}'_p = \left(\bigcup_0^a \frac{y'}{a} dy' \right) \cup \left(\bigcup_a^{2a} \frac{2a-y'}{a} dy' \right). \quad (6)$$

После перехода к операндам, записанным через искусственно вводимые переменные x' , y' в соответствии с выражениями (5) и (6), они оказываются приведенными к началу отсчета, что иллюстрируется рис. 3.

В соответствии с аксиоматикой теории вероятностей функция плотности вероятностей суммы $p^{\tilde{C}}(z' = x' + y')$ может быть сформирована как результат интегрирования по переменной x' произведений вероятностных значений $p^{A'}(x')$ и $p^{B'}(y')$ для значений $y' = z' - x'$, при этом

$$p^{A'}(x') = \begin{cases} p_1^{A'}(x') = \frac{x'}{a^2}, & \text{если } 0 \leq x' \leq a; \\ p_2^{A'}(x') = \frac{(2a-x')}{a^2}, & \text{если } a \leq x' \leq 2a; \end{cases} \quad (7)$$

$$p^{B'}(y') = \begin{cases} p_1^{B'}(y') = \frac{y'}{a^2}, & \text{если } 0 \leq y' \leq a; \\ p_2^{B'}(y') = \frac{(2a-y')}{a^2}, & \text{если } a \leq y' \leq 2a, \end{cases}$$

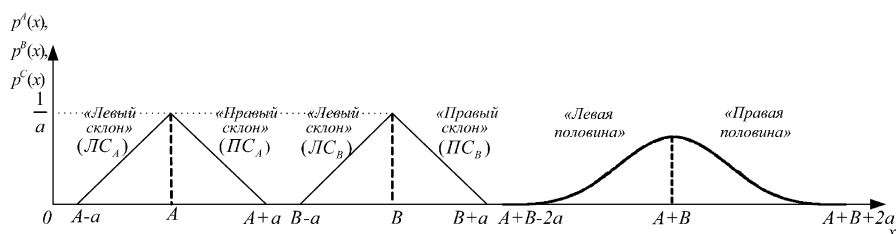


Рис. 2. Операнды, представленные вероятностными числами

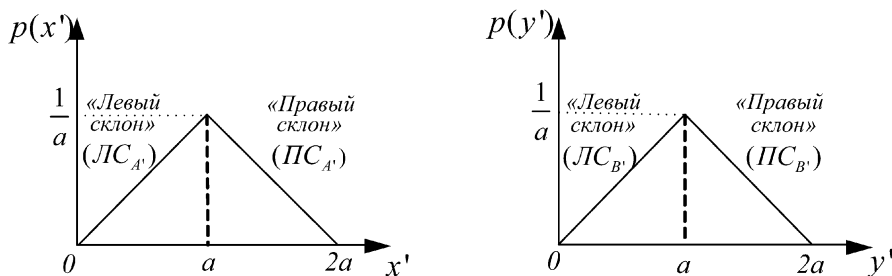


Рис. 3. Вероятностные слагаемые $\tilde{A}'_p$, $\tilde{B}'_p$, приведенные к началу отсчета

где $\frac{x'}{a^2}$ — уравнение прямой, описывающее "левый склон" (ЛС_{A'}) $\tilde{A}'_p$; $\frac{(2a-x')}{a^2}$ — уравнение прямой, описывающее "правый склон" (ПС_{A'}) $\tilde{A}'_p$; $\frac{y'}{a^2}$ — уравнение прямой, описывающее "левый склон" (ЛС_{B'}) $\tilde{B}'_p$; $\frac{(2a-y')}{a^2}$ — уравнение прямой, описывающее "правый склон" (ПС_{B'}) $\tilde{B}'_p$.

Для нахождения результата $\tilde{C}'_p$ вероятностными методами, которые сводятся к интегрированию, следует определиться с пределами интегрирования. Пределы интегрирования находятся с учетом границ $\tilde{C}'_\mu$. Для определения границ $\tilde{C}'_\mu$ перейдем от возможностной суммы $\tilde{C}_\mu$ к возможностной сумме $\tilde{C}'_\mu$, записанной через переменную z' , что иллюстрируется рис. 4.

Для треугольных чисел при $0 \leq x' \leq 2a$, $0 \leq y' \leq 2a$ интервал $0 \leq z' \leq 4a$ может быть разделен на два одинаковых интервала:

$$0 \leq z' \leq 2a; \quad (8)$$

$$2a \leq z' \leq 4a, \quad (9)$$

соответствующих "левой половине" и "правой половине" $\tilde{C}'_\mu$ (рис. 4).

Так как сумма двух симметричных нечетких чисел является также симметричным нечетким числом [3], достаточно рассмотреть получение только "левой половины" $\tilde{C}'_\mu$, которая представлена выражением (8).

Для получения пределов интегрирования разделим интервал (8) на два участка:

$$0 \leq z' \leq a; \quad (10)$$

$$a \leq z' \leq 2a, \quad (11)$$

для которых интегрирование по x' , как будет показано далее, должно проводиться с разными выражениями подынтегральных функций, записанных через x' и $y' = z' - x'$.

Определим пределы интегрирования для участка (10), для которого выполняются соотношения

$$0 \leq z' \leq a; \quad (12)$$

$$0 \leq x' \leq a(\text{ЛС}_{x'}); \quad (13)$$

$$0 \leq y' \leq a(\text{ЛС}_{y'}). \quad (14)$$

Справа от неравенств символически указаны функции плот-

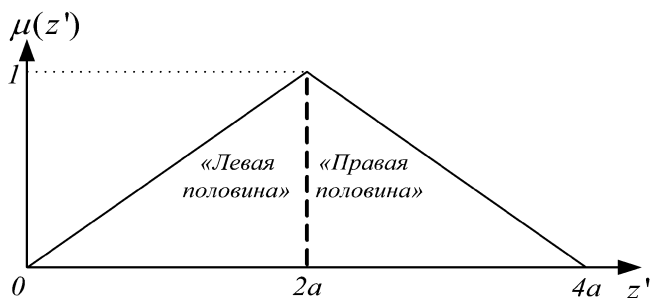


Рис. 4. Возможностная сумма $\tilde{C}_\mu$, приведенная к началу отсчета

ностей вероятностей в соответствующих им интервалах. Функции, являющиеся треугольными, описываются прямыми, соответствующими левым сторонам треугольников, которые названы склонами (ЛС $_{x'}$) и (ПС $_{x'}$) (см. рис. 2).

Складывая неравенство $0 \leq y'$ из левой части (14) с $x' = x'$, получаем $x' + 0 \leq x' + y'$. Заменяя в правой части полученного неравенства $x' + y'$ на z' , приходим к $x' \leq z'$. С учетом того, что из (13) $0 \leq x'$ и из (12) $z' \leq a$, переходим от последнего неравенства к следующему выражению:

$$0 \leq x' \leq z' \leq a'. \quad (15)$$

Выражение (15) определяет искомые пределы интегрирования по переменной x' на участке (10).

Определим пределы интегрирования для участка (11). Значения величины z' на этом интервале получаются путем сложения x' и y' , лежащих на следующих трех интервалах, соответствующих различным комбинациям функций плотностей вероятностей $p^{A'}(x')$ и $p^{B'}(y')$:

$$1) 0 \leq x' \leq a \text{ (ЛС}_{x'}); \quad (16)$$

$$a \leq y' \leq 2a \text{ (ПС}_{y'}); \quad (17)$$

$$2) 0 \leq x' \leq a \text{ (ЛС}_{x'}); \quad (18)$$

$$0 \leq y' \leq a \text{ (ЛС}_{y'}); \quad (19)$$

$$3) a \leq x' \leq 2a \text{ (ПС}_{x'}); \quad (20)$$

$$0 \leq y' \leq a \text{ (ЛС}_{y'}). \quad (21)$$

Вычислим пределы интегрирования для участка (11) при интервалах изменения величин x' и y' , представленных выражениями (16) и (17).

Складывая неравенство $a \leq y'$ из левой части (17) с $x' = x'$, получаем $x' + a \leq x' + y'$. Заменяя в правой части полученного неравенства $x' + y'$ на z' , приходим к $x' + a \leq z'$. Вычитая из обеих частей последнего неравенства величину a , получим следующее выражение: $x' \leq z' - a$. С учетом того, что из (16) $0 \leq x'$ и из (11) $z' \leq 2a$, переходим от последнего неравенства к следующему выражению:

$$0 \leq x' \leq z' - a \leq a. \quad (22)$$

Выражение (22) определяет искомые пределы интегрирования по переменной x' на участке (11) при изменении величин x' и y' согласно выражениям (16) и (17).

Вычислим пределы интегрирования для участка (11) при интервалах изменения величин x' и y' , представленных выражениями (18) и (19).

Складывая неравенство $y' \leq a$ из правой части (19) с $x' = x'$, получаем $x' + y' \leq x' + a$. Заменяя в левой части полученного неравенства $x' + y'$ на z' , приходим к $z' \leq x' + a$. Вычитая из обеих частей последнего неравенства величину a , получим следующее выражение: $z' - a \leq x'$. С учетом того, что из (18) $x' \leq a$ и из (11) $a \leq z'$, переходим от последнего неравенства к следующему выражению:

$$0 \leq z' - a \leq x' \leq a. \quad (23)$$

Выражение (23) определяет искомые пределы интегрирования по переменной x' на участке (11) при изменении величин x' и y' согласно выражениям (18) и (19).

Вычислим пределы интегрирования для участка (11) при интервалах изменения величин x' и y' , представленных выражениями (20) и (21).

Складывая неравенство $0 \leq y'$ из левой части (21) с $x' = x'$, получаем $x' + 0 \leq x' + y'$. Заменяя в левой части полученного неравенства $x' + y'$ на z' , приходим к $z' \leq x' + a$. Вычитая из обеих частей последнего неравенства величину a , получим следующее выражение: $z' - a \leq x'$. С учетом того, что из (20) $x' \leq a$ и из (11) $a \leq z'$, переходим от последнего неравенства к следующему выражению:

$$0 \leq z' - a \leq x' \leq a. \quad (24)$$

Выражение (24) определяет искомые пределы интегрирования по переменной x' на участке (11) при изменении величин x' и y' согласно выражениям (20) и (21).

Теперь по известным значениям $p^{A'}(x)$ и $p^{B'}(y')$, характеризующим плотность распределения вероятностей значений слагаемых x' и y' , определим плотность распределения вероятностей $p_i^{\tilde{C}'}(z') = p_i^{\tilde{C}'}(x' + y')$ значений их суммы z' . Эти действия заслуживают более подробного рассмотрения.

Функцию $p^{\tilde{C}'}(z')$ найдем в результате интегрирования значений произведений плотностей вероятностей слагаемых, приводящих к одинаковым результатам на рассматриваемых участках в первой половине области значений z' (значения второй половины области могут быть легко получены с учетом симметрии).

На 1-м участке (10) в интервале $0 \leq z' \leq a$ для слагаемых, удовлетворяющих соотношениям (13) и (14), получаем:

$$\begin{aligned} p^{\tilde{C}'}(z') &= \int_0^{z'} p_1^{A'}(x') p_1^{B'}(z' - x') dx' = \\ &= \int_0^{z'} \frac{x'}{a^2} \frac{z' - x'}{a^2} dx' = \frac{z'^3}{6a^4}. \end{aligned} \quad (25)$$

На 2-м участке (11) величина z' получается путем интегрирования плотностей вероятностей $p^{A'}(x')$ и $p^{B'}(y')$, описываемых разными комбинациями выражений (16) и (17), (18) и (19), (20) и (21), которые образуют три участка с пределами интегрирования по x' (22), (23) и (24) соответственно:

$$\begin{aligned} p^{\tilde{C}'}(z') &= \int_0^{z'-a} p_1^{\tilde{A}'}(x') p_2^{\tilde{B}'}(2a - (z' - x')) dx' + \\ &+ \int_{z'-a}^a p_1^{\tilde{A}'}(x') p_1^{\tilde{B}'}(z' - x') dx' + \int_a^{z'} p_2^{\tilde{A}'}(2a - x') \times \\ &\times p_1^{\tilde{B}'}(z' - x') dx' = \int_0^{z'-a} \frac{x'}{a^2} \frac{(2a - (z' - x'))}{a^2} dx' + \\ &+ \int_{z'-a}^a \frac{x'}{a^2} \frac{z' - x'}{a^2} dx' + \int_a^{z'} \frac{2a - x'}{a^2} \frac{z' - x'}{a^2} dx' = \\ &= \frac{12az'^2 - 12a^2z' + 4a^3 - 3z'^3}{6a^4}. \end{aligned} \quad (26)$$

После получения значений интегральных функций перейдем от $\tilde{C}'_p$ к истинному значению $\tilde{C}_p$. Для этого, подставляя в полученные выражения интегральной функции (25), (26) вместо z' значения x из (4), получим следующие соотношения:

$$f_1^{\tilde{C}_p}(x) = \frac{(x - A - B + 2a)^3}{6a^4}, \quad (27)$$

где $f_1^{\tilde{C}_p}(x)$ является функцией, описывающей результат сложения, при изменении значения $A + B - 2a \leq x \leq A + B - a$;

$$\begin{aligned} f_2^{\tilde{C}_p}(x) &= \\ &= \frac{12a(x - A - B + 2a)^2 - 12a^2(x - A - B + 2a) +}{6a^4} \rightarrow \\ &\rightarrow \frac{+ 4a^3 - 3(x - A - B + 2a)}{6a^4}, \end{aligned} \quad (28)$$

где $f_2^{\tilde{C}_p}(x)$ есть функция, описывающая результат сложения, при изменении значения $A + B - a \leq x \leq A + B$.

Представленные выше соотношения описывают только "левую половину" результирующего ве-

роятностного числа $\tilde{C}_p$. Так как результирующее нечеткое число $\tilde{C}_p$ является симметричным, получение соотношений, описывающих его "правую половину", может быть реализовано с использованием преобразований координат.

"Правая половина" вероятностного числа $\tilde{C}_p$ может быть получена переносом функций, описывающих "левую половину" числа $\tilde{C}_p$, вместе с осью симметрии этого числа (далее называемой осью симметрии) в точку начала координат; поворотом "левой половины", прилегающей слева к оси ординат, относительно оси симметрии (совпадающей после выполненного переноса с осью ординат) и переносом полученной "правой половины" в исходную точку.

Для переноса функций, описывающих "левую половину" числа $\tilde{C}_p$, вместе с осью симметрии этого числа в точку начала координат необходимо прибавить к аргументу функций, представленных соотношениями (27) и (28), величину сдвига $A + B$, т. е. вместо аргумента x следует указать $x + A + B$. После этого поворачиваем "левую половину", находящуюся слева от оси симметрии, для чего умножим аргумент x на -1 и в результате получим функции, симметричные относительно оси x , описывающие "правую половину": $A + B - x$. Перенесем ось симметрии с полученной "правой половиной" в исходную точку, т. е. прибавим к аргументу x значение $A + B$ и в результате получим следующее значение аргумента функций, описывающих "правую половину", а именно: $2A + 2B - x$. Подставляя вновь полученное значение аргумента в соотношения (27) и (28) и проводя преобразования, которые здесь не приведены, получим соотношения, описывающие "правую половину" результирующего числа $\tilde{C}_p$ на интервалах $A + B \leq x \leq A + B + a$ и $A + B + a \leq x \leq A + B + 2a$. Ниже представлены эти соотношения:

$$\begin{aligned} f_3^{\tilde{C}_p}(x) &= \\ &= \frac{12a(2a - x + A + B)^2 - (24a - 12x + 12A + 12B)a^2 +}{6a^4} \rightarrow \\ &\rightarrow \frac{- 3(2a - x + A + B) + 4a^3}{6a^4}, \end{aligned} \quad (29)$$

где $f_3^{\tilde{C}_p}(x)$ является функцией, описывающей результат сложения, при изменении значения $A + B \leq x \leq A + B + a$;

$$f_4^{\tilde{C}_p}(x) = \frac{(A + B + 2a - x)^3}{6a^4}, \quad (30)$$

где $f_4^{\tilde{C}_p}(x)$ представляет собой функцию, описывающую результат сложения, при изменении значения $A + B + a \leq x \leq A + B + 2a$.

Качественное сопоставление результатов сложения возможностей и вероятностных нечетких треугольных чисел приводит к следующим выводам:

- на интервале определения возможностей и вероятностных сумм точки с максимальным значением возможностной и вероятностной функций принадлежности совпадают;
- возможностная и вероятностная функции принадлежности, характеризующие степень размытости сумм возможностей и вероятностных треугольных чисел, оказываются существенно различающимися;
- отмеченные существенные различия придают актуальность сравнительной оценке погрешностей представления степени размытости результатов при выполнении вероятностных и

возможностных операций, проводимых как с треугольными, так и с другими видами нечетких чисел.

Список литературы

1. Алтуний А. Е., Семухин М. В. Модели и алгоритмы принятия решений в нечетких условиях. Тюмень: Изд-во Тюменского государственного университета, 2000. 352 с.
2. Борисов А. Н., Алексеев А. В., Меркурьева Г. В. и др. Обработка нечеткой информации в системах принятия решений. М.: Радио и связь, 1989. 304 с.
3. Борисов А. Н., Крумберг О. А., Федоров И. П. Принятие решений на основе нечетких моделей. Рига: Зинатне, 1990. 184 с.
4. Вентцель Е. С. Теория вероятностей: Учеб. для вузов. 5-е изд. стер. М.: Высш. шк., 1998. 576 с.
5. Дюбуа Д., Прад А. Теория возможностей. Приложения к представлению знаний в информатике. М.: Радио и связь. 1990. 288 с.



КОМПЬЮТЕРНАЯ ГРАФИКА

УДК 004.92:004.272.2

А. Н. Милов,

ГУП НПЦ "ЭЛВИС" г. Москва

(<http://multicore.ru>) quattro@elvees.com

Два подхода к организации масштабируемых графических архитектур стандарта OpenGL на базе цифровых процессоров обработки сигналов

Рассматривается вопрос об организации интерфейса OpenGL для построения масштабируемых графических архитектур на базе цифровых процессоров обработки сигналов и методах организации потоковых вычислений в таких архитектурах. Приводятся результаты реализации двухпроцессорной графической системы с внешним интерфейсом OpenGL на базе систем на кристалле серии "Мультикор".

Введение

ГУП НПЦ "ЭЛВИС" является разработчиком собственной IP-ядерной (*intellectual property*) платформы проектирования систем на кристалле "Мультикор". Она включает в себя широкий набор программируемых ядер и других аппаратных средств [1]:

- RISC-ядро включает сопроцессоры CP0 и CP1 (MIPS32) и является R4000-совместимым. Ядро име-

ет собственный кэш программной памяти, который позволяет значительно ускорить загрузку команд из внешней памяти;

- DSP-ядра являются высокопроизводительными 32-битными устройствами, обрабатывающими данные в различных форматах с фиксированной и плавающей точкой. Они имеют типичную для устройств DSP "гарвардскую" архитектуру по потокам и форматам обрабатываемых данных и позволяют осуществлять до двух вычислительных операций и до двух пересылок в течение одного командного цикла. Кроме того, система инструкций DSP позволяет наращивать производительность системы по SIMD-типу [2];
- контроллер PCI (PMSC — *PCI Master Slave Controller*) предназначен для обмена данными между шиной PCI в соответствии со спецификацией PCI Local Bus Specification. Rev.2.2 и любой областью внутренней и внешней памяти системы;
- линковые порты, интерфейсно совместимые с ADSP-21160 и имеющие пропускную способность до 60 Мбайт/с;
- порт внешней памяти (32-битные шины адреса и данных), осуществляющий связь системы с широким набором внешних устройств памяти (SDRAM, FLASH, SRAM);
- контроллеры прямого доступа в память (DMA), позволяющие эффективно организовывать цепочки одно- или двумерных передач между внутренней памятью и линковыми портами, внешней памятью, PCI;
- таймеры и другие стандартные устройства сопряжения (Serial port, UART и т. д.).

Системы на кристалле "Мультикор" разрабатываются для коммерческих, военных и космических применений.

Как известно, проблема визуализации трехмерной графики для бортовых систем решена на недостаточном

уровне, так как используемые в бортовых условиях вычислительные устройства являются недостаточно производительными, а большинство из них вообще не предназначено для вычислений такого уровня.

Вместе с тем, в области компьютерной графики в последние годы прослеживается тренд к переходу от преимущественно аппаратной обработки данных к программируемому и масштабируемому архитектурам.

Например, появившаяся в 2002 г. технология программируемых графических процессоров (*shaders*) [3] позволила программировать разнообразные эффекты на графических процессорах, что является, по сути, ничем иным, как введением программируемого конвейера. Преимущества таких систем очевидны — гибкость и переносимость программируемых эффектов.

Вместе с тем, нельзя не отметить выход в 2006 г. графического ускорителя G80 от nVidia, архитектура которого включает 128 универсальных потоковых процессоров (SP — *Stream Processor*), работающих на частоте 1,35 ГГц и способных выполнять простые операции MAD (умножение и сложение) и MUL (умножение) за один командный цикл [4]. Описанные SP-процессоры могут выполнять одну из трех графических стадий с входными данными: вершинные преобразования, сборку примитивов, растеризацию. Они также могут объединяться в кластеры по SIMD-типу и обрабатывать входные данные много быстрее, что является относительно легким инструментом повышения общей производительности системы. Более того, эта кластерная система является универсальной, т. е. может быть запрограммирована для практически любых вычислений, абсолютно не связанных с графикой вообще.

Безусловно, архитектура системы включает также блоки аппаратной фильтрации текстур, фрагментные процессоры и общую кэш-память данных (*cache crossbar*), без которых немисливо обеспечить необходимую производительность для современной графической системы бытового уровня.

В данной статье рассмотрены классификация вызовов стандарта OpenGL в контексте реализации масштабируемых графических архитектур на базе цифровых процессоров обработки сигналов DSP (*Digital Signal Processors*), методы организации потоков графических вычислений в таких архитектурах и приведены результаты реализации графической библиотеки OpenGL для систем на кристалле серии "Мультикор".

Поскольку графическая система на базе систем "Мультикор" создавалась для бортовых применений, в ней была заложена следующая основная функциональность OpenGL:

- задание модельно-видовой и проекционной матриц трансформаций модели;
- задание области вывода кадрового буфера;
- описание трехмерной модели посредством всех поддерживаемых спецификацией OpenGL примитивов;
- задание нормалей к вершинам, цветов или текстурных координат;
- задание и масштабирование текстур с размерами 2^n , $n \in N$, $n \leq 7$ (т. е. до 128×128 текселей);

- задание (и включение отбраковки) лицевых граней примитивов (*culling*);
- создание списков вызова (*display lists*);
- задание параметров и пространственного положения источника света;
- работа непосредственно с кадровым буфером.

OpenGL как стандарт взаимодействия с графическими системами

OpenGL (*Open Graphics Library*) — это открытый и мобильный стандарт для описания двух- и трехмерных сцен с использованием унифицированных команд [5]. Впервые он был введен в 1992 г. компанией *Silicon Graphics* с целью стандартизировать взаимодействие между программами и графическими устройствами. С момента выхода первой спецификации OpenGL обогатился множеством дополнений и расширений, но сохранил преемственность версий. Например, программа, написанная в соответствии со спецификацией OpenGL версии 1.2, работает одинаково во всех последующих версиях спецификации.

Кроме этого, OpenGL является открытым международным стандартом с открытой документацией и исходными кодами, что позволило реализовать графические библиотеки, поддерживающие OpenGL, для множества самых различных платформ и операционных систем: Windows, Linux, Solaris, MacOS, Alpha, PalmOS и др. Таким образом, OpenGL является, по сути, единственным межплатформенным стандартом взаимодействия пользователей программ с графическими системами.

Стандарт предусматривает архитектуру взаимодействия "клиент—сервер" при организации такого взаимодействия. Это означает, что клиент формирует вызовы OpenGL и посылает их на сервер для обработки и формирования окончательного изображения, при этом клиент и сервер OpenGL могут быть даже разными вычислительными устройствами, соединенными, например, через локальную сеть.

Вместе с тем, аппаратно-программная модель OpenGL предусматривает программное исполнение вызовов OpenGL, если таковые не реализованы аппаратно.

Основные операции конвейера OpenGL приведены на рис. 1.

Заметим, что приведенный конвейер является скорее логическим устройством архитектур OpenGL, чем иллюстрацией их аппаратного устройства. Более того, аппаратное устройство каждого графического ускорителя уникально ввиду постоянного совершенствования аппаратуры и алгоритмов. Поэтому при построении архитектур OpenGL широко распространено понятие *контекста OpenGL*.

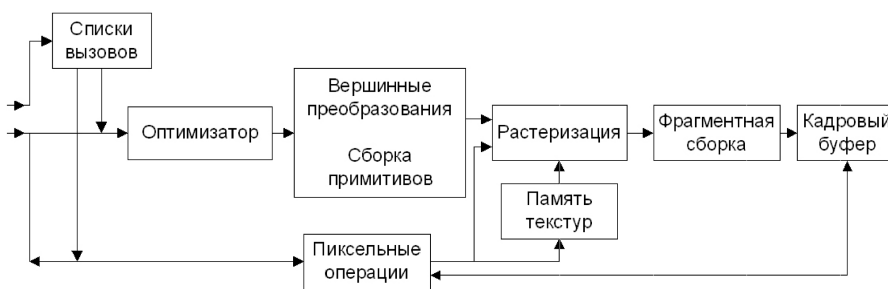


Рис. 1. Операции конвейера OpenGL

Контекст OpenGL — это упорядоченный набор данных, формируемый и изменяемый вызовами OpenGL, который содержит информацию о текущих настройках сцены, созданных клиентом OpenGL, и однозначно определяющий состояние графической системы между вызовами. Он может быть реализован в виде программно доступной структуры данных как в области памяти клиента OpenGL, так и сервера. Предлагаемая классификация вызовов OpenGL предусматривает эффективное разделение контекста между памятью клиента и памятью сервера.

Разделяемый интерфейс OpenGL

В случае организации разделяемого интерфейса OpenGL для многопроцессорных графических архитектур первоочередной задачей является организация эффективных потоков данных и вычислений в системе при соблюдении положений стандарта OpenGL.

Для решения задачи предлагается классификация вызовов по влиянию на конвейер и контекст OpenGL (табл. 1).

Интерфейсные и контекстные вызовы могут быть реализованы исключительно в рамках ресурсов клиента OpenGL, поскольку их исполнение фактически сводится к чтению и записи по соответствующему смещению контекста.

Для модельных и конвейерных вызовов необходимы некоторые комментарии. Известно, что производительность большинства графических систем с классическими конвейерами растеризации в высокой степени коррелирует с числом и конфигурацией входящих примитивов для растеризации. Так, для задания параметров сцены визуализации в большинстве случаев необходимо не более одного соответствующего контекстного вызова OpenGL, однако для формирования реалистичного изображения необходимы, как правило, десятки и сотни тысяч примитивов, на вид которых влияют параметры контекста. Кроме того, контекст может изменяться в процессе визуализации, однако стандарт запрещает его изменение внутри пары вызовов glBegin/glEnd. В связи с этим предлагается формировать промежуточные блоки

для обработки в моменты вызовов glEnd, которые включают в себя:

- набор контекстной информации, соответствующей блоку данных, который включает матрицы преобразований координат, параметры освещения, отбраковки примитивов, размера области вывода и пр.;
- массивы координат вершин, текстурных координат, цветов и нормалей к вершинам для текущей модели.

Приведенные блоки данных содержат всю необходимую информацию для самостоятельной обработки любым из процессоров в системе с последующим формированием структур растеризации.

Собственно ограничение, которым мы воспользовались, и было введено в стандарт OpenGL для того, чтобы позволить каждой реализации OpenGL эффективно проводить буферизацию вершинных и параметрических данных.

Таким образом, предлагаемый подход к организации клиент-серверной архитектуры позволяет формировать из непрерывной последовательности вызовов OpenGL отдельные независимые блоки данных и осуществлять их конвейерную обработку. Вместе с тем, такое разбиение решает еще одну серьезную проблему при концентрации основных вычислений в рамках ресурсов DSP-ядер — относительно малый размер внутренней памяти данных, который обязывает проводить блочную обработку потока данных.

Тогда схематичная диаграмма взаимодействия внутренних компонентов OpenGL примет вид, представленный на рис. 2. Диспетчер конвейера обработки осуществляет постановку нового блока данных в очередь обработки процессора, который занимается формированием текущего кадра.

Конвейерные вызовы OpenGL выполняют следующие действия:

- приостановка добавления в очередь новых блоков данных;
- ожидание окончания обработки последнего переданного кадра сопоставленным ему процессором;
- операции чтения/записи с соответствующим буфером на плате;
- возобновление очереди конвейера.

Таблица 1

Предлагаемая классификация вызовов OpenGL

Параметр	Вызовы OpenGL			
	Интерфейсные	Контекстные	Модельные	Конвейерные
Критерий объединения в группу	Взаимодействие с программой пользователя без участия сервера OpenGL	Формирование контекста OpenGL: текстурирование, матрицы, освещение...	Задание вершин, нормалей, цветов и текстурных координат модели и их передача серверу OpenGL	Для выполнения необходим останов конвейера растеризации
Действие на конвейер растеризации	Нет (взаимодействует только с контекстом OpenGL без отложенных вызовов)	Добавление очередного буфера данных для обработки без останова конвейера		Осуществляется останов конвейера, затем выполняются соответствующие операции
Действие на контекст OpenGL	Чтение	Чтение/запись	Нет	
Примеры	glGetError, glIsTexture, glIsList, glIsEnabled, glGetFloatv, glGetLightfv...	glMatrixMode, glTexImage2D, glViewport, glLightfv, glClear, glEnable, glDisable, glFrontFace, glCullFace...	glBegin, glEnd, glVertex, glColor, glNormal, glTexCoord, glFlush	glDrawPixels, glReadPixels, glCopyPixels, glFinish

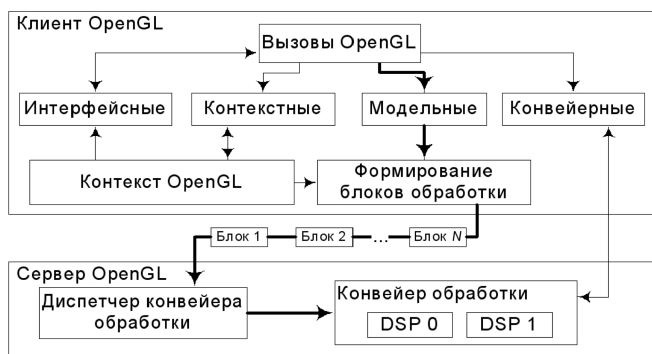


Рис. 2. Диаграмма взаимодействия внутренних компонентов OpenGL

В связи с тем, что для выполнения конвейерных вызовов требуется останов конвейера обработки, эффективность конвейерных систем снижается для таких вызовов. Впрочем, последнее утверждение справедливо для всех современных графических ускорителей.

Остановимся более подробно на устройстве сервера OpenGL.

Организация конвейера обработки данных в многопроцессорных графических системах

Предложенная классификация вызовов OpenGL позволила существенно сократить набор функций внешнего интерфейса, поддерживаемого непосредственно каждым микроконтроллером "Мультикор" в системе, а значит, упростить реализацию и избежать множественного исполнения общих операций для всех микроконтроллеров. Список этих функций и их описание приведены в табл. 2.

На транспортном уровне системы каждой приведенной функции соответствует одноименное сообщение. Со-

общение имеет заголовок и тело. Заголовок кодирует функцию сообщения и атрибут применения. В теле функции содержатся параметры обработки сообщения, соответствующие его типу. Например, тело сообщения ClearBuffer содержит индекс буфера для очистки (кадровый, Z-буфер и пр.) и цвет очистки.

При анализе различных вариантов организации многопроцессорной системы принимались во внимание несколько условий:

- обработка входных вызовов OpenGL согласно стандарту должна происходить в порядке очереди. Это означает, что никакой из вызовов OpenGL не должен исполниться раньше, чем любой предшествующий ему;
- необходимо осуществить распараллеливание исходной задачи между процессорами максимально эффективно, чтобы получить наилучший результат производительности;
- промежуточный обмен данными между процессорами нежелателен ввиду малых темпов передачи данных.

Вначале рассмотрим принципиальную маркированную сеть Петри [6], описывающую работу одного процессора по формированию конечного изображения (рис. 3). Имеем множество состояний системы $S = \{s_1, s_2, s_3, s_4\}$, где s_1 — состояние готовности системы к обработке входящего сообщения; s_2 — состояние работы функций Render, LoadTexture, LoadFragment или ClearBuffer; s_3 — работа функции Rasterize; s_4 — выгрузка готового кадрового буфера.

Также имеем множество переходов $T = \{t_1, t_2, t_3, t_4, t_5\}$, начальную маркировку $M = \{1, 0, 0, 0\}$ и отношение инцидентности

$$F = \{\langle s_1, t_1, s_2 \rangle, \langle s_1, t_2, s_3 \rangle, \langle s_2, t_5, s_1 \rangle, \langle s_3, t_3, s_4 \rangle, \langle s_4, t_4, s_1 \rangle\}.$$

Число срабатываний переходов t_1 и t_5 пропорционально числу примитивов в сцене OpenGL. Переход t_2 осуществляется при вызовах glFinish или glFlush. При этом запускается процесс растеризации обработанных данных после функции Render. По его окончании система переходит в состояние s_4 , при котором запускается DMA канал выгрузки сформированного кадрового буфера в контроллер видео, а система переходит в начальное состояние.

Система находится в состоянии s_1 до момента получения нового сообщения для обработки. Это может быть любое сообщение табл. 2.

Общая производительность в такой системе ограничена только производительностью отдельных стадий обработки кадра.

Рассмотрим теперь более сложный случай, когда в системе работают два процессора, имеющие один общий ресурс — выходной кадровый буфер. С помощью маркера-семафора право работы с ним переходит от одного процессора к другому. Принципиальная схема сети Петри для такой системы представлена на рис. 4. Подсети N_1 и N_2 соответствуют двум процессорам в системе. Начальная маркировка соответствует состояниям готовности обоих процессоров (s_{11}, s_{21}) и принадлежности маркера-семафора первому процессору (s_{25}).

Условные обозначения:

s_{15}, s_{25} — состояния доступности ресурса кадрового буфера после выгрузки соответственно первым и вторым процессором;

t_{13}, t_{23} — переходы получения права работы с кадровым буфером.

Таблица 2

Базовые операции, поддерживаемые каждым из процессоров

Функция	Атрибут	Описание функции
ClearBuffer	Локальный	Очистка соответствующего буфера заданным значением (используется один из каналов DMA). Может быть применена для кадрового буфера, буфера глубины и пр.
Render	Локальный	Обработка входного блока данных, которая включает вершинные преобразования, сборку примитивов, отбраковку нелицевых граней, отсечение невидимых элементов, расчет освещения методом Гуро по нормальям. В результате формируются структуры растеризации (фрагменты раstra) для следующей стадии (см. Rasterize)
Rasterize	Локальный	Растеризация фрагментов в конечный кадровый буфер в соответствии со сформированными фрагментными данными на стадии Render
LoadTexture	Глобальный	Загрузка новой текстуры в локальную память во внутреннем формате, совместимом с форматом кадрового буфера
LoadFragment	Глобальный	Загрузка в кадровый буфер входного растрового фрагмента данных во внутреннем формате (вызовы glDrawPixels, glBitMap)

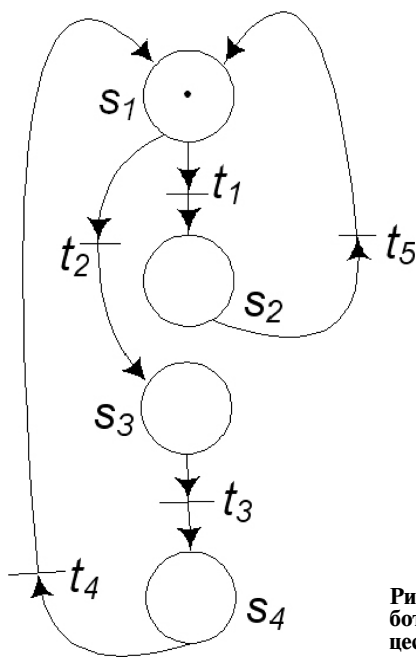


Рис. 3. Сеть Петри обработки кадра одним процессором

Отношения инцидентности для этих переходов могут быть формализованы следующими кортежами: $\langle s_{25} + s_{13}, t_{13}, s_{14} + s_{11} \rangle$, $\langle s_{15} + s_{23}, t_{23}, s_{24} + s_{21} \rangle$.

Они обеспечивают выполнение первого условия: гарантируется, что кадр с номером $n + 1$ не будет выдан раньше кадра с номером n . Вместе с тем, осуществляется параллельная конвейерная обработка соседних кадров в очереди при условии использования функции `glFlush` (она разрешает конвейеризацию вызовов внутри интерфейса OpenGL). В каждый момент времени модельные вызовы OpenGL передаются для обработки только одному из процессоров. Назовем его *активным*. Тогда действие функции `glFlush` в такой системе сводится к смене активного процессора. При вызове функции `glFlush`, однако, не прекращается обработка данных в процессоре, который был активным на момент вызова. Таким образом, запаздывание конвейера в такой системе будет составлять не более двух кадров. Легко показать также, что приведенная сеть Петри является 1-ограниченной, т. е. безопасной. Промежуточный обмен данными между процессорами отсутствует, так как они обрабатывают разные кадры.

Эффективность параллельной обработки кадров (по сути, оптимальность переходов t_{13}, t_{23}) подтверждена результатами профилирования двухпроцессорной системы на базе микросхем 1892BM3T (MC-12) при обработке тестовой сцены, состоящей из 1000 треугольников (см. рис. 9, в, см. вторую сторону обложки). Диаграмма задержек каждого из процессоров для первых 500 кадров сцены приведена на рис. 5. Она отражает актуальные задержки в выполненных RISC-ядром инструкциях при ожидании состояний s_{15}, s_{25} . Средняя задержка для обоих процессоров составила около 5 тыс. инструкций (включая инструкции чтения ячейки маркера-семафора). Визуально заметна также разница в задержках между первым и вторым процессорами на каждом кадре. Она объясняется топологией архитектуры самой системы: путь данных от первого процессора к выходному кадровому буферу (в терминах представленной сети Петри — время перехода t_{14}) несколько длиннее, чем от второго: на нем присутствуют арбитражные шины передачи данных.

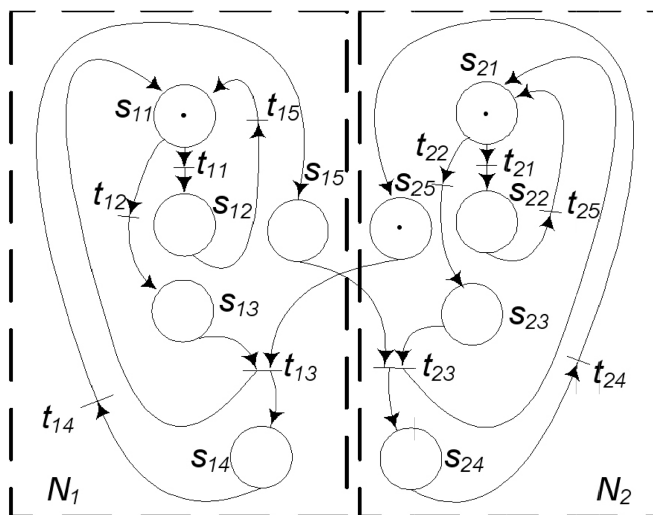


Рис. 4. Сеть Петри двухпроцессорной графической системы

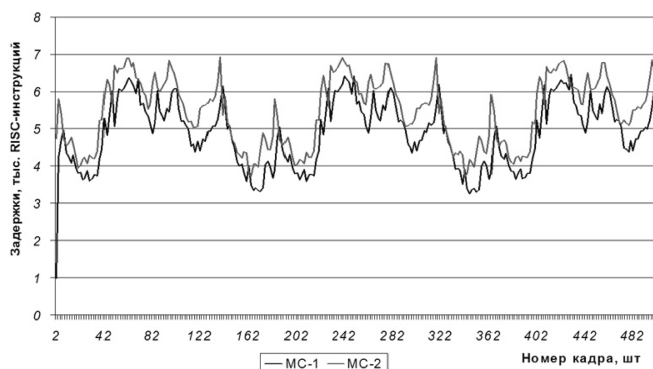


Рис. 5. Диаграммы задержек переходов t_{13} для двух микросхем 1892BM3T, работающих в общей графической системе. MC-1 и MC-2 — соответственно первый и второй процессоры системы

Исходя из того, что рабочая частота микросхем составляет 80 МГц, а работа системы профилировалась в режиме с включенной программной кэш-памятью RISC-ядра, можно заключить, что эффективность такого распараллеливания вычислительной задачи достаточно высока — порядка 99 %.

Также было проведено профилирование работы системы в режиме, при котором входящие кадры обрабатывал один процессор. На рис. 6 приведены результаты прироста производительности двухпроцессорной системы относительно однопроцессорной.

На рис. 6 видно, что производительность двухпроцессорной системы возрастает с увеличением вычислительной нагрузки на процессоры (а именно, с увеличением числа треугольников в сцене). И это вполне логично, так как при малом числе треугольников основная нагрузка лежит на транспорте кадрового раstra на выход системы.

Альтернативный подход: использование ресурсов RISC-ядра

В рамках создания графической библиотеки OpenGL для систем на кристалле серии "Мультикор" был также реализован и проанализирован другой подход к построению архитектуры многопроцессорной графической сис-

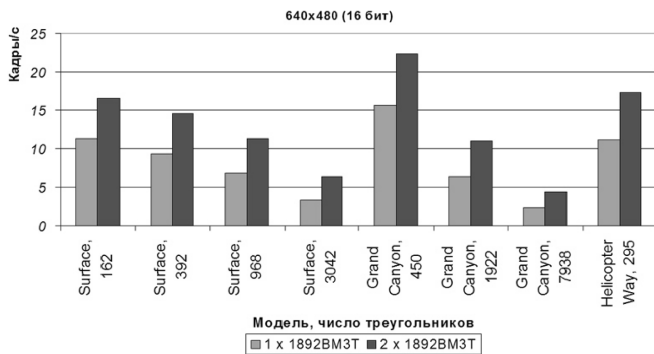


Рис. 6. Прирост производительности в двухпроцессорной системе

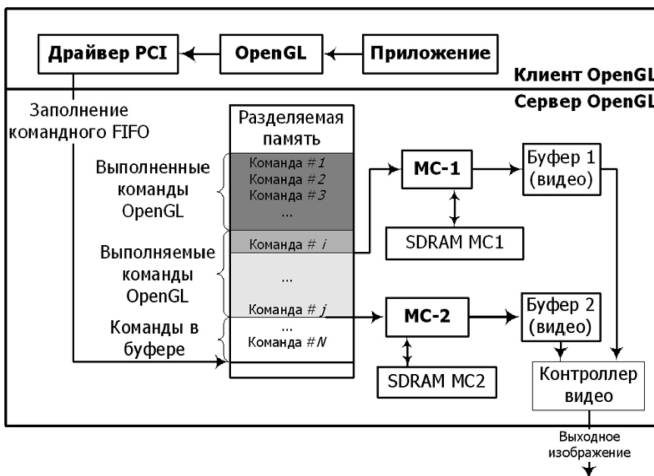


Рис. 7. Диаграмма работы системы с дублированием контекста OpenGL

темы. В основе его лежит принцип параллельной работы ядер RISC и DSP в системе на кристалле.

Основная идея подхода состоит в том, что вызовы OpenGL и формирование исходных буферов обработки могут быть реализованы в рамках возможностей RISC-ядра, а непосредственную обработку сформированных буферов проводит, как и в предыдущем случае, DSP.

Для этого была разработана транспортная система, включающая общий для двух процессоров командный буфер FIFO (*First In — First Out*), расположенный в разделяемой памяти. Все вызовы OpenGL были разделены на две основные группы:

- глобальные вызовы (подлежат исполнению на обоих процессорах);
- локальные вызовы (исполняются только активным процессором).

К глобальным вызовам в данном случае относятся интерфейсные, контекстные и конвейерные вызовы. К локальным функциям относятся функции задания модели (модельные, за исключением *glFlush*).

Алгоритм работы системы может быть описан следующим образом:

1. Клиент OpenGL кодирует текущий вызов, осуществляет запись в командный буфер FIFO и обновляет указатель буфера записи.

2. Каждый из процессоров выполняет чтение обновленного указателя буфера записи, а также чтение самого закодированного вызова OpenGL.

3. Если вызов является глобальным, то он исполняется обоими процессорами.

4. Если вызов является локальным, а процессор — активным, то он исполняется. В противном случае он игнорируется.

Диаграмма работы системы представлена на рис. 7 (MC-1 и MC-2 — соответственно первый и второй процессоры системы).

В описанной системе, по сути, имеет место дублирование контекста OpenGL на каждом из процессоров. Основные достоинства такого подхода:

- уменьшение вычислительной нагрузки на клиента OpenGL;
- возможность эффективной организации RISC-ядром SIMD-режимов обработки данных в DSP).

Из недостатков подхода следует отметить усложнение программного обеспечения, реализуемого непосредственно на каждом из процессоров, а также избыточность входных данных командного FIFO, которая составляет для разных вызовов от 20 до 50 %. Последняя обусловлена добавлением транспортной системой к каждому вызову OpenGL соответствующего заголовка.

Результаты и выводы

В статье рассмотрен вопрос об эффективной организации архитектуры и потоков данных в многопроцессорных графических системах, имеющих разделяемый интерфейс OpenGL. Обсуждены достоинства и недостатки двух подходов к реализации библиотеки OpenGL для программируемых цифровых процессоров обработки сигналов на примере систем на кристалле серии "Мультикор". Оба представленных подхода были реализованы в рамках создания двухпроцессорной графической системы с единым внешним интерфейсом OpenGL на базе микросхем 1892BM3T (рис. 8, см. вторую сторону обложки).

Графическая система разработана в ГУП НПЦ "ЭЛВИС" для поддержки современных бортовых вычислительных комплексов и предназначена для визуализации в интерактивном режиме трехмерных и двумерных сцен, описанных по стандарту OpenGL. Она имеет внешнее устройство сопряжения со стандартной шиной PCI версии 2.2 для входных команд OpenGL и выходной контроллер видео, соответствующий стандарту Fibre Channel.

Предложенная классификация вызовов OpenGL актуальна не только для систем, масштабируемых посредством увеличения числа микросхем на печатной плате, но также для эффективной организации обработки сцен OpenGL несколькими ядрами внутри одной микросхемы (в том числе, DSP-ядрами, работающими в режиме SIMD).

Некоторые результаты работы системы приведены на рис. 9 (см. вторую сторону обложки).

Список литературы

1. **Микросхема** интегральная 1892BM3T. Руководство пользователя. М.: ГУП НПЦ "ЭЛВИС", 2005.
2. **DSP-ядро Elcore-x4**. Система инструкций. М.: ГУП НПЦ "ЭЛВИС", 2006.
3. **Kessenich J., Baldwin D., Rost R.** The OpenGL® Shading Language, 3Dlabs Inc. Ltd., 2006.
4. **NVIDIA GeForce 8800 Architecture Technical Brier**. nVidia Corp, 2006.
5. **Segal M., Akeley K.** The OpenGL® Graphics System: A Specification (Version 1.4), Silicon Graphics Inc., 2002.
6. **Котов В. Е.** Сети Петри. М.: Наука, 1984.

А. А. Орлов, канд. техн. наук,

Е. Е. Канунова,

Муромский институт

Владимирского государственного университета

Цифровая обработка текста на изображениях рукописей как линейчатых объектов

Введены понятия и построены теоретические основы интегрирования растровых изображений по сегменту линии. Получен детектор полосы, который построен на основе второй производной гауссида. Разработан алгоритм выделения полосовых объектов. Проведены экспериментальные исследования на тестовых изображениях. Алгоритм апробирован на реальных сценах рукописей.

Введение

Изображение является основным источником информации для человека. Для большинства классов изображений наиболее ценными данными является препарат протяженных линейчатых объектов. Таковыми являются, например, контуры на изображениях. По границам (контурам) человек распознает и анализирует форму объектов на рассматриваемой сцене.

Сообразно с этим в области компьютерной обработки изображений существует огромное число работ, связанных с выделением и анализом границ [1—5]. Задача выделения контуров объектов решается как детектирование перепадов яркости в области изображения на основе градиентных фильтров. Градиент характеризует скорость изменения величины (в данном случае яркости) и тем самым является основным признаком границ образов.

Имеются также сцены, на которых непосредственно присутствуют изображения объектов в виде полос. Это изображения рукописей, треков, сварных швов, ребер на флюорограммах и многое другое. Для повышения качества обработки таких изображений возникает необходимость использования специфических геометрических признаков [6].

Целью работы является построение фильтра для детектирования полосовых образов на рукописях для их последующего анализа.

Разработка математической модели исходного изображения

Пусть $\varphi(x, y) = 0$ — уравнение кривой на плоскости R^2 , так что функция $\varphi: R^2 \rightarrow R$ возвращает значение, модуль которого численно равен расстоянию до этой кривой.

Определение. Множество точек, расположенных вдоль кривой $\varphi(x, y) = 0$ на расстоянии, не превышающем значение d (рис. 1),

$$S = \{(x, y) \mid |\varphi(x, y)| \leq d\} \quad (1)$$

назовем областью полосы (или просто полосой), заданной кривой φ (сердцевинной полосой).

Составим модель изображения, содержащего полосу (1). Пусть $\gamma: R^2 \rightarrow R$ — функция яркости некоторого изображения в непрерывном пространстве R^2 .

Определение. Изображение, заданное функцией $\gamma: R^2 \rightarrow R$, будем называть фоном.

Определение. Изображением полосы назовем сцену, заданную характеристической функцией $\chi: R^2 \rightarrow \{0, 1\}$:

$$\chi(x, y) = \begin{cases} 1, & (x, y) \in S \\ 0, & \text{иначе} \end{cases} = \begin{cases} 1, & |\varphi(x, y)| \leq d \\ 0, & \text{иначе} \end{cases}$$

Определение. Профилем полосы в точке (x, y) будем называть сечение поверхности $z = \chi(x, y)$ нормальной плоскостью в пространстве R^3 к кривой $\varphi(x, y) = 0$ в точке (x, y) .

Ясно, что профиль изображения полосы (1) будет иметь прямоугольную форму.

Обобщим понятие полосы, видоизменив ее профиль. Для этого будем полагать, что полоса состоит (точнее, интегрируется) из бесконечного множества ее профилей, расположенных вдоль кривой $\varphi(x, y) = 0$.

Пусть $A(x)$ — профиль полосы, так что

$$\lim_{x \rightarrow \pm\infty} A(x) = 0.$$

Тогда характеристическая функция изображения полосы профиля $A(x)$ определяется как

$$\chi(x_0, y_0) = \int_{\varphi(x, y) = 0} A(\sqrt{(x-x_0)^2 + (y-y_0)^2}) ds.$$

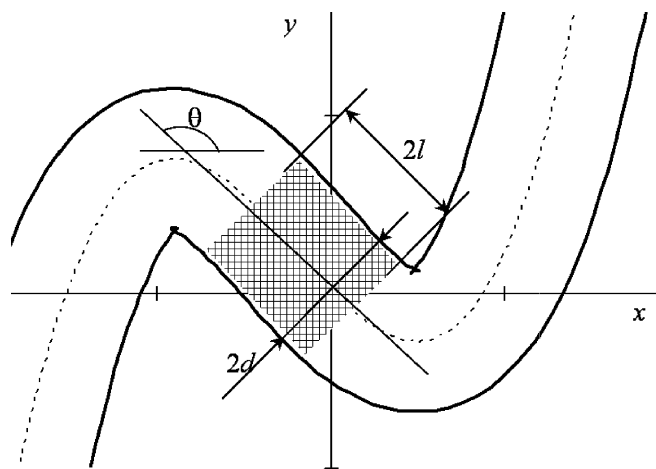


Рис. 1. Параметры полосы

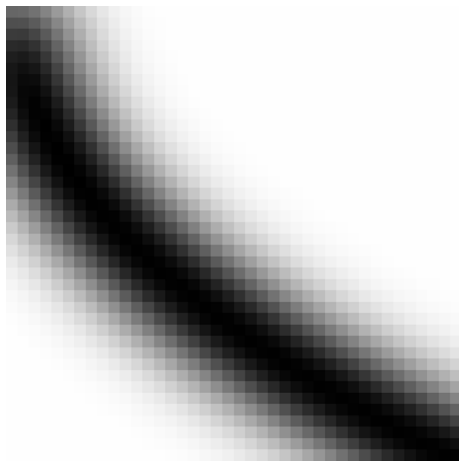


Рис. 2. Изображение полосы

Пример изображения полосы, где $A(x)$ — гауссиан, приведен на рис. 2.

В качестве модели исходного изображения возьмем алгебраическую сумму

$$f(x, y) = \gamma(x, y) + k\chi(x, y)$$

как наложение изображения полосы на фон, где k — коэффициент, характеризующий контраст полосы на фоне.

Вычисление признаков полосы

Решим задачу вычисления параметров полосы, заданной функцией $f: R^2 \rightarrow R$. Для этого разобьем полосу на сегменты.

Определение. Под сегментом полосы в точке (x, y) ($(x, y) \in \varphi$) будем понимать область полосы S , ограниченной прямыми, полученными параллельным переносом нормали к кривой φ в точке (x, y) на расстояние l (см. рис. 1).

Определение. Значение $2l$ назовем длиной сегмента.

Таким образом, каждый сегмент полосы характеризуется шириной $2d$, длиной $2l$, позицией (x, y) и ориентацией θ .

Аппроксимируем сегмент полосы второй производной гауссоида (рис. 3):

$$s(x, y) = \left(1 - \frac{x^2}{\sigma^2}\right) e^{-\frac{x^2 + y^2}{2\sigma^2}}, \quad (2)$$

где σ — параметр, соответствующий полуширине полосы.

Определение. Функцию $s(x, y)$ будем называть детектором сегмента полосы (или просто детектором полосы).

Введем соответствующую функцию комплексного переменного $s(r) = s(x + iy) = s(x, y)$, на основе которой выполним вращение поверхности $s(x, y)$ относительно оси аппликат на угол θ , умножив комплексный аргумент на $e^{i\theta}$:

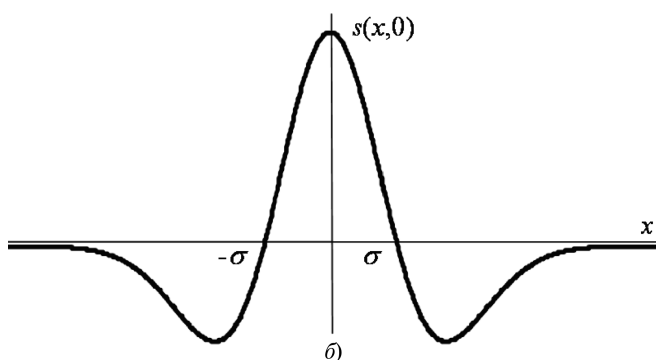
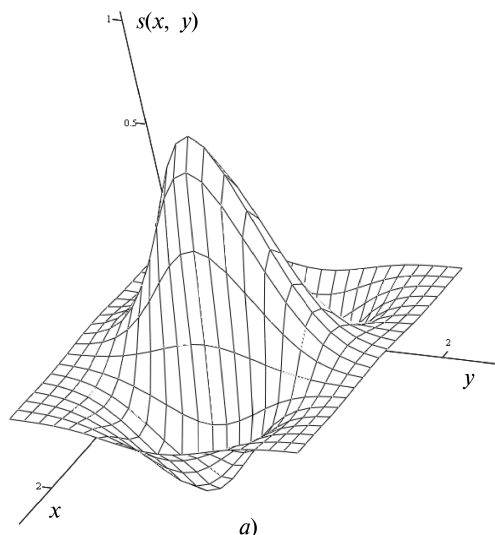


Рис. 3. Детектор полосы

$$s(x, y, \theta) = \text{Rot}_\theta s(x, y) = s(re^{i\theta}),$$

где Rot_θ — оператор вращения вокруг оси Oz на угол θ .

Теорема. Функция $s(x, y, \theta)$ разложима на три составляющих:

$$s(x, y, \theta) = s_1(x, y)\cos 2\theta + s_2(x, y)\sin 2\theta + s_3(x, y), \quad (3)$$

где $s_1(x, y)$, $s_2(x, y)$ — нормированные гауссоидом гиперболические параболоиды (рис. 4, а и б соответственно):

$$s_1(x, y) = -\frac{x^2 - y^2}{2\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}, \quad (4)$$

$$s_2(x, y) = \frac{xy}{\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}, \quad (5)$$

а $s_3(x, y)$ — лапласиан гауссоида (рис. 4, в):

$$s_3(x, y) = \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) e^{-\frac{x^2 + y^2}{2\sigma^2}}. \quad (6)$$

Доказательство. Обозначим гауссоид как

$$\Gamma(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}.$$

Так как поверхность гауссоида симметрична оси аппликат, ясно, что $\text{Rot}_\theta \Gamma(x, y) = \Gamma(x, y)$. Используя это, получаем:

$$\begin{aligned} s(x, y, \theta) &= s(\mathbf{r}e^{i\theta}) = s[(x + iy)e^{i\theta}] = \\ &= s(x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta) = \\ &= \left(1 - \frac{(x\cos\theta - y\sin\theta)^2}{\sigma^2}\right) \Gamma(x, y); \end{aligned}$$

$$\begin{aligned} (x\cos\theta - y\sin\theta)^2 &= x^2\cos^2\theta - 2xy\sin\theta\cos\theta + \\ &+ y^2\sin^2\theta = x^2\frac{1+\cos 2\theta}{2} - xy\sin 2\theta + y^2\frac{1-\cos 2\theta}{2} = \\ &= \frac{x^2}{2} + \frac{x^2}{2}\cos 2\theta - xy\sin 2\theta + \frac{y^2}{2} - \frac{y^2}{2}\cos 2\theta = \\ &= \frac{x^2 - y^2}{2}\cos 2\theta - xy\sin 2\theta + \frac{x^2 + y^2}{2}; \end{aligned}$$

$$\begin{aligned} s(x, y, \theta) &= \left[-\frac{x^2 - y^2}{2\sigma^2}\cos 2\theta + \frac{xy}{\sigma^2}\sin 2\theta + \right. \\ &\left. + \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) \right] \Gamma(x, y). \end{aligned}$$

Раскрыв скобки, мы приходим к равенству (3). Теорема доказана.

Сопоставим параметр σ с параметрами d и l (т. е. возьмем $d = l = \sigma$).

В качестве сравнительной оценки исходного изображения в точке (x_0, y_0) с детектором полосы (2) выполним скалярное произведение (свертку)

$$h(\theta) = \int_{-\infty-\infty}^{\infty} \int_{-\infty-\infty}^{\infty} f(x - x_0, y - y_0) s(\mathbf{r}e^{i\theta}) dx dy. \quad (7)$$

Определение. Формулу (7) назовем интегральным преобразованием по сегменту полосы.

Вследствие корреляционных свойств преобразования (7) спектральная функция $h(\theta)$ будет принимать максимальное значение в точке θ_0 тогда и только тогда, когда поверхность функции исходного изображения будет наилучшим образом повторять поверхность детектора.

Таким образом, для определения θ_0 достаточно решить следующую задачу оптимизации:

$$h(\theta) \rightarrow \max;$$

$$-\frac{\pi}{2} \leq \theta < \frac{\pi}{2}. \quad (8)$$

Для получения решения сформулируем следующие теоремы.

Теорема. Функция вида

$$u(\alpha) = a\cos\alpha + b\sin\alpha + c$$

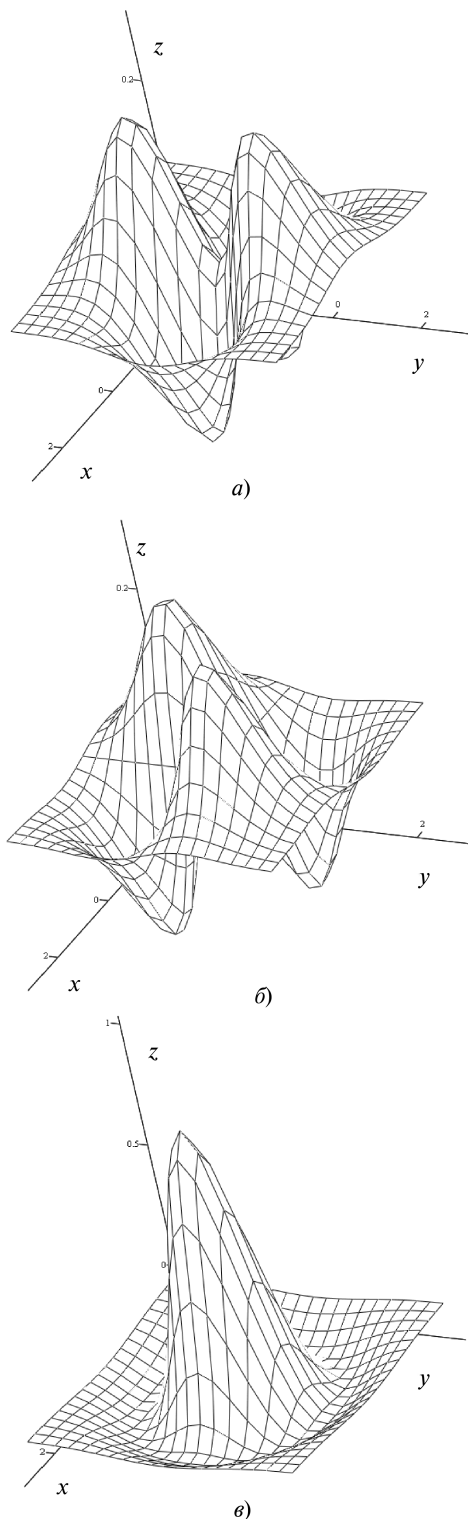


Рис. 4. Составляющие детектора полосы

принимает максимальное значение

$$\max_{\alpha} u(\alpha) = \sqrt{a^2 + b^2} + c$$

при аргументе, равном углу наклона вектора (a, b) ($\alpha = \angle(a, b)$).

Здесь угол наклона вычисляется по формуле

$$\angle(a, b) = \begin{cases} \arctg \frac{b}{a}, & a > 0; \\ \arctg \frac{b}{a} + \text{sign}(b)\pi, & a < 0; \\ \text{sign}(b)\frac{\pi}{2}, & a = 0. \end{cases}$$

Доказательство. Запишем функцию $u(\alpha)$ через скалярное произведение векторов:

$$u(\alpha) = a \cos \alpha + b \sin \alpha + c = \begin{pmatrix} a \\ b \end{pmatrix} \cdot \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} + c.$$

Так как скалярное произведение равно модулю одного вектора, умноженному на проекцию другого вектора на направление первого, то

$$\begin{pmatrix} a \\ b \end{pmatrix} \cdot \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} = \left| \begin{pmatrix} a \\ b \end{pmatrix} \right| \cdot \cos \varphi,$$

где φ — угол между векторами (a, b) и $(\cos \alpha, \sin \alpha)$.

Получаем

$$u(\alpha) = \left| \begin{pmatrix} a \\ b \end{pmatrix} \right| \cdot \cos \varphi + c.$$

Отсюда следует, что $u(\alpha)$ достигает наибольшего значения, когда $\cos \varphi = 1$, т. е. при $\varphi = 0$. Таким образом, наибольшее значение равно $|(a, b)| + c = \sqrt{a^2 + b^2} + c$ и обеспечивается при совпадении углов наклона векторов (a, b) и $(\cos \alpha, \sin \alpha)$. Теорема доказана.

Теорема. Спектральная функция $h(\theta)$, получаемая по формуле (7), принимает максимальное значение в точке

$$\theta_0 = \frac{1}{2} \angle(S_1, S_2),$$

так что

$$h(\theta_0) = \sqrt{S_1^2 + S_2^2} + S_3,$$

где S_1, S_2, S_3 — свертки исходного изображения $f(x, y)$ с функциями $s_1(x, y)$, $s_2(x, y)$ и $s_3(x, y)$ соответственно:

$$S_i = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x - x_0, y - y_0) s_i(x, y) dx dy, \quad i = 1, 2, 3.$$

Доказательство. Обозначим свертку функций как

$$f \otimes s = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x - x_0, y - y_0) s(re^{i\theta}) dx dy.$$

На основании (3) получаем

$$h(\theta) = f \otimes s = f \otimes [s_1(x, y) \cos 2\theta + s_2(x, y) \sin 2\theta + s_3(x, y)] = (f \otimes s_1) \cos 2\theta + (f \otimes s_2) \sin 2\theta + f \otimes s_3 = S_1 \cos 2\theta + S_2 \sin 2\theta + S_3.$$

По предыдущей теореме $\max_{\theta} h(\theta) = \sqrt{S_1^2 + S_2^2} + S_3$ в точке $2\theta_0 = \angle(S_1, S_2) \Leftrightarrow \theta_0 = \frac{1}{2} \angle(S_1, S_2)$.

Теорема доказана.

Решение задачи оптимизации (8) показывает, что алгоритм выделения полосовых объектов сводится к вычислению трех свертков функции исходного изображения $f: R^2 \rightarrow R$ с функциями (4), (5) и (6).

При реализации алгоритма в дискретном пространстве будем помещать значения функций $s_1(x, y)$, $s_2(x, y)$ и $s_3(x, y)$ в матрицы порядка $(3\sigma + 1) \times (3\sigma + 1)$. Размеры матрицы обусловлены правилом трех сигм.

Экспериментальное исследование

Проведем экспериментальные исследования. В качестве тестового изображения будем использовать сцену, содержащую: полосовые объекты различной толщины, образующие символы; горизонтальные параллельные отрезки, расположенные вблизи друг от друга на разных расстояниях; площадной объект произвольной формы (рис. 5, а).

На рис. 5, б—е представлены результаты обработки тестового изображения, выполненные на основе интегрального преобразования по сегменту полосы (7). Взято пять значений ширины сегмента $2\sigma = 1, 2, 3, 4, 5$ для рис. 5, б, в, г, д, е соответственно.

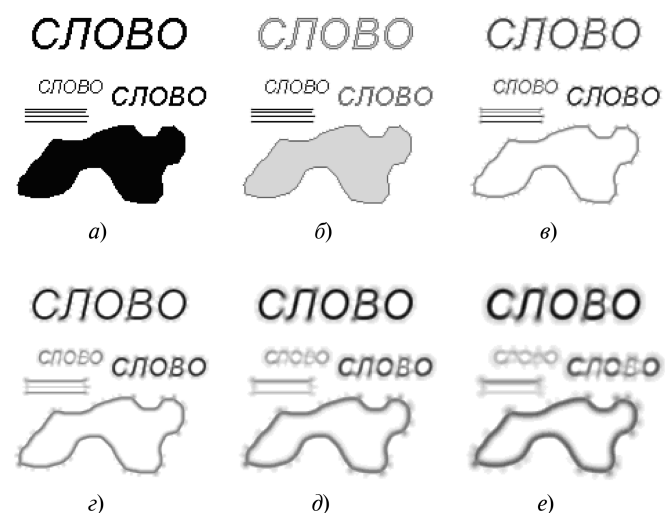
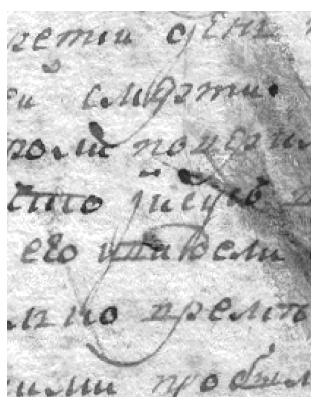
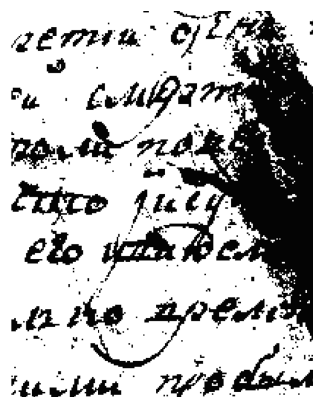


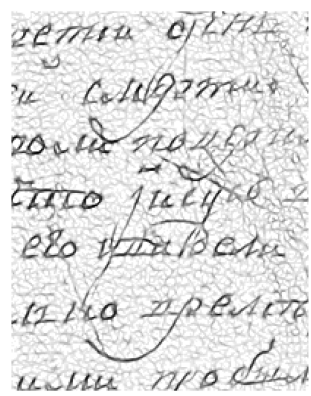
Рис. 5. Обработка тестового изображения



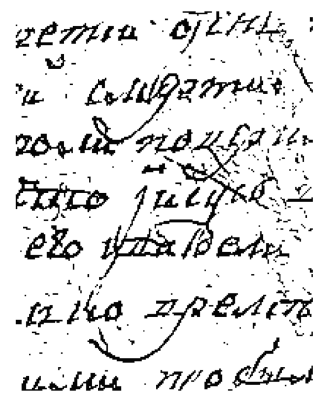
а)



б)

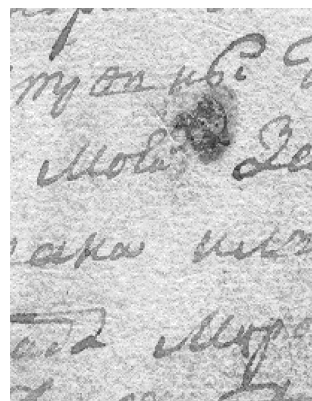


в)

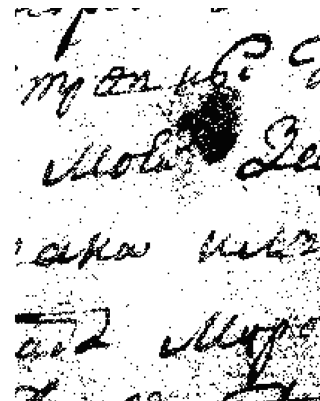


г)

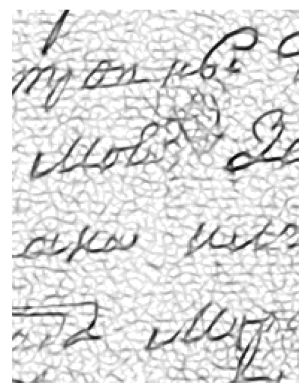
Рис. 6. Обработка рукописи



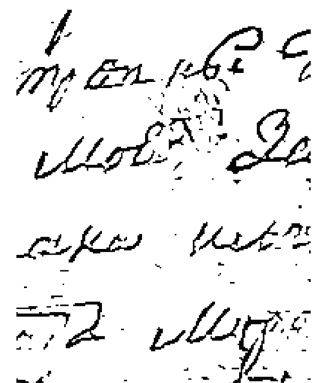
а)



б)



в)



г)

Рис. 7. Обработка рукописи

Анализ результатов показывает:

- при $2\sigma = 1$ выделяются полосы шириной в один пиксель, подчеркиваются (но в меньшей степени) края полос большей ширины и контур площадного объекта;
- при $2\sigma = 2$ выделяются полосы шириной в два пикселя;
- при $2\sigma = 3$ выделяются полосы шириной в три пикселя, происходит слияние отрезков, расположенных друг от друга на один пиксель;
- при $2\sigma > 3$ выполняется смазывание и слияние имеющихся на тестовом изображении объектов.

Таким образом, практические результаты говорят о том, что разработанный алгоритм можно применять для выделения полосовых объектов. При этом ширина выделяемых полос должна быть равна удвоенному параметру σ детектора (производной гауссоида).

Выполним оценку результатов алгоритмов на реальных изображениях старинных рукописей (рис. 6, а и 7, а). Видно, что исходные изображения являются некачественными (изображения мало контрастные, неравномерный фон, присутствие пятен). Необходимо отметить, что полезной информацией здесь является именно

рукописный текст, который построен из образов полосовых объектов. Поставим цель — их выделение.

Результат бинаризации исходных изображений, т. е. отделения текста от фона, приводит к неудовлетворительным результатам (рис. 6, б и 7, б) — из-за имеющихся на сцене дефектов часть информации становится потерянной.

Статистический анализ рукописей показывает, что ширина полосы составляет около двух пикселей. Возьмем $\sigma = 1$. Результаты применения на рукописях разработанного алгоритма представлены на рис. 6, в и 7, в. Алгоритм обеспечивает значительное подавление дефектных областей, устраняет разрывы в тексте рукописи, вызванные аддитивным шумом. Это можно наблюдать на изображениях (рис. 6, г и 7, г), полученных бинаризацией фильтрованных от дефектов сцен.

Таким образом, можно отметить, что практические эксперименты показывают хороший результат и подтверждают правильность теоретических исследований, заключающихся в использовании структурных признаков исходных изображений данного класса.

Результатами работы можно считать следующее. Введены понятия и построены теоретические основы интегрирования растровых изображений по сегменту полосы. Разработан алгоритм выделения полосовых объектов. Практические эксперименты показали хороший результат и подтвердили правильность теоретических исследований. Алгоритм апробирован на реальных сценах рукописей. Следует также отметить, что построенная теория интегрирования вдоль кривых может служить основой для создания более сложных и качественных алгоритмов, использующих большее число структурных признаков изображений.

1. Бакут П. А., Колмогоров Г. С. Сегментация изображений: Методы выделения границ областей // Зарубежная радиоэлектроника. 1987. № 10. С. 25—47.
2. Методы компьютерной обработки изображений / Под ред. В. А. Сойфера. М.: ФИЗМАТЛИТ, 2003. 784 с.
3. Прэтт У. Цифровая обработка изображений: Пер. с англ. М.: Мир, 1982. Кн. 2. 480 с.
4. Grigorescu C., Petkov N., Westenberg M. Contour and boundary detection improved by surround suppression of texture edges // Image and Vision Computing. 2004. 22. P. 609—622.
5. Meer P., Georgescu B. Edge detection with embedded confidence // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2001. 23. P. 1351—1365.
6. Лабунец В. Г., Чернина С. Д. Теория и применение преобразования Хоха // Зарубежная радиоэлектроника. 1987. № 10. С. 48—56.

УДК 004.021,004.421

С. В. Мельман,

Институт автоматизации и процессов управления
ДВО РАН, г. Владивосток
gruzd@dvo.ru

Визуализация объемов в задачах анализа физических полей синоптических объектов

Рассмотрены вопросы разработки системы визуализации объемов применительно к физическим полям синоптических объектов. Приведены требования к визуализации синоптических данных, программная структура системы и типы обрабатываемых данных. Описаны реализованные в системе методы визуализации скалярных и векторных 3D-полей, как статических, так и изменяющихся во времени, в контексте визуализации физических полей синоптических объектов. Приведены примеры работы системы на модельных и реальных данных.

Введение

Программно-алгоритмические средства визуализации объемов являются важным инструментом в задачах исследования динамики атмосферы и океана, связанных с обработкой больших объемов пространственных данных и компьютерным моделированием. Разработка таких средств ведется давно, несмотря на большой спектр реализуемых на сегодня возможностей [1—6], потребность в совершенствовании методов/алгоритмов визуализации сохраняет свою актуальность, поскольку растут объемы обрабатываемых данных и повышаются требования к скорости и эффективности визуализации.

Необходимость в разработке специализированных средств 3D-графики в настоящей работе определяется

следующей спецификой требований со стороны рассматриваемого выше спектра атмосферно-океанологических задач: многоплановый характер рассматриваемых физических полей скалярного и векторного типа с необходимостью комбинированной визуализации; многоплановая анимация — движение камеры в сцене, динамика скалярного поля, анимация векторного поля и их комбинации; необходимость в наращиваемом наборе разных методов/алгоритмов визуализации и интерактивности с учетом многоплановости физических полей атмосферы и океана и исследуемых синоптических объектов; повышение скорости графической обработки данных за счет применения аппаратных графических возможностей.

Описываемая в настоящей статье графическая система направлена на повышение наглядности визуализации больших объемов пространственных данных, получаемых с искусственных спутников Земли и метеорологических станций в целях изучения синоптических объектов (течения, вихри, циклоны и др.), построения математических моделей [7] и прогноза их динамики. В конечном счете, визуальный анализ и интерпретация получаемых данных расширяет возможности метеорологов при изучении погодных явлений и способствует повышению качества прогнозов погоды. Наряду с визуализацией данных в системе решаются и задачи уточнения, восполнения и регуляризации метеорологических и спутниковых данных.

1. Структура и функциональные возможности системы визуализации

Универсальная программная оболочка. С учетом основного требования к разрабатываемой системе визуализации, — возможности наращивания и модификации алгоритмической базы объемной визуализации, — была разработана структура, представленная на рис. 1.

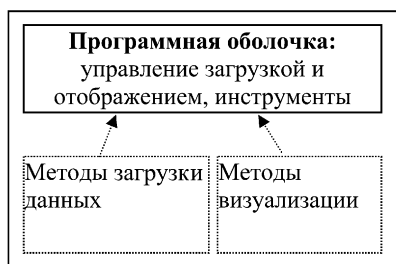


Рис. 1. Структура программной оболочки

Основу ее составляет универсальная программная оболочка, посредством функций которой пользователь управляет загрузкой данных (главным образом, данных спутниковых измерений), выбором методов визуализации и инструментов для проведения визуального анализа данных. В настоящей версии это методы визуализации статических и динамических скалярных и векторных полей.

Загрузка, интерпретация, интерполяция, аппроксимация или другие действия восполнения данных в конкретный момент времени скрыты от пользователя. Процесс визуализации динамических полей в конечном результате представляется пользователю в виде анимационного фильма со шкалой времени, перемоткой вперед/назад, паузой и т. д.

Подключаемые модули. Возможности свободного наращивания и модификации методов визуализации и методов работы с различными форматами данных реализованы поддержкой сменных модулей — плагинов (*plug-in*) (рис. 1). В настоящей версии программы реализована поддержка двух типов подключаемых модулей: модули загрузки данных и модули визуализации.

Для каждого вида модулей была разработана спецификация взаимодействия и потоков данных. Для модулей загрузки определяется место хранения данных и методы доступа к хранилищу из различных частей программного комплекса. Для модулей визуализации была определена процедура интерактивных настроек параметров вывода.

Основным требованием к системе визуализации синоптических данных является не только необходимость одновременной визуализации нескольких видов данных (температура, влажность, давление, ветер), но и одновременное применение нескольких методов визуализации для одного набора данных (объемный рендеринг + изоповерхности + плоские срезы). В условиях обработки большого количества данных был разработан метод разделения данных на "общие — частные". Необходимость подобного разделения возникает в связи с перекрестным использованием независимых модулей. Например: один и тот же модуль загрузки данных участвует в загрузке двух разных полей (загрузчик TOVS для температуры и влажности). Для каж-

дого из полей данных применяется по два модуля визуализации (объемный текстурный рендеринг + метод многочастичной анимации), следовательно, модуль загрузки работает одновременно с двумя полями данных и каждый из модулей визуализации работает с двумя полями данных. Таким образом, загрузчик и модули визуализации должны "помнить" о загруженных полях, чтобы правильно ими управлять и правильно их визуализировать. Решением задачи стало разделение данных на данные загрузки и данные пары "загрузка—визуализация". Каждый отдельный процесс загрузки данных создает область частных данных, и каждая пара также имеет свои частные данные. Частные данные — это небольшой набор параметров модуля. Общие данные — это скалярное или векторное поле со всеми своими атрибутами. Схема выбора загрузчиков и пар представлена на рис. 2.

Модули визуализации. При разработке схемы и спецификации интерфейса обмена параметрами между отдельными блоками учитывалось, что модуль визуализации может визуализировать данные любого загрузчика соответствующего типа. Модуль визуализации для своей работы не требует дополнительных сведений о настройках других модулей системы или каких-либо дополнительных данных, кроме данных, загруженных модулем загрузки, и параметра "скорость анимации", он общий для всех модулей и является связью скорости динамического поля и мультипликации. Каждый модуль должен предоставлять пользователю возможность настройки внутренних параметров визуализации в режиме диалога. Возможность использования одновременно нескольких модулей визуализации позволяет использовать несколько узкоспециализированных алгоритмов в комплексе, что может значительно повысить наглядность.

В текущей версии проекта реализованы три алгоритма визуализации, применяемые для скалярных данных, и ранее разработанный автором алгоритм визуализации векторных полей [15]. Предполагается развитие алгоритмической базы за счет разработки новых и включения известных алгоритмов.

Работа с данными. Условно разделим все виды обрабатываемых данных на три группы:

- скалярные поля (поля температур, влажности);
- векторные поля (движение воздушных масс);
- в последнюю группу включаются модули загрузки специфической информации для повышения наглядности при визуализации данных (топографические данные по планете, графические полигональные объекты и др.).

Выбранные загрузчики: 1. Точечный заряд 2. Данные TOVS	Выбранные пары загрузчик – визуализатор: (2-1) Данные TOVS – Метод объемной ТВ (2-1) Данные TOVS – Метод ВИ (1-3) Точечный заряд – Метод объемной ТВ (1-3) Точечный заряд – Метод ВИ
--	---

Рис. 2. Выбор модулей загрузки и пар для визуализации

Каждый модуль загрузки работает только с определенной группой данных, с определенным типом и только от определенного источника данных. Такая строгая направленность требует разработок множества модулей, но вместе с тем, они способны точно учитывать специфику обрабатываемых данных. Результатом работы модуля загрузки является не только загрузка, но и интерпретация данных и приведение их к формату, принятому в приложении. Таким образом, с этим набором данных способен работать любой модуль визуализации, относящийся к той же группе.

Загрузка и предварительная обработка данных. Для загрузки реальных полей температуры и влажности воздуха, создаваемых Японским Метеорологическим Агентством по данным радиометров HISR/AMSU спутников POES, NOAA, был разработан специальный модуль загрузки. Так как эти данные представляют собой продолжительные наблюдения в течение месяца, то была реализована динамическая загрузка данных, когда пользователь с помощью шкалы времени может выбрать серию для визуализации. Поскольку данные являются разреженными и нерегулярными (в пространстве и во времени), необходимо их восполнение. Например, поле температуры определяется нерегулярными значениями замеров температуры в точках исследуемого объема, задаваемых широтой, долготой и высотой. Для восполнения нерегулярности в пространстве использовалась триангуляция Делоне. Второй способ восполнения данных относится к замерам, промежутки между которыми достигали от 2 до 11 ч. Здесь применялась линейная интерполяция между двумя восстановленными полями скаляров.

Для рассматриваемого приложения характерны большие объемы данных, их разреженность и нерегулярность. Отсюда возникает необходимость в предварительной обработке по интерполяции, аппроксимации, регуляризации и восполнению данных. Этот этап обработки может выполняться как на ПЭВМ, так и на многопроцессорной вычислительной системе MBC-1000 — для повышения скорости обработки. Конечный выход — данные непосредственно для рендеринга. Поэтому реализована и схема с применением параллельных вычислений: предварительная обработка на MBC-1000 и визуализация с использованием аппаратного рендеринга на Windows-машине (рис. 3). Предусмотрены также возможности оптимизации и ускорения



Рис. 3. Схема использования MBC-1000

обработки данных с использованием многоядерных процессоров и графических ускорителей. К последним применима технология "шейдеров", использование которой позволило некоторые вычисления над полями скаляров и векторов (например, динамическое вычисление аномалий температуры и влажности) переложить на графический процессор. Применение технологии "шейдеров" дает также дополнительные возможности по совершенствованию алгоритмов объемной визуализации.

Функциональные возможности: визуальный анализ данных. Создаваемая система может использоваться для визуализации различных пространственных статических и динамических полей скалярного и векторного типа. Вместе с тем, она прежде всего ориентирована на применение в исследованиях синоптических объектов в задачах динамики атмосферы и океана. Основные прикладные функциональные направленности при этом следующие:

- визуальное обнаружение важнейших пространственно-временных характеристик атмосферы, в частности, влажности (интегральной и на различных уровнях), положение тропопаузы;
- визуализация изотемпературных поверхностей атмосферы;
- интерактивный анализ атмосферных океанических аномалий, определение их сущности, происхождения, положения и времени жизни.

Для обеспечения указанных проблемно-ориентированных возможностей в интерфейсе системы предусматриваются следующие графические функции:

- темпоральные и пространственные срезы;
 - гибкое управление интерфейсом;
 - интерактивное управление модулями загрузки и визуализации;
 - интерактивная обработка данных.
- К инструментарию относятся такие режимы как:
- управление камерой ("Arcball", свободный полет, плоский режим, автоматическое слежение за динамическими объектами);
 - управление данными (отсечение скаляров, фильтрация);
 - управление визуализацией;
 - управление динамическими полями.

2. Алгоритмы визуализации объемов

В настоящей версии системы реализованы три алгоритма анимационной визуализации объемов, применяемые для скалярных данных, и один алгоритм для визуализации векторных данных:

- 1) объемной текстурной визуализации (ТВ);
- 2) объемной многочастичной визуализации (МВ);
- 3) визуализации изоповерхностей (ВИ);
- 4) многочастичной анимации векторных полей (МЧА) [15].

Объемная текстурная визуализация. Метод объемной визуализации базируется на использовании библиотеки OpenGL и аппаратно поддерживаемых 3D-текстур. Для

подготовки этапа рендеринга в рассматриваемом объеме строится семейство секущих плоскостей, параллельных плоскости экрана. Поле скалярных данных по предопределенной схеме интерпретируется как 3D-текстура, и графический акселератор берет на себя дальнейшую интерполяцию данных между ячейками сетки. Закраска плоскостей происходит средствами 3D-акселератора с использованием подготовленной и загруженной в видеопамять 3D-текстуры. При визуализации данных можно оперировать параметрами "цвет" и "прозрачность", "освещенность". Вычислительные затраты метода сводятся к вычислению трехмерных текстурных координат, которые присваиваются вершинам примитивов, являющихся приближением секущих плоскостей. Для этого над текстурными координатами проводятся матричные преобразования:

$$\mathbf{M} = \mathbf{M}_T^{-1} \cdot \mathbf{M}_V^{-1} \cdot \mathbf{M}_T \cdot \mathbf{M}_S,$$

где $\mathbf{M}_V$ — текущая видовая матрица; $\mathbf{M}_T$ — матрица прямого сдвига центра координат; $\mathbf{M}_S$ — матрица масштабирования.

За счет использования прозрачности, реализованной в 3D-ускорителях, все, не относящиеся к исследуемому объему пиксели, остаются невидимыми, в результате на экране формируется реалистичное изображение. Ниже приведены примеры работы описанных методов визуализации на тестовых (см. рис. 5, 7, 8 на четвертой стороне обложки) и реальных (см. рис. 9) данных.

Многочастичная визуализация. Входными данными для работы алгоритма является задание скалярного поля в рассматриваемом объеме. Идея алгоритма заключается в отображении на экран частиц (в нашем случае это сферы малого диаметра) в соответствии со значением скаляра в рассматриваемых точках объема. Для реализации наглядности отображения скалярного поля варьируются следующие параметры: цвет частиц; освещенность; вероятность появления. Реализуется анимационная визуализация, при которой происходит постоянная смена положений частиц (мельтешение) — в результате, наблюдатель по характеру распределения частиц на последовательности кадров может судить о структуре скалярного поля. При реализации предложенного метода использовался закон распределения, рассчитывающий вероятность появления частицы в объеме:

$$A = F(s),$$

где A — вероятность появления частицы в объеме; F — функция зависимости; s — значение скаляра.

Функцию F задает пользователь. Как вариант предлагается $F = s^n$, где n задает пользователь. Так же как и функцию вероятности, пользователь выбирает функцию раскраски частиц в зависимости от значений скалярного поля. Например: $R = s$; $G = 1 - s$; $B = s$, где R , G , B — компоненты цвета; s — значения скаляра, приведенного к интервалу $[0, 1]$ в данной точке пространства (см. рис. 6, 8, 9 на четвертой стороне обложки).

Для достижения наилучшего визуального эффекта пользователь может выбирать число частиц и их размер.

Визуализация изоповерхностей. Для конвертирования воксельного представления изоповерхностей в полигональное был реализован метод "марширующих кубиков" [14], что обеспечило аппаратную поддержку визуализации изоповерхностей скалярных полей.

3. Практическое применение. Оценка эффективности

Для оценки эффективности текущей версии системы были проведены эксперименты по визуализации как модельных, так и реальных данных. Для получения модельных данных была построена эмпирическая модель циклона на основании известных общих закономерностей динамики отдельных параметров и анализа данных о циклонах района Японского моря. Использовались динамические модельные данные развития аномалий в циклоне: скалярные поля — влажность, температура, давление; векторные данные — ветер. Иллюстрации к работе системы приведены на рис. 4—9 (см. четвертую сторону обложки).

Рис. 4 демонстрирует пользовательский интерфейс системы, диалоги и окно визуализации. Визуализация и анимация происходят в отдельном окне. Конфигурация компьютера, на котором проводились тесты: AMD Athlon 3200+, 1 Гбайт memory, video — Ge Force 6600 512 Мбайт. В зависимости от ракурса, набора данных и методов визуализации скорость смены кадров варьировалась от 1 до 100 кадров в секунду.

На рис. 5 приведен пример статической визуализации (алгоритм объемного текстурного рендеринга) аномалий температуры в циклоне. Аномалией считается отклонение от нормального значения. Таким образом, можно наблюдать центр циклона и зону максимальных аномалий, так называемое "ядро".

На рис. 6—9 демонстрируются кадры анимационной визуализации.

На рис. 6 показаны аномалии влажности на данных модели циклона. Используется многочастичный алгоритм объемной визуализации. Наблюдается скопление влаги на высоте около 2000 м.

На рис. 7 визуализируются модельные данные циклона: аномалии давления и ветер в циклоне. Визуализация осуществляется алгоритмом объемного текстурного рендеринга (поле давления) в сочетании с алгоритмом многочастичной анимации векторного поля (ветер). На рис. 8 дополнительно показано скалярное поле влажности, которое визуализируется многочастичным алгоритмом.

На рис. 9 приведен пример визуализации реальных данных пространственного поля температуры. Визуализация обеспечивается одновременной работой трех алгоритмов: объемного текстурного рендеринга, многочастичного алгоритма и визуализации изоповерхностей. В качестве источника реальных данных использовались пространственные поля температуры/влажности возду-

ха, создаваемые Японским Метеорологическим Агентством по данным радиометров HISR/AMSU спутников POES NOAA. Были взяты данные, полученные в период с 1 по 31 августа 2000 г. в районе Японского моря:

- координаты (16,0—41,0) северной широты (126,3—164,3) восточной долготы;
- высота от 0 до 22000 м над уровнем моря;
- температура от $-77,8$ до $28,5$ °C;
- PRECIPITABLE WATER 0,02...4,89 г/(см²).

Визуальный анализ в данном случае позволяет обнаружить тропопаузу на высоте около 18000 м над уровнем моря.

Оценивая полученные на этапе тестирования результаты с точки зрения конечного пользователя, можно отметить, что программа позволяет визуально обнаруживать важнейшие пространственно-временные характеристики атмосферы, в частности, влажность (интегральную и на различных уровнях), положение тропопаузы. В том числе крайне полезной представляется возможность визуализации изотемпературных поверхностей атмосферы и возможность использования данного приложения для интерактивного анализа атмосферных/океанических аномалий, определения их сущности, происхождения, положения и времени жизни.

Заключение

В работе представлен действующий прототип системы визуализации пространственных данных статических и динамических скалярных и векторных полей. Система ориентирована на применение в задачах исследования динамики атмосферы и океана в качестве инструмента автоматизации научных исследований. Структура системы предусматривает развитие алгоритмической базы статической и анимационной визуализации, расширение типов используемых данных. В текущей версии реализованы: алгоритм многочастичной визуализации; алгоритм визуализации объема на базе применения 3D-текстур; визуализация изоповерхностей с предварительной конвертацией воксельного представления в полигональное методом "марширующих кубиков", алгоритм многочастичной анимации векторных полей. Тестирование системы на модельных и некоторых реальных данных показало ее практическую эффективность.

Вместе с тем, полученные результаты носят незавершенный характер и требуют дальнейшего развития, а именно: расширения алгоритмической базы визуализации объемов; совершенствования графического интерфейса и средств визуального анализа; поиска новых решений интегрированной визуализации скалярных и векторных полей; повышения скорости обработки данных с применением параллельных вычислений; расши-

рения средств конвертации данных для обработки различных типов данных; дальнейшей апробации на реальных данных тропических циклонов и других синоптических объектов.

* * *

Выражаю особую благодарность за поддержку и помощь д-ру техн. наук В. А. Бобкову, канд. техн. наук А. И. Алексанину и С. Е. Дьякову (ИАПУ ДВО РАН).

Работа выполнена при финансовой поддержке Президиума РАН (Программа фундаментальных исследований РАН № 14, раздел II) и ДВО РАН (грант раздела III).

Список литературы

1. **Научная** визуализация. Институт физико-технической информатики. 2005 г. // <http://www.icpt.su/?fl=341>
2. **Геофизический** центр Российской академии наук. 2005 г. // http://www.gcras.ru/hist_r.htm
3. **The Earth Simulator Center**. 2006 // <http://www.es.jamstec.go.jp/esc/eng/>
4. **Аниканов А. А., Потий О. А.** Проблемы и подходы к решению задачи визуализации данных о течениях в природных объектах // Научная визуализация в прикладных задачах. М.: Изд-во МГУ. 2003. С. 20—22.
5. **The Scientific Visualization Group**, Institute for System Programming of Russian Academy of Sciences. <http://www.ispras.ru/~3D/eng/>; gallery: <http://www.ispras.ru/~3D/eng/problems/gallery.htm>
6. **Центр "Фобос"**. 1998—2006, ФОБОС. Все права защищены. <http://fobos.gismeteo.ru>.
7. **Щебенькова А. А.** Стримлеты (streamlets): возможный путь преодоления криволинейного эффекта Гиббса // Труды Международной конференции по вычислительной механике и современным прикладным программным системам. г. Алушта, 2005. С. 203—204.
8. **Шевцов М. Ю., Долговесов Б. С.** Система объемной визуализации реального времени на базе стандартного графического акселератора // GraphiCon'2005. Р. 401—405.
9. **Belotserkovskii O. M., Chechetkin V. M., Oparin A. M.** Visualization of hydrodynamic calculations // Труды межд. конф. по компьютерной графике и машинному зрению "Графикон-2003". Москва, 2003.
10. **Montani C., Scateni R., Scopigno R.** A modified look-up table for implicit disambiguation of marching cubes // Visual Computer. 1994. № 10(6). Р. 353—355.
11. **Ягель Р.** Рендеринг объемов в реальном времени // Открытые системы. #05. 1996.
12. **Drebin R. A., Carpenter L., Hanrahan P.** Volume Rendering // Computer Graphics. August 1988. 22(4). Р. 65—74.
13. **Шевцов М. Ю.** Объемная визуализация научных данных на базе VolumePro 1000 // Труды Конференции-конкурса "Технологии Microsoft в информатике и программировании". 2004. С. 68—70.
14. **Montani C., Scateni R., Scopigno R.** A modified look-up table for implicit disambiguation of marching cubes // Visual Computer. 1994. 10(6). Р. 353—355.
15. **Мельман С. В., Гриняк Т. М.** Анимация трехмерных стационарных векторных полей // Электронный журнал "Исследовано в России". 2004. 202. С. 2149—2155. <http://zhurnal.ape.relarn.ru/articles/2004/202.pdf>



ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

УДК 006.922+004.855

А. В. Манцивода, д-р физ.-мат. наук, проф.,
Иркутский государственный университет (ИГУ),

А. А. Малых, канд. физ.-мат. наук,
Центр новых информационных технологий ИГУ

Достижения в Интернете и будущее информационной среды российского образования

Благодаря реализации блистательных инновационных идей, которые существенно продвинули вперед возможности глобального информационного пространства, Интернет постепенно переходит в новую стадию своего развития. В статье делается обзор этих идей и анализируется степень их полезности для информационной среды российского образования. На примерах показывается, что эти идеи способны придать нашему образовательному информационному пространству новое качество и сократить дисбаланс между достаточно богатой коммуникационной основой и относительно бедным содержательным наполнением.

О трудностях информатизации

Начнем с простого эксперимента. Сравним объем информации, опубликованной на сайтах двух вузов из разряда ведущих — российского и американского. Эти учебные заведения по своему научному и образовательному потенциалу находятся в одной весовой категории (вузы не называем, поскольку сравниваем ситуации). С помощью поисковой системы Google подсчитаем число документов тех видов, в которых обычно публикуются учебные пособия и материалы, научные статьи и презентации (см. таблицу).

Итак, объем содержательных материалов, доступных с точки зрения Google на сайте российского вуза, составляет лишь 1,7 % от соответствующего объема у вуза американского (измерения проводились в начале июня 2006 г.). Можно предположить, что к американскому и российскому вузу у Google разное отношение. И все равно — такая разница вызывает большую обеспокоенность. Российская система образования активно интегрируется в глобальное образовательное пространство, а информатизация является существенным компонентом конкурентоспособности.

Вернемся к американским вузам. Возьмем, к примеру, полусеместровый курс "Методология программиро-

вания" весеннего семестра 2006 г. [1] в Стэнфордском университете. Представительство курса на сайте *stanford.edu* содержит в сумме 182 PDF-файла, включающие конспекты лекций, контрольные, практики, ответы и решения. Кроме того, веб-страницы содержат 21 презентацию лекций, а также черновик книги "Искусство и наука Джава" профессора Э. Робертса, ведущего этот курс. На сайте имеются и примеры программных разработок. Материалы курса полностью открыты и доступны всем желающим.

Конечно, не везде курсы получают столь насыщенную информационную поддержку. Однако и в других солидных американских и европейских учебных заведениях образовательный процесс, в том числе и очный, имеет весьма развитый информационный компонент. Наши же дела иногда обстоят по-другому. Предложим несколько объяснений этому.

У нас учебный процесс реально не строится на информационных технологиях. Это всего лишь возможность, которую можно использовать, но без которой можно и обойтись. А в этом случае срабатывает принцип минимальной достаточности. Поскольку российский преподаватель не поставлен в такие условия, которые вынуждают его активно пользоваться информационными технологиями, он, конечно, будет экономить на необязательных вещах. Тем более, что еще не изжитая такая болезнь, как "компьютеробоязнь". В американских университетах все строится через Интернет, а информационная поддержка вплетена в учебный и научный процесс. Можно сказать, что информатизация образования у американцев проходит на микроуровне — через каждого преподавателя и студента. Каждый преподаватель является генератором виртуального контента.

Мы в целом более консервативны, чем наши американские коллеги, и с большим подозрением относимся к активному использованию информационных технологий в учебном процессе.

Нам часто свойственна переоценка значимости собственных материалов и наработок. Мы предпочитаем не выставлять их в открытом доступе, в то время как для американских коллег открытая публикация является правилом (публичность обеспечивает ответственное отношение к делу).

Тип ресурса	Российский вуз	Американский вуз	Соотношение, %
Формат PDF	37 000	3 010 000	1,23
Формат PowerPoint	582	89 500	0,65
Формат DOC	19 300	98 200	19,6
Формат PostScript	807	123 000	0,66
Итого	~58 000	~3 320 000	1,7

Наконец, у российских и американских преподавателей и студентов разные уровни оснащенности рабочих мест. У многих наших студентов (особенно в провинции) даже нет компьютеров. Это затрудняет внедрение информационных технологий в ядро учебного процесса. Вместе с тем, процесс обучения пока не мотивирует студента вкладывать заработанные деньги в свое будущее, покупая, например, компьютер и доступ в Интернет, вместо траты денег на развлечения.

Возникает четыре взаимосвязанных вопроса:

- Как повысить мотивацию российского преподавателя (а значит, и студента) к использованию информационных технологий?
- Как подвигнуть преподавателя к созданию собственного контента в информационной образовательной среде?
- Как интегрировать информационные технологии в учебный процесс?
- Как насытить информационную среду российского образования качественным и богатым контентом?

Очевидно, что основные усилия должны быть сфокусированы на микроуровне — уровне отдельного субъекта учебного процесса. Образно говоря, нужно информатизировать не вузы и факультеты, а преподавателя. Задача нелегкая, поскольку имеет серьезный психологический компонент — проблему мотивирования. Простой разработкой электронных образовательных ресурсов эту задачу не решить.

С этой точки зрения представляется весьма полезным проанализировать опыт современных веб-компаний, которым с помощью набора инновационных идей удалось дотянуться до массового пользователя, вовлечь массового пользователя в бизнес-процессы как активного игрока, выстроить соответствующую мотивационную конструкцию.

Веб¹ нового поколения

Начнем с анализа достижений компании Google, поисковый сервис которой уже использовался нами выше. После многих лет доминирования Microsoft компании Google первой удалось на равных сражаться, и не только сражаться, но и агрессивно атаковать Microsoft. Рассмотрим основные источники ее успехов [2].

Веб вместо персонального компьютера. Одной из главных причин доминирования Microsoft всегда был контроль над платформой (операционной системой) персонального компьютера — Windows. Приложения работают на компьютере настолько успешно, насколько позволяет им платформа. Это всегда ставило конкурентов Microsoft в неравные условия. Во-первых, Microsoft очень неохотно раскрывала информацию о Windows, необходимую для создания эффективных приложений. Компания также активно пользовалась другим конкурентным преимуществом — возможностью включать свои приложения непосредственно в стандартный пакет

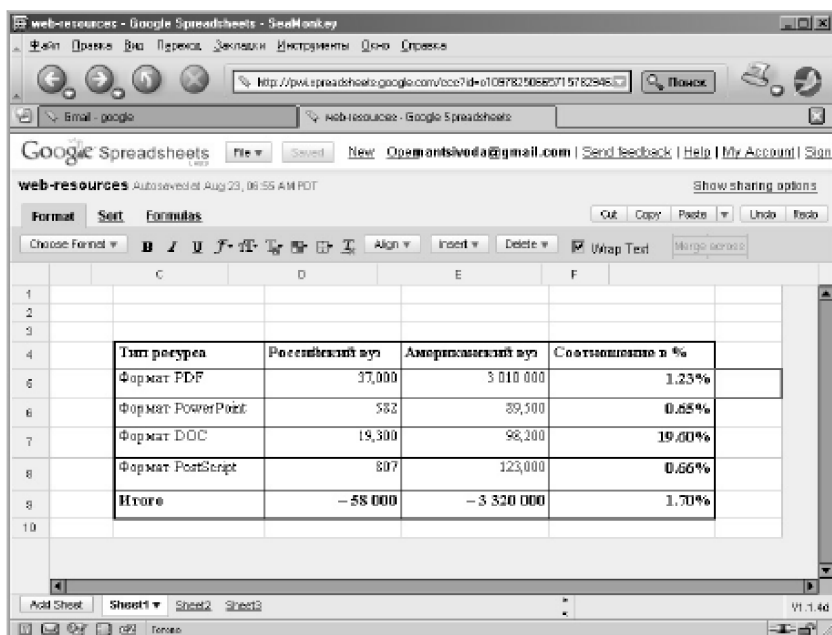
Windows. Кроме того, через Windows всегда можно было диктовать политику стандартов. Поэтому получалось, что приложения конкурентов боролись не только с приложениями Microsoft, но и с ее платформой. А приложение победить платформу не может по определению.

Google исповедует совершенно иные принципы работы. В отличие от локальной Windows, для Google платформой служит весь веб. Другими словами, веб интерпретируется как "глобальный компьютер", а наш небольшой ПК на столе является лишь окном в него. А для того, чтобы быть окном, нужен лишь навигатор (браузер). Это означает, что разработки Google уже не зависят от конкретной операционной системы, установленной на персональном компьютере, поскольку функционируют вне ее. Таким образом, ситуация кардинально меняется: борьба Microsoft и Google уже не является конкуренцией между платформой и приложением. Это конкуренция между двумя платформами. При этом глобальная веб-платформа имеет несравнимо более высокий потенциал, чем локальная в силу своей грандиозности и спектра возможностей. Конечно, имеются и значительные проблемы, связанные с доступом к Интернету и со сложностями работы через навигатор. Но эти проблемы последовательно и неуклонно решаются.

Сервисы вместо программ. Людям, имеющим компьютер, хорошо известно, с чего начинается работа с новой программой — с установки программного обеспечения. Сначала вставляем в компьютер компакт-диск с дистрибутивом (или скачиваем дистрибутив из Интернета), устанавливаем приложение на компьютер и только затем запускаем приложение и работаем с ним. Такая последовательность свойственна локальной платформе. С веб-платформой все по-другому. Ее основными игроками являются не приложения, а *сервисы*. Сервисы уже функционируют где-то там в глубинах Интернета, и все, что необходимо сделать, это подключиться к ним — с помощью хорошо известной процедуры регистрации. Зарегистрировавшись, пользователь может работать с сервисом так, как будто это приложение, установленное на его компьютере.

Развитые интерфейсы. Эффективная работа со сложной системой невозможна без богатых средств взаимодействия системы с пользователем. В частности, качественный веб-сервис невозможен без хорошего интерфейса, позволяющего работать с ним так же плодотворно, как с локальным приложением. Для веб-сервисов это непростая задача, поскольку их интерфейсы должны работать через навигатор. А возможности навигаторов ограничены. Поэтому до последнего времени между локальными интерфейсами и веб-интерфейсами существовала большая разница. Дело сдвинулось после запуска компанией Google почтовой службы GMail, за которой быстро последовал Google Maps. Оба этих веб-сервиса имеют богатый пользовательский интерфейс и такой же уровень взаимодействия с пользователем, как у локальных приложений. В своей статье [3] Дж. Дж. Гарретт для обозначения подхода Google к созданию интерфейсов использовал термин Ajax. Он писал: "Ajax — не технология. На самом деле, это несколько процветающих технологий, объединенных новыми мощными способами". Получаемый от их правильного соединения синергетический эффект позволяет достигать нужных результатов.

¹ Веб (всемирная паутина, www) — глобальное информационное пространство, основанное на физической инфраструктуре Интернета и протоколе передачи данных HTTP. Наиболее распространенный формат обмена данными в вебе — HTML-страница.



Интерфейс таблиц в Google Docs

Термин Ajax мгновенно стал очень популярным, что является косвенным свидетельством того, что "вопрос назрел". На рисунке приведен внешний вид сервиса Google Spreadsheets, построенный на Ajax. Google Spreadsheets — это близкий аналог всем известного MS Excel, но реализованный как веб-сервис. Благодаря Ajax-интерфейсу с ним работать так же легко, как и с Excel. Но как веб-сервис, Google Spreadsheets пользуется всеми преимуществами сетевой работы. На нем, например, можно реализовать совместные проекты с доступом группы пользователей или разработчиков к общим таблицам. И устанавливать на компьютере ничего не надо — достаточно набрать в навигаторе <http://docs.google.com> и настроить собственную среду для работы с документами и таблицами в стиле сетевого офиса.

Конечно, в наиболее сложных случаях Ajax не может конкурировать с локальными приложениями (пока трудно, например, представить веб-интерфейс для графического редактора Adobe Photoshop). У Google также есть системы, которые надо скачивать из Интернета и устанавливать на персональный компьютер. Но эти системы, как правило, служат лишь дополнением к веб-сервисам.

Использование коллективного разума. В Интернете дрейфуют огромные и неупорядоченные объемы данных. Еще больше данных и знаний суммарно находятся в головах у пользователей Интернета. И тот, кому удастся воспользоваться этим грандиозным источником, имеет большие шансы на успех.

Системы, с которыми мы работаем на персональном компьютере, обычно реализуют интеллектуальный потенциал своих разработчиков, их идеи и технологии. Такие приложения основаны на асимметричной схеме, когда разработчик только дает, а пользователь только получает. Современные веб-сервисы основаны на совершенно ином подходе — на принципе сотрудничества. Именно этот принцип стал основой первоначального успеха поисковой системы Google. В то время, как другие

поисковые системы применяли мудреные схемы, анализирующие текст веб-страниц, Google воспользовался коллективным опытом. Он просто стал подсчитывать, сколько на данную веб-страницу имеется ссылок из других веб-страниц, и анализировать, в каком контексте это делается. Другими словами, Google воспользовался той оценкой веб-ресурсов, которую дали им пользователи Интернета, и это сразу позволило ему вырваться вперед в конкурентной борьбе.

Сегодня имеется множество веб-сервисов, основанных на использовании коллективного разума. Пример — "народная" энциклопедия Wikipedia [4], основанная на неочевидной, но очень успешной идее о том, что статьи энциклопедии могут добавляться любым веб-пользователем, а корректироваться — любым другим. Задача же самой Wikipedia как веб-ресурса — помочь тем, кто что-то знает, превратить свои знания в статью энциклопедии, а тем, кто хочет что-то знать, — помочь найти эту информацию. Другими словами,

Wikipedia действует как посредник, который накапливает информацию в процессе взаимодействия людей.

Можно упомянуть, например, Cloudmark — систему, фильтрующую спам. Вместо того, чтобы непосредственно анализировать спам-тексты, Cloudmark анализирует и обобщает персональные мнения пользователей электронной почты о том, что есть и что не есть спам, и действует на основании этого анализа. Cloudmark работает эффективнее многих систем, базирующихся на анализе самих посланий.

Подход, связанный с привлечением коллективного разума, оказался чрезвычайно плодотворным. Главная задача — найти такой способ взаимодействия с пользователями, чтобы они в процессе своей работы приносили сервису добавленную стоимость, поделившись с ним своим опытом, информацией, знаниями. При этом очевидно, что лишь небольшое число пользователей пойдет на то, чтобы добавлять стоимость приложению через явные средства. Поэтому основная трудность состоит в том, чтобы добавление стоимости стало побочным эффектом обычной работы с сервисом. Наиболее эффективное решение здесь — организовать сервис как площадку взаимодействия людей, обменивающихся своими знаниями и ресурсами.

Информационное обеспечение (infoware). Рассмотрим пример. Компания MapQuest была пионером в категории картографических веб-сервисов в 1995 г. Но когда Yahoo!, затем Microsoft, и, наконец, Google решили выйти на этот рынок, они легко предложили конкурентоспособные сервисы путем простого лицензирования тех же самых данных.

Противоположная ситуация сложилась у Интернет-магазина Amazon.com. Как и у конкурентов, первоначальная база данных у Amazon была взята у R. R. Bowker — регистрационного провайдера международного стандарта нумерации книг ISBN. Но в отличие от MapQuest, Amazon неустанно уточнял информацию, добавлял пре-

доставляемые издателями данные, такие как картинки обложек, оглавление, предметный указатель и примеры материалов. Более того, он привлекал своих потребителей к партнерству, давая им возможность аннотировать данные и делиться опытом (т. е. пользовался "коллективным разумом"). Кончилось тем, что после десяти лет работы именно Amazon, а не Bowker, является первичным источником для библиографических данных о книгах, используемых студентами и библиотекарями, а как Интернет-магазин — имеет на порядок больше клиентов, чем конкуренты. Если бы MarQuest сделала то же самое, например, привлекая своих пользователей к аннотированию карт и направлений, это привело бы к добавлению стоимости ресурса. В этом случае конкурентам было бы намного труднее выйти на данный рынок.

Этот пример показывает, что основная стоимость многих современных веб-сервисов составляет не программное, а *информационное* обеспечение. Для поисковых систем — это базы индексирования, другие сервисы собирают адреса пользователей, данные о календаре событий, отзывы о товарах и т. д. В такой ситуации программное обеспечение играет хотя и важную, но вспомогательную роль. Поэтому его код часто делают открытым даже коммерческие компании. А вот информационное обеспечение охраняют весьма тщательно.

Длинный хвост. Перечисленные выше качества веб-компаний нового поколения позволяют им реализовать новую *бизнес-модель* работы. Классическая бизнес-модель выстраивается на следующих принципах:

- веб предназначен для публикации, а не для партнерства;
- рекламодатели, а не потребители, должны контролировать ситуацию;
- в Интернете доминируют топовые веб-сайты.

Новая бизнес-модель исходит из того, что маленькие сайты (совокупность огромного количества малых ресурсов часто называют "длинным хвостом" веба) составляют основную массу содержимого Интернета, а узкие маркетинговые ниши с небольшой емкостью рынка определяют основную массу возможных приложений. Поэтому новые веб-компании формируют сервисы, обеспечивающие самоорганизацию потребителей — для того, чтобы заполучить весь веб, вплоть до самых окраин, а не только центр (длинный хвост, а не только голову).

Рассмотрим DoubleClick — Интернет-компанию, работающую с 90-х годов прошлого века. Она специализируется на разработке систем для Интернет-рекламы и ориентирована на крупных клиентов. Ее бизнес-модель основана на явном заключении и ведении договоров, продаже и инсталляции программного обеспечения. В результате за все годы DoubleClick сделала около 2000 успешных инсталляций своей системы. Новые же сервисы, такие как AdSense, имеют уже сотни тысяч, если не миллионы, рекламодателей. Причина — эти сервисы строятся как автоматизированные посредники, соединяющие рекламодателей и тех, кто хочет разместить у себя на сайте рекламу, чтобы заработать деньги. В результате рекламный сервис стал доступен даже маленьким компаниям, которые и составляют основную массу веба. По старым схемам (подписание договоров, приобретение и установка ПО) такой сервис небольшим компаниям слишком накладен, а значит, недоступен.

Некоторые права защищены (Some rights reserved). Защита интеллектуальной собственности ограничивает повторное использование ресурсов, а также делает невозможным активное вовлечение массового пользователя в процесс работы и экспериментирование. Поэтому когда преимущества исходят из коллективной работы, а не частных ограничений, очень выгодно снять лишние правовые барьеры. Рекомендуется следовать существующим Открытым стандартам и применять лицензии с наименьшим числом ограничений, при этом разрабатывать такой код, который легок для хакеров и доступен для повторного использования.

Список нововведений веб-компаний нового поколения можно продолжить. Их суммарный эффект таков, что его можно охарактеризовать как революцию в информатике. Например, исчезает (точнее, совершенно меняется) фундаментальное понятие "жизненного цикла программ". В классическом варианте жизненный цикл коммерческой системы состоит из стадии альфа-версии (совсем сырой), бета-версии (сырой, но с полноценным функционалом), отладки с помощью ограниченного круга пользователей, доводки до коммерческой версии, превращения в "коробку" (единицу продажи), продвижения на рынке, сбора информации о практике использования системы, и, наконец, работы над следующей версией, которая выпускается в лучшем случае через полгода-год. По новой же схеме рекомендуется выпускать в широкое использование еще сырые сервисы ("принцип вечной беты"), которые дорабатываются в активном взаимодействии с пользователями. Сервис не имеет никакого физического воплощения (коробки). Продвижение на рынок осуществляется по схеме вирусного маркетинга — рекомендации распространяются непосредственно от одного пользователя к другому (например, Google никогда не рекламировал свои продукты). Что касается выпуска новых версий, то, в частности, главный разработчик сервиса Flickr.com Кол Хендерсон признавался, что они разворачивают новые средства каждые полчаса. Именно по этой причине в недавней статье в Zdnet [5] Фил Вейнрайт заключил, что Microsoft не сможет победить Google: "Бизнес-модель Microsoft зависит от того, чтобы каждый делал апгрейды на своем компьютере раз в два или три года. Google зависит от тех, кто проверяет, что новенького появляется в их компьютерной среде раз в день". Мы же со своей стороны считаем, что ни одно из направлений еще не утерало своего потенциала, и в масштабном столкновении идей и парадигм нас ждут новые открытия.

Веб-сервисы для системы образования

Всемирное образовательное пространство все больше и больше становится *сетевым пространством*. Сетевым в самом широком смысле — системой взаимодействующих людей, организаций и ресурсов, распределенных по всему миру. Основным инструментом для формирования сети служат информационные технологии. Их развитие позволило строить гибкие и масштабируемые сетевые среды. Каждая сетевая среда представляет собой гармоничное сочетание *пассивных* компонентов (источников информации) и *активных* компонентов — процес-

сов, которые обеспечивают работу и взаимодействие участников, действующих в сети.

Сетевая образовательная среда позволяет модернизировать учебные методики, расширяет и выравнивает уровни доступа людей к образовательным услугам, способствует появлению инновационных образовательных схем. Сети *соединяют* людей, например, в виртуальном общении преподавателя и студента. Сети *объединяют* людей — для реализации совместных проектов и создания профессиональных сообществ. Сети *накапливают информацию* — в образовательных ресурсах, виртуальных библиотеках, базах данных и знаний. Сети делают информацию *доступной*, например, через поисковые системы. Сети способствуют *реализации потенциала* своих участников, давая им возможность публиковать свое мнение, иметь доступ к информации и реализовывать профессиональные амбиции.

Построение информационной сети — лучший способ объединения разрозненных и удаленных друг от друга объектов. Например, Иркутский госуниверситет обладает системой научных станций, разбросанных на огромной сибирской территории вокруг озера Байкал. В эту систему заложен значительный сетевой потенциал. Действуя как единая сеть, научные станции могут проводить сложные исследовательские, мониторинговые работы и, конечно, вести подготовку студентов. В частности, здесь можно реализовывать сетевые образовательные проекты, распределив группу студентов по научным станциям для решения общих задач. Такой подход будет чрезвычайно полезен для формирования у студентов как специальных, так и общих компетенций (например, навыков командной работы). Уникальность природы региона делает такой подход очень заманчивым.

Встает вопрос: как организовать взаимодействие удаленных друг от друга команд студентов? Ответ очевиден — через информационную среду с предоставлением научным станциям доступа в мировое информационное пространство. Для успешной работы команд нужна как пассивная поддержка (обеспечение доступа к информационным ресурсам), так и активная (функциональная) поддержка. Например, частью активной поддержки может стать *система управления объектами*. Она позволяет планировать деятельность по реализации проекта, назначать работы участникам группы, отслеживая при этом выполнение работ, обеспечивать обмен сообщений между сотрудниками. Кроме того, она предоставляет возможность формирования списков дел, которые необходимо выполнить (так называемых ToDo-листов), обеспечивает накопление данных и совместную разработку документов.

Систему управления проектами лучше всего реализовать в виде *веб-сервиса*. Эффективность такого подхода подтверждена мировой практикой (например, сервис *basecamp* компании 37signals поддерживает сегодня более миллиона проектов). Веб-сервисы хороши и для поддержки других сетевых процессов. В целом, они играют ключевую роль в реализации активных сетевых компонентов.

Всем знакомы стандартные веб-сервисы, такие как электронная почта, поисковые системы, сервисы доступа к информации (библиотеки и репозитории), коммуникативные сервисы (чаты, форумы и блоги). Они реа-

лизуют в сети наиболее универсальные функции, необходимые, в частности, и для системы образования. Но для того, чтобы эффективнее решать образовательные задачи, мы должны опираться также и на *специализированные* сервисы. Кроме упомянутого выше проект-менеджмента, можно указать много других полезных специализированных систем. Рассмотрим, например, еще один сервис, взяв для разнообразия совсем иную задачу — разработку и распространение библиотечной системы для школ (ведение каталога, читателей и т. д.). В принципе, такие системы имеются на рынке, но в виде обычных программ, работающих на отдельном компьютере (или, в лучшем случае, в локальной сети без использования преимуществ глобальной информационной среды). Сравним обычный и сетевой подходы. Сначала проанализируем, что получается при реализации локальной библиотечной системы, не использующей веб. Можно предположить следующий порядок действий:

- Некоторая компания разрабатывает программную систему поддержки школьных библиотек и представляет ее на рынок в виде "коробки".
- Школа или иная организация, которой приглянулась эта система, выплачивает деньги фирме, после чего "коробка" оказывается в школе.
- Системный администратор школы (или лицо, его замещающее) устанавливает эту систему на компьютере библиотекаря или в школьной сети. Теперь задача системного администратора — следить, чтобы данная система функционировала бесперебойно и у библиотекаря не было проблем.
- Библиотекарь после изучения документации проводит заполнение базы данных библиотечной системы. Начинается рутинная работа.

В результате этой деятельности на *микроуровне* (уровне школы и ее сотрудников) получаем следующее:

- для инсталляции и штатной поддержки системы необходим компетентный системный администратор;
- для установки библиотечной системы школе нужен сервер или отдельный школьный компьютер с высокой степенью защищенности;
- уровень "выживаемости" системы не во всех учебных заведениях высок: компьютер могут украсть, сломать, данные случайно стереть или испортить;
- библиотекарь остается один на один с системой. Регулярные механизмы поддержки его работы отсутствуют;
- в целом, эффективная работа системы и ее "выживаемость" зависит от уровня квалификации как администратора, так и библиотекаря.

На *макроуровне* (уровне системы образования в целом) имеем:

- данные о школьных библиотеках "рассыпаны" по стране — по маленьким локальным базам данных, которые недоступны для повторного использования, внешнего доступа и обработки;
- система ориентирована на индивидуальную работу, ее применение не ведет к созданию сообщества пользователей и обмену опытом;
- в разных школах могут использоваться системы с несовместимыми форматами представления данных. Такие данные очень трудно объединить, а при отказе школы от системы зачастую гибнет и наработанная база;

- задача обновления (апгрейда) является тяжелой — систему необходимо индивидуально обновлять в каждой школе;
- система не связана с использованием Интернета. Это недостаток, поскольку не привлекаются возможности глобального информационного пространства и не выстраивается соответствующая мотивация. Наполнение изолированной базы данных не ведет к содержательному наполнению образовательной информационной среды в целом;
- для фирмы-разработчика такая схема весьма приятна — тысячам школ можно продать тысячи коробок.

Теперь обратимся к возможностям, предоставляемым современными сервисными технологиями. В этом случае события могли бы развиваться следующим образом:

- Компания (или рабочая группа) разрабатывает библиотечный сетевой сервис, способный одновременно работать в удаленном режиме со множеством школ.
- Данный сервис публикуется в Интернете на одном или нескольких серверах.
- Школы подключаются к сервису (регистрируются). Для этого не нужно никаких "коробок" и установок локальных или сетевых программ — библиотекарь работает через навигатор. Но нужен стабильный канал в Интернете. Сам сервис должен обладать свойством масштабируемости, чтобы эффективность его работы не зависела от числа пользователей.
- Библиотекарь формирует с помощью сервиса базу данных. Для него эта база выглядит как отдельная и локальная. На самом деле все работают с единой большой базой данных, но имеют доступ только к своей информации.
- Выход на стабильное использование системы. На основе обратной связи сервис постоянно обновляется и улучшается. Формируется сообщество библиотекарей-пользователей сервиса. Повторное использование и обмен данными.

Подведем некоторые итоги.

1. Сервисный вариант более стабилен, поскольку его работа организуется специально подготовленными профессионалами, строго реализующими технологии, например, резервного копирования. При проблемах с компьютерами в школе ее библиотечные данные не страдают.

2. Снижаются требования к необходимому уровню квалификации, поскольку не нужны усилия на поддержку системы (надо только установить навигатор). Правильно построенный Ajax-интерфейс, сценарии работы, а также участие в виртуальном сообществе библиотекарей облегчают новым пользователям входение в работу.

3. При появлении новых версий сервиса нет нужды в обновлениях на школьных компьютерах, поскольку вся работа проводится на серверах. Сервис постоянно развивается за счет оптимизации программного обеспечения и развития базы данных — с активным участием сообщества пользователей.

4. Общая база данных является ценным источником информации для мониторинга, статистических оценок, отслеживания текущей ситуации в школах. База данных является значимым контентом для информационно-образовательной среды в целом.

5. Через единый сервис легче обеспечивать целостность информационной среды и следование стандартам.

Очень важно, что благодаря сервису организуется неформальное сообщество пользователей, которое постоянно повышает квалификацию своих отдельных участников за счет обмена информацией, знаниями, опытом. Поддержка такого сообщества должна быть существенной частью самого сервиса. Хорошо известен феномен: когда набирается критическая масса пользователей, сообщество начинает оказывать благотворное влияние как на каждого своего участника, так и на сам сервис, формируя общее понимание того, в каком направлении сервис должен развиваться.

Проанализируем, каким образом в библиотечном проекте срабатывают веб-принципы, которые разбирались нами выше.

Веб вместо персонального компьютера и сервисы вместо пакетов программ. Эти принципы реализуются в полной мере и имеют важное значение, поскольку снижают требования к уровню квалификации работников школы, требования к технике (достаточно, чтобы работал навигатор). Повышается стабильность, поскольку информация поддерживается централизованно профессиональными службами сервиса.

Развитые интерфейсы. С точки зрения интерфейсов, библиотечный сервис является нетривиальным, требует применения Ajax-подхода и тщательно проработанных сценариев работы пользователей.

Использование коллективного разума и информационное обеспечение. Здесь при правильной организации работы преимущества могут быть очень большими. Коллективно созданная база данных библиотек самоценна и несет значительную информацию интегрального характера. Каждый библиотекарь каждым своим действием добавляет стоимость всему сервису, а значит, и всей информационной среде образования, поскольку введенные им данные расширяют контентную массу и могут использоваться другими участниками. Сложенные вместе, эти данные являются важным источником информации и для управленческих служб. Они делают ситуацию с библиотечным обслуживанием в школах прозрачной и понятной. Кроме того, в рамках виртуального библиотечного сообщества, собравшегося вокруг сервиса, будет порождаться дополнительный контент. Это могут быть аннотации книг, обмен опытом, форумы по библиотечному делу в целом, информация издательств, выпускающих школьную литературу.

Длинный хвост. Сервис направлен в первую очередь на массового пользователя, который не обязан владеть мощными ресурсами и высокой квалификацией. Сервис должен обеспечить постепенное "втягивание" пользователей в свою орбиту (правильно построенные веб-сервисы в инструкциях почти не нуждаются). База данных сервиса формируется как сумма большого числа маленьких баз данных школьных библиотек. Бизнес-модель сервиса может быть реализована через участие издательств и других фирм, заинтересованных в продвижении своих продуктов. В рамках сервиса можно также организовать книжный Интернет-магазин, а также службу заказов.

Некоторые права защищены. Вопрос авторских прав должен быть явно оговорен в лицензии по использованию сервиса. В частности, для эффективной работы уместно было бы, чтобы при формировании собственных баз данных можно было использовать виртуальные биб-

лиографические карточки, аннотации и другую информацию, созданную другими пользователями.

Библиотечный и подобные ему веб-сервисы обеспечивают регулярное наполнение среды контентом и обладают рядом важных качеств:

- контент создается не в режиме "активные разработчики — пассивные потребители". Каждый пользователь является как разработчиком, так и потребителем, происходит накопление коллективного опыта и знаний;
- процесс создания контента сопровождается самоорганизацией образовательного сообщества на базе профессиональных интересов;
- форма хранения этого контента делает его общедоступным и доступным (в отличие от россыпи локальных баз данных с неизбежными повторами, ошибками и недоступностью). Срабатывает принцип "достаточное число глаз делает ошибки мелкими" — совместными усилиями контент корректируется и уточняется;
- централизованная работа резко снижает затраты на организацию данных в соответствии с общепринятыми международными стандартами, такими как IMS [8];
- большая часть контента будет представлять собой активные сервисы, которые служат не только для информационной, но и для функциональной поддержки.

Все эти качества способны сыграть ключевую роль в развитии образовательной системы РФ как сетевой среды.

Знания в образовательной сети

Если оценивать текущее состояние информационной среды российского образования, то можно увидеть в ней дисбаланс двух видов. Во-первых, это серьезное несоответствие между вполне развитой коммуникационной инфраструктурой (которая расширится по мере реализации президентской программы подключения школ к Интернету [6]) и недостаточным содержательным наполнением (хотя здесь есть ряд прекрасных проектов, см., например, [7]). Ко второму виду относится дисбаланс между пассивными и активными компонентами информационной среды. Если процесс накопления ресурсов движется вперед, то образовательные сервисы развиты очень слабо. Актуальной является задача создания активного пространства, обеспечивающего полноценную и разностороннюю поддержку преподавателю и студенту в реализации их *функций*.

Развитая полнофункциональная сетевая среда является многообещающей основой для больших и маленьких образовательных инноваций и дает возможность вывести информационную поддержку на качественно новый уровень. Поэтому данная публикация посвящена, в первую очередь, активным сетевым инструментам. Тем не менее и *пассивный* компонент достоин значительной модернизации на базе достижений последних лет. В первую очередь, это связано с сетевыми технологиями работы со *знаниями*, которые обеспечивают не только проведение элементарных операций хранения, передачи и поиска информации (как в случае *данных*), но и определенный уровень интеллектуальной обработки. Задача

"интеллектуализации" очень непростая, требует поиска компромиссов между выразительностью и эффективностью, особенно в сетевых случаях. Потребовался не один десяток лет, прежде чем сформировалось общее понимание, позволяющее использовать эти технологии в массовом порядке. В 2004 г. дело дошло даже до принятия открытого стандарта формата представления знаний, основанного на понятии онтологии [12].

Использование сетевых технологий представления и обработки знаний в научных и образовательных сетях — одно из наиболее интересных и многообещающих приложений. Знания — основа образовательного процесса. Сетевая поддержка образовательной и исследовательской работы должна сочетаться с методами управления знаниями, системами метаописаний и классификаций. Например, за многие годы исследований озера Байкал накопились огромные массивы разрозненных публикаций и данных. Новейшие технологии обработки знаний позволяют обеспечить регулярный процесс перевода этой бесценной информации в сетевую среду, провести уточненную классификационную работу, сформировать по данной тематике агрегированные базы знаний и продвинутые информационно-справочные системы. Такая системная среда является исключительно интересной основой как для инновационных образовательных проектов, так и для научной деятельности.

Подведем некоторые итоги. Весьма распространенным является мнение о том, что главная цель сегодня — накопить в сетевой среде российского образования как можно больше научных и образовательных ресурсов. И это, конечно, правильно. Но двигаться к этой цели следует, учитывая впечатляющие достижения последних лет, в частности, в области представления знаний и создания сетевых сервисов. Стоит задача аккумуляции огромного интеллектуального потенциала российского образовательного сообщества в сетевом пространстве, развитие сетевого партнерства как внутри учебных заведений, так и между ними. Причем работу с пассивным контентом следует дополнять активной разработкой системы веб-сервисов, реализующих процессы в виртуальном научном и образовательном пространстве и поддерживающих работу исследователя, преподавателя и студента. Сочетание этих двух векторов развития окажет благотворное влияние на ту роль, которую играет информационная среда в системе российского образования.

В ЦНИТ ИГУ в течение ряда лет ведется активное развитие методов и технологий сетевого представления знаний и их обработки через систему веб-сервисов [9, 10]. В настоящее время нами разрабатываются подходы к применению этих результатов в создании образовательных веб-сервисов [11] на основе концептуальных положений, обзор которых дан выше.

Список литературы

1. **Course CS106A** — Programming Methodology. <http://stanford.edu/class/cs106a/>, имеется частичный русский перевод курса весны 2006 <http://teacode.com/temp/cs106a-ru/>.
2. **Tim O'Reilly**. What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software // <http://www.oreilynet.com/pub/a/oreilly/tim/news/2006/09/30/what-is-web-20.html>, September 2005.

3. **Garrett J. J.** Ajax: a New Approach to Web Applications // Adaptive Path Publication, February 2005, <http://www.adaptive-path.com/publications/essays/archives/000385.php>

4. **Wikipedia**, the Free Encyclopedia that Anyone Can Edit. <http://en.wikipedia.org>, русскоязычная версия <http://ru.wikipedia.org>.

5. **Wainwright P.** Why Microsoft Can't Best Google // ZDNet Blogs. Software as Services. August 2005, <http://blogs.zdnet.com/SAAS/?p=13>.

6. **Программа** подключения школ к Интернету в рамках приоритетного национального проекта "Образование". <http://www.rost.ru/projects/education/ed2/ed25/aed25.html>.

7. **Федеральный** портал "Российское образование". <http://edu.ru>. Единое окно доступа к образовательным ресурсам <http://window.edu.ru>.

8. **IMS** Global Learning Consortium. <http://imglobal.org/>.

9. **Манцивода А. В., Малых А. А.** Представление знаний в Интернете семантическое программирование // Системная информатика. Новосибирск: Наука, 2006. Вып. 10. С. 174—202.

10. **Mantsivoda A., Malykh A. and Lipovchenko V.** Logic Programming in Knowledge Domains // In: Proceedings of 22nd International Conference on Logic Programming, Seattle, USA, August 2006, Lecture Notes in Computer Science. Springer, 2006. Vol. 4079.

11. **Разработка** распределенной системы создания, метаописания, поиска и использования образовательных ресурсов на основе открытых стандартов. — Проект в рамках программы "Развитие научного потенциала высшей школы (2006—2008 годы)", подраздел 3.2.3, Рук. А. В. Манцивода.

12. **OWL** Web Ontology Language. Overview. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>.

УДК 621.385.63

Л. А. Серков, канд. физ.-мат. наук, доц.,
Академия управления и предпринимательства,
г. Екатеринбург

Синергетическое моделирование образовательных процессов

Описывается синергетический подход к образовательным процессам на примере модели экономического цикла, описывающей колебания численности студентов, усваивающих образовательные программы, образовательных ресурсов и вакансий для выпускников вузов и допускающей режимы квазипериодического и детерминированного хаотического поведения траекторий. Подробно проанализированы условия возникновения этих режимов. Исследована возможность самоорганизации в осциллирующих системах.

Введение

Построение и анализ синергетических моделей управления образовательными системами становится в последнее время составной частью наукоемких технологий. Это происходит вследствие того, что существующие системы управления образованием недостаточно эффективны. Вместе с тем, все образовательные объекты являются открытыми системами и обладают способностями к самоорганизации и адаптации, поэтому представляется перспективным подход к управлению вузами через распознавание, анализ, прогнозирование и управление процессами самоорганизации. Этот подход является главной составной частью системного управления и предполагает рассмотрение системы управления организацией как целостной совокупности элементов. Любое образовательное учреждение — это открытая сис-

тема взаимодействующих и управляемых частей (подразделений, людей и т. д.), работающая с определенной стратегией, миссией и имеющая в своем распоряжении определенные ограниченные ресурсы. Роль математического моделирования в описании и прогнозе развития высшей школы не вызывает сомнений. В этой сфере модели нужны для выявления "узких мест", возникающих в ходе развития системы. В случае построения адекватных моделей могут быть проанализированы альтернативные образовательные стратегии и дан прогноз развития системы при различных управляющих воздействиях. Кроме того, в ходе верификации моделей становится ясно, какой информацией об исследуемой системе мы не располагаем. Это может стать важным стимулом для проведения дальнейших социологических, психологических, экономических исследований системы высшего образования России. Наконец, модели, в отличие от общих подходов, допускают более конкретную и аргументированную критику.

Теоретические исследования в области системной динамики проводятся в разных направлениях. Одно из них — моделирование динамических закономерностей на различных иерархических уровнях. Все динамические модели, как аналитические, так и допускающие лишь вычислительный эксперимент, можно условно разделить на две группы. К первой относятся модели экономического роста, ко второй — модели экономического цикла или, в более широком смысле, экономических колебаний [1].

Колебательная динамика в большей мере соответствует реальным процессам по сравнению с поведением траекторий моделей роста. Математическое описание циклов позволяет решать ряд важных задач. В их числе — моделирование и, как

следствие, изучение поведения параметров циклов под воздействием меняющихся значений факторов. Реальные колебательные процессы с большой условностью можно считать циклическими, имея в виду лишь последовательное чередование стадий цикла, а не строгую их периодичность. Поэтому для более реалистического описания подобных процессов необходимо создать модели, позволяющие выявлять хаотическую динамику экономических переменных. Системы с хаосом демонстрируют одновременно и хорошую управляемость и удивительную пластичность: система чутко реагирует на внешние воздействия, сохраняя при этом тип движения. Комбинация управляемости и пластичности, по мнению многих исследователей, является причиной того, что хаотическая динамика является характерным типом поведения для многих жизненно важных систем (в том числе для подсистем живых организмов). При этом с позиций синергетического подхода данные модели должны описывать способность систем к самоорганизации. В предыдущей работе автора приведена одна из таких моделей, описывающая слияния и поглощения компаний [2].

Целью настоящей статьи является рассмотрение синергетического подхода к образовательным процессам на примере циклической модели, описывающей колебания численности студентов, усваивающих учебные программы, и потребности в выпускаемых специалистах. Модель допускает режимы квазипериодический, резонансный и детерминированного хаотического поведения траекторий. Каждый из этих режимов отвечает определенным модельным поведениям в усвоении студентами образовательной информации. Отметим, что кроме чисто методологического значения предлагаемая модель достаточно реалистична и может использоваться для объяснения многих закономерностей исследуемых процессов.

Описание и анализ модели

Прежде чем приступить к описанию модели, определимся с некоторыми понятиями. *Образовательной системой* в дальнейшем будем считать систему, в которой осуществляются образовательные процессы. Отметим также, что говоря об образовательной системе, мы будем подразумевать, прежде всего, организацию обучения и образовательные процессы в высших учебных заведениях.

Образовательные учреждения являются открытыми системами, взаимодействующими с окружающей средой. В роли последней выступают система экономического законодательства, сис-

тема ценностей и менталитета общества, система региональных ограничений и преимуществ. Заметим также, что главным потребительским свойством образовательной услуги является ее качество. При этом управление качеством образования существенно отличается от управления качеством, например, продукции тем, что качество готовой продукции непосредственно не зависит от ее потребителя. Качество же получаемого образования в значительной мере зависит от самого потребителя этой услуги (т. е. от учащегося), в том числе от его довузовской подготовки. Более подробно концепция производственной функции высшего образования рассмотрена в [3]. На практике процесс обучения организуется как периодически действующая система. Кроме того, выпускники вузов должны повышать свой образовательный уровень после окончания учебных заведений (периодически и непериодически). Таким образом, модель образовательного процесса должна описывать сложную динамику исследуемой системы, включая равновесные режимы (устойчивые предельные циклы, фокусы), а также широкий набор режимов детерминированного хаоса. Очевидно, что число студентов, усваивающих образовательные программы, предложенные вузом, колеблется при изменениях объема этих информационных ресурсов, уровня квалификации и усердия студентов, уровня их мотивации, связанного с востребованностью выпускников вуза на рынке труда.

В рамках данной концепции динамическими переменными предлагаемой модели (применительно к вузу) являются число u информационных образовательных ресурсов (образовательных программ, предлагаемых вузом), число v усваивающих эти программы студентов и число w вакансий на рынке труда для выпускников вуза. Исследуемая модель является обобщением и вертикальным расширением хорошо известной модели — "хищник — жертва" (модель Лотка—Вольterra) [4]. Модель применима преимущественно к вузам и описывает колебания численности студентов, усваивающих предложенные им и выбранные ими (дополнительное образование) образовательные программы и колебания их востребованности на рынке квалифицированных специалистов (считаем квалифицированным специалистом выпускника, работающего по специальности или не по специальности, но выполняющего служебные обязанности, требующие навыков и знаний в объеме вуза). При этом полагаем, что востребованными на рынке труда являются не все выпускники вуза, положительно освоившие учебные программы, а лишь их часть, которая, в частности, успешно может применять полученные знания на практике.

Запишем уравнения модели. Пусть прирост образовательной информации $u(t)$, используемой вузом, определяется информационным потоком u'_+ (где верхний индекс означает производную по времени), поступающим в популяцию студентов с постоянной скоростью γ . Убывание образовательных ресурсов определяется потоком u'_- и происходит за счет старения информации в результате ее потребления студентами и специалистами, повышающими квалификацию после окончания вуза. Кроме того, в потоке u'_- следует учесть релаксационный член, отражающий релаксацию информационных ресурсов к стационарному равновесному состоянию в отсутствие взаимодействия студентов и специалистов с информационными ресурсами. Общий информационный поток $u' = u'_+ - u'_-$.

Аналогично, динамика студентов, усваивающих образовательные программы, $v' = v'_+ - v'_-$, где прирост студенческой популяции v'_+ пропорционален произведению числа студентов на объем информационных ресурсов, предлагаемых им (считаем, что чем больше образовательных программ предлагает вуз, тем большее число студентов изучает их). Убыль этой популяции v'_- пропорциональна произведению числа студентов, закончивших обучение, на потребителей выпускников вузов (как принято в модели "хищник—жертва") и релаксационному члену.

Наконец, динамика специалистов на рынке труда $w' = w'_+ - w'_-$, где прирост специалистов определяется поступлением в их ряды выпускников вузов (пропорционален произведению популяции специалистов на выпускников вузов) и повышением образовательного уровня или переквалификацией самих специалистов (пропорционален произведению их популяции на объем информационных ресурсов). Убыль данной популяции w'_- определяется релаксационным членом, отражающим кроме прочих факторов, также факт утраты квалификации, естественной старости и смерти. Заметим, что усвоение образовательных программ зависит как от уровня начальной подготовки студентов, так и от квалификации студентов, приобретаемой ими в ходе обучения и постоянно увеличивающейся от курса к курсу. Этот факт также должен быть отражен в уравнениях предлагаемой модели.

Таким образом, запишем уравнения модели в виде

$$u' = -u + \gamma - \chi_1 u - \alpha_1 u w; \quad (1)$$

$$v' = -v + m\lambda(t)u - \beta v_\tau w - \mu_1 v_\tau; \quad (2)$$

$$w' = -\delta(w - w^{st}) + \alpha_2 u w + \beta v_\tau w + \mu_2 v_{\tau 1}; \quad (3)$$

$$m' = f(m) + I(t). \quad (4)$$

В уравнениях (1)—(4) u', v', w', m' — производные по времени; μ_1, μ_2 — коэффициенты. Уравнение (1) описывает прирост с постоянной скоростью γ и убыль в результате потребления (старение информации) студентами и специалистами информационных ресурсов (обучающих программ) со скоростями $\chi_1(t)$ и α_1 соответственно. Переменная m характеризует квалификацию обучающегося.

Уравнение (2) описывает изменение численности студентов v , усваивающих со скоростью $\lambda(t)$ предоставленную им информацию в объеме u .

Уравнение (3) описывает изменение численности квалифицированных специалистов на рынке труда, в ряды которых "вливаются" востребованные рынком выпускники вузов.

Первые слагаемые в уравнениях (1)—(3) описывают релаксацию динамических переменных к стационарному равновесному состоянию в отсутствие взаимодействия студентов с информационными ресурсами и специалистов с выпускниками (в первых двух уравнениях стационарные значения переменных u, v равны нулю, а в третьем — стационарное значение $w = w^{st}$).

Параметр δ характеризует скорость релаксации рынка труда к стационарному состоянию в отсутствие выпускников вузов, т. е. устойчивость этого рынка к нехватке квалифицированных кадров. Чем меньше значение δ , тем меньше скорость релаксации, тем устойчивее рынок трудовых ресурсов, тем более высокие требования предъявляются к выпускникам вузов. Таким образом, параметр δ можно считать параметром, ответственным за мотивацию обучающихся.

В уравнениях (1)—(2) скорости релаксации информационных ресурсов и обучающихся студентов равны единице. В уравнениях (1)—(3) коэффициенты α, β являются постоянными скоростями "поглощения" информации подготовленными специалистами (специалисты тоже, как и студенты, повышают свой образовательный уровень) и востребованности окончивших вуз студентов на рынке квалифицированных специалистов соответственно. При этом используются предположения, принятые в теории "хищник—жертва" (например, изменение популяции студентов и специалистов за время dt пропорционально произведению их числа на объем информации и т. д.).

Уравнение (4) описывает изменение квалификации m студента в процессе обучения, влияющей на степень усвоения образовательной информации [5]. Параметр $I(t)$ в этом уравнении характеризует усилия, вкладываемые системой образования в обучение студента [5]. В простейшем случае в линейном приближении примем $I(t) =$

$= I_0 = \text{const}$, $f(m) = am$, $a < 0$ (хотя это довольно грубое приближение и справедливо лишь в начале процесса обучения). Коэффициент a показывает, насколько легко студент осваивает новый материал. В вычислениях примем $a = -0,1$; $I_0 = 0,1$; $m_0 = 0,1$, где m_0 — начальный уровень квалификации студента при поступлении в вуз. Переменная $v_\tau = v(t - \tau)$, τ — время запаздывания, равное периоду обучения студентов (студент должен пройти весь курс обучения, прежде чем он будет востребован на рынке труда). Последние слагаемые в уравнениях (2)—(3) описывают, соответственно, окончание обучения студентов, не востребованных пока рынком труда, и приход этих студентов через некоторое время ($\tau_1 > \tau$) на этот рынок.

Специфика российской системы образования состоит в одновременном преподавании студентам большого числа дисциплин в течение всего семестра и сдачи зачетов и экзаменов студентами по ним в течение сессии (двух—трех недель). При такой организации учебного процесса получается, что "от сессии до сессии живут студенты весело", а потом за несколько ночей авральным методом проглатывают материал, чтобы донести его до зачета или экзамена (в отличие от модульной системы обучения за рубежом, при которой предполагается освоение одновременно лишь одной дисциплины, сдачу экзамена по ней и затем переход к другой). Исходя из сказанного считаем, что переменная $\lambda(t)$ имеет периодическую зависимость от времени

$$\lambda = \chi_2(1 + 0,5\cos(\omega t)),$$

где χ_2 характеризует амплитуду усвоения студентами информации; $\omega = 2\pi/T$ — частота колебаний; T — период колебаний, принимаемый равным 0,5 учебного года (от начала семестра до конца сессии, $t = 0$ соответствует началу зимней сессии). Заметим, что параметры $\chi_1 \ll \chi_2$, $\alpha_1 < \alpha_2$, так как убыль информационных ресурсов (старение) в результате использования их студентами и специалистами происходит гораздо медленнее, чем усвоение этих ресурсов (ресурсы должно освоить большое число студентов и специалистов, прежде чем они обесценятся).

Таким образом, система уравнений (1)—(4) описывает неавтономный осциллятор, допускающий различные режимы поведения: от периодического до квазипериодического и хаотического. Интегрирование этой системы уравнений проводилось методом Рунге—Кутты 4-го порядка с шагом по времени, равным 0,001 при следующих значениях параметров: $\alpha_1 = 0,05$; $\alpha_2 = 0,5$; $\beta = 0,6$; $w^{st} = 0,03$; $\chi_1 = 0,01$; $\gamma = 1,5$; $\mu_1 = 0,1$; $\mu_2 = 0,07$. Параметры χ_2 (параметр прилагаемых студентами усилий) и δ (параметр мотивации) яв-

ляются управляющими (бифуркационными) и варьируются в пределах: χ_2 от 0,1 до 5; δ от 0,5 до 15. Время запаздывания τ (и τ_1) также является бифуркационным параметром. Исследование влияния времени запаздывания (при изменении τ и τ_1 от 0,1 до 5) на характер динамических режимов показало, что характер этих режимов не меняется, а происходит лишь параметрический сдвиг их проявления. Так как предлагаемая модель описывает динамические режимы качественно, то в дальнейшем будем считать время запаздывания равным нулю, что намного ускоряет вычисления (так как позволяет перейти от бесконечномерной к конечномерной системе уравнений).

Сначала исследуем систему уравнений (1)—(4) для случая постоянной квалификации студентов $m = 1$, $a = -1$, $I_0 = 1$.

Исследование модельных уравнений проводилось по компьютерной технологии двух- и однопараметрического исследований [6]. На рис. 1 (см. вторую сторону обложки) приведена двухпараметрическая бифуркационная диаграмма динамических режимов исследуемой модели (параметры χ_2 , δ , максимальный период исследуемых циклов равен 35), а на рис. 2 — однопараметрический срез этой диаграммы при значении $\delta = 9,23$. Из анализа и сравнения двух приведенных бифуркационных диаграмм отчетливо виден переход от предельного цикла к двумерному тору (на рис. 2

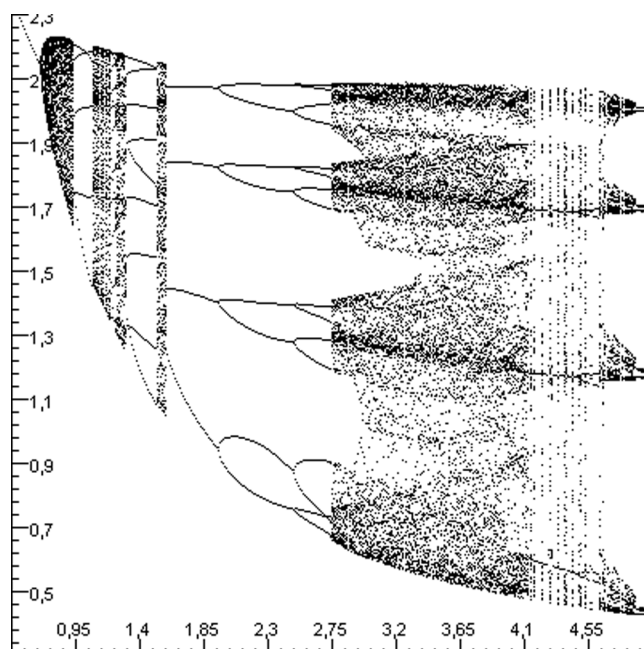


Рис. 2. Однопараметрическая бифуркационная диаграмма динамических режимов при $m = 1$, $\delta = 9,23$. По оси абсцисс отложены значения параметра χ_2 , по оси ординат — переменной u (информационный ресурс)

при $\chi_2 \approx 0,5$) и режимы квазипериодических колебаний на торе.

Характер квазипериодических режимов зависит от числа вращения θ , определяющего отношение базовых частот колебаний [6]. Если θ рационально, то имеет место резонанс на торе (и, соответственно, периодические колебания). При иррациональном значении числа вращения движение на торе будет эргодическим, при котором траектории плотно заполняют всю поверхность тора. Области резонансов на двухпараметрической диаграмме (см. рис. 1 на второй стороне обложки) имеют форму языков, опирающихся острым концом на соответствующие точки линии рождения тора. Эти области называют языками, или клювами, Арнольда [6].

Для каждого выбранного пути движения в пространстве параметров характерна своя последовательность бифуркаций, связанных с возникновением и исчезновением различных резонансов на торе. Возникновение резонансных циклов на торе однозначно связано с явлением синхронизации частот в отношении, соответствующем числу вращения. На рис. 1 отчетливо видна область однопериодного предельного цикла при всех варьируемых значениях δ и значениях χ_2 от 0,5 до 0,75 и области периодических режимов (соответствующие периоды обозначены целыми числами). Кроме языков Арнольда на этом рисунке присутствуют области квазипериодических режимов (окрашены белым цветом) и области детерминированного хаоса (окрашены пестрым цветом) в интервале изменения χ_2 от 1,75 до 2,5. Отметим, что переход к хаосу от двумерного тора происходит по сценарию Фейгенбаума путем удвоения периода резонансных циклов (на рис. 2 в интервале изменения χ_2 от $\sim 1,9$ до $\sim 2,75$).

Характерной особенностью языков Арнольда на рис. 1 и 2 является то, что внутри языков присутствуют бифуркационные области удвоения периода. Заметим также, что с ростом параметра δ , характеризующего устойчивость рынка трудовых ресурсов, области, занятые языками Арнольда, увеличиваются в размерах (см. рис. 1), но периоды резонансных циклов, как и ожидалось, уменьшаются. При этом в промежутках между широкими языками видны очень узкие языки Арнольда, соответствующие длиннопериодным циклам. Кроме того, как видно на рис. 1, ширина языков возрастает с ростом интенсивности внешнего периодического воздействия χ_2 .

Исследуем систему уравнений (1)–(4) при изменяющейся в процессе обучения квалификации m студентов. Двухпараметрическая бифуркационная диаграмма при переменном m (от 0,1 до 1) в общих чертах не отличается от диаграммы, представленной на рис. 1. Отличие проявляется в смещении бифуркационных границ удвоения периода к хаосу (из внутренней области языков). Осо-

бенно отчетливо это видно на рис. 3 (см. вторую сторону обложки), на котором в увеличенном виде изображена часть диаграммы в интервале изменения δ от 4,0 до 7,0. Кроме того, изменяется форма и размеры языков Арнольда. Они уменьшаются в размерах.

Анализ результатов вычислений

Как уже говорилось выше, на практике процесс обучения организуется как периодически действующая система. В рамках предлагаемой модели переход к режимам предельных циклов с возрастающими периодами через хаотические стадии происходит в полном соответствии со схемой, отмеченной Д. С. Чернавским [7]. В общем виде эта схема совпадает с известной триадой, предложенной двести лет назад, в трудах Гегеля:

тезис—антитезис—синтез.

Здесь "тезис" и "синтез" можно понимать как стадии упорядочения, причем "синтез" содержит большее количество информации и соответствует более сложному высокоэнтропийному порядку. При этом информацию будем понимать как выбор одного из множества вариантов и запоминание его. "Антитезис" можно понимать как противоположность упорядочению, т. е. хаотическую стадию, которую Д. С. Чернавский называл "перемешивающим" слоем [7]. Наличие "перемешивающего" слоя является необходимым условием для генерации информации в системе и соответствует творческой активности личности.

Особенно отчетливо смена режимов по этой схеме проявляется на рис. 2, на котором видна смена динамических режимов с изменением амплитуды χ_2 . При этом предельный цикл периода можно считать, например, режимом успешного усвоения лишь одной дисциплины. Этот режим проявляется при маленьких значениях амплитуды колебаний χ_2 , когда студенты практически не прикладывают никаких усилий для усвоения курсов. Квазипериодичность проявляется в том, что студенты одновременно изучают несколько учебных курсов, сам процесс обучения носит периодический характер, но усвоение каждой дисциплины описывается индивидуальной траекторией и отсутствует процесс согласования (синхронизации) между изучением отдельных курсов (циклы не повторяются).

Такая синхронизация проявляется при определенных значениях параметров χ_2 и δ , определяющих области, занятые резонансными циклами. Эти области интересны тем, что внутри них может происходить, например, согласование процесса усвоения нескольких дисциплин (число дисциплин равно периоду резонансного цикла). Например, на рис. 2 при значении $\chi_2 = 0,95$ может происходить согласование усвоения пяти дисциплин,

при $\chi_2 = 1,4$ — девяти, т. е. с увеличением усилий студентов χ_2 появляются ("синтезируются" по триаде Гегеля) более высокоинформативные области, т. е. области с более сложным упорядочением (с более высоким периодом циклов). Кроме того, как уже говорилось выше, размеры областей согласования увеличиваются, а период циклов уменьшается с понижением устойчивости рынка труда к нехватке выпускников вузов на нем (параметр δ).

Режим детерминированного хаоса, проявляющийся при значениях χ_2 от $\approx 2,75$, характеризуется странным аттрактором. Этот аттрактор отличается от спирального наличием выступов (складок). Особенный интерес представляет область перехода от резонансных циклов к детерминированному хаосу (кромка хаоса). Так как уже говорилось, что подобный переход происходит по механизму удвоения периода, то эта область является наиболее подходящей для творческой активности студентов (успешного усвоения множества дисциплин). Именно в этих областях расположены очень узкие длиннопериодные языки Арнольда, соответствующие максимумам усвоения информации. Более реалистична ситуация изменяющейся в процессе обучения квалификации m студентов. В этом случае границы областей удвоения периодов (кромка хаоса) смещаются в область хаоса (рис. 3), т. е. в область генерации информации (творческой активности студентов).

Кроме того, режим детерминированного хаоса интересен своей пластичностью, т. е. управляя этим режимом, можно легко перейти к режимам

периодических движений (с различными периодами). Таким образом, варьируя параметры, характеризующие периодичность усвоения студентами информации (χ_2) и устойчивость рынка труда (δ), можно регулировать процесс обучения.

Заключение

Предлагаемая авторами синергетическая модель описывает различные режимы функционирования образовательных процессов: от периодических и квазипериодических до режимов детерминированного хаоса. В рамках этой модели показано, что, варьируя управляющие параметры системы уравнений, описывающих модель, можно управлять процессом усвоения студентами образовательной информации.

Список литературы

1. Колемаев В. А. Математическая экономика. М.: Юнити. 2002. 400 с.
2. Буланичев В. А., Серков Л. А. Модельный подход к самоорганизующимся системам с детерминированным хаосом // Информационные технологии. 2006. № 7. С. 48—53.
3. Буланичев В. А., Серков Л. А. Модельный подход к функционированию вузов как самоорганизующихся систем // Информационные технологии. 2006. № 3. С. 68—73.
4. Смит Дж. М. Модели в экологии. М.: Мир, 1976. 184 с.
5. Капица С. П., Курдюмов С. П., Малинецкий Г. Г. Синергетика и прогнозы будущего. М.: УРСС, 2003. 288 с.
6. Анищенко В. С. и др. Нелинейные эффекты в хаотических и стохастических системах. М.—Ижевск: Ин-т компьютерных исследований. 2003. 530 с.
7. Чернавский Д. С. Синергетика и информация. Динамическая теория информации. М.: УРСС, 2004. 288 с.

УДК 11:004.89

Л. Р. Черняховская, д-р техн. наук, проф.,
И. Б. Герасимова, канд. техн. наук, К. Р. Нугаева
Уфимский государственный авиационный
технический университет

Поддержка принятия решения по управлению качеством образовательного процесса на основе хранилища знаний университета

Рассмотрена проблема повышения качества образования в высших учебных заведениях Российской Федерации за счет совершенствования системы управления образовательными процессами. Предлагается подход к обеспечению информационной поддержки принятия решений на основе хранилища знаний университета.

Введение

Одной из актуальных задач в области образования является повышение его качества. Проблема качества подготовки является центральной в аспекте востребованности специалистов национальной экономикой и международного признания российских степеней и квалификаций. Выпускаемые вузом специалисты — очень трудоемкая "продукция", производство которой невозможно без эффективного управления учебным заведением. Система отечественного высшего образования стоит на пороге интеграции с образовательными системами стран-участников Болонского соглашения, следовательно, необходимо обеспечение соответствия качества образовательного процесса международным стандартам.

Образовательная система выпускает специалистов с высшим образованием, часть из которых в дальнейшем пополняет кадровый потенциал для

научно-исследовательской инновационной системы. Научно-исследовательская система тоже производит особый вид продукции — новые знания, получаемые в ходе выполнения фундаментальных и прикладных научных исследований. Знания как один из компонентов интеллектуального потенциала используются как в инновационной системе, так и в образовательном процессе.

Решение задач управления в сложных системах в настоящее время лежит в сфере проектирования корпоративных информационных систем. Современные университеты не уступают по сложности крупным производственным предприятиям, а потому их нормальное функционирование уже невозможно без внедрения в систему управления вузом корпоративной информационной системы. У крупного предприятия и вуза много общего: сопоставимы финансовые показатели, численность персонала, филиальная сеть и т. п. Однако в управлении университетом есть особенности, которые препятствуют внедрению комплексных информационных систем, используемых на производстве. Решением проблемы является разработка оригинальной университетской информационно-управляющей системы, обеспечивающей поддержку принятия решений на основе хранилища знаний университета.

Качество образовательного процесса

Образование — это процессы и одновременно результаты этих процессов, представленные в виде определенного набора знаний, навыков, умений, культурных и нравственных установок, приобретаемых личностью. Применительно к высшему образованию — это процессы и виды деятельности вузов, направленные на подготовку специалистов в той или иной области на основе передачи студентам определенной совокупности теоретических и практических знаний, необходимых для реализации успешной профессиональной деятельности. Образование, как и любой процесс или результат деятельности человека, обладает определенным качеством [1]. Проблема заключается не только в том, чтобы разработать оптимальную для вузов модель качества подготовки специалистов, эта модель еще должна отвечать запросам общества (заказчика, работодателя).

Рассматривая качество образования "изнутри", как показатель качества функционирования высшего учебного заведения, и "извне", как показатель подготовки специалиста, функционирующего в социуме, мы приходим к выводу, что интегральный показатель комплексной оценки качества образования складывается из внутренней оценки качества образования, как качества потенциала и качества процесса образования, и

внешней оценки качества образования, как качества результата образования [1].

Последовательная иерархическая декомпозиция каждого из этих интегральных свойств, основанная на результатах онтологического анализа, позволяет определить множество свойств объектов образовательного процесса, которые могут быть найдены на основе измеряемых данных в образовательном процессе или оценены экспертно. Далее каждому из перечисленных элементов ставится в соответствие шкала оценок, состоящая из 5 типов оценок: низкая, ниже среднего, средняя, выше среднего и высокая. В числовом выражении этим оценкам соответствуют следующие диапазоны: 0—0,2; 0,21—0,4; 0,41—0,6; 0,61—0,8; 0,81—1. По данной шкале каждому элементу дается оценка на основании имеющихся знаний и результатов мониторинга образовательного процесса.

Таким образом, в замкнутой системе оценка качества образования осуществляется с двух сторон:

1) внутренняя оценка качества образования сводится к оценке предоставляемых образовательных услуг, что зависит от потенциальных возможностей университета, и к самооценке знаний выпускников;

2) внешняя оценка качества связана с оценкой уровня профессиональных знаний выпускников вузов и их соответствия требованиям, предъявляемым потребителями (работодателями).

Жизненный цикл процесса подготовки специалиста

Инновации в стратегическом управлении образованием приводят к открытию новых специальностей, направлений, что, в свою очередь, требует создания новых учебных курсов. Динамика изменения требований к подготовке специалистов в новых социально-экономических условиях вызывает потребность в мобильных учебных системах. Жизненный цикл знаний, используемых в процессе подготовки специалиста, имеет вид спирали (рис. 1).

При формировании новых направлений подготовки специалистов получение и освоение новых знаний происходит на основе уже приобретенных знаний. Знания — главная ценность в деятельности университета. Университетские знания формируются в результате методических разработок и научно-исследовательской деятельности преподавателей и студентов. Накопленные за многие годы знания при эффективном использовании повышают качество образовательных процессов. На сегодня большая часть методических материалов университета представляет собой множество

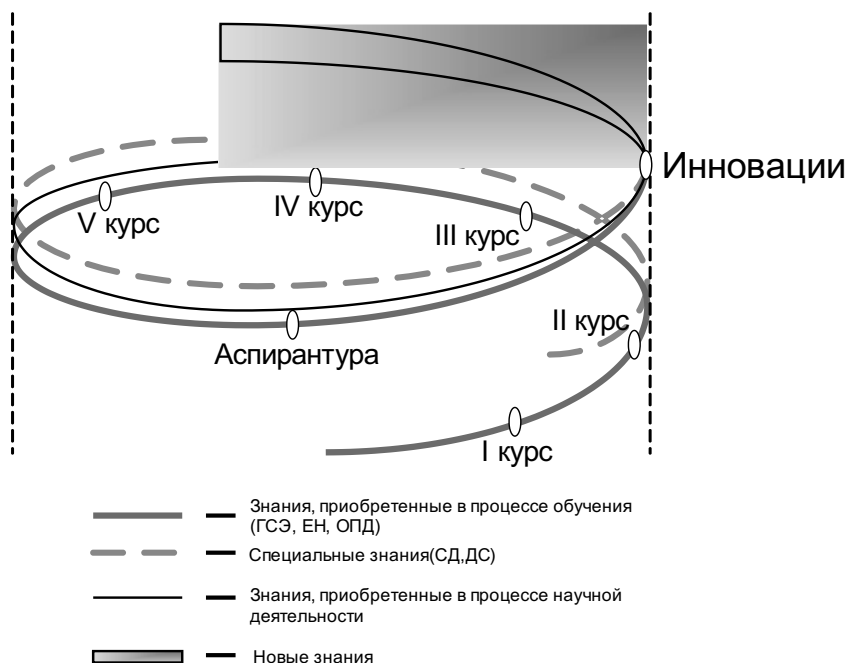


Рис. 1. Жизненный цикл подготовки специалиста

слабоструктурированных, в большей части бумажных документов, не используемых в полном объеме. Для решения поставленной задачи предлагается разработка интеллектуальной системы поддержки подготовки специалиста (ИСППС) на основе методологии объектно-когнитивного анализа [2] (рис. 2). Основные функции ИСППС — оценка качества образовательных процессов на основе предложенной методики критериев качества, поддержка принятия решений по управлению качеством образовательных процессов на всех этапах жизненного цикла подготовки специалистов, обеспечение обратной связи со средой потребителей образовательных услуг.

Основными модулями разрабатываемой ИСППС являются онтологическое хранилище знаний и модуль формирования решений на основе онтологии. Для перехода от данных к знаниям используется интеллектуальный анализ данных, в результате которого имеем жесткие классы объектов образования. Знания хранилища регулярно пополняются новыми требованиями к знаниям специалистов, которые поступают в систему управления образовательными процессами со стороны конечных потребителей — работодателей. Эти требования анализируются экспертом с помощью ИСППС и вносятся в хранилище знаний в виде новых фрагментов знаний, которые будут в дальнейшем использоваться при разработке учебно-методического комплекса для конкретной специальности по конкретной дисциплине.

Предложена методика разработки ИСППС, основой которой является объектно-когнитивный

анализ (ОКА) предметной области, интегрирующий методы объектно-ориентированного анализа, онтологического анализа и семантической сети представления знаний.

Объектно-ориентированный анализ — способ анализа, изучающий требования к системе с точки зрения будущих классов и объектов, на основе словаря предметной области [3].

Онтологический анализ — это уровень анализа знаний, в основе которого лежит описание предметной области в терминах сущностей, отношений между ними и действий над сущностями [3].

Семантический анализ — это анализ предметной области, направленный на описание и идентификацию базовых элементов предметной области, установление взаимосвязей (отношений) между ними и определение характеристик отношений [4].

Результатами объектно-когнитивного анализа являются формальные описания отношений между абстрагированными понятиями и сущностями, являющимися базовыми объектами предметной области (когнитивными элементами), в терминах предметно-ориентированного тезауруса. В соответствии с предложенной методикой начальным этапом разработки ИСППС является моделирование образовательного процесса. Затем проводится онтологический анализ предметной области и разрабатывается онтология предметной области. На следующем этапе осуществляется формирование моделей представления знаний и синтез алгоритмов поддержки принятия решений с использованием хранилища знаний.

Моделирование интеллектуальной системы поддержки подготовки специалистов

При моделировании ИСППС использовались специальные методы и средства объектно-ориентированного моделирования предметной области, разработанные для проектирования информационных систем, чтобы воссоздать концептуальную модель экспертов в формализованной модели представления знаний. Предлагается использование объектно-ориентированного подхода к моделированию учебного процесса, которое базируется на анализе профессиональной деятельности и разработке требований к специалистам, формируемых на основе регламентирующих (стандарт специальности, типовой учебный план, типовые рабочие программы) и дидактических материалов

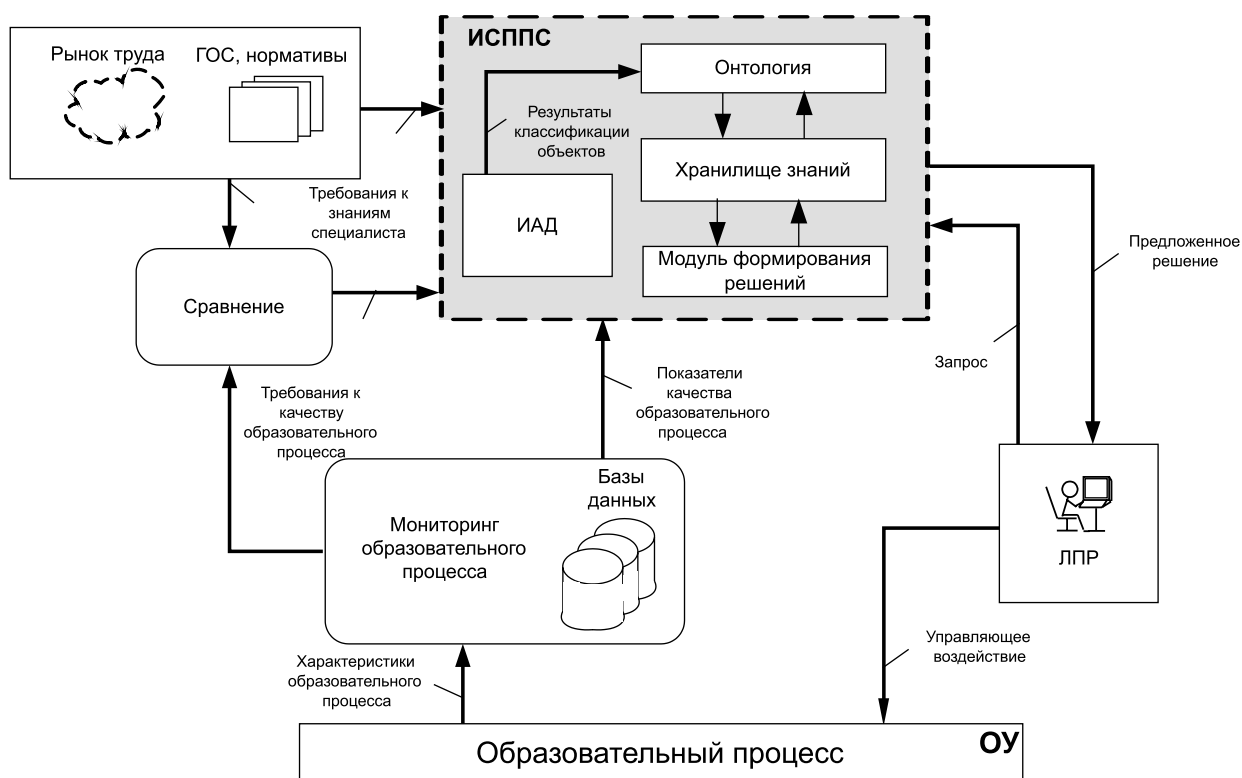


Рис. 2. Контур управления качеством образовательных процессов

(учебники, учебные пособия, научно-техническая литература), потребностей рынка труда конкретного региона и мирового уровня знаний специалистов в данной предметной области. Рынок труда представляет собой службы занятости, кадровые агентства, потребителей знаний специалистов, которые являются работодателями. Моделирование проводилось на языке *Unified Modeling Language* и среде *Rational Software 2003*. В соответствии с методологией объектно-ориентированного анализа и проектирования первым этапом является анализ требований, который подразумевает выделение процессов и требований, и их формулировку в виде прецедентов.

Внешние требования к знаниям подготавливаемых специалистов формируются на основе требований общества (руководители коммерческих, государственных, научных организаций и учреждений). Внутренние требования к знаниям специалистов представлены требованиями государственного образовательного стандарта (ГОС) по данной специальности и соответствуют критериям качества знаний студента, установленным при изучении учебных дисциплин и подготовки выпускных квалификационных работ. Модель выявленных требований была разработана с использованием моделирования требований в *Rational Requisite Pro 2003*. В модели требований каждый учебный модуль соединяется с соответ-

ствующей дисциплиной (посредством изучения которой эти знания приобретаются будущим специалистом) отношением "родитель—потомок", и при этом он содержит: ссылки на базовые знания, т. е. на фундаментальные работы, в которых даны определения ключевых терминов и основы теории; ссылки на близкие по содержанию знания; ссылки на соответствующие требования в области знаний, умений и навыков специалистов.

Для формирования связей терминов и дисциплин создана специальная матрица с помощью средств *Coverage Analysis*, в которой установлены связи основных терминов предметной области и специальных дисциплин. Для этого в проекте сформирована матрица соответствия терминологии предметной области и специальных дисциплин (рис. 3).

Иерархическая структура учебных модулей использована для разработки виртуальных учебников и контрольных заданий для аттестации учащихся, разработки учебных планов для новых специальностей и корректировки для имеющихся.

В процессе моделирования устанавливаются парадигматические отношения между когнитивными элементами учебного процесса (причинно-следственные, отношения сходства), а также отношения обобщения, ассоциации, зависимости и реализации, необходимые для разработки ком-

Relationships: - direct only	D1: Спец...	D1.1: Ин...	D1.2: За...	D1.3: Сре...	D1.4: Упр...	D1.5: Ауд...	D1.6: Фин...	D1.7: Упр...	D1.8: Упр...	D1.8.1:...	D1.8.2:...	D1.8.3:...	D1.8.4:...	D1.8.5:...	D1.8.6:...	D1.8.7:...	D1.9: Сер...
TERM1: качество																	
TERM1.2: система...																	
TERM1.2.1: уровень...																	
TERM1.2.1.1:...																	
TERM1.2.1.1.1:...																	
TERM1.2.1.1.2:...																	
TERM1.2.1.1.2.1:...																	
TERM1.2.1.1.3:...																	
TERM1.2.1.1.4:...																	
TERM1.2.1.1.5:...																	
TERM1.2.1.1.6:...																	
TERM1.2.1.1.7:...																	
TERM1.2.1.1.7.1:...																	
TERM1.2.1.1.8:...																	
TERM1.2.2: планирова...																	
TERM1.2.2.1: цели...																	
TERM1.2.2.1.1:...																	
TERM1.2.2.1.1.1:...																	
TERM1.2.2.1.1.1.1																	
TERM1.2.2.1.1.1.2																	
TERM1.2.2.1.1.1.3																	

Рис. 3. Матрица соответствия терминологии предметной области и специальных дисциплин

плекса объектно-ориентированных моделей облика специалиста по управлению качеством.

Полученные научные результаты использовались при подготовке специальных дисциплин для специальности "Управление качеством", будут полезными при разработке учебно-методического материала (УММ) по дисциплинам специальности. Программный продукт *Rational Requisite Pro* 2003 интегрирован с линейкой программных продуктов *Rational Software* 2003, что позволило перенести структурированные требования к образовательному процессу на диаграмму прецедентов *Rational Rose* 2003 в виде концептуальной модели и приступить к следующим уровням объектно-ориентированного моделирования ИСППС.

На основе объектно-когнитивного анализа предметной области разработан комплекс объектно-ориентированных моделей структуры и динамики функционирования информационной системы поддержки принятия решений. Модели содержат описания абстрагированных понятий и сущностей, являющихся базовыми объектами предметной области $a_i \in A$, и определяют отношения между ними. Отношения $a_i R^Z a_j$, $R^Z \in R$, принадлежат множеству отношений $R = \{R^A, R^H, R^D, R^F, R^C, R^S\}$, которое включает определенные в UML подмножества отношений ассоциации R^A , обобщения R^H , зависимости R^D , реализации R^F ,

а также парадигматические отношения новых стереотипов, а именно каузальные отношения R^C и отношения сходства R^S .

Моделирование проводится как "поуровневый спуск" от концептуальной модели к логической. Логическая модель образовательного процесса позволяет определять два различных взгляда на системы: статический и динамический. Статическая модель выражается диаграммами классов (*Class diagram*). Именно диаграммы классов служат основой для генерации программного кода на целевом языке программирования (рис. 4).

Разработка учебно-методического комплекса подготовки специалистов осуществляется преподавателями с участием методистов кафедры. Учебно-методический отдел университета и

учебно-методический совет по специальностям осуществляют контроль методического обеспечения учебного процесса и разработки нормативно-справочной документации. Учебно-методические материалы хранятся в базе данных. С помощью моделей динамики были отражены алгоритмы функционирования ИСППС.

На основе результатов объектного анализа и моделирования была разработана модель онтологии предметной области учебного процесса подготовки специалистов по специальности 220500 "Управление качеством"; с использованием терминов модели онтологии описаны контекстные условия определения и использования некоторых терминов в описании дисциплин.

Онтология образовательного процесса

Предложенный подход к организации поддержки принятия решений по управлению качеством образовательных процессов предполагает анализ данных о проблемных ситуациях и выявление присутствующих в них детерминированных и вероятностных структур. Для того чтобы перейти от разрозненных описаний проблемных ситуаций к поддержке принятия решений, необходимо осуществить переход от данных к знаниям на основе интеллектуальных методов анализа данных.

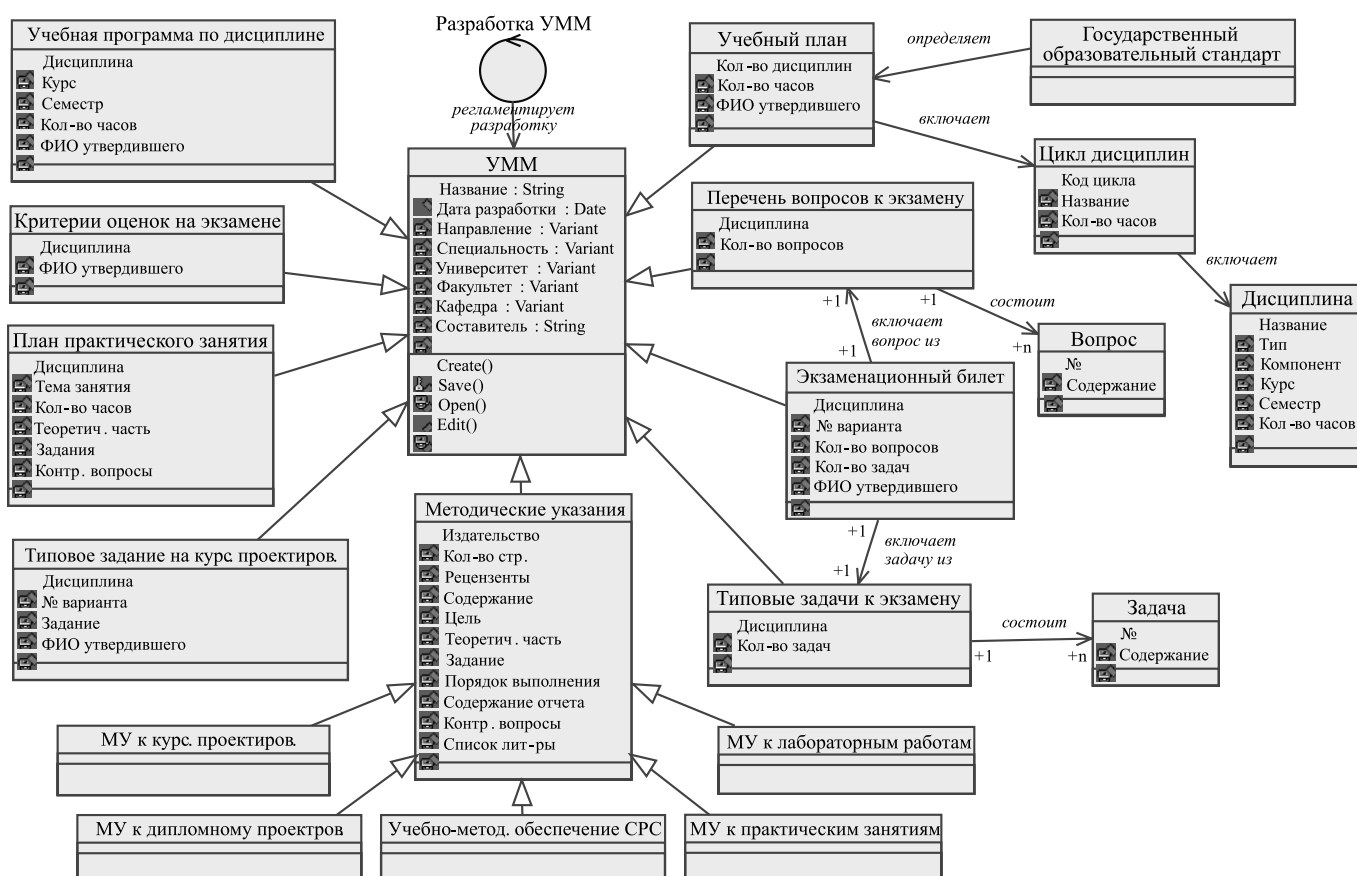


Рис. 4. Структура УММ

Современными средствами тезаурусной технологии являются информационно-программные инструментальные средства лингвистического анализа, использование которых позволяет автоматизировать ряд рутинных операций, например составление словаря терминов текста с указанием частот их встречаемости в источнике. Для кластерного анализа прецедентов управления качеством образования формируется матрица типа "термин—документ", отражающая вклад каждого из терминов словаря предметной области в раскрытие содержания каждого из включенных в рассмотрение документов. Затем была получена матрица $DT = \{dt_{ij}\}$, элементами которой являются значения бинарной переменной, отображающей факт присутствия (отсутствия) терминов в соответствующих описаниях. Далее была проведена кластеризация признаков посредством иерархического агломеративного метода.

Анализ полученных результатов позволил выделить укрупненные классы. С использованием полученной структуры в качестве ориентира, дальнейший анализ проводился с помощью итеративного метода k средних, в результате которого были определены усредненные характеристики полученных 10 кластеров. Таким образом, ис-

пользование методов кластерного анализа в сложных условиях, обусловленных особенностями описаний ситуаций, требующих управления, позволило выделить относительно устойчивую классификационную структуру в данных о прецедентах и сформировать информационное пространство для поиска решений в базе знаний. Далее экспертным образом была сформирована выборка описаний управляющих воздействий, включающая классы типовых решений, соответствующих выделенным категориям управления качеством образования.

Структурирование знаний о качестве образовательного процесса предлагается проводить с помощью онтологического анализа, в основе которого лежит описание предметной области в терминах сущностей, отношений между ними, и действий над сущностями. Таким образом, в процессе структуризации онтология образовательного процесса представляется в виде иерархической системы.

Онтология задает единое информационное пространство, в котором интегрируются различные модели представления знания об образовательном процессе, знания о конкретной области подготовки специалистов, представленное в форме онтологии $Onto^{edu}$ и $Onto_i^{app}$, правила управ-

ления образовательным процессом и прецеденты конкретных проблемных ситуаций, требующих принятия решений. Онтология образовательного процесса включает онтологию верхнего уровня (метаонтологию) $Onto^{meta}$, онтологию высшего образования $Onto^{edu}$ и предметные онтологии подготовки специалистов по различным специальностям $Onto_i^{app}$:

$$Onto = \langle Onto^{meta}, Onto^{edu}, \{Onto_i^{app}\}, Inf^F \rangle,$$

где Inf^F — модель машины вывода, ассоциированной с онтологической системой $Onto$.

Сущностями метаонтологии $Onto^{meta}$ являются такие понятия, как "объект", "атрибут", "значение", "отношение" и т. п. Онтология $Onto^{edu}$ оперирует понятиями, характеризующими процесс подготовки специалистов высшего образования (например, "университет", "преподаватель", "обучаемый", "учебный курс" и др.), и организуется по образцу метаонтологии $Onto^{meta}$.

Предметная онтология $Onto_i^{app}$ содержит понятия, характеризующие семантику особенностей подготовки специалистов по конкретным специальностям ("образовательный стандарт", "дисциплина специализации", "информационная технология" и др.), структурированные в соответствии с иерархией, установленной для конкретной специальности. Структура рассматриваемых предметных областей предполагает наличие в онтологии отношений наследования, статической агрегации, а также нескольких типов ассоциативных парадигматических отношений: отношений агрегации, причинно-следственных отношений, отношений сходства, отношений семантического подобия. Наполненная предметная онтология $Onto_i^{app}$ может рассматриваться как компонент хранилища знаний при работе с конкретной предметной областью и являться, в свою очередь, шаблоном для построения динамического компонента базы знаний, изменяющегося при переходе от исследования одной конкретной задачи к другой.

Концептуальная модель позволяет визуализировать с помощью UML 2.0 семантически связанную структуру предметной области и выявлять новые классы, атрибуты, их значения и отношения между ними. Данная модель позволяет объединить понятия из нормализованного словаря предметной области, полученного на основе лингвистического анализа методик, инструкций по управлению качеством образования и глоссария, сформированного сначала из объектных моделей в семантической сети со множеством связей. Затем выявленные в семантической сети отношения упорядочиваются на основании тезауруса с выделенными классами отношений. Далее на концептуальной модели к ней добавляются роли, кото-

рые делают онтологию семантически богатой структурой.

Полученная концептуальная модель онтологии позволила описать разрабатываемую онтологию по степени абстракции в соответствии с уровнями иерархии. На верхнем уровне онтологии выявлены классы "качество", "образовательный процесс", "управление", "структура". Этот уровень модели характеризуется общими знаниями о процессе. Затем в каждом из классов находятся подклассы, представляющие средний уровень онтологии знания о специфике образовательного процесса по конкретной специальности. Нижним уровнем иерархии в онтологии являются экземпляры классов. Таким образом, общие знания выявлены и сформированы на основании разработанной концептуальной модели онтологии.

Специальные знания о конкретных проблемных ситуациях представлены в форме прецедентов, основной отличительной особенностью которых является их объектная структура и интеграция с обобщенными (абстрагированными) знаниями предметной области. Метаонтология является верхним уровнем в иерархии разрабатываемых онтологий. Разработанные онтологии предметной области и прецедентов основаны на парадигматических и синтагматических отношениях, что позволяет осуществлять логический вывод на основе предложенных правил принятия решений и аксиом. Они формируются с экспертами на основании разработанной модели онтологии, модели динамики состояний оценки качества подготовки специалистов, а также проведенного кластерного анализа данных по управлению качеством образовательного процесса.

Пример правила оценки качества разработки учебно-методических документов для специальности в форме аксиомы на языке SWRL представлен ниже.

*Correct(?UMK, ?Spec, ?Disc) ← has(?LearnProg, ?Stat, ?KFr),
has(?KritMark), has(?ExamQuest, ?KFr), has(?Literat, ?Year),
has(?LearnCourse, ?Vid, ?KFr), has(?MetodMatLR, ?KFr),
has(?MetadMatCR).*

Здесь *Correct(UMK, Spec, Disc)* — это корректный комплекс учебно-методических документов для конкретной дисциплины конкретной специальности, содержащий весь необходимый комплект фрагментов знаний по данной дисциплине.

Алгоритм логического вывода на правилах выполняется на основе дескриптивной логики.

Процесс разработки учебного плана организуется в соответствии с ГОС, а также учитывая требования со стороны работодателя (конечного потребителя знаний специалиста), которые влияют на состав и структуру дисциплин. Помимо этого учитываются внутренние факторы, такие как возможность объединения занятий в "потоки".

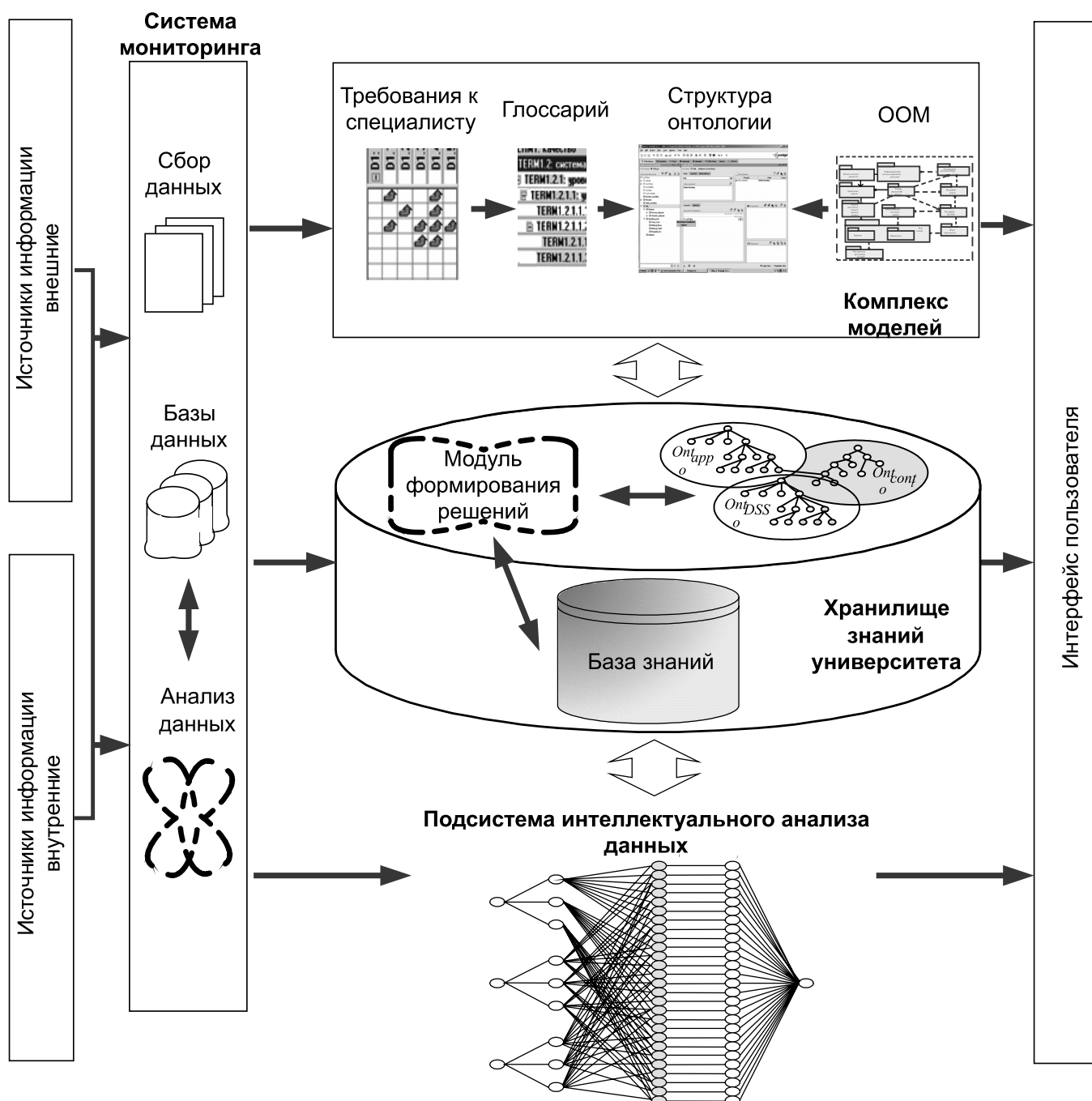


Рис. 5. Интеллектуальная система поддержки подготовки специалиста на основе хранилища знаний университета

Онтология прецедентов представляет собой нижний уровень иерархии онтологий: после метаонтологии и онтологии предметной области. Структура прецедентов является объектно-ориентированной. Каждый прецедент ассоциирован с ситуацией, требующей принятия решения по управлению качеством. Визуализация онтологии осуществляется средствами *Protégé* 3.1 на основе результатов анализа и моделирования предметной области.

Хранилище знаний

Практическая ценность любой системы, основанной на знаниях, в первую очередь, зависит от используемой базы знаний предметной области. Современная обучающая среда должна быть адаптивной, что обуславливает соответствующие требования к формированию, наполнению, использованию, адаптации и обучению базы знаний ИСППС. Для аккумуляции университетских знаний по управлению качеством образователь-

ного процесса предлагается использовать технологию хранилища знаний (рис. 5).

Хранилище знаний (*Knowledge Warehouse*) предназначено для интеграции и хранения знаний, представленных в форме логических утверждений, онтологии поддержки принятия решений и модуля прецедентов проблемных ситуаций. Таким образом, хранилища знаний подобны виртуальным складам, где знания распределены по большому числу серверов [5]. Модуль формирования решений реализует поиск решений в хранилище знаний в соответствии с типом модели представления знаний (дедуктивный вывод на онтологии и правилах или поиск по аналогии на множестве прецедентов). С помощью хранилища знаний предложенная ИСППС решает следующие задачи: разработка учебных планов для новых и их корректировка для имеющихся специальностей; разработка методического обеспечения (МО) по новым дисциплинам; обновление МО для имеющихся дисциплин; аттестация; тестирование (проверка уровня знаний учащихся); оценка качества образования с использованием предложенных критериев; разработка компьютерных обучающих систем, обеспечивающих возможность дистанционного обучения.

Результаты

Разработаны принципы представления знаний с использованием языка объектно-ориентированного моделирования UML (*Unified Modeling Language*). Формализация ИСППС для процесса управления качеством образования происходит на основе результатов анализа и моделирования предметной области. Формируются структуры предметной онтологии и онтологии прецедентов, разрабатываются правила и аксиомы. На основании выделенных групп экспертами были сформу-

лированы правила распознавания класса проблемной ситуации и записаны на языке формализации онтологических правил SWRL (*Semantic Web Rule Language*). В соответствии с разработанными правилами была построена онтология прецедентов, содержащая прецеденты управления качеством.

Заключение

Таким образом, задача обеспечения качества образовательного процесса — задача многокритериальная, решаемая в условиях существующей неопределенности. Для ее решения необходимы экспертные знания. Использование разработанной онтологической базы знаний, содержащей онтологию предметной области, онтологию прецедентов и правила распознавания проблемных ситуаций, позволяет увеличить полноту информационного обеспечения, делает возможным доступ к информации распределенных пользователей с помощью соблюдения стандартов Semantic Web; помогает управляющим образовательным процессом принимать более качественные решения в проблемных ситуациях.

Список литературы

1. **Коротков Э. М.** Управление качеством образования: Учеб. пособие для вузов. М.: Академический Проект; Мир, 2006. 320 с.
2. **Бадамшин Р. А., Ильясов Б. Г., Черняховская Л. Р.** Проблемы управления сложными динамическими объектами в критических ситуациях на основе знаний. М.: Машиностроение, 2003. 240 с.
3. **Гаврилова Т. А., Хорошевский В. Ф.** Базы знаний интеллектуальных систем. СПб: Питер, 2000. 384 с.
4. **Вагин В. Н.** Дедукция и обобщение в системах принятия решений. М.: Наука, Гл. ред. физ.-мат. лит., 1988. 384 с.
5. **Spinning the Semantic Web. Bringing the World Wide Web to it Full potential** / Ed. by D. Fensel, J. Hendler, N. Lieberman, W. Wahlster. The MIT Press. Cambridge, Massachusetts, London, England, 2006. 469 p.

Новые книги

Гуревич И. М. Законы информатики — основа строения и познания сложных систем. 2-е изд. М.: ТОРУС ПРЕСС, 2007. 400 с.

Второе издание книги посвящено основным законам и принципам информатики. Дано определение сложных систем и описаны их свойства. Определены ограничения на физически реализуемые преобразования, принцип информационной эквивалентности инерциальных систем координат. Дан анализ неопределенности описания и измерения наблюдаемых, логики квантовой механики. Оценены информационные характеристики сцепленных состояний. Приведены оценки процессов формирования информации Вселенной с ее структурными элементами, формирования частиц в инфляционной Вселенной. Дана классификация и проведен анализ массы (энергии) на основе информационных характеристик. Показано, что информационное взаимодействие — пятый вид фундаментальных взаимодействий. Проведено изучение познаваемости сложных систем, познаваемости Вселенной. Оценены предельные возможности познания. Сформулирована новая парадигма познания.

CONTENTS

Chernetsova E. A. <i>The Method of Computer Aided Processing Images Represented by Pixel Intensity Array on the Base of Neural Networks</i>	2
Korsunov N. I., Yudin A. A. <i>Solving of Thermal Conductivity Problem by Using Neuronet Computer Technologies</i>	6
Karpenko A. P., Fedoruk V. G. <i>An Overview of Multi-Criteria Optimization Software Systems. I. Systems Developed in Russia</i>	15
Gagarin A. V. <i>Intellectual Algorithm of Parameters Optimization of Resource-Consuming Models to Real-World Conditions</i>	23
Petunin A. A., Mukhacheva E. A., Filippova A. S. <i>Method of Rectangular Approximation for Solving the Problems of Non-Regular Figured Cutting-Packing</i>	28
Aksarin M. V., Tarasyk M. V. <i>Model of Covert Channel in Relational Multilevel Secure Databases</i>	32
Shniperov A. N. <i>Synthesis and Analysis High-Speed Symmetric Cryptosystem on the Basis of Operated Operations</i>	36
Mandrikov E. A., Kulev V. A., Shalyto A. A. <i>Application of Genetic Algorithms for Making of Controlling Automation in the Task about «Flibs»</i>	42
Dolgov A. I., Korochentsev D. A. <i>Addition of Possibility and Probabilistic Fuzzy Triangular Numbers</i>	46
Milov A. N. <i>Two Approaches to Organization of Scalable OpenGL Graphic Architectures Based on DSP Processors</i>	51
Orlov A. A., Kanunova E. E. <i>Digital Processing the Text on the Manuscript Scenes as Curve Objects</i>	57
Melman S. V. <i>Volume Rendering in Analysis of Physical Fields of Synoptic Objects</i>	62
Mantsivoda A. V., Malykh A. A. <i>Achievements on the Web and the Perspectives of the Russian Education Information System</i>	67
Serkov L. A. <i>Synergetic Modelling of Educational Processes</i>	74
Cherniakhovskaya L. R., Gerasimova I. B., Nugaeva K. R. <i>Decision Support System for Educational Quality Management on the Base of University Knowledge Warehouse</i>	79

Адрес редакции:

107076, Москва, Стромьинский пер., 4/1

Телефон редакции журнала **(495) 269-5510**

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Художник *В.Н. Погорелов*.
Технический редактор *О. А. Ефремова*. Корректор *Н. Д. Лельчук*

Сдано в набор 09.11.2007. Подписано в печать 17.12.2007. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.
Усл. печ. л. 10,78. Уч.-изд. л. 12,85. Заказ 20. 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