

IT-ТЕХНОЛОГИИ: РАЗВИТИЕ И ПРИЛОЖЕНИЯ

**Сборник докладов
Международной научно-технической конференции
(Владикавказ, декабрь 2019)**

Владикавказ 2019

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
СЕВЕРО-КАВКАЗСКИЙ ГОРНО-МЕТАЛЛУРГИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ)

IT-ТЕХНОЛОГИИ: РАЗВИТИЕ И ПРИЛОЖЕНИЯ

Сборник докладов
Международной научно-технической конференции
(Владикавказ, декабрь 2019)

Владикавказ 2019

УДК 004
ББК 73
И92

И92 **ИТ-технологии: развитие и приложение:** Сборник докладов Международной научно-технической конференции (Владикавказ, декабрь 2019) / Под ред. д-ра техн. наук Гроппена В. О.; Северо-Кавказский горно-металлургический институт (государственный технологический университет). – Владикавказ: Северо-Кавказский горно-металлургический институт (государственный технологический университет). Изд-во «Терек», 2019. – 178 с.

ISBN 978-5-9500071-6-3

В сборник включены лучшие доклады, представленные на Международную НТК «ИТ-технологии: развитие и приложения» (Владикавказ, декабрь 2019). В рамках основных направлений исследований были предложены: новые методы решения задач дискретной оптимизации на основе композитных алгоритмов; подходы к решению многокритериальных задач с булевыми переменными на основе свертки критериев методом эталонов; технологии оптимизации программного кода на основе новых критериев качества, учитывающих структуру программного алгоритма; новый способ оценки эффективности средств криптозащиты в многоканальных телекоммуникационных системах.

УДК 004
ББК 73

ISBN 978-5-9500071-6-3

© ФГБОУ ВО СКГМИ (ГТУ), 2019
© Авторы статей, 2019

СОДЕРЖАНИЕ

ТЕХНОЛОГИИ ПРИНЯТИЯ РЕШЕНИЙ

<i>Алексеева В. А., Санталов А. А.</i> Использование нейронных сетей NARX открытого и закрытого циклов для прогнозирования на фондовом рынке	5
<i>Гроппен В. О., Датиев А. А., Берко А. А., Буймистров В. А.</i> Экспериментальная проверка сравнительной эффективности традиционных и композитных методов неявного перебора применительно к задачам с булевыми переменными	12
<i>Будаева А. А., Артамонов М. А.</i> Экспериментальный анализ эффективности использования эталонов при решении многокритериальных оптимизационных задач с булевыми переменными методами типа ветвей и границ	21
<i>Носова Т. Н.</i> Система поддержки принятия решений, предназначенная для формирования нечеткой оценки деятельности куратора вуза.....	27
<i>Рябова А. А., Куценко П. А.</i> Использование искусственного интеллекта при создании систем рекомендаций на сайте	36
<i>Гусейнзаде Ш. С.</i> Моделирование с применением интегрированных раскрашенностохастических сетей Петри на базе CPNTools.....	44

ТЕХНОЛОГИИ ОПТИМИЗАЦИИ ПОЛЬЗОВАТЕЛЬСКИХ КОДОВ

<i>Томаев М. Х., Миронян Р. А.</i> Формализации метода статического каэширования функций.....	54
<i>Томаев М. Х., Губиев Д. А.</i> Экстремальные подходы к выбору метода размещения пользовательских данных в многоуровневой оперативной памяти ЭВМ.....	58
<i>Мирошников А. С., Буймистров В. А.</i> Оптимизация создания экземпляров классов методом переноса кода установки начальных значений свойств класса из блока конструктора класса в секцию инициализации.....	61

ТЕХНОЛОГИИ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

<i>Проскурин А. Е., Токарева И. В.</i> Оптимизация метода распознавания изображений на основе алгоритма скелетизации.....	66
---	----

ТЕХНОЛОГИИ ЗАЩИТЫ ИНФОРМАЦИИ

- Бабенко А. А.** Разработка формальной модели исследования технических средств защиты информации в государственных информационных системах 73
- Ибрагимов Б.Г., Мамедов Т.Г.** Анализ методов криптозащиты информации в многоканальных телекоммуникационных системах 81

ТЕХНОЛОГИИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

- Касьянов В. Н., Касьянова Е. В.** Языковые и программные средства облачного параллельного программирования 90
- Калиниченко А. В., Габанова М. М.** Моделирование процесса перколяции на многопроцессорных системах и возможность применения теории перколяции для анализа компьютерных сетей 98
- Пекунов В. В.** Параллельное решение задачи автоматического достраивания порождающей модели программы на базе конструирующих XPath-запросов 105

ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В ОБРАЗОВАНИИ

- Оськин Д. А., Оськин А. Ф.** Организация процесса интеллектуального анализа образовательных данных 112
- Даурова А. А., Астахова Л. Г.** К вопросу о повышении эффективности преподавательской деятельности 120

ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ДЛЯ РЕШЕНИЯ АКТУАЛЬНЫХ ТЕХНИЧЕСКИХ ЗАДАЧ

- Клячкин В. Н., Жуков Д. А.** Применение агрегированных классификаторов для диагностики функционирования технического объекта 126
- Коляда А. А., Коляда Н. А., Протасеня С. Ю.** Алгоритм параметризации пороговой криптосхемы разделения секрета с применением минимально избыточной модулярной системы счисления 136
- Рябова А. А., Курдов К. Р.** Адаптация корпоративных информационных систем к потребностям конкретного предприятия 145
- Орлова Н. С.** Моделирование режимов виброкипения толстых слоев 151
- Черняев А. П.** Точные решения математической модели однокамерного сердца и условия ритмичной работы сердечной камеры 163

УДК 004.032.26

ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ NARX ОТКРЫТОГО И ЗАКРЫТОГО ЦИКЛОВ ДЛЯ ПРОГНОЗИРОВАНИЯ НА ФОНДОВОМ РЫНКЕ

Алексеева В. А., канд. техн. наук, доцент, v.fashutdinova@mail.ru

Санталов А. А., anton.santalov1995@gmail.com

Ульяновский государственный технический университет,

г. Ульяновск, Россия, www.ulstu.ru

Аннотация. В статье проводится сравнение эффективности применения нейронных сетей при прогнозировании на фондовом рынке. В рамках работы разработаны и протестированы нейронные сети NARX открытого и закрытого цикла. Определено, что возможно применение нейросети NARX закрытого цикла в качестве дополнительного инструмента краткосрочного прогнозирования при использовании стратегии скальпинга.

Ключевые слова: нейронная сеть NARX, нейронная сеть открытого цикла, нейронная сеть закрытого цикла, фондовый рынок.

USING OPEN-LOOP AND CLOSED-LOOP NARX NEURAL NETWORKS TO FORECAST STOCK MARKET

Alekseeva V. A., Santalov A. A.

Abstract. *This paper compares the effectiveness of using neural networks in the stock market forecasting. Open-loop and closed-loop NARX neural networks are developed and tested. The conclusion is as follows: using closed-loop NARX neural networks as an additional tool for short-term forecasting using the scalping strategy is possible.*

Keywords: *NARX neural network, open-loop neural network, closed-loop neural network, stock market*

Введение

Нейронные сети – один из самых популярных классов алгоритмов для машинного обучения. Нейронные сети позволяют автоматизировать процесс прогнозирования в сложных профессиональных задачах.

Отсутствие знаний о точных зависимостях между входными и выходными данными заставляет прибегнуть к исследованию этих зависимостей нейронными сетями. Вследствие особенностей применяемых методов зависимости между исходными данными определяются приближенно, поэтому предсказания, основанные на таких исследованиях, имеют определенную погрешность.

В финансовом анализе нейронные сети чаще всего применяются для прогнозирования, создания собственных индикаторов, алгоритмического трейдинга и моделирования рисков. Главной задачей данного исследования является анализ эффективности применения нейронных сетей при прогнозировании на фондовом рынке.

Исследуется возможность применения нейросети NARX закрытого цикла в качестве дополнительного инструмента краткосрочного прогнозирования при использовании стратегии скальпинга. Скальпинг (англ. Scalping – снимать скальп) – название одной из стратегий внутридневных спекулятивных операций на фондовом рынке, особенностью которой является закрытие сделки при достижении небольшой прибыли в несколько пунктов. Как правило, сделка при этом совершается в небольшой промежуток времени. Скальпинг наиболее распространён на срочных рынках (рынках производных финансовых инструментов). Следствием скальпинга является заключение большого количества сделок, каждая из которых приносит небольшой результат (небольшие прибыль или убыток).

Описание данных и архитектуры нейросети

В качестве исходной выборки использовались открытые данные с сайта МосБиржи (<https://www.moex.com/>) по акциям YNDX (Yandex CIA) за две недели торгов с 5 октября по 18 октября 2018 года. Выбор был обусловлен слабой информационной активностью вокруг компании «Яндекс» в данный промежуток времени, что должно было обеспечить пониженное влияние инфоповодов и повышенное влияние внутренних зависимостей рынка на стоимость акций компании. Выборка содержала в себе временной ряд из 5216 наблюдений 5 показателей, связанных с изменением стоимости акций. Первые четыре показателя $x(t)$ (объем совершенных сделок, цена открытия, наименьшая цена, наибольшая цена) являлись входными переменными, в то время как пятый показатель – $y(t)$ (цена закрытия) – не только являлся прогнозируемой выходной переменной, но также и передавался на вход нейронной сети для вычисления следующего прогноза.

Решение задачи сводилось к разработке нейронной сети, предсказывающей цену закрытия акции, к выбору параметров сети и оценке эффективности ее прогнозов.

Поставленная задача является задачей регрессии, поэтому для ее решения была использована нейросеть NARX (*англ.* Nonlinear Autoregressive exogenous model – нелинейная авторегрессионная экзогенная модель). Для создания нейронной сети использовалась функция *narxnet* из пакета MATLAB [1]. Обучение производилось функциями *trainlm* (алгоритм Левенберга – Марквардта [2]) и *trainrp* (алгоритм Resilient backpropagation или метод эластичного распространения ошибок [3]), которые были выбраны из-за высокой скорости сходимости, несмотря на большие затраты памяти, чем у других функций обучения.

Выбор количества слоев для нейронной сети можно обосновать эвристическим правилом [4], гласящим, что количество связей в нейросети не должно превышать 10 % от размера выборки. Для данного исследования использовался вариант деления временного ряда на три последовательных блока (обучения, валидации и теста) в соотношении 70 : 15 : 15, и если учитывать размер только обучающей выборки, то с помощью несложных подсчетов можно получить, что количество связей в данной нейронной сети не должно превышать 522. Были выбраны и реализованы несколько архитектур нейросетей, которые удовлетворяют такому количеству связей: двухслойная с 22 нейронами в каждом слое – [22 22], трехслойная с 16 нейронами – [16 16 16] и четырехслойная с 13 нейронами – [13 13 13 13].

Каждая из этих архитектур была использована для реализации нейронных сетей, работающих в трех режимах работы – прогноз на один шаг в открытом цикле, прогноз по всей тестовой выборке в закрытом цикле и прогноз на несколько шагов в закрытом цикле. Примеры нейронной сети открытого и закрытого циклов представлены на рисунке 1.

Прогноз на один шаг вперед осуществлялся нейронной сетью открытого цикла. Открытый цикл подразумевает постоянное использование входных данных для прогнозирования. Причем NARX позволяет установить задержку отклика нейронов, и для проектируемой нейронной сети был оставлена стандартная предлагаемая задержка, равная 2 временным периодам. Это позволяет ограничить конечный отклик на входные данные.

Нейронная сеть закрытого цикла была получена из нейронной сети открытого цикла с помощью функции *closeloop* [1]. Закрытый цикл

подразумевает, что отклик нейронной сети с выхода напрямую передается на вход нейронной сети. Таким образом, он используется не только для получения выходных данных, но и для прогнозирования значения цены в следующий временной период. Этот прием позволяет повысить точность. В таком цикле было осуществлено прогнозирование по всей тестовой выборке с использованием только показателей первых двух временных периодов и последующих выходных откликов.

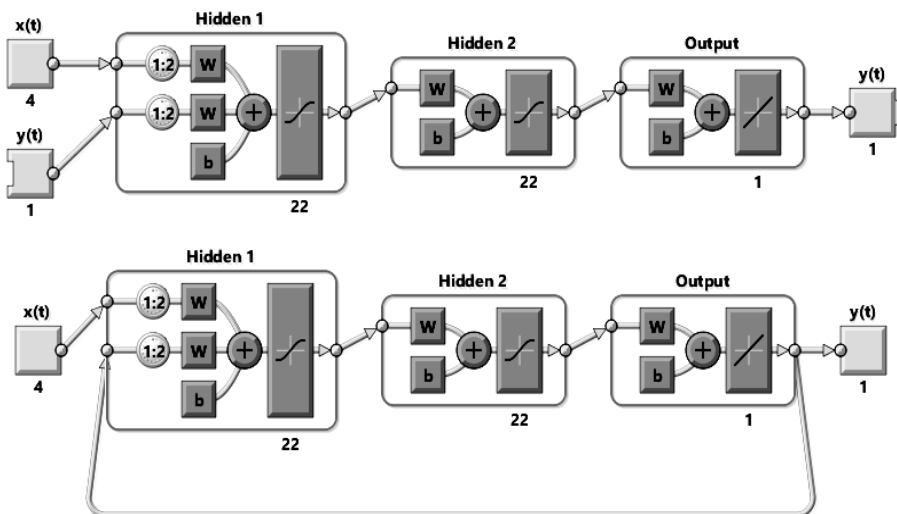


Рис. 1. Примеры нейронной сети открытого (сверху) и закрытого (снизу) циклов (Hidden 1, 2 – скрытые слои с 22 нейронами в каждом; Output – выходной слой)

Прогноз на несколько шагов вперед также использует закрытый цикл, что позволяет производить прогноз даже при отсутствии входных данных. Было произведено прогнозирование цены на 5 временных периодов, с использованием выходного отклика и однократным использованием входных данных.

Комбинации выбранных алгоритмов обучения, архитектур и режимов работы дало 6 нейронных сетей, протестированных в трех режимах работы. Качество прогнозирования каждой из них оценивалось по двум критериям: средней квадратической ошибке MSE и среднеквадратическому отклонению σ , определяемых по формулам:

$$MSE = \frac{1}{N} \sum_{t=1}^N (Y_t - \hat{Y}_t),$$

где N – количество наблюдений;

Y_t – значение временного ряда в момент времени t ;

\hat{Y}_t – предсказываемое значение временного ряда в момент времени t .

$$\sigma = \sqrt{MSE}.$$

Анализ результатов

В рамках проведенной работы была исследована точность прогнозов нейронной сети, примененной для предсказания цен акций на фондовой бирже. Для 6 нейронных сетей и 3 режимов работы были получены значения критериев качества (MSE и σ), представленные в таблице 1.

Таблица 1

Значения критериев качества

Режим работы	Архитектура нейронной сети											
	Алгоритм <i>trainlm</i>						Алгоритм <i>trainrp</i>					
	[22 22]		[16 16 16]		[13 13 13 13]		[22 22]		[16 16 16]		[13 13 13 13]	
	MSE	σ	MSE	σ	MSE	σ	MSE	σ	MSE	σ	MSE	σ
1	1388.7	37.3	234.9	15.3	100.2	10.0	785.9	28.0	1669.3	40.9	10655.1	103.2
2	55641.4	235.9	9224.7	96.1	21299.0	145.9	60339.9	245.6	6726.3	82.0	10035.9	100.2
3	398.9	20.0	272.9	16.5	545.9	23.4	3379.8	58.1	192.4	13.9	42.2	6.5

(1 – режим прогнозирования на один шаг в открытом цикле;

2 – режим прогнозирования по всей тестовой выборке в закрытом цикле;

3 – режим прогнозирования на несколько шагов в закрытом цикле).

Из таблицы видно, что среднеквадратическое отклонение (далее – СКО) предсказанной цены при прогнозировании по всей тестовой выборке в закрытом цикле значительно больше (в 6,3–14,6 раз) в пяти из шести вариантов нейронной сети, чем при прогнозировании на один шаг в открытом цикле, а в оставшемся варианте нейронной сети СКО лишь незначительно меньше (на 2,9 %). Это можно объяснить тем, что преимущества от повторного использования выходного отклика для прогнозирования показателей в последующих временных периодах

потерялись из-за большого объема тестовой выборки (15 % от наблюдений 5216 временных периодов за вычетом двух периодов, использованных в качестве изначальных входных данных), во время предсказания значений которой накопилась ошибка прогнозирования.

При учете того, что значения $y(t)$ лежали в пределах от 2100 до 2355, полученные результаты в случае прогнозирования в открытом цикле на один шаг удовлетворительные.

В случае прогнозирования нейронной сетью на 5 шагов вперед в закрытом цикле можно наблюдать ожидаемое снижение СКО по сравнению с прогнозированием по всей тестовой выборке в закрытом цикле. Малое количество прогнозов, использующих только выходной отклик, позволило в полной мере использовать преимущество от повторного использования отклика. В половине случаев (3 из 6 вариантов нейросети) многоступенчатое прогнозирование в закрытом цикле показало лучшее качество прогнозирования, чем прогнозирование на один шаг вперед в открытом цикле.

Из полученных результатов можно сделать вывод, что потенциально нейронная сеть может быть эффективна в случае предсказания цены акций, отличающихся относительной стабильностью, так как ее применение может сильно затруднять вызванные влиянием информационных поводов колебания цен, которые невозможно предсказать по значениям показателей в предыдущие временные периоды. Таким образом, при удовлетворительной точности и малой глубине прогнозирования нейронная сеть закрытого цикла, разработанная в данной исследовательской работе, могла бы использоваться только в качестве дополнительного инструмента при совершении торговых операций на малых временных интервалах с небольшими значениями риска и прибыли, что отвечает особенностям такого способа биржевой торговли, как скальпинг.

Разработанные нейронные сети предсказывают конкретную цену закрытия, однако на фондовой бирже получение дохода в первую очередь определяет рост или падение цены. Поэтому представляет интерес выделение задачи прогноза тренда и задачи прогноза величины изменения цены по отдельности. Предсказанное направление тренда позволит определить вкладываться в рост или падение цены, а спрогнозированное изменение цены акции – определить порог, после превышения которого заявку следует снимать с торгов. Поэтому при совершении сделок в небольшой промежуток времени по акциям, которые имеют выраженные частые колебания цены, вначале предсказание направления тренда, а после этого значения, на которое изменится цена, будет более удобным и

простым. Также такой подход может показать большую эффективность по сравнению с предсказанием только конечного значения цены акции.

Заключение

Результаты проведенного исследования показали, что нейронная сеть NARX закрытого цикла может применяться в качестве дополнительного инструмента краткосрочного прогнозирования при использовании стратегии внутридневных спекулятивных операций на фондовом рынке – скальпинге, но не при долгосрочном предсказании цен акций.

В дальнейшем планируется проверить эффективность раздельного прогнозирования направления движения цены акции и величины изменения цены.

Литература

1. *Mark Hudson Beale, Martin T. Hagan, Howard B. Demuth Neural Network Toolbox. User's Guide.* 2017.

2. *Пархоменко С. С., Леденёва Т. М.* Обучение нейронных сетей методом Левенберга – Марквардта в условиях большого количества данных // *Вестник ВГУ.* 2014. № 2. С. 98–104.

3. *Осоков Г. А., Филимонов А. В.* Динамическая оптимизация структуры персептронов. Дубна: ОИЯИ, 2002.

4. *Медведев В. С., Потемкин В. Г.* Нейронные сети. MATLAB 6. М.: ДИАЛОГ-МИФИ, 2002. 496 с.

5. *Клячкин В. Н., Кувайскова Ю. Е., Алексеева В. А.* Статистические методы анализа данных. М.: Финансы и статистика, 2016. 240 с.

6. *Санталов А. А.* Диагностика технического состояния системы водоочистки с использованием нейронных сетей // *Прикладная математика и информатика: современные исследования в области естественных и технических наук. Материалы IV-й научно-практической Международной конференции (школы-семинара) молодых ученых: в 2 частях.* 2018. – Издательство: Качалин Александр Васильевич. С. 237–242.

7. *Санталов А. А., Жуков Д. А.* Диагностика технического состояния системы с применением нейросетевых методов // В сборнике: *Перспективные информационные технологии (ПИТ 2018). Труды Международной научно-технической конференции / Под редакцией С. А. Прохорова.* 2018. С. 202–205.

**ЭКСПЕРИМЕНТАЛЬНАЯ ПРОВЕРКА
СРАВНИТЕЛЬНОЙ ЭФФЕКТИВНОСТИ ТРАДИЦИОННЫХ
И КОМПОЗИТНЫХ МЕТОДОВ НЕЯВНОГО ПЕРЕБОРА
ПРИМЕНИТЕЛЬНО К ЗАДАЧАМ
С БУЛЕВЫМИ ПЕРЕМЕННЫМИ**

Гроппен В. О., д-р техн. наук, профессор, groppen@mail.ru
Датиев А. А., канд. техн. наук, профессор, atsamaz.datieff@yandex.ru
Берко А. А., магистрант, aleksej-berko@yandex.ru
Буймистров В. А., магистрант, vovabmail@gmail.com
Северо-Кавказский горно-металлургический институт
(государственный технологический университет),
г. Владикавказ, Россия

Аннотация. В работе приведены результаты экспериментальной оценки эффективности композитных версий алгоритмов неявного перебора применительно к экстремальным задачам с булевыми переменными. Получены аналитические зависимости, позволяющие оценить и прогнозировать выигрыш во времени поиска решения этими алгоритмами по сравнению с соответствующими алгоритмами-предшественниками, осуществляющими неявный перебор.

Ключевые слова: алгоритм, неявный перебор, время счета, экспериментальная проверка, эффективность, динамическое программирование, методы типа ветвей и границ.

***EXPERIMENTAL VERIFICATION OF THE COMPARATIVE
EFFECTIVENESS OF TRADITIONAL AND COMPOSITE IMPLICIT
ENUMERATION METHODS APPLIED TO PROBLEMS
WITH BOOLEAN VARIABLES***

Groppen V. O., Datiev A. A., Berco A. A., Booimistrov V. A.,

Abstract. The paper presents the results of an experimental evaluation of the efficiency of composite versions of implicit enumeration algorithms applied to extreme problems with Boolean variables. Analytical dependences are obtained that allow one to estimate and predict the time gain of solutions search by these algorithms in comparison with the corresponding parent-algorithms performing implicit enumeration.

Keywords: algorithm, implicit iteration, computation time, experimental verification, efficiency, dynamic programming, branches and bound type methods.

Введение

Ниже приведены результаты экспериментальной оценки эффективности композитных версий алгоритмов неявного перебора применительно к экстремальным задачам с булевыми переменными:

$$\begin{cases} F = \sum C_i z_i \rightarrow \max(\min); \\ \forall j: \sum_i b_{i,j} z_i Q_j a_j; \\ \forall j: Q_j \in \{\leq, \geq, >, <, =\}; \\ \forall i: z_i = 1, 0, \end{cases} \quad (1)$$

где вектор $Z = \{z_1, z_2, \dots, z_n\}$;

$\forall i, C_i, b_{i,j}, a_j$ – константы, причем далее полагаем, что все константы и коэффициенты неотрицательны.

Поскольку эти исследования являются продолжением и развитием исследований, приведенных в [1–3] применительно к задаче о ранце, ниже используются обозначения, определения, методы вычисления оценок и отсекающие «неперспективных» планов, принятые в вышеназванных изданиях. Далее, в разделах 2–4, применительно к решению задачи (1), приводятся оценки эффективности композитных версий методов типа ветвей и границ и динамического программирования по отношению к их «классическим» версиям [3–6]. При этом, чтобы исключить влияние на эти оценки технических характеристик использованных аппаратных средств (таких, как тактовая частота процессора, объем оперативной памяти, пропускная способность общей шины и т. п.), оценивался выигрыш во времени поиска решения, вызванный применением композитных алгоритмов по сравнению с использованием соответствующих «классических» процедур на тех же аппаратных средствах и при решении тех же задач.

Описание использованных алгоритмов вычисления оценок и отсекающих неперспективных направлений поиска на дереве ветвлений приводятся далее.

Методы вычисления оценок и отсеечения неперспективных направлений

Верхняя оценка величины F в (1) при условии, что в базис введено подмножество

$$Z' \in Z$$

переменных, значениям которых соответствует вершина x_k дерева поиска, ниже определяется выражением:

$$\Delta(x_k) = \Delta(Z') = \begin{cases} \sum_{i \in I_1(x_k)} C_i z_i + \sum_{j \in I(Z) \setminus I_1(x_k)} C_j, & \text{если все условия выполнены;} \\ -\infty \text{ при } F \rightarrow \max; +\infty \text{ при } F \rightarrow \min - & \text{в противном случае,} \end{cases} \quad (2)$$

где $I_1(x_k)$ обозначает множество индексов введенных в базис булевых переменных, значения которых определяются положением $x_k \in X$ на дереве поиска;

$I(Z)$ – множество индексов всех булевых переменных решаемой задачи.

Аналогично определяется нижняя оценка $\delta(I_1)$ величины F задачи (1), отвечающая вершине x_k дерева поиска:

$$\delta(x_k) = \begin{cases} \sum_{i \in I_1(x_k)} C_i z_i, & \text{если все условия (1) выполнены;} \\ -\infty, & \text{при } F \rightarrow \max; +\infty \text{ при } F \rightarrow \min - \text{ в противном случае.} \end{cases} \quad (3)$$

Еще одной характеристикой вершины x_k дерева поиска является вектор ресурсов $\mu(x_k) = \{\mu_1(x_k), \mu_2(x_k), \dots, \mu_m(x_k)\}$:

$$\forall j: \mu_j(x_k) = a_j - \sum_{i \in I_1(x_k)} b_{i,j} z_i. \quad (4)$$

Таким образом, каждой вершине x_k дерева поиска, построенного композитной версией алгоритма, реализующего неявный перебор векторов переменных задачи (1), можно поставить в соответствие вектор $V(x_k) = \{\Delta(x_k), \delta(x_k), \mu(x_k)\}$, опираясь на который можно сформулировать правило отсеечения подмножеств «неперспективных» векторов переменных. Так, если на дереве поиска существуют две вершины x_k и x_p , такие, что справедливо хотя бы одно из приводимых ниже условий а), б):

$$а) \begin{cases} I(x_k) = I(x_p); \\ F \rightarrow \min; \\ \forall j : \mu_j(x_k) \leq \mu_j(x_p); \\ \delta(x_k) \geq \delta(x_p), \end{cases} \quad (5)$$

или

$$б) \begin{cases} I(x_k) = I(x_p); \\ F \rightarrow \max; \\ \forall j : \mu_j(x_k) \leq \mu_j(x_p); \\ \Delta(x_k) \leq \Delta(x_p), \end{cases} \quad (6)$$

то подмножество векторов переменных, отвечающих вершине x_k , можно исключить из дальнейшего рассмотрения.

Далее описания всех алгоритмов интерпретируются как построение дерева ветвлений, причем экспериментальная проверка их эффективности осуществлялась по следующей схеме:

а) число переменных n пробегает ряд значений от n_{\min} до n_{\max} ;

б) при каждом фиксированном n число ограничений m меняется в диапазоне от 1 до m_{\max} ;

в) при каждом фиксированном сочетании n и m методом Монте-Карло генерируются коэффициенты и константы для десяти задач, каждая из которых решается традиционной реализацией одного из методов неявного перебора и его композитной модификацией;

г) для каждой фиксированной комбинации n и m фиксируется минимальное, максимальное и среднее время поиска решения каждым алгоритмом;

д) по экспериментальным данным строятся аналитические зависимости верхней границы, нижней границы и среднего времени поиска решения от величин n и m .

Отношения этих времен, соответствующих традиционной и композитной реализации метода неявного перебора, позволяют судить об относительной эффективности последней.

Методы типа ветвей и границ, осуществляющие фронтальный спуск по дереву ветвлений

Алгоритм 2.1

1. На множестве висячих вершин $X_1 \subseteq X$ построенной части дерева ветвлений $G(X, U)$ выбирается вершина x_j с наилучшей оценкой.

Если это осуществляется на первой итерации, то такой вершиной *a priori* считается корневая вершина дерева.

2. Если выбранной вершине отвечает равенство $I_1 = I$, то перейти к шагу 5, в противном случае – к следующему шагу.

3. Ветвление осуществляется из выбранной на шаге 1 последней итерации вершины x_j . Новое множество висячих вершин дерева вновь обозначаем X_1 .

4. Вычисляются оценки, отвечающие висячим вершинам построенного на предыдущем шаге «куста». Для этого можно воспользоваться (2), если целевая функция системы (1) является максимизируемой и (3) – если целевая функция минимизируется. Перейти к шагу 1.

5. Алгоритм закончен. Вектор переменных, соответствующий выбранной вершине, является оптимальным.

Аналогичное описание композитной версии этого алгоритма приводится ниже.

Алгоритм 2.2

1. На множестве не вычеркнутых висячих вершин построенной части дерева ветвлений $G(X, U)$ выбирается вершина x_j с «наилучшей» первой компонентой вектора $V(x_j)$. Если это осуществляется на первой итерации, то такой вершиной *a priori* считается корневая вершина дерева.

2. Если выбранной вершине отвечает равенство $I_1 = I$, то перейти к шагу 6, в противном случае – к следующему шагу.

3. Ветвление осуществляется из выбранной на шаге 1 последней итерации вершины x_j . Новое множество висячих вершин дерева вновь обозначаем X_3^T .

4. Для каждой висячей вершины построенного на предыдущем шаге «куста» x_j формируется вектор $V(x_j)$. Для формирования первой компоненты этого вектора используется процедура (2), для определения второй компоненты вектора $V(x_j)$ используется (3), остальные компоненты вычисляются по формуле (4).

5. Если на множестве висячих вершин существует вершина x_k , для которой справедливо хотя бы одно из условий (5) или (6), то эта вершина вычеркивается. Перейти к шагу 1.

6. Алгоритм закончен. Вектор переменных, соответствующий выбранной вершине, является оптимальным.

Максимальная $\eta_{2\max}$, минимальная $\eta_{2\min}$ и средняя величина выигрыша η_2 во времени счета алгоритмом 2.2 по сравнению с алгоритмом 2.1 определяется эмпирическими зависимостями вида ($4 \leq n \leq 25$; $1 \leq m \leq 15$):

$$\eta_{2\max} \approx 8,08 \cdot 10^{-12} \cdot m^{8,24} \cdot \exp(0,77 \cdot n); \quad (7)$$

$$\eta_{2\min} \approx 0,00926 \cdot 1,2^m \cdot \exp(0,366 \cdot n \cdot m^{-0,45}); \quad (8)$$

$$\eta_2 \approx 5,68 \cdot 10^{-6} \cdot 1,7^m \cdot \exp(0,80 \cdot n \cdot m^{-0,5}). \quad (9)$$

Коэффициент корреляции ρ_3 применительно к зависимостям (7)–(9) находится в диапазоне 0,24–0,44.

Модифицированные методы типа ветвей и границ, осуществляющие фронтальный спуск по дереву ветвлений

Алгоритм 3.1

1. В базис вводится единовременно $g = \text{entire}[0,5(n + 1)]$ переменных, что соответствует построению первых g ярусов дерева ветвлений и появлению множества $X_1 \subseteq X, |X_1| = 2^g$, g – висячих вершин. Перейти к шагу 5.

2. На множестве висячих вершин $X_1 \subseteq X$ построенной части дерева ветвлений $G(X, U)$ выбирается вершина x_j с наилучшей оценкой.

3. Если выбранной вершине отвечает равенство $I_1 = I$, то перейти к шагу 6, в противном случае – к следующему шагу.

4. Ветвление осуществляется из выбранной на шаге 2 последней итерации вершины x_j . Новое множество висячих вершин дерева вновь обозначаем X_1 .

5. Вычисляются оценки, отвечающие висячим вершинам построенного на предыдущем шаге «куста». Для этого можно воспользоваться (2), если целевая функция системы (1) является максимизируемой, и (3) – если целевая функция минимизируется. Перейти к шагу 2.

6. Алгоритм закончен. Вектор переменных, соответствующий выбранной вершине, является оптимальным.

Композитная версия этого алгоритма отличается наличием процедуры отсечения вершин дерева ветвлений, удовлетворяющих условиям (5) или (6).

Алгоритм 3.2

1. В базис вводится одновременно $g = \text{entire}[0,5(n+1)]$ переменных, что соответствует построению первых g ярусов дерева ветвлений и появлению множества $X_1 \subseteq X, |X_1| = 2^g$, висячих вершин. Перейти к шагу 5.

2. На множестве не вычеркнутых висячих вершин $X_1 \subseteq X$ построенной части дерева ветвлений $G(X, U)$ выбирается вершина x_j с наилучшей оценкой.

3. Если выбранной вершине отвечает равенство $I_1 = I$, то перейти к шагу 6, в противном случае – к следующему шагу.

4. Ветвление осуществляется из выбранной на шаге 2 последней итерации вершины x_j . Новое множество висячих вершин дерева вновь обозначаем X_1 .

5. Вычисляются оценки и ресурсы, отвечающие висячим вершинам построенного на предыдущем шаге «куста». В первом случае для этого можно воспользоваться (2), если целевая функция системы (1) является максимизируемой, и (3) – если целевая функция минимизируется, во втором – системой (4). Если среди висячих вершин существует вершина x_k , удовлетворяющая условию (5) или (6), то она вычеркивается. Перейти к шагу 2.

6. Алгоритм закончен. Вектор переменных, соответствующий выбранной вершине, является оптимальным.

Максимальная $\eta_{3\max}$, минимальная $\eta_{3\min}$ и средняя величина выигрыша η_3 во времени счета алгоритмом 3.2 по сравнению с алгоритмом 3.1 определяется эмпирическими зависимостями вида ($4 \leq n \leq 25$; $1 \leq m \leq 15$):

$$\eta_{3\max} \approx (-1,84 \cdot 10^{-3} \cdot m^2 + 0,0249 \cdot m + 0,5) \cdot \exp\{n(0,0029 \cdot m^2 - 0,05 \cdot m + 0,37)\}; \quad (10)$$

$$\eta_{3\min} \approx (-0,007 \cdot m^2 + 0,09 \cdot m + 0,5) \cdot \exp\{n \cdot (6,6 \cdot 10^{-4} \cdot m^2 - 0,01 \cdot m - 0,01)\}; \quad (11)$$

$$\eta_3 \approx (-0,001 \cdot m^2 + 0,036 \cdot m + 0,33) \cdot \exp\{n \cdot (18 \cdot 10^{-4} \cdot m^2 - 0,038 \cdot m + 0,26)\}; \quad (12)$$

Коэффициент корреляции ρ_3 применительно к зависимостям (10)–(12) находится в диапазоне 0,42–0,55.

Динамическое программирование

Содержательно решение задачи (1) с помощью динамического программирования может быть проиллюстрировано последовательным,

ярус за ярусом, построением дерева ветвлений, стратегия которого существенно отличается от стратегии, описанной в приведенных выше алгоритмах.

Алгоритм 4.1

1. На множестве висячих и не помеченных вершин $X' \subseteq X_2$ построенной части дерева ветвлений $G_2(X_2, U_2)$ выбирается вершина x_j . Если таковой вершины нет, то перейти к шагу 4. Если выбор осуществляется на первой итерации, то такой вершиной *a priori* считается корневая вершина дерева.

2. Осуществляется ветвление из выбранной на шаге 1 последней итерации вершины x_j и вычисляются компоненты каждого вектора $V(x_k)$, отвечающего всем висячим вершинам построенного «куста». Для определения первой компоненты этого вектора используется (3), остальные компоненты вычисляются в соответствии с (4).

3. Все вершины куста, построенного на шаге 2 последней итерации, помечаются, перейти к шагу 1, иначе к следующему шагу (4).

4. Убираются все пометки подмножества висячих вершин дерева ветвлений.

5. Если на множестве висячих вершин существует вершина x_k , для которой справедливо хотя бы одно из условий (5) или (6), то эта вершина вычеркивается.

6. Если число введенных в базис переменных висячих вершин дерева ветвлений равно числу переменных решаемой задачи, то перейти к следующему шагу, в противном случае перейти к шагу 1.

7. На множестве не вычеркнутых висячих вершин выбирается вершина x_k с наилучшей первой компонентой соответствующего ей вектора $V(x_k)$.

8. Алгоритм закончен. Вектор переменных Z , соответствующий выбранной на шаге 7 вершине x_k , является оптимальным.

Композитная версия алгоритма 4.1 отличается от него следующими параметрами:

1. Каждой вершине x_j дерева ветвлений ставится в соответствие вектор $V(x_j)$, аналогичный тому, который использовался в алгоритмах 2.2 и 3.2.

2. Если на i -й итерации i -у ярусу построенной части дерева ветвлений принадлежат две такие вершины x_j и x_q , у которых вторая компонента вектора $V(x_j)$ «лучше» первой компоненты вектора $V(x_q)$, то вершина x_q вычеркивается.

Экспериментальная зависимость величины максимального выигрыша во времени поиска решения η_4 при решении задачи размерности n ($2 < n < 31$) композитной версией алгоритма 4.1 по сравнению с этим алгоритмом определяется линейным выражением вида:

$$\eta_4 = 0,267 + 0,826 \cdot n, \text{ коэффициент корреляции } \rho_4 = 0,901.$$

Заключение

Экспериментально подтверждена высокая эффективность композитных версий методов неявного перебора, гарантирующих глобально оптимальные решения экстремальных задач с булевыми переменными. В ряде случаев величина выигрыша во времени счета превышала 10^2 .

Литература

1. *Гроппен В. О.* Композитные алгоритмы поиска глобально оптимальных решений экстремальных задач с булевыми переменными // Материалы семинара “ИТ-технологии: теория и практика”. Владикавказ: Изд-во СКГМИ (ГТУ), 2017. С. 21–38.

2. *Гроппен В. О., Датиев А. А., Пановская К. В.* Экспериментальный анализ эффективности композитных версий методов неявного перебора применительно к задаче о ранце // Материалы семинара “ИТ-технологии: теория и практика”. Владикавказ: Изд-во СКГМИ (ГТУ), 2018. С. 3–20.

3. *Датиев А. А.* Экспериментальный анализ эффективности композитной версии метода неявного перебора применительно к задаче о ранце // Сборник докладов XV-й Международной научно-технической конференции “ИТ-технологии: развитие и приложения”. Владикавказ. Изд-во «ВЕСТА», 2018. С. 271–276.

4. *Land A. H., Doig A. G.* An automatic method of solving discrete programming problems // *Econometrica*. 1960. Vol. 28. No. 3. Pp. 497–520.

5. *Marchand Hugues, Martin Alexander, Weismantel Robert, Wolsey Laurence.* Cutting planes in integer and mixed integer programming (PDF) // *Discrete Applied Mathematics*. 2002. № 123. Pp. 387–446.

6. *Bellman R.* The Theory of Dynamic Programming // The RAND Corporation. 1954.

**ЭКСПЕРИМЕНТАЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ
ИСПОЛЬЗОВАНИЯ ЭТАЛОНОВ ПРИ РЕШЕНИИ
МНОГОКРИТЕРИАЛЬНЫХ ОПТИМИЗАЦИОННЫХ ЗАДАЧ
С БУЛЕВЫМИ ПЕРЕМЕННЫМИ
МЕТОДАМИ ТИПА ВЕТВЕЙ И ГРАНИЦ**

Будаева А. А., канд. техн. наук, budalina@yandex.ru

Артамонов М. А., магистрант

Северо-Кавказский горно-металлургический институт

(государственный технологический университет),

г. Владикавказ, Россия

Аннотация. В работе приведены экспериментальные результаты оценки эффективности решения многокритериальных задач дискретной оптимизации с булевыми переменными совмещением метода эталонов с алгоритмами типа ветвей и границ.

Ключевые слова: эталон, метод ветвей и границ, многокритериальная оптимизация, булевы переменные.

***EXPERIMENTAL ANALYSIS OF THE EFFICIENCY
OF USING STANDARDS IN SOLVING MULTICRITERIA
OPTIMIZATION PROBLEMS WITH BOOLEAN
VARIABLE METHODS OF THE BRANCH AND BOUND TYPE***

Budaeva A. A., Artamonov M. A.

Abstract. The paper presents the experimental results of evaluating the effectiveness of solving multi-criteria discrete optimization problems with Boolean variables by combining the method of standards with algorithms of the branch and bound type.

Keywords: standard, branch and bound method, multi-criteria optimization, Boolean variables.

Введение

Широко известно применение эталонов как одного из методов решения многокритериальных задач [1, 2], а также их использование в рамках методов типа ветвей и границ – в этом случае термин «эталон»

заменяется словом «оценка». В [3–8] предлагается ряд новых технологий решения задач, охватывающих практически все области теории принятия решений, причем ядром, объединяющим предлагаемые подходы, является использование эталонов. Ниже эффективность одной из этих технологий – решение многокритериальных задач дискретной оптимизации с помощью комбинации метода эталонов и одного из алгоритмов типа ветвей и границ проверяется экспериментально. Цель – экспериментально оценить характер зависимостей затрат времени счета и объема используемой оперативной памяти от размерности решаемой задачи. При этом используются следующие обозначения и определения.

Обозначения и определения

Далее используются следующие обозначения:

\vec{X}

– вектор переменных;

I – множество индексов переменных ($|I| = n$);

I_1 – множество индексов введенных в базис переменных ($|I_1| \leq |I|$);

$C_{j,i}$ – коэффициент j -го критерия, отвечающий i -й переменной;

X_k – множество значений, принимаемых k -й переменной;

\vec{F}_i

– i -й критерий ($i = 1, 2, \dots, n$);

$\vec{\varphi}_j$

– j -е ограничение;

K_i – величина, соответствующая наилучшему значению i -го критерия.

Основные положения

Далее будем полагать, что решаются задачи с булевыми переменными, i -й критерий является аддитивным с неотрицательными коэффициентами, $F_i(\vec{X}) \rightarrow \max$ и $|I_1| \geq 1$, а оценка его отклонения от величины K_i , равная $\beta_i(I_1)$ определяется выражением:

$$\beta_i(I_1) = \left[K_i - \left(\sum_{j \in I_1} C_{i,j} x_{i,j} + \sum_{j \in I \setminus I_1} C_{i,j} \right) \right]^2. \quad (1)$$

Аналогично, если в задаче с булевыми переменными i -й критерий является аддитивным с неотрицательными коэффициентами,

$F_i(\vec{X}) \rightarrow \min$ и $|I_1| \geq 1$, то далее полагаем, что оценка его отклонения от величины K_i определяется выражением:

$$\beta_i(I_1) = \left[K_i - \sum_{j \in I_1} C_{i,j} x_{i,j} \right]^2. \quad (2)$$

Далее формальная постановка многокритериальной задачи с булевыми переменными имеет вид:

$$\left\{ \begin{array}{l} \forall i : \sum_j C_{i,j} x_{i,j} \rightarrow \max (\min); \\ \forall q : \varphi_q(\vec{X}) \leq b_q; \\ \forall k : x_k \in \{0,1\}. \end{array} \right. \quad (3)$$

Тогда величина эталона K^i определяется решением однокритериальной задачи вида:

$$\left\{ \begin{array}{l} K_i = \sum_j C_{i,j} x_{i,j} \rightarrow \max (\min); \\ \forall q : \varphi_q(\vec{X}) \leq b_q; \\ \forall k : x_k \in \{0,1\}. \end{array} \right. \quad (4)$$

Решая задачу (2) применительно к каждому критерию, получаем вектор

$$\vec{K} = \{K_1, K_2, \dots, K_n\}, \quad (5)$$

которому в n -мерном пространстве критериев соответствует «идеальный» объект «а».

Если критерии однородны, то есть измерены в совпадающих шкалах, то дистанция между объектами определяется, как расстояние в евклидовом пространстве, т. е. как квадратный корень из суммы квадратов разностей [5]. Таким образом, задача (3) заменяется системой вида:

$$\left\{ \begin{array}{l} \Delta = \sqrt{\sum_{i=1}^n [K_i - F_i(\vec{X})]^2} \rightarrow \min; \\ \forall j: \varphi_j(\vec{X}) \leq b_j; \\ \forall k: x_k \in \{0,1\}; \quad \vec{X} = \{x_1, x_2, \dots, x_m\}. \end{array} \right. \quad (6)$$

Можно показать, что вектор переменных задачи (6) является Парето-оптимальным решением задачи (3) [6]. Более того, если нормировать значения целевых функций системы (3) таким образом, чтобы все они были безразмерны и заключены в диапазоне $\{0 - 1\}$:

$$\forall i: f_i(\vec{X}) = \frac{F_i(\vec{X}) - F_{i \min}(\vec{X})}{F_{i \max}(\vec{X}) - F_{i \min}(\vec{X})}, \quad (7)$$

то, подставляя левые части системы (7) в (6) вместо $F_i(\vec{X})$, легко убедиться [5], что вектор переменных полученной системы тоже является оптимальным по Парето решением системы (1).

Так как оптимальные векторы переменных в системе (8):

$$\left\{ \begin{array}{l} \delta = \Delta^2 = \sum_{i=1}^n [K_i - \sum_j C_{i,j} x_{i,j}]^2 \rightarrow \min; \\ \forall q: \varphi_q(\vec{X}) \leq b_q; \\ \forall k: x_k \in \{0,1\}; \quad \vec{X} = \{x_1, x_2, \dots, x_m\}, \end{array} \right. \quad (8)$$

и в системе (6) совпадают, далее для решения системы (3) используется (8). Для того, чтобы доказать применимость методов типа ветвей и границ для решения системы (8), достаточно показать, что по мере спуска по дереву ветвлений оценка величины δ , определяемая компонентами (9) и (10), не улучшается. Справедлива следующая теорема (доказательство приводится в [8]), использованная ниже в методе типа ветвей и границ при наборе статистики:

Если в рамках принятых выше допущений $I_1 \subseteq I_2$, то оценка $\beta_j(I_1)$ «лучше», чем оценка $\beta_j(I_2)$.

Постановка и результаты экспериментов

В ходе экспериментов использовался ПК со следующими характеристиками (табл. 1):

Таблица 1

Характеристики ПК

Производитель процессора	AMD
Тип процессора	FX 8350
Количество ядер процессора	8
Макс. такт. частота	4 ГГц
Кэш память	6 Мб
Оперативная память (RAM)	8 Гб
Тип памяти	DDR3
Частота памяти	1200 МГц
Производитель видеокарты	NVIDIA
Графический контроллер	GeForce GTX 760
Жесткий диск (HDD)	1024 Гб
Операционная система	Windows 10 (64 bit)

Целями эксперимента являлись зависимости времени счета и объема используемой оперативной памяти от числа переменных и числа целевых функций системы (3). Переменные менялись в диапазоне 2–20, число целевых функций – в диапазоне 1–10. Для каждого сочетания количества переменных и целевых функций проводилось по 10 экспериментов и из них определялись средние, максимальные и минимальные величины времени счета и объема используемой памяти, соответствующие этому сочетанию. Ниже, на рисунках 1 и 2, представлены зависимости среднего времени (в мс) решения задачи (3) и среднего объема затраченной при этом оперативной памяти (в Кб) от числа переменных и количества целевых функций.

Спецификой использованного метода типа ветвей и границ являлись процедуры спуска по дереву поиска и вычисления оценок. В качестве первой использовался фронтальный спуск по дереву ветвлений, в качестве второго – сумма коэффициентов целевой функции, отвечающих введенным в базис переменным.

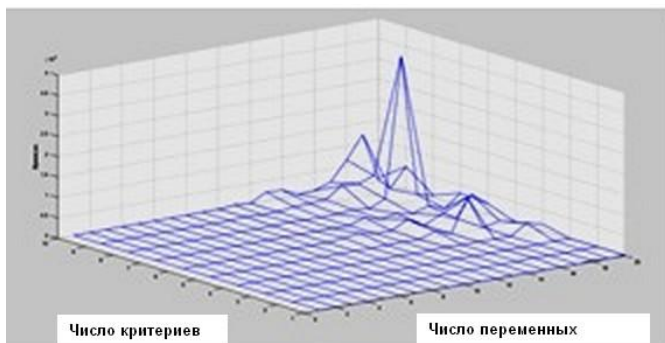


Рис. 1. Экспериментальная зависимость среднего времени счета от размерности задачи

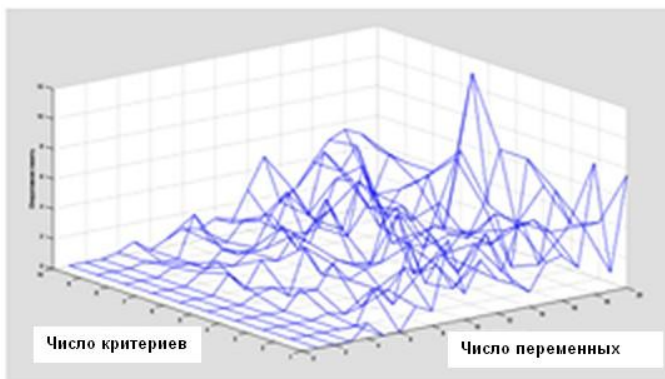


Рис. 2. Экспериментальная зависимость среднего объема использованной RAM от размерности задачи

Сравнивая эти графики можно допустить, что они описываются различными аналитическими зависимостями: зависимость времени поиска решения от размерности задачи (рис. 1) ближе к экспоненциальной зависимости, в то время как зависимость объема использованной RAM от тех же параметров (рис. 2) ближе к полиномиальной.

Литература

1. Podinovski V. V., Podinovskaya O. V. New Multicriterial Decision Rules in Criteria Importance Theory // ISSN 1064-5624. Dollady Mathematics. 2013. Vol. 88. № 1. Pp. 486–488.

2. *Joseph J. Moder, Salah E. Elmaghraby*. Handbook of Operations Research. Vol. 1: Foundations and Fundamentals. Van Nostrand Reinhold Company. 1978. Pp. 513–548.

3. *Гроппен В. О.* Принципы принятия решений с помощью эталонов // Автоматика и телемеханика. 2006. № 4. С. 167–184.

4. *Будаева А. А., Гроппен В. О.* Выбор оптимальной технологии ранжирования // Устойчивое развитие горных территорий. 2014. № 3. С. 3–7.

5. *Groppen V. O.* New Solution Principle for Multi-criteria Problems Based on Comparison Standards: Models, Algorithms, Applications // Applications to Industrial and Societal Problems. CIMNE, Barcelona, Spain. 2008. Pp. 201–209.

6. *Groppen V. O.* New Solution Principles of Multi-Criteria Problems Based on Comparison Standards. www.arxiv.org/ftp/math/papers/0501/0501357.pdf. 2004.

7. *Будаева А. А.* Использование метода эталонов для решения задач дискретной многокритериальной оптимизации // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2015. Т. 15. № 1. С. 22–27.

8. *Гроппен В. О, Будаева А. А.* Эталонные как уникальный инструмент постановки и решения задач теории принятия решений // Материалы семинара «IT-технологии: теория и практика». Владикавказ: Изд-во СКГМИ (ГТУ), 2017. С. 3–20.

УДК 004.891.2

СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ, ПРЕДНАЗНАЧЕННАЯ ДЛЯ ФОРМИРОВАНИЯ НЕЧЕТКОЙ ОЦЕНКИ ДЕЯТЕЛЬНОСТИ КУРАТОРА ВУЗА

Носова Т. Н., старший преподаватель, ntn.mgtu@mail.ru
Магнитогорский государственный технический университет
им. Г. И. Носова,
г. Магнитогорск, Челябинская обл., Россия

Аннотация. В работе рассматривается процесс разработки, моделирования и кодирования системы поддержки принятия решения при формировании экспертных нечетких заключений на базе конкретных числовых показателей при оценке качества выполняемых работ персоналом организации.

Ключевые слова: система поддержки принятия решения, лингвистические переменные, фазификация, функции принадлежности, степень принадлежности, правила логического вывода, оценка деятельности персонала.

***DECISION SUPPORT SYSTEM, INTENDED FOR FORMING
A FUZZY ASSESSMENT
OF THE UNIVERSITY CURATORS ACTIVITY***

Nosova T. N.

Annotation. *The paper discusses the process of developing, modeling and coding a decision support system in the formation of expert fuzzy conclusions based on specific numerical indicators in assessing the quality of work performed by the staff of the organization.*

Keywords: *decision support system, linguistic variables, phasification, membership functions, degree of belonging, inference rules, personnel performance evaluation.*

Введение

В настоящее время математический аппарат нечеткой логики широко используется в системах принятия решений в условиях нечеткости входных и выходных данных. Методы нечеткой логики нашли многочисленные применения в задачах моделирования и управления технологическими процессами, в области управления персоналом и ряде других областей. Математическая теория нечетких множеств позволяет формализовать нечеткие понятия и знания, оперировать ими и осуществлять нечеткие выводы [1].

Не до конца определенные понятия постоянно присутствуют в повседневной человеческой жизни: расстояние до машины – «невелико», человек – «достаточно молодой», оценка результатов профессиональной деятельности – «высокая» и т. п. Так, при оценке результатов профессиональной деятельности, с определенной долей достоверности, можно признать результаты работы «хорошими» или «плохими». Экспериментально показано [2–4], что использование нечеткой логики в управлении (нечеткое управление) дает лучшие результаты по сравнению с традиционными алгоритмами управления. Нечеткая логика позволяет более точно представить человеческий подход к решению многих прикладных задач; системы нечеткой логики легко адап-

тируются к изменяющимся условиям и достаточно просто проектируются.

В представляемой автором работе рассматривается процесс разработки, моделирования и кодирования системы поддержки принятия решения при формировании экспертных нечетких заключений на базе конкретных числовых показателей при оценке качества выполняемых работ персоналом организации. При формировании итоговых оценок эффективности работы персонала эксперту приходится учитывать большое количество факторов: знание предмета, способность к получению новых знаний (повышение профессиональной квалификации), научная деятельность, публикационная активность, умение работать в командных проектах, генерировать новые идеи, умение обосновывать свои выводы и т. д. Очевидно, что не все эти качества можно измерить и оценить в конкретных числовых показателях. Корпоративные документы, регламентирующие процесс формирования итоговой оценки той или иной деятельности, сформулированы в терминах «лингвистических» (нечетких) понятий. Например: признать результаты работы сотрудника – удовлетворительными, хорошими, отличными; вынести – поощрение или порицание; назначить – премию или штраф, повысить или снизить оплату; выдать управляющие инструкции и т. п. Это создает естественные предпосылки для использования аппарата нечеткой логики в целях формализации правил и построения системы поддержки принятия решений [5].

Постановка задачи

Наиболее продуктивным подходом к решению задачи управления в условиях нечеткости входных и/или выходных данных является использование теории нечетких множеств и нечеткой логики.

Разработанное программное приложение поддержки принятия решения тестировалось на примере предметной области «Оценка качества работы куратора студенческой группы». Работа преподавателя-куратора оценивается по нескольким показателям, каждый из которых в разной степени влияет на общий рейтинг эффективности. Итоговая оценка работы куратора может принять один из вариантов нечеткой качественной характеристики – результаты работы признать: «неудовлетворительными», «удовлетворительными», «хорошими», «отличными» и в соответствии с этим сформировать дальнейшие поощрительные или мотивирующие управляющие воздействия. В качестве входных факторов, влияющих на итоговую оценку деятельности кура-

тора, выбраны характеристики, подлежащие количественному измерению:

- количество проведенных кураторских часов в семестре;
- количество совместных мероприятий со студентами за семестр;
- количество посещений общежития за семестр;
- процент потерь контингента;
- средняя оценка деятельности куратора по опросам обучаемых.

Последний показатель является усредненной и наиболее субъективной оценкой, тем не менее, позволяющей учесть мнение студентов и организовать обратную связь. При математическом описании решения задачи, перечисленные выше факторы влияния на итоговый показатель эффективности, использованы в качестве перечня лингвистических переменных, термами которых является множество возможных значений: {«высокий», «средний», «низкий»} – уровень оценки показателя. В процессе нечеткого логического вывода каждому терм-значению ставится в соответствие значение оценочного коэффициента с учетом степени принадлежности этого числового показателя к тому или иному терму. Затем, формируется нечеткая итоговая оценка деятельности куратора на базе средневзвешенного значения всех оценочных коэффициентов.

Используемый математический аппарат и моделирование предметной области

С точки зрения классической теории множеств элемент данных либо «присутствует» в наборе, либо «отсутствует» в нем – без каких-либо двусмысленностей и вариантов интерпретации. Теория нечетких множеств, напротив, допускает для элемента данных: возможность «иногда» присутствовать или «иногда» отсутствовать в наборе до определенной степени. В классической теории множество определяется как набор уникальных элементов, для которых характеристическая функция принимает одно из двух дискретных значений: 0 (элемент не принадлежит множеству) или 1 (элемент принадлежит множеству). Нечеткое множество определяется функцией принадлежности, значением которой может быть любое действительное число в диапазоне от 0 до 1. Промежуточные значения означают принадлежность множеству с какой-то степенью достоверности. Напомним [6], что лингвистическая переменная – это числовая переменная, имеющая имя (давление, температура, количество и т.д.), с которой связаны лингвистиче-

ские описания. Лингвистическая переменная характеризуется набором свойств:

$$(X, T(X), U),$$

где X – название лингвистической переменной;

$T(X)$ обозначает терм-множество переменной X , т. е. множество названий лингвистических значений переменной X , причем каждое из таких значений является нечеткой переменной со значениями из универсального множества U .

Терм-множеством (term set) называется множество всех возможных значений, которые способна принимать лингвистическая переменная (давление – высокое/низкое, температура – низкая, средняя, высокая).

При определении нечеткого множества количество и характер нечетких переменных субъективны для каждой отдельной задачи [7].

Входным и выходным сигналам в нечеткой логической системе соответствуют логико-лингвистические переменные, значения которых определяются термами-множествами. База знаний нечеткой логической системы состоит из продукционных правил, определяющих зависимость между входными и выходными термами-множествами, и функций принадлежности, показывающих степень соответствия реальных величин понятиям, определяемых термами-множествами.

Традиционный алгоритм, применяемый при разработке нечеткой логической системы, следующий:

1. Фаззификация входных величин (перевод реальных значений в значение логико-лингвистических переменных).

2. Нечеткий логический вывод.

3. Дефаззификация – обратное преобразование нечеткого множества лингвистических выходных переменных в числовые значения, выполняемое после формирования нечеткого логического вывода (получение реальных значений выходных переменных).

Фаззификация заключается в переводе точных значений входных сигналов в значение логико-лингвистических переменных, используя для этого функции принадлежности [8].

Для всех лингвистических переменных рассматриваемого модельного примера определены атомарные термы: значение показателя «низкое», «среднее», «высокое», с границами, задающими поведение функций принадлежности.

Наборы свойств лингвистических переменных приведены в табл. 1.

Свойства предметной области «Деятельность куратора»

	X – названия лингвистических переменных	U – универсальное множество значений	T(X) – терм-множество		
			низкое	среднее	высокое
1	Количество кураторских часов в сем.	[0; 40]	10; 18	15; 19; 30; 33	27; 32; 40
2	Количество совместных мероприятий со студентами в сем.	[0; 20]	6; 10	8; 11; 15; 20	12; 17; 20
3	Количество посещений общежития в сем.	[0; 40]	10; 18	15; 19; 30; 33	27; 32; 40
4	Процент потерь контингента, %	[0; 50]	0,05; 0,13	0,06; 0,07; 0,14; 0,17	0,16; 0,3; 0,5
5	Средняя оценка студентов, %	[0; 1]	0,2; 0,35	0,2; 0,5; 0,75; 0,9	0,6; 0,85; 1

Приведенные в табл. 1. значения выбраны в результате диалога с преподавательским сообществом и руководством организации, а также могут корректироваться в зависимости от изменяющихся условий. Описываемый программный продукт настраивается на изменение количества и характера лингвистических переменных, так как для него эти показатели являются входными данными для обработки. В процессе фазсификации – преобразования числовых значений в степени принадлежности к нечеткому множеству – точные значения входных влияющих параметров переведены в значения (термы) логико-лингвистических переменных, с использованием функций принадлежности. Функция $\mu_A(x)$ для любого входного значения x связана со степенью его принадлежности к множеству A . Значение степени принадлежности находится в диапазоне $[0; 1]$. На рис. 1а, б приведены примеры функций принадлежности для входных параметров «Количество кураторских часов в сем.» и «Процент потерь контингента».

Для описания поведения терма «низкое количество» показателя выбрана z-образная функция принадлежности, формируемая по закону:

$$\mu(x) = \begin{cases} 1, & x \leq a; \\ \frac{b-x}{b-a}, & a < x \leq b; \\ 0, & x > b; \end{cases}$$

для термов «среднее» и «высокое количество» – трапециевидная функция принадлежности:

$$\mu(x) = \begin{cases} 0, & x \leq a; \\ \frac{x-a}{b-a}, & a < x \leq b; \\ 1, & b < x \leq c; \\ \frac{d-x}{d-c}, & c < x \leq d; \\ 0, & x > d. \end{cases}$$

Для оценивания конкретных характеристик разработана нечеткая модель оценивания по схеме Мамдани. Каждый фактор оценивания представлен лингвистической переменной, для которой определено терм-множество значений.

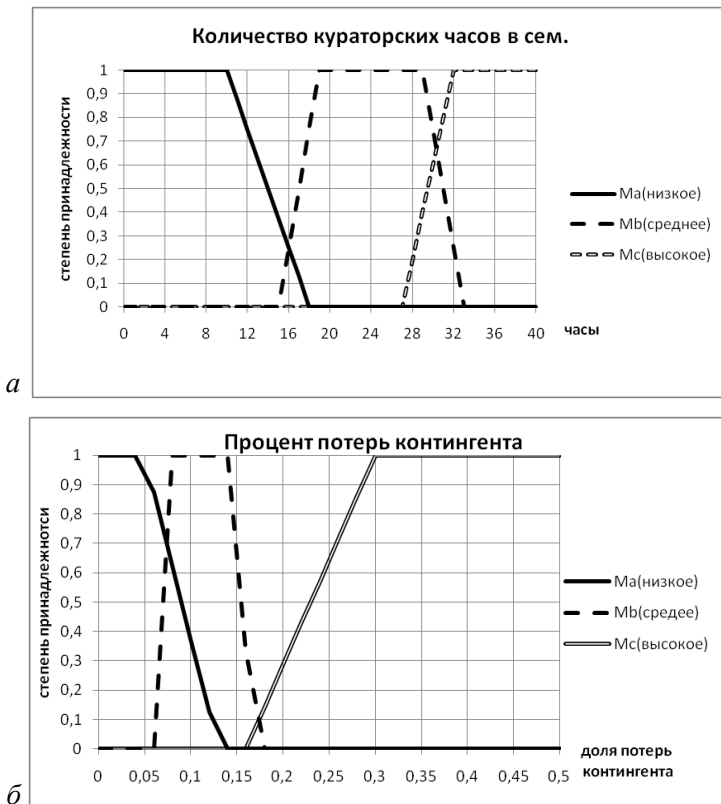


Рис. 1. Функции принадлежности входного параметра: а – «Количество кураторских часов в сем.» термам «низкое», «среднее», «высокое» значение; б – «Процент потерь контингента» термам «низкое», «среднее», «высокое» значение

Для построения функций принадлежности был выбран прямой метод. Он удобен при решении задач, для которых свойства физических величин могут быть измерены. При необходимости более точного определения функции принадлежности можно воспользоваться последующим анализом результатов для коррекции нечёткой модели.

Следует обратить внимание, что некоторые конкретные числовые значения входных влияющих параметров могут одновременно принадлежать разным термам с разной степенью достоверности. Например, значение «Количество кураторских часов», равное 16, принадлежит терму «низкое» со степенью достоверности, равной 0,14, и терму «среднее» со степенью достоверности, равной 0,25. В таких ситуациях использована операция объединения нечетких множеств.

Объединением двух нечетких множеств A и B называется некоторое нечеткое множество C , заданное на этом же универсуме X со следующей функцией принадлежности:

$$\mu_C(x) = \max\{\mu_A(x), \mu_B(x)\},$$

называемое также *max-объединением*.

На этапе нечеткого логического вывода сформулированы универсальные правила перевода термов логических переменных в значения оценивающего числового критерия:

$$\text{Критерий}_i = \mu(x) \cdot \begin{cases} 0,5, & \text{если значение – низкое,} \\ 0,8, & \text{если значение – среднее,} \\ 1, & \text{если значение – высокое.} \end{cases}$$

где $\mu(x)$ – степень принадлежности числового показателя к терму с учетом *max-объединения*.

Затем вычисляется итоговый показатель эффективности как средневзвешенное значение всех критериев с учетом весовых коэффициентов.

$$\begin{aligned} \text{Итоговый показатель эффективности} &= \\ &= \frac{\sum_{i=1}^n \text{критерий}_i \cdot \text{весовой коэффициент}_i}{\sum_{i=1}^n \text{весовой коэффициент}_i}, \end{aligned}$$

где n – количество входных параметров, влияющих на общую оценку.

Весовые коэффициенты на первом этапе могут быть приняты равными 1. В дальнейшем, их изменение может усиливать/ослаблять

влияние того или иного критерия на общую оценку эффективности деятельности сотрудников.

И наконец, на последнем этапе значения числового итогового показателя эффективности, принадлежащего интервалу $[0; 1]$, переводятся в значения итоговой нечеткой оценки со значениями: Работу куратора признать «плохой – вынести порицание», «удовлетворительной», «хорошей», «отличной – назначить премию» и т.п. со стандартной шкалой оценивания.

Заключение

На основе выше описанного алгоритма разработано программное приложение на языке C#. Созданный программный продукт предназначен для моделирования нечетких систем на алгоритмическом уровне в поведенческой прикладной области – в системах поддержки принятия решения при многофакторной оценке деятельности сотрудников учебных заведений (кураторов студенческих групп). При необходимости представляемый программный продукт легко адаптируется для любых предметных областей в задачах формирования разнообразных качественных показателей и итоговых оценочных заключений на базе многокритериальных исходных данных. Приложение может применяться в различных предметных областях, для формирования рейтингов производственной и образовательной деятельности при итоговой оценке эффективности работы персонала [9]. Программа может быть использована в качестве экспертной системы для оценки деятельности сотрудников различных профессиональных областей, а также при преподавании дисциплин «Управление персоналом», «Информационные системы», «Нечеткая логика и ее приложения» и т. п. [10].

Литература

1. *Пегат А.* Нечеткое моделирование и управление: пер. с англ. М.: Бином, 2009. 798 с.
2. *Абдурахманов Д. Б.* Методика интегральной оценки инвестиционных рисков строительного предприятия с нечеткой логикой обработки экспертных данных // Экономика строительства. 2014. № 4 (28). С. 34–39.
3. *Степанов В. И., Миргородская О. Ю.* Нечеткая логика – инструмент управления профессиональными рисками // Нормирование и оплата труда в промышленности. 2013. № 1. С. 64–68.

4. *Клименков Г. В., Кукор Б. Л.* Комплексная оценка процесса стратегического управления промышленностью региона // Вестник УГНТУ. Наука, образование, экономика. Серия: Экономика. 2017. № 3 (21). С. 42–49.

5. *Марценюк М. А., Поляков В. Б.* Нечеткий алгоритм многофакторной оценки рейтинга студента // Прикладная информатика. № 5 (53). С. 41–44.

6. Основы теории нечетких множеств [Электронный ресурс]. – Режим доступа: <https://www.intuit.ru/studies/courses/87/87/info>. Заглавие с экрана. (Дата обращения: 17.03.2019).

7. *Чернов В. Г.* Основы теории нечетких множеств: Учебное пособие / Владимирский государственный ун-т. Владимир: Изд-во Владимирского государственного ун-та, 2010. 96 с. ISBN 978-5-9984-0055-1.

8. Технология нечеткой логики ООО "ДинСофт" [Электронный ресурс]. – Режим доступа: <http://www.dynsoft.ru/contacts.php>. Заглавие с экрана. (Дата обращения: 17.03.2019).

9. *Носова Т. Н., Калугина О. Б.* Использование алгоритма битовых шкал для увеличения эффективности поисковых запросов, обрабатывающих данные с низкой избирательностью // Электротехнические системы и комплексы. 2018. № 1 (38). С. 63–67.

10. *Носова Т. Н.* Методологические аспекты изучения дисциплины «Информационные технологии. Базы данных» // Педагогическая информатика. 2017. № 2. С. 99–109.

УДК 004.89

ИСПОЛЬЗОВАНИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ПРИ СОЗДАНИИ СИСТЕМ РЕКОМЕНДАЦИЙ НА САЙТЕ

Рябова А. А., канд. техн. наук, доцент, alinaryabova@gmail.com

Куценко П. А., студент, zebartcoc@gmail.com

Российский экономический университет им. Г. В. Плеханова,

филиал в г. Пятигорске,

г. Пятигорск, Россия

Аннотация. Статья посвящена исследованию рекомендательных систем и алгоритмов выдачи результата. В ходе исследования был проанализирован алгоритм YouTube. Сделан вывод о том, что создание рекомендательных систем и применение искусственного интел-

лекта существенно упрощает жизнь обычным пользователям интернета.

Ключевые слова: система рекомендаций, искусственный интеллект, YouTube, алгоритм, коллаборативная фильтрация, Google Brain, TensorFlow.

THE USE OF ARTIFICIAL INTELLIGENCE IN CREATING SYSTEMS OF RECOMMENDATIONS ON THE SITE

Ryabova A. A., Kutcenko P. A.

Abstract. The article is devoted to the study of recommender systems and algorithms for issuing results. The study analyzed the YouTube algorithm. It is concluded that the creation of recommender systems and the use of artificial intelligence greatly simplifies the life of ordinary users of the Internet.

Keywords: recommendation system, artificial intelligence, YouTube, algorithm, collaborative filtering, Google Brain, TensorFlow.

Введение

Каждый день многие из нас что-то ищут в интернете. Некоторые ищут что-то целенаправленно, некоторые просто блуждают по сети. И те и другие преследуют определенную цель: найти то, что может им понравиться или пригодиться в дальнейшем.

Не так давно, еще в прошлом веке, о появлении новых товаров или услуг узнавали из почтовых рассылок, либо необходимо было посетить места продажи. Сейчас же все намного проще: этот процесс ускорился в десятки (а то и сотни) раз с появлением сначала телевидения, а затем и интернета. Какой фильм сейчас в тренде? Какие акции в магазинах? О чем пишет пресса? Об этом всем и не только мы узнаем с помощью специальных сервисов и веб-сайтов.

Хотелось бы уделить внимание одной интересной теме: применение искусственного интеллекта при создании систем рекомендаций на сайте. Почему это важно? Развитие искусственного интеллекта (далее по тексту – ИИ) дает много возможностей. Его применение в интернете существенно упрощает жизнь и помогает находить информацию намного быстрее.

Системы рекомендаций. Для чего и как работает?

Рекомендательные системы – это комплексы алгоритмов, программы и сервисы, основная задача которых – предсказать, какие объ-

екты (товары или услуги) будут интересны пользователю, имея информацию о его профиле либо иные данные. То есть задача такой системы – подсказать пользователю, что может быть ему также интересно. А для чего собственно нужен ИИ? Разберемся подробнее.

Различают четыре основных типа рекомендательных систем:

1. Коллаборативная фильтрация – рекомендации основаны на истории оценок как самого пользователя, так и других пользователей. Как это работает? Допустим, что пользователю *N* понравился фильм, который вы еще не смотрели. Этот пользователь похож на вас и ему нравятся такие же фильмы, как и вам. Этот фильм по некоторым условиям похож на те фильмы, которые вы уже смотрели. Значит, фильм может понравиться и вам. Примером подобного сервиса является last.fm. Какие плюсы? Теоретически высокая точность. Минусы? Высокий порог входа. Если не знать ничего об интересах пользователя, рекомендации будут практически бесполезны. Многие пользователи будут уходить.

2. Системы, основанные на контенте – рекомендации основаны на данных, собранных о каждом конкретном товаре. Как это работает? Посетителю рекомендуются объекты, похожие на те, которыми данный пользователь интересовался ранее или которые уже приобретал. Как узнать, что один объект похож на другой? Исходя из содержимого этих объектов. Примерами подобных сервисов являются Prismatic, Surfingbird. Какие плюсы? Такие рекомендации работают даже тогда, когда система ничего не знает о данном пользователе. Какие минусы? Коэффициент точности вначале очень низок.

3. Системы, основанные на знаниях – рекомендации основаны на знаниях предметной области (а не о каждом товаре). Такой тип рекомендаций имеет более высокую точность, предлагая пользователю то, что ему необходимо. Система изучает взаимосвязи между объектами, учитывая ряд дополнительных опций, которые относятся к индивидуальным свойствам конкретного пользователя. Такими свойствами могут быть, например, пожелания или исходные данные социальных сетей. Примеры условий: вы купили плеер? Тогда вам возможно пригодятся наушники. Вы купили хорошую камеру? Вот вам скидка 10 % на сумку для нее. Данную практику применяют крупные интернет-магазины. Какие плюсы? Например, это позволит исключить ситуацию «Вы только что купили квартиру в Москве? Вероятно, вот эти 5 квартир в Москве вам тоже пригодятся!» Какие минусы? Очень сложно разработать такую систему и собрать данные.

4. Существует также и четвертый вид рекомендательных систем – гибридные. Такие рекомендации основаны на комбинировании кол-

лаборативных и контентных подходов. Это позволяет избежать недостатков, свойственных каждой системе. Примером условий может быть любой запрос, поскольку ограничение только в воображении создателей. Одним из лучших сервисов в применении данного подхода является Netflix. Погрешность по 5-балльной шкале составляет примерно 0,86 балла. Самым большим минусом является самая высокая сложность разработки такой системы.

Итак, мы увидели, что для каждого вида систем существуют свои сложности. Возникает вопрос: а что же лучше? Точного ответа нет, поскольку каждый сервис старается создать рекомендации на свой вкус и цвет (как говорится).

Существует два уровня систем рекомендаций:

1-й уровень – это глобальные оценки: анализ особенностей и предпочтений, которые изменяются очень медленно; интересных страниц; зависимости от характерных пользовательских черт, например – пола, места проживания и т. д.

2-й уровень – кратковременные тренды и быстрые изменения интереса во времени.

Чтобы составить качественные рекомендации, необходимо использовать явный или неявный сбор данных:

- При явном сборе от пользователя необходимо получить заполненные анкеты для выявления предпочтений. Минус данного метода в том, что достаточно сложно заставить пользователя поставить оценку.

- При неявном сборе сохраняются действия пользователя: что смотрел, какой товар добавлен в корзину, что комментировал или купил. Составление рейтингов при таком методе, происходит автоматически. Изъян данного метода – неопределенность: если пользователь посмотрел товар, неизвестно, понравился ли ему этот товар или нет; если пользователь не купил товар – неизвестно, по какой причине он отказался его покупать.

От теории к практике. Применение

Хотелось бы затронуть реальные проекты. Разберем рекомендательные системы, на примере видеохостинга YouTube. Смею предположить, что, по крайней мере, каждый третий человек хоть раз заходил на этот сайт. Без сомнений, YouTube является основой интернета сегодня. За последние годы ресурс стал настолько мощным, что это даже страшит. Он с точностью предсказывает, какие видео могут понравиться пользователям и делает это лучше, чем раньше. Сейчас не

все вспомнят, но когда-то YouTube был немного больше, чем просто инфраструктура: он давал возможность быстро и просто встраивать видео на других сайтах, а по мере роста ресурса стал еще и хранилищем для архивных телевизионных клипов и вирусных хитов. На сегодняшний день это самый известный сервис наравне с Wikipedia: если вам напомнить о «харлем шейке», то вы точно пойдете смотреть все ролики с бредовыми танцами и застрянете там до конца дня.

«Мы знали, что люди приходят на YouTube только тогда, когда они знают, что хотят найти. Но мы хотели привлечь их, даже если они не знают, чего конкретно хотят», – говорит технический руководитель YouTube Джим МакФадден. Удивительно, но большую популярность этому сервису дал именно алгоритм систем рекомендаций. Цель разработчиков – как можно дольше удержать посетителя на своем сайте. Для этого совершенствовались алгоритмы поиска и подбора рекомендаций.

Подходяще подобранные рекомендации YouTube – это плоды работы GoogleBrain. Подразделение, которое работает с технологиями искусственного интеллекта, примкнуло к YouTube в 2015 году, хотя и не было первопроходцем в этой области. До этого компания использовала систему машинного обучения Sisyphus для формирования рекомендаций, тем не менее новая технология неконтролируемого обучения превысила все прошлые старания.

«Одна из главных особенностей этой технологии – обобщение. Если раньше я смотрел выступление комика, то в рекомендациях появлялось еще одно видео от этого комика. Модель GoogleBrain же найдет похожие видео с другими комиками, еще более смешными – она способна составлять не такие очевидные цепочки», – говорит МакФадден.

Наверняка вы и сами замечали, как точно YouTube может «угадать», что понравится вам. Например, алгоритм Brain отображает более короткие ролики для пользователей мобильного приложения и более длинные для тех, кто смотрит YouTube в приложении на телевизоре. Этот алгоритм догадался, причем безошибочно, что изменение рекомендаций по длине видео в зависимости от платформы будет способствовать росту среднего времени просмотра.

Основа всего – алгоритм

Благодаря активным внедрениям информационных систем с признаками интеллектуальности многие предприятия стали владельцами коллекций информационных ресурсов самой различной направленности.

сти и природы, каждая из которых управляется собственным программным обеспечением со специфическим интерфейсом [2].

Алгоритмы YouTube используют время просмотра – как долго пользователи просматривают видео. В этом есть смысл, поскольку YouTube не хочет, чтобы посетители тратили время на просмотр видео, которое никому не интересно.

Это значительно более тонкий коэффициент, чем просто «как долго вы просматривали видео». Алгоритм принимает во внимание множество разных факторов и оценивает их соответствующим образом: удержание посетителя, действия от нажатий, взаимодействие со зрителем и некоторые другие условия из-за кулис, которые мы никогда не видим. После этого YouTube адаптирует эти условия к вашему профилю, чтобы рекомендовать видео, которые, возможно, будут интересны вам.

Как и в случае других продуктов компании Google, в сервисе YouTube начали переход на внедрение глубоких нейронных сетей для основной массы проблем, связанных с обучением. Разработанная система рекомендаций базируется на технологии GoogleBrain. Она была недавно опубликована в открытый доступ как TensorFlow. Данная технология представляет собой гибкую среду для экспериментирования с разнообразными архитектурами глубоких нейронных сетей и использования крупномасштабного распределенного обучения. Модели, созданные инженерами Google, исследуют приблизительно один миллиард параметров и научены на сотнях миллиардов примеров.

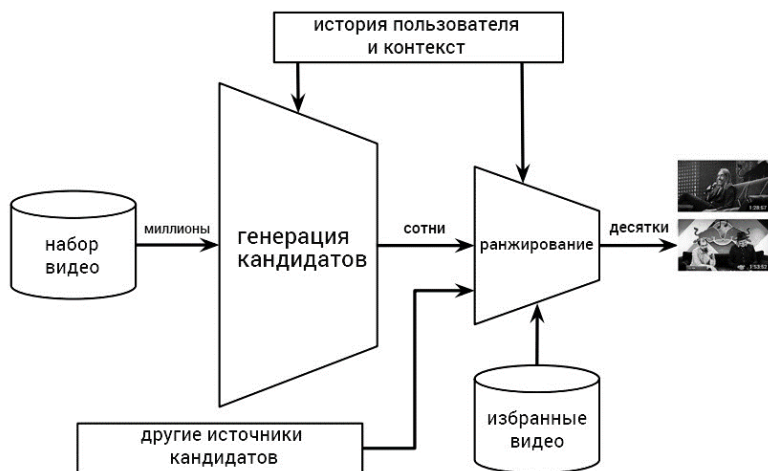


Рис. 1. Рекомендательный алгоритм YouTube

Система складывается из двух нейронных сетей: первая – для генерации «кандидатов» для просмотра, а другая – для их сортировки по рейтингу. Сеть генерации кандидатов собирает действия из истории активности пользователя YouTube в качестве входной информации и отделяет сотни небольших подкатегорий видеороликов из очень большого набора [1].

По факту сеть генерации кандидатов дает возможность отобрать ролики, которые смотрели пользователи с аналогичными интересами. Соответствие между пользователями формируется по тому, как много одинаковых видео они посмотрели (по ID видеороликов), по их поисковым запросам, а также по возрасту, полу и т. д.

Во время процесса генерации кандидатов алгоритм Google Brain большой объем видео с YouTube «просеивает» до сотен видеороликов, которые могут быть интересны пользователю. Прежде для этой цели использовался метод матричной факторизации, и первые версии модели нейронной сети повторяли это поведение с помощью неглубоких сетей, которые использовали только ранее просмотренные видео. Инженеры Google назвали свой подход нелинейным обобщением техник факторизации.

И несмотря на то что в YouTube существуют механизмы обратной связи (нравится/не нравится, опросы и т. д.), для обучения модели нейронной сети разработчики применяют неявное взаимодействие посетителя с системой в виде просмотров, где до конца просмотренное видео считается положительным примером, а показанное в выдаче, но не просмотренное – негативным. Выбор на этом уровне базируется на гораздо более объемной истории пользователя, что позволяет подбирать наиболее точные рекомендации, при случае когда явных отзывов недостаточно.

Чтобы представить пользователю небольшой список «лучших» рекомендаций, важно точно разделять «кандидатов» по их условной важности для пользователя. Сеть ранжирования выполняет данную задачу, устанавливая для каждого видео собственную оценку и применяя большой набор характеристик, которые описывают видео и пользователя. Ролики с самой высокой оценкой сортируются по важности и выдаются пользователю. Двухэтапный подход дает возможность разработчикам работать с миллионами роликов и при этом быть уверенными, что подобранный для каждого пользователя комплект рекомендаций будет персонализирован и интересен.

Огромное количество видеоконтента загружается на серверы YouTube каждую секунду, поэтому для такого сервиса весьма важно

представлять и «свежие» видеоролики, чтобы содействовать активности пользователей. Кроме отслеживания новых видеороликов, разработчики стараются улучшать систему в направлении автоматического поиска и распространения популярного контента. Популярность видео может сильно измениться со временем, однако усреднённая вероятность просмотра данного ролика за период обучения в несколько недель будет иметь вид мультиномиального распределения по всему набору контента.

Заключение

Если ранее подобные системы рекомендаций мы могли наблюдать только на крупных площадках и сервисах, то сегодня можно встретить такое даже в обычных интернет-магазинах. Существует распространённый стереотип, который мешает широкому применению в бизнесе рекомендательных систем. Многим кажется, что в реальности внедрение рекомендательных алгоритмов – это очень сложный процесс, который требует глобальных изменений в процессе сбора и обработки данных, а также изменения бизнес-процессов, логики и так далее. Эти сомнения совершенно не обоснованы, поскольку на самом деле рекомендательные системы могут быть очень даже полезны для ведения успешного бизнеса. Мы разобрали на некоторых примерах, что такое возможно и сервисы только благодаря этому могут стать популярными. А где же брать данные? Обычно, чтобы начать рекомендовать вполне достаточно тех данных, которые уже собираются.

На примере видеохостинга YouTube мы обнаружили, что для создания рекомендаций нет ничего сложного. Конечно, очень важен опыт в разработке подобных систем. Очень часто программисты изобретают колесо, что намного усложняет и замедляет процесс создания и внедрения таких систем. Они пишут программный код и отлаживают его, обучают нейронные сети – очень много времени затрачено зря. Если говорить об ИИ, то на помощь приходит открытая программная библиотека для машинного обучения TensorFlow от Google.

Я считаю, что наше будущее за искусственным интеллектом. Сегодня уже ни для кого не секрет, что ИИ может многие процессы автоматизировать. В будущем, я считаю, рекомендательные системы станут еще лучше. Поисквые алгоритмы и алгоритмы выдачи результата смогут предсказывать, что желает пользователь получить в данный момент.

Литература

1. Deep Neural Networks for YouTube Recommendations. URL: <http://static.googleusercontent.com/media/research.google.com/ru//pubs/archive/45530.pdf> (Дата обращения 22.03.2019)

2. Рябова А. А., Липовая Т. Б. Применение INTERNET-технологий в интеллектуальных информационных системах // Современные методы интеллектуального анализа данных в экономических, гуманитарных и естественно-научных исследованиях: Материалы Международной научно-практической конференции. Российский экономический университет им. Г. В. Плеханова, филиал в г. Пятигорске Ставропольского края. Пятигорск, 2016. С. 80–86.

УДК 519.95

МОДЕЛИРОВАНИЕ С ПРИМЕНЕНИЕМ ИНТЕГРИРОВАННЫХ РАСКРАШЕННОСТОХАСТИЧЕСКИХ СЕТЕЙ ПЕТРИ НА БАЗЕ CPNTools

Гусейнзаде Ш. С., канд. техн. наук, доцент,
shahla.huseynzade@gmail.com

Сумгаитский государственный университет,
г. Сумгаит, Республика Азербайджан

Аннотация. В статье вводятся расширения структуры стохастических сетей Петри (ССП) для удобства решения конкретных задач моделирования интеллектуального управления динамическими системами. Описаны наиболее распространенные классы СПП, проведено их сравнение и определены недостатки. Недостатком является то, что не учитывается совместимость множества параметров и характеристик, которые отражают вероятностные показатели структурных компонент, таких как функции инцидентностей, позиции, переходы, маркировки и дуги. Эти недостатки существенно ограничивают возможности исследователя. С целью преодоления этих проблем ниже сформирована интегрированная СПП на основе представленных типов. Разработан алгоритм функционирования интегрированной СПП. С целью качественного анализа процесса и интерпретирования динамики меток, в представленную интегрированную СПП введена возможность построения раскрашенной структуры: функция цвета, емкость

маркеров в позициях с учетом цветов. На примере функционирования технической системы разработана интегрированная раскрашенная стохастическая сеть Петри (РССП), – модель процесса возникновения и устранения неисправностей в виде графа ССП на CPNTools. На основе известных статистических данных об интенсивностях возникновения отказов и длительностях операций поиска неисправностей, замены и ремонта отказавшего блока, сформированы структурные элементы СП со стохастическими и раскрашенными параметрами. При компьютерной реализации алгоритма получена последовательность срабатываемых переходов на изменения маркировки графа. Модель интегрированной РССП разрешает конфликтные ситуации и предотвращает тупиковые состояния.

Ключевые слова: стохастические сети Петри; вероятность маркировок; множество переходов; раскрашенные сети Петри; функции инцидентностей; начальная маркировка; тупиковые ситуации.

MODELLING WITH THE APPLICATION OF INTEGRATED COLORED STOCHASTIC PETRI NETS BASED ON CPN Tools

Huseynzade Sh. S.

Abstract. *The article introduces extensions of the structure of stochastic Petri net (SPN) for the convenience of solving specific problems of intelligent control modeling of dynamic systems. The most common classes of SPN are described, compared and deficiencies are identified. The disadvantage is that there is not taken into account the compatibility of the set of parameters and characteristics, which reflect probabilistic indicators of structural components, such as incidence functions, positions, transitions, markings and arcs. These shortcomings significantly limit the ability of the researcher. In order to overcome these problems, an integrated SPN was formed on the basis of the types presented. The algorithm of functioning of the integrated SPN is developed.*

Using the example of a module in a flexible production system, a model has been developed for the process of occurrence and elimination of faults in a technical system in the form of a PN graph.

The algorithm of functioning of the integrated SPN is developed. In order to qualitatively analyze the process and interpret the dynamics of tags, the possibility of constructing a colored structure is introduced into the presented integrated SPN: the function of color, the capacity of markers

in the positions with regard to colors. Using the example of the functioning of a technical system, there has been developed a model of the process of emergence and elimination of faults in the formintegrated colored stochastic Petri net (CSPN) of a graph of PN on CPN Tools. On the basis of known statistics on failure rates and durations of troubleshooting operations, replacement and repair of a failed unit, structural elements of the PN with stochastic parameters are formed. With the computer implementation of the algorithm, a sequence of triggered transitions and changes in the marking of the graph are obtained. The integrated CSPN model resolves conflict situations and prevents dead-end states.

Keywords: *stochastic Petri net; labeling probability; the set of transitions; colored Petri net; incidence functions; initial labeling; dead-end states.*

Введение

В теории моделирования систем имеют место исследования стохастических процессов, степень неопределенности которых, с течением времени может изменяться. В таких случаях принимаются некоторые предположения к условиям функционирования данных процессов, в которых допускаются некоторые потенциально присутствующие ошибки в количественной оценке вероятности событий.

Путем взаимодействия различных видов специальных сетей Петри образуются комбинированные сети Петри (СП). Они отличаются наибольшей сложностью, обладая характеристиками всех своих «родителей», интегрируют их возможности. В частности, интегрированная РССП, являясь комбинацией раскрашенных и стохастических СП, обладает преимуществами и тех, и других [1].

Аппарат ССП позволяет построить модель интеллектуального управления динамических систем, в которой на структуру накладываются стохастические параметры и логические условия взаимодействия процессов. Отдельным структурным элементам, таким как позиции, переходы, дуги, маркеры можно присвоить стохастические параметры. В том числе переходам ССП сопоставляются условные вероятности их срабатывания.

В настоящее время имеется много расширений ССП. Например, разработаны ССП с непрерывным и дискретным временем, разными типами временных задержек переходов, приоритетами и ингибиторными дугами [2, 3]. Так, определен класс обобщенных ССП с экспоненциально распределенной задержкой. Время может располагаться в позициях, метках, дугах и/или переходах.

Для SimNet время срабатывания устанавливается в соответствующий переход [4]. Имеются также другие расширения структуры и поведения ССП для удобства решения конкретных задач моделирования.

Стохастические сети Петри определяются парой $M_s = (c, \mu^s)$, где $c = (P, T, I, O)$ описывает структуру сети, а отображение $\mu^s : P \rightarrow V_s = [0, 1]$ присваивает каждой позиции вероятностей наличия фишек $\mu^s(p_i)$ [5]. В этой разработке заданы вероятностные показатели маркировок. Недостатком этой разработки является то, что не учитываются вероятностные показатели структурных компонент, таких как позиции, переходы, дуги.

В непрерывно-временной ССП, заданной пятеркой $N = (P, T, W, \Omega, M)$, переходам присвоены $\Omega: T \rightarrow IR_+$ – функции темпов переходов [6]. В этой разработке заданы вероятностные показатели переходов. Недостатком этой разработки является то, что не учитываются вероятностные показатели маркировок, позиций, дуг.

В существующих модификациях ССП стохастические параметры одновременно применены только к одному из структурных элементов, что существенно ограничивает возможности исследователя. Учитывая сложность реальных систем, параллелизм процессов, взаимные синхронизации и блокировки и множество вероятностных параметров, появляется необходимость расширения выразительных средств моделирования, что делает необходимым разработку новых интегрированных ССП и усовершенствование существующих, в которых стохастические параметры одновременно применены к нескольким структурным элементам.

Представление интегрированных ССП

В настоящей работе представляется разработанная модификация, приводящая к интеграции вышепоказанных видов ССП, в которой стохастические параметры применены к двум структурным элементам одновременно: к переходам и к маркерам в позициях. Модификация названа как интегрированная ССП и сформирована нижеследующим образом: интегрированная стохастическая сеть Петри определяется пятеркой

$$N_S^I = (P, T, W, \Omega, \mu^s),$$

где $P = \{p_1, p_2, \dots, p_n\}$, $n > 0$ – конечное непустое множество позиций;
 $T = \{t_1, t_2, \dots, t_m\}$, $m > 0$ – конечное непустое множество переходов;

$W: (P \times T) \cup (T \times P) \rightarrow N$ – функция весов дуг между позициями и переходами, и наоборот (N – натуральные числа);

$I: P \times T \rightarrow (0, 1, \dots)$; $O: T \times P \rightarrow (0, 1, \dots)$ – соответственно функции входных и выходных вмешательств);

$\Omega: T \rightarrow IR_+$ – функция темпов переходов ($IR_+ = [0; \infty)$ – непрерывная временная шкала);

отображение $\mu^s: P \rightarrow V_s = [0, 1]$ присваивает каждой позиции вектор распределения вероятностей наличия фишек $\mu^s(p_i)$.

Неопределенность наличия фишек описывается векторами распределения вероятностей каждой позиции. Перераспределение наличия фишек отражается в длине и компонентах векторов распределения. Определение элементов вектора распределения с вычислительной точки зрения вызывает некоторые трудности при считывании функции темпов (скоростей) переходов и времени пребывания в маркировке. С целью преодоления этой проблемы разработан алгоритм вычисления вероятности срабатывания, времени пребывания в маркировке разрешенного перехода и элементов вектора распределения вероятностей при изменении маркировки, после срабатывания перехода. Разработанный алгоритм состоит из следующих шагов:

Шаг 1. Создание матрицы входных и выходных инцидентов и темпов переходов:

$$F = \{h_{ij}\}, \quad H = \{h_{ji}\}, \quad \Omega = \{\theta_j\},$$

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

Шаг 2. Создание начальной маркировки:

$$\mu = \{\mu_i\}, i = \overline{1, n}.$$

Шаг 3. Определение матрицы:

$$S = \{s_c\},$$

где $c = \overline{0, m}$ для фиксации номеров разрешенных переходов.

Шаг 4. Поиск разрешенного перехода:

а) выбираются все $f_{ij} \neq 0$, при $i = \overline{1, n}$;

б) для каждого фиксированного i должен быть $\exists \mu_i \neq 0$.

Шаг 5. Если для перехода t_j условие срабатывания выполняется, фиксируется номер этого разрешенного перехода:

$$c = c + 1; \quad s_c = j;$$

Шаг 6. Индекс j увеличивается на единицу: $j = j + 1$. При $j \leq m$ осуществляется переход к пункту a шага 4, в противном случае, если $c = 0$, то выводится сообщение о тупиковом состоянии, завершается поиск и осуществляется переход к пункту 13.

Шаг 7. Выбор перехода t_j с наибольшим темпом θ_j среди фиксированных переходов, при $j = s_i$, где $i = \overline{1, c}$;

Шаг 8. Вычисление вероятности срабатывания разрешенного перехода t_j в маркировке μ_i :

$$PF(t_j) = \frac{\theta_j}{\sum_{i=1}^c \theta_{s_i}}.$$

Шаг 9. Вычисление среднего времени пребывания в маркировке μ_i :

$$SJ(t_j) = \frac{1}{\sum_{i=1}^c \theta_{s_i}}.$$

Шаг 10. Формирование распределения вероятностей каждой входной позиции после срабатывания перехода t_j :

$$\mu'_i = \mu_i - f_{i,j};$$

Шаг 11. Формирование вектора распределения вероятностей каждой выходной позиции после срабатывания перехода:

$$\mu'_i = \mu_i + h_{j,i};$$

Шаг 12. По выбору пользователя или процесс продолжается и осуществляется переход к шагу 5, или процесс останавливается и осуществляется переход к шагу 13.

Шаг 13. Конец алгоритма.

Интегрированные ССП могут быть применены для моделирования различных стохастических процессов более широкого класса. Од-

нако при этом рассматриваемая сеть не лишена некоторых недостатков. Нет деклараций и пометок сети, которые дают возможность качественно характеризовать процессы и интерпретировать динамику меток сети.

Представление интегрированных РССП

Раскрашенные сети Петри (РСП) отражают логическую последовательность событий, взаимодействие параллельных процессов и позволяют проследивать потоки информации. Преимущество этого типа моделей состоит в простоте понимания на инженерном уровне и быстроте чтения графических образов. В системе CPNTools (Colored Petri Nets Tools) воспроизведение динамики меток на экране дисплея позволяет проследить ход и качественные характеристики процессов и эффективно интерпретировать динамику раскрашенных сетей [7]. Раскрашенная сеть состоит из трех частей: структуры сети, деклараций и пометки сети. Структура сети аналогична традиционной структуре. Таким образом, необходимо добавить в сеть возможность построения раскрашенных структур, таких как:

- C – функция цвета маркера, определяющая в данном случае цвет каждого из маркеров $M(p_i)$ для позиций сети;
- V – условия срабатывания переходов в зависимости от цвета маркера;
- K – емкость маркеров в позициях с учетом C .

Каждое дополнительное свойство сети усложняет ее понимание, но при этом сокращает ее размерность, что, в конечном счете, упрощает процесс моделирования.

На сегодняшний день существует большое число программных продуктов, позволяющих проводить моделирование в режиме реального времени с помощью сетей Петри. Из них выделяют моделирующую систему CPNTools, которая в полной мере поддерживает раскрашенные СП. Более того, CPNTools также позволяет использовать свойства стохастических и временных сетей Петри при построении моделей. Задание стохастичности в нем возможно лишь с помощью процедур программирования на языке CPNML.

Поэтому целесообразна разработка подхода моделирования интегрированных РССП с помощью процедур программирования на языке CPNML на базе CPNTools.

В приводимом ниже примере рассмотрена модель функционирования технической системы, в которой требуется смоделировать про-

цессы возникновения и устранения в ней неисправностей. Известны статистические данные об интенсивностях возникновения отказов и длительностях таких операций, как поиск неисправностей, замена и ремонт отказавшего блока.

С целью разработки модели функционирования представленной системы с применением интегрированной РССП, соответственно к этапам формирования СП, определяются основные структурные элементы.

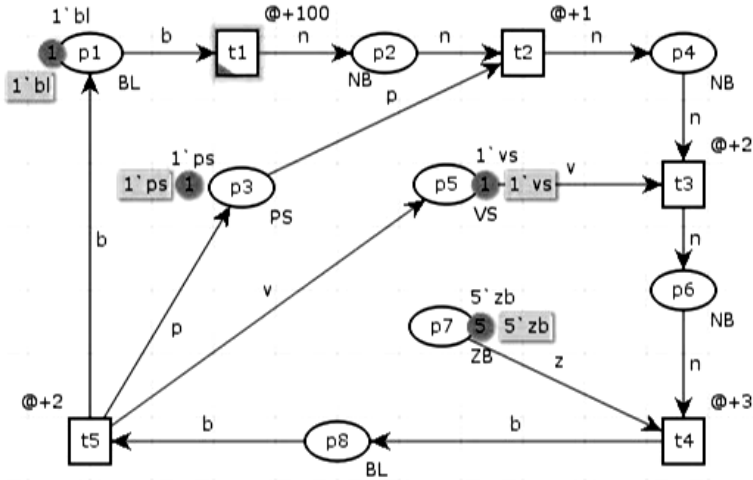


Рис. 1. Граф РИССП технической системы

Позиции: p_1 – в системе имеются блоки, число = m ; p_2 – имеются неисправные блоки; p_3 – поисковая система свободна; p_4 – обнаружен неисправный блок; p_5 – восстанавливающая система свободна; p_6 – восстанавливается система; p_7 – имеются запасные блоки, число = n ; p_8 – заменен блок.

Переходы: t_1 – отказ системы; t_2 – поиск неисправного блока; t_3 – начинается восстановление системы; t_4 – замена блоков; t_5 – заканчивается восстановление системы.

Случайным может быть число неисправных блоков, темпы появления отказов, темпы поиска неисправных блоков и восстановление блоков и т. п. При срабатывании перехода в РССП в определенный момент времени фишки изымаются из входных позиций перехода и мгновенно помещаются в выходные. Каждому переходу $t \in T$ сопоставляется темп $\Omega(t)$, являющийся параметром экспоненциального распределения.

В рассмотренном примере имеются позиции $P = \{p_1, p_2, \dots, p_7\}$ и переходы $T = \{t_1, t_2, t_3, t_4\}$. На основе функции входной и выходной инцидентности строятся дуги между позициями и переходами СП [8].

Матрица темпов переходов $\Omega = \{\theta_j\}$, где $j = \overline{1, m}$ представляется вектором модельных времен: (100; 1; 2; 3; 5).

Начальная маркировка $\mu_0 = (\mu_{0,1}; \mu_{0,2}; \mu_{0,3}; \mu_{0,4}; \mu_{0,5}; \mu_{0,6}; \mu_{0,7}; \mu_{0,8})$ представляется векторами: $\mu_0 = (1,0; 0,9; 0,7; 0,8; 0,7; 0,8; 0,7; 0,9)$.

Условия запуска переходов представлены в виде кодового выражения на языке ML CPN, к которому применены объявленные константы, операции и функции.

```
Standart Declarations
colset UNIT=unit;
colset BOOL
colset INT
colset INTINF
colset TIME
colset BL=unit with bl;
colset NB=unit with nb;
colset ZB=unit with zb;
colset VS=unit with vs;
colset PS=unit with ps;
colset REAL
colset STRING
var b:BL;
var n:NB;
var z:ZB;
var v:VS;
var :PS;
```

При компьютерной реализации алгоритма получена последовательность событий в виде сработавших переходов $\delta = (t_1, t_2, t_3, t_4, t_5)$ и изменения маркировки графа. Алгоритм разрешает конфликтные ситуации и предотвращает образование тупиковых состояний в интегрированной ССП.

Заключение

На разработанной модели можно отрабатывать принципы управления, соответствующие различным ситуациям: выявлять недостатки, тупиковые состояния систем и вносить корректировки.

Потенциальные области применения разработанного подхода можно найти при моделировании когнитивных процессов, при проектировании интеллектуального управления динамическими системами со стохастическим характером параметров.

Литература

1. *Борисов В. В., Круглов В. В., Федулов А. С.* Нечеткие модели и сети. М.: Телеком, 2012. 284 с.
2. *Florin G., Natkin S.* Les reseaux de Petri stochastiques // *Technique et Science Informatique*. 1985. Vol. 4. N 1. P. 143–160.
3. *Ларкин Е. В., Ивутин А. Н., Костомаров Д. С.* Методика формирования сети Петри – Маркова для моделирования когнитивных технологий // *Известия ТулГУ. Технические науки*. 2013. Вып. 9. Ч. 1.
4. *Jensen K., Podelski A.* Tools and algorithms for the construction and analysis of systems // *International Journal on Software Tools for Technology*. Springer-Verlag, Berlin, Heidelberg. Volume 8. Issue 3. June 2006.
5. *Лескин А. А., Мальцев П. А., Спиридонов А. М.* Сети Петри в моделировании и управлении. Л.: Наука, 1989.
6. *Тарасюк И. В.* Стохастические сети Петри – формализм для моделирования и анализа производительности вычислительных процессов // *Системная информатика*. 2004. Вып. 9. С. 135–194. <http://itar.iis.nsk.su/files/itar/pages/spnsinf.pdf>
7. *Jensen K., Kristensen L. M.* Coloured Petri Nets: Modelling and Validation of Concurrent Systems // 1st Springer Publishing Company, Incorporated. 2009. 384 p.
8. *Питерсон Дж.* Теория сетей Петри и моделирование систем. М.: Мир, 1984.

ТЕХНОЛОГИИ ОПТИМИЗАЦИИ ПОЛЬЗОВАТЕЛЬСКИХ КОДОВ

УДК 681.343.001

ФОРМАЛИЗАЦИИ МЕТОДА СТАТИЧЕСКОГО КЭШИРОВАНИЯ ФУНКЦИЙ

Томаев М. Х., канд. техн. наук, доцент, tmxwork@mail.ru

Миронян Р. А., студент

Северо-Кавказский горно-металлургический институт

(государственный технологический университет),

г. Владикавказ, Россия

Аннотация. Описывается подход для улучшения производительности пользовательских программных кодов, в основе которого метод статического кэширования функций с целочисленным аргументом. Обосновывается оптимизационный характер задачи. Предлагается дискретная модель, в которой формально выражены критерии качества пользовательского кода, а также ограничение в виде накладных расходов, которые требуются для реализации описываемого подхода.

Ключевые слова: программа, оптимизация, модель, качество, метод, экстремальные, кэширование, память.

FORMALIZATION OF FUNCTIONS STATIC CACHING METHOD

Tomaev M. H., Mironyan R. A.

Abstract. *The article describes an approach, permitting to improve the performance of user program codes, which is based on the method of static caching of functions with an integer argument. The optimization nature of the problem is justified. A discrete model is proposed in which the criteria for the quality of a custom code are formally expressed, as well as a restriction in the form of overhead costs that are required to implement the described approach.*

Keywords: *program, optimization, model, quality, method, extreme, caching, memory.*

Введение

Методы оптимизации программного кода [1; 2] сохраняют свою актуальность, несмотря на быстрое развитие аппаратных средств вы-

числительных систем. В ряду оптимизационных подходов особое место занимают методы так называемого экстремального программирования, которые характеризуются высокими затратами вычислительных ресурсов (как правило, оперативной памяти) для достижения часто небольшого прироста производительности. Использование подобных методов оправдывает себя в случаях, когда производительность является критически важным показателем качества программного продукта [1]. Так как объем быстрой памяти ЭВМ является достаточно дорогим ресурсом, то, очевидно, актуальной становится задача оптимального распределения оперативной памяти между такими участками программного кода, модификация которых даст наибольший эффект. В работе предлагается модель, формально описывающая подобный подход в виде дискретной экстремальной задачи, в рамках которой критерии качества сформулированы в виде целевой функции, максимизирующей выигрыш в производительности, который может быть получен в результате оптимизации, а ограничение выражено в виде суммарных затрат быстрой памяти, общий объем которых не должен превысить верхнюю границу доступной оперативной памяти.

Метод статического кэширования результатов выполнения функций с целочисленными аргументами

Примером подобной модели является задача оптимального статического кэширования функций с целочисленным аргументом, суть которой можно продемонстрировать на следующем примере: Исходный код пользователя на языке C++, представленный в Листинге 1, содержит функцию F1 с целочисленным аргументом x , которая вызывается N раз, причем каждый раз в функцию передается значение, сгенерированное с помощью стандартного генератора случайных чисел – функции `rand`, возвращающей псевдослучайные целые значения из интервала $[0, \text{RAND_MAX}]$.

Листинг 1:

```
double F1(int x)
{   return sin(x+1)/ cos(x*2) + log10(x+1);
}
void main()
{   double Sum;
    for (int i=0; i< 100000; i++) Sum += F1(rand());
    cout<<"Average = " << (double)Sum/100000.0<<endl;
}
```


После применения метода статического кэширования получим новый, более производительный код (Листинг 2):

Листинг 2:

```
int F1(int x)
{ static bool CacheStatus[RAND_MAX+1];
  static double Cache[RAND_MAX+1];
  if (CacheStatus[x]==false)
  {
    CacheStatus[x] = true;
    Cache[x] = (sin(x+1)/ cos(x*2) + log10(x+1));
  }
  return Cache[x];
}
void main()
{ doubleSum;
  for (int i=0; i< 100000; i++) Sum += F1(rand());
  cout<<"Average = " << (double)Sum/100000.0<<endl;
}
```

Для обслуживания кэша в функции F1 используются 2 статических массива:

1) CacheStatus – массив булевых значений. Элемент CacheStatus[x] равен true, если значение функции для аргумента «x» уже вычислено, и нулю – в противном случае.

2) Cache – массив вычисленных и помещенных в кэш значений.

Оптимизационную задачу, описывающую процесс выбора оптимальной стратегии статического кэширования в пользовательской программе, можно представить в виде следующей модели (1):

$$\left\{ \begin{array}{l} F_1 = \sum_{i=0}^M (z_i f_i) \rightarrow \max; \\ f_i = \begin{cases} (N_i - Q_i)t_i, & \text{если } N_i \geq Q_i; \\ 0, & \text{если } N_i < Q_i. \end{cases} \\ \sum_{i=0}^M CacheSize_i \leq V; \\ z_i = 1, 0; i = 1, 2, \dots, M; \end{array} \right. \quad (1)$$

где z_i – булева переменная (неизвестная в данной модели), равная единице, если к i -ой функции применяется статическое кэширование, и нулю – в противном случае;

N_i – количество вызовов i -ой функции;

Q_i – минимально возможное количество вызовов функции, необходимое для формирования кэша;

$CacheSize_i$ – размер кэша, необходимый для статического кэширования значений i -й функции;

V – верхняя граница доступного объема оперативной памяти.

Целевая функция модели (1) максимизирует суммарный выигрыш в производительности, полученный от статического кэширования функций пользовательской программы. Результатом решения данной задачи является оптимальная стратегия кэширования, определяющая, какие функции следует модифицировать, а какие – нет. Модель (1) является задачей дискретной оптимизации, для решения которой можно воспользоваться комбинаторными процедурами (прямым перебором, методами типа ветвей и границ и другими).

Заключение

Реализация оптимизационного алгоритма, осуществляющего оптимизацию в соответствии со стратегией, вычисляемой согласно модели (1), позволит расширить набор доступных разработчикам инструментов, улучшающих качество проектируемого ПО.

Литература

1. Гроппен В. О. Принципы принятия решений с помощью этапов // Автоматика и телемеханика. 2006. № 4. С. 167–184.

2. Гроппен В. О., Томаев М. Х. Модели, алгоритмы и средства программной поддержки проектирования оптимальных программных продуктов // Автоматика и телемеханика. 2000. Вып. 11.

3. Томаев М. Х. Модели экстремального программирования // Информационные технологии и системы. Наука и практика. 2009. С. 150–152.

4. Томаев М. Х. Технологии глобальной оптимизации пользовательских программных кодов // Автоматизация и управление в технических системах. 2015. № 3 (15). С. 16–30.

ЭКСТРЕМАЛЬНЫЕ ПОДХОДЫ К ВЫБОРУ МЕТОДА РАЗМЕЩЕНИЯ ПОЛЬЗОВАТЕЛЬСКИХ ДАННЫХ В МНОГОУРОВНЕВОЙ ОПЕРАТИВНОЙ ПАМЯТИ ЭВМ

Томаев М. Х., канд. техн. наук, доцент, tmxwork@mail.ru

Губиев Д. А., магистрант

Северо-Кавказский горно-металлургический институт
(государственный технологический университет),
г. Владикавказ, Россия

Аннотация. В статье предлагается способ перемещения данных из динамической области данных в более быструю стековую. Обосновывается оптимизационный характер метода. Предлагается дискретная модель, описывающая задачу.

Ключевые слова: программа, оптимизация, модель, качество, метод, экстремальные, кэширование, память.

EXTREME APPROACHES TO THE CHOICE OF METHOD FOR PLACING USER DATA IN A MULTI-LEVEL MAIN MEMORY OF A COMPUTER

Tomaev M. H., Gubiyev D. A.

Abstract. *The article proposes a way to move data from a dynamic data area to a faster stack. The optimization nature of the problem is justified. A discrete model is proposed, described the task.*

Keywords: *program, optimization, model, quality, method, extreme, caching, memory.*

Введение

Проблема оптимального размещения данных [1; 2] актуальна в связи с существенной разницей во времени выделения необходимых пользователю приложению блоков данных, обусловленной типом памяти, выбранной для размещения. Для каждого класса быстрой (оперативной) памяти характерны признаки, влияющие на производительность.

В работе [5] показана возможность оптимизации размещения данных заменой стековых данных статическими. Настоящая работа посвящена методу оптимизации, основанном на перемещении информации из динамической области в стек приложения.

Перемещение динамических данных в стек приложения

Современные компиляторы не имеют оптимизационных средств, позволяющих решить проблему низкой производительности менеджера динамической памяти. Главным препятствием замене динамических данных на стековые либо статические является непредсказуемость размера динамических объектов. Глобальная (статическая), а также стековая память способны размещать данные только фиксированной (известной на момент компиляции кода) длины. В связи с этим замена динамических элементов на данные фиксированной длины требует некоторой предварительной подготовки. Статистический анализ размеров динамических объектов может дать приблизительное значение верхней границы размеров динамических объектов – эту величину можно использовать для безопасного размещения статического элемента данных в стековом с помощью подхода, наглядно представленного в следующем примере:

Исходный код пользователя на языке C++ представлен в листинге 1.

Листинг 1:

```
void F1(int x, double &left, double &right)
{   double *temp = new [x];
    for (int i=0; i<x; i++) temp[i] = x * return sin(i+1)/ log10(i+1);
    std::sort( temp, temp, temp + x);  left = temp[0]; right = temp[x-1];
    delete [] temp;
}
void main()
{   double L, R;  F1(rand(), L,R);   std::cout <<L<<R<<std::endl;
}
```

Пусть в результате статистического анализа тестовых запусков программы известно, что приближенное значение верхней границы длины массива «temp» равно 32700 элементов. Тогда безопасная модификация функции F1, перемещающая данные из динамической области в стек, приведет исходный код к следующему виду (Листинг 2).

Листинг 2:

```
void F1(int x, double &left, double &right)
{   double stack_data[32700];
    double *temp;
    if ( x >= 32700 ) temp = new [x];
    else      temp = stack_data;
    for (int i=0; i<x; i++) temp[i] = x * return sin(i+1)/ log10(i+1);
```

```

std::sort( temp, temp, temp + x); left = temp[0]; right = temp[x-1];
if (temp!=stack_data) delete [] temp;
}
void main()
{ doubleL, R; F1(rand(), L,R); std::cout <<L<<R<<std::endl;
}

```

Выделенные «жирным» шрифтом строки соответствуют проведенным модификациям. Как видно из Листинга 2, выполненные изменения гарантируют безопасную устойчивую работу приложения, которая не зависит от достоверности значения верхней границы исходного динамического массива.

Так как размер стека приложения является постоянной величиной, задача перемещения данных из динамической памяти в стековую носит очевидный оптимизационный характер и может быть выражена следующим образом:

$$\begin{cases} F_1 = \sum_{i=0}^M (z_i f_i) \rightarrow \max; \\ f_i = \frac{v_i}{s^{heap}} - \frac{v_i}{s^{stack}}; \\ \sum_{i=0}^M CacheSize_i \leq V; \\ z_i = 1, 0; i = 1, 2, \dots, M; \end{cases} \quad (1)$$

где z_i – булева переменная, равная единице, если к i -й динамический элемент данных перемещается в стек, и нулю – в противном случае;

$CacheSize_i$ – размер кэша, необходимый для статического кэширования значений i -й функции;

V – верхняя граница доступного объема стека потока.

Задача (1) решается для каждой функции приложения в отдельности, т. к. каждой новой функции верхнего уровня доступен весь объем стека. Целевая функция модели (1) максимизирует суммарный выигрыш в производительности одной функции, полученный от перемещения динамических элементов данных в стек. Результатом решения данной задачи является оптимальная стратегия размещения данных, определяющая, какие данные следует перемещать в стек, а какие – нет. Модель (1) является задачей дискретной оптимизации, для реше-

ния которой можно воспользоваться комбинаторными процедурами (прямым перебором, методами типа ветвей и границ и другими).

Заключение

Реализация оптимизационного алгоритма, осуществляющего оптимизацию в соответствии со стратегией, вычисляемой согласно модели (1), позволит расширить набор доступных разработчикам инструментов, улучшающих качество проектируемого ПО.

Литература

1. *Гроппен В. О.* Принципы принятия решений с помощью эталонов // Автоматика и телемеханика. 2006. № 4. С. 167–184.
2. *Гроппен В. О., Томаев М. Х.* Модели, алгоритмы и средства программной поддержки проектирования оптимальных программных продуктов // Автоматика и телемеханика. 2000. Вып. 11.
3. *Томаев М. Х.* Модели экстремального программирования // Информационные технологии и системы. Наука и практика 2009. С. 150–152.
4. *Томаев М. Х.* Технологии глобальной оптимизации пользовательских программных кодов // Автоматизация и управление в технических системах. 2015. № 3 (15). С. 16–30.
5. *Томаев М. Х.* Средства автоматизации оптимизационных преобразований исходных кодов программных систем // Электронный научный журнал: Программные продукты системы и алгоритмы. 2018. Вып. 3. ISSN 2311-6749. DOI: 10.15827/2311-6749.28.319

УДК 004.272

ОПТИМИЗАЦИЯ СОЗДАНИЯ ЭКЗЕМПЛЯРОВ КЛАССОВ МЕТОДОМ ПЕРЕНОСА КОДА УСТАНОВКИ НАЧАЛЬНЫХ ЗНАЧЕНИЙ СВОЙСТВ КЛАССА ИЗ БЛОКА КОНСТРУКТОРА КЛАССА В СЕКЦИЮ ИНИЦИАЛИЗАЦИИ

Мирошников А. С., канд. техн. наук, доцент, mirandrey@mail.ru
Буймистров В. А., магистрант, vovabmail@ya.ru
Северо-Кавказский горно-металлургический институт
(государственный технологический университет),
г. Владикавказ, Россия

Аннотация. В статье описан алгоритм, выполняющий оптимизацию создания экземпляров классов на языке программирования C++ методом переноса кода установки начальных значений свойств класса из блока конструктора класса в секцию инициализации и программный комплекс, позволяющий оптимизировать пользовательский исходный код в соответствии с методом оптимизации.

Ключевые слова: C#, C++, оптимизация кода, класс, конструктор класса, секция инициализации.

OPTIMIZATION OF CREATING INSTANCES OF CLASSES BY TRANSFERRING THE INSTALLATION CODE OF THE INITIAL VALUES OF CLASS PROPERTIES FROM THE CLASS CONSTRUCTOR BLOCK TO THE INITIALIZATION SECTION

Miroshnikov A. S., Buymistrov V. A.

Abstract. *The article describes an algorithm that optimizes the creation of instances of classes in the C++ programming language by transferring the code for setting the initial values of the class properties from the class constructor block to the initialization section and a software package that allows you to optimize the user source code in accordance with the optimization method.*

Keywords: *C#, C++, code optimization, class, class constructor, initialization section.*

Введение

В настоящее время по-прежнему сохраняет актуальность тематика оптимизации программного обеспечения. Особое значение она имеет в области разработки систем с высокими требованиями к качеству программного обеспечения. Сюда можно отнести научные и военные системы, датчики, разнообразные встраиваемые системы и другие.

Математическая модель

Основная идея формулируемого подхода – создание экземпляров классов методом переноса кода установки начальных значений свойств класса из блока конструктора класса в секцию инициализации.

Выигрыш от оптимизации с помощью секции инициализации можно сформулировать следующим образом:

$$\left\{ \begin{array}{l} F = \sum_{i=1}^M \left(N_i \left(\sum_{j=1}^{S_i} \left((1 - b_{ij}) z_{ij} t_{C_{ij}}^{\text{constructor}} \right) \right) \right); \\ \forall (i, j): z_{ij} \geq b_{ij}; \\ z_{ij} = \overline{0, 1}. \end{array} \right. \quad (1)$$

где z_{ij} – булева переменная, равная единице, если j -ая переменная i -го класса должна быть размещена в секции инициализации (неизвестная, задается пользователем);

M – количество классов;

N_i – количество экземпляров i -го класса, используемых в программе;

S_i – количество переменных в составе i -го класса;

b_{ij} – булева переменная, равная единице, если j -ая переменная i -го класса уже размещена в секции инициализации (перед оптимизацией);

C_{ij} – индекс класса, используемого в качестве типа данных для j -ой переменной i -го класса;

$t_{C_{ij}}^{\text{constructor}}$ – время работы конструктора без параметров при инициализации по умолчанию j -ой переменной i -го класса.

В ходе реализации практической части была разработана программа, осуществляющая поиск решения описанного метода оптимизации.

Программная реализация оптимизации и эксперимент

Для оценки эффективности программного продукта были проведены эксперименты.

Эксперименты проводились на машине со следующими характеристиками: процессором Intel(R) Core(TM) i7-4770 CPU @ 3.4 GHz и 32 ГБайта оперативной памяти.

Исходный код эксперимента основан на коде из листинга 1. В рабочей функции производился многократный запуск создания экземпляров неоптимизированного и оптимального классов, замеры времени создания экземпляров классов и вывод результатов эксперимента.

На рис. 1 представлены результаты эксперимента в виде графика, каждая точка которого представляет собой вычисленный выигрыш – разницу между временем выполнения неоптимизированной программы и временем выполнения оптимизированной программы.

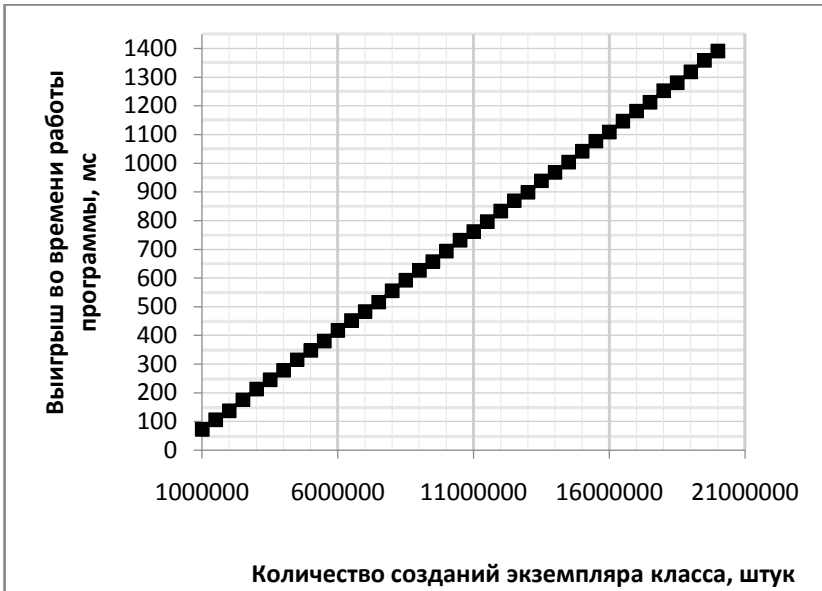


Рис. 1. Выигрыш во времени работы программы, представленный как разность времени работы программы до оптимизации и времени работы программы после оптимизации

В результате аппроксимации выявлена зависимость линейного характера:

$$y = 6,93827E - 05x + 4,39212E - 01, \quad (2)$$

где y – выигрыш во времени работы программы;

x – количество созданий экземпляра класса.

Проведенные эксперименты подтвердили эффективность реализованного метода оптимизации.

Заключение

Ценность полученных в результате исследований результатов в том, что получен специализированный программный продукт, осуществляющий оптимизацию создания экземпляров классов методом переноса кода установки начальных значений свойств класса из блока конструктора класса в секцию инициализации. Приводятся особенности метода оптимизации и результаты эксперимента, подтверждающие эффективность реализованного метода оптимизации. Получен-

ный программный продукт можно использовать для демонстрации оптимизационных методов на лабораторных работах по курсу «Проектирование оптимального программного обеспечения» либо для оптимизации программных продуктов на языке «С++».

Литература

1. *Томаев М. Х.* Технологии глобальной оптимизации пользовательских программных кодов // Автоматизация и управление в технических системах. 2015. № 3. С. 16–30. DOI: 10.12731/2306-1561-2015-3-2. URL: auts.esrae.ru/15-277.

2. *Гроппен В. О., Томаев М. Х.* Модели, алгоритмы и средства программной поддержки проектирования оптимальных программных продуктов // Автоматика и телемеханика. 2000. Вып. 11.

УДК 004.81

ОПТИМИЗАЦИЯ МЕТОДА РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ НА ОСНОВЕ АЛГОРИТМА СКЕЛЕТИЗАЦИИ

Проскурин А. Е., канд. техн. наук, доцент, alex_5_4@mail.ru

Токарева И. В., ст. преп., ticher_@bk.ru

Северо-Кавказский горно-металлургический институт

(государственный технологический университет),

г. Владикавказ, Россия

Аннотация. В данной статье рассматривается исследование методов распознавания изображений. За основу был взят метод, использующий алгоритм скелетизации, который был улучшен с помощью оптимизационных подходов.

Ключевые слова: машинное зрение, распознавание изображений, скелетизация, оптимизация.

OPTIMISATION OF METHOD OF IMAGE RECOGNITION BASED ON THE ALGORITHM OF CARCASS

Proskurin A. E., Tokareva I. V.

Abstract. This article discusses the study of image recognition methods. The method using the skeletalization algorithm was taken as a basis, which was improved with the help of optimization approaches.

Keywords: machine vision, image recognition, skeletalization, optimization.

Введение

Задача распознавания изображений в настоящее время является одной из актуальных задач в науке, технике и других областях, а также является неотъемлемой частью работы в области искусственного интеллекта. Исследование задач такого типа зародилось во второй половине прошлого века в пик популярности исследований нейронных сетей. Ярким примером было создание знаменитой модели восприятия информации мозгом («перцептрон») Фрэнком Розенблаттом.

Для решения задачи данной области требуются достаточно большие вычислительные ресурсы, поэтому одним из направлений в вычислительной технике является повышение эффективности за счет использования оптимизационных задач.

В теории развития задач распознавания можно отметить два основных направления:

- первое направление рассматривает способность распознавания графической информации у разнообразных видов биологических объектов и построение математических моделей работы нейронов мозга;

- вторым направлением принято считать развитие методов и теорий построения устройств, используемых для решения отдельных задач в прикладных целях.

Одной из таких прикладных задач является система распознавания текстовой информации при переводе печатного и рукописного текста в электронную форму, которые широко используются в современных системах автоматизированного документооборота.

Эта задача является одной из самых непростых и ресурсоемких в данной области. Сложность можно объяснить тем фактом, что даже человек, читая незнакомый рукописный текст, без учета контекста допускает незначительный процент ошибок.

Для оцифровки текстов при работе с бумажными носителями информации в основном используют оптические сканеры. При этой операции возникают различные искажения оригинального текста в виде помех, шумов, бликов, смазанностей изображения и т. п.

Далее в процессе распознавания отсканированного текста происходит поэтапная разбивка на блоки в виде абзацев, предложений, слов и символов. При этом для любого символа выполняется локализация и выделение, на основе которого составляется матрица. Она в дальнейшем подвергается обработке с целью снижения искажений на основе операций сглаживания, фильтрации и нормализации. В итоге образуются характерные признаки, на основе которых происходит классификация введенного изображения.

В задаче распознавания текста используют множество методов, к наиболее распространенным можно отнести следующие:

1. *Алгоритм скелетизации.* В данном методе используются бинарные образы, на основе которых происходит построение так называемых «скелетов» в виде каркаса символа состоящего из ребер и связывающих их узлов. В процессе определения соотношений ребер и узлов может быть создана некоторая матрица соответствия идентифи-

цируемому объекту. Данный метод имеет несколько видов реализаций.

2. *Инвариантные числа.* Данный метод основан на применении некоторых чисел, позволяющих не учитывать размер и расположение символов, связанное с их ориентацией. Это в дальнейшем позволяет создать таблицу соответствия данных чисел конкретному объекту.

3. *Поточечное процентное сравнение с эталоном.* Для распознавания, кроме указанных преобразований, дополнительно производится предварительные аффинные преобразования объекта. После чего выполняется сравнение с заранее созданной базой эталонов изображений.

Наиболее подходящим для реализации алгоритмом можно считать метод «скелетизации». Однако и он имеет некоторые недостатки в виде ошибок в распознавании некоторых «сложных» символов. Для таких случаев необходимо использовать различные фильтры, позволяющие избавиться от вышеуказанного недостатка.

Для построения скелета символа были созданы различные методы. К наиболее известным можно отнести следующие:

1. Метод Щепина.
2. Скелетизация с применением шаблонов.
3. Волновой метод.

Из перечисленных методов был выбран волновой метод.

Данный метод заключается в анализе пути прохождения сферической волны по изображению (рис. 1). На каждом шаге анализируется смещение центра масс точек, образующих новую генерацию волны, относительно его предыдущих положений.

Метод состоит из следующих шагов:

- построение скелета изображения с помощью сферической волны;
- оптимизация полученного скелета.

Рис. 1. Прохождение сферической волны по изображению

Отслеживание линий изображения производится путем отслеживания перемещения центра отрезка, образуемого крайними точками генерации волны (рисунок 2). После отслеживания возможно сглаживание отрезков.

Выявление увеличения «ширины» волны и разделения волны на дочерние позволяет установить точку предполагаемого соединения двух отрезков. Определение увеличения «ширины» волны производится путем сравнения «ширины» очередной генерации волны и ее среднего значения за N предыдущих генераций (N задается заранее). Причем мы получаем две точки (А, В) трассируемого отрезка. После разделения волны на две полуволны, мы получаем еще две пары точек (С, D) и (Е, F). Точка соединения отрезков лежит в шестиугольнике ABCDEF и первоначально устанавливается как центр масс этого многоугольника (рисунок 3). Коррекция возлагается на оптимизацию скелета изображения.

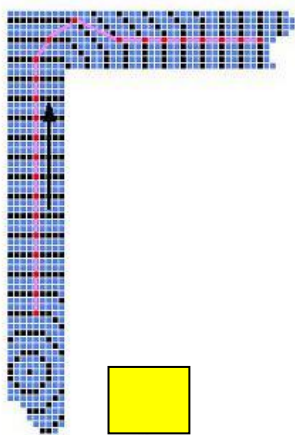


Рис. 2. Отслеживание линий изображения

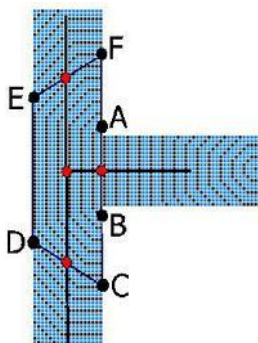


Рис. 3. Определение места соединения отрезков

Полученный скелет изображения не является оптимальным. Это связано прежде всего с тем, что мы имеем дело с растровым изображением, а значит, изображение имеет искажения тем большие, чем меньше размер изображения в символах.

Для уменьшения влияния искажений на получаемый скелет необходимо произвести его оптимизацию. В получаемом скелете возможно представление одного отрезка некоторой последовательностью ребер. Избавиться от этого можно анализом последовательности ребер, оценивая получающееся отклонение линии от прямой. В случае если отклонение находится в допустимых пределах, необходимо последовательность ребер заменить на одно.

Для оптимизации скелета просматриваются окрестности выделенных точек соединения отрезков, т. е. таких точек, где наблюдается разделение волны на полуволны.

Визуально скелет выглядит как утончение исходной формы образа объекта до набора одномерных линий. Полученные с помощью волнового алгоритма скелеты растровых изображений можно оптимизировать, уменьшив число точек, из которых состоит скелет. Тем самым снизится «похожесть» оптимизированного скелета с полученным после скелетизации скелетом, но снизится объем памяти, необходимый для хранения скелета. Формируем задачу поиска оптимального количества точек в алгоритме выделения скелета растрового изображения [4]:

$$\left\{ \begin{array}{l} \sum_j^m Z_j * \sum_i^n \sqrt{\frac{A_{ij} + B_{ij} + C_{ij}}{2} * \left(\frac{A_{ij} + B_{ij} + C_{ij}}{2} - A_{ij}\right) * \left(\frac{A_{ij} + B_{ij} + C_{ij}}{2} - B_{ij}\right) * \left(\frac{A_{ij} + B_{ij} + C_{ij}}{2} - C_{ij}\right)} \rightarrow \min \\ \forall j: Z_j = 1,0. \\ \sum_j^m Z_j = 1. \end{array} \right.$$

где A_{ij}, B_{ij}, C_{ij} – стороны треугольников,
 Z_j – булева переменная, обозначающая выборку.

Формальная постановка описывает задачу поиска оптимального количества точек в алгоритме выделения скелета растрового изображения. Очевидно, что любую задачу такого рода можно решить полным перебором, однако для большинства прикладных задач анализ 2^n вариантов даже с применением современных ЭВМ требует слишком больших затрат времени. Для повышения быстродействия вычислений лучше всего использовать оптимизационные методы, позволяющие сократить объем перебора. Наиболее популярными из них являются методы типа ветвей и границ и динамическое программирование.

Допустим, после скелетизации изображения русской заглавной буквы «Г» получен скелет, изображенный на рис. 4.

Полученный скелет можно оптимизировать, сократив количество точек, содержащихся в нем. Для этого строим треугольники, как

показано на рис. 5, где S_1, S_2, S_3 и S_4 обозначают площади треугольников.



Рис. 4. Пример скелета

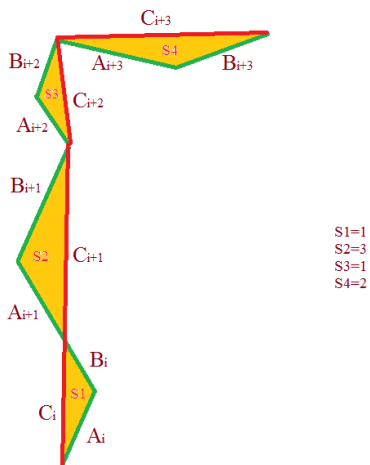


Рис. 5. Треугольники, построенные поверх скелета

Далее в соответствии с формулой (1) решаем оптимизационную задачу методом ветвей и границ (рис. 6).

Изначально скелет состоял из 7 рёбер и 8 точек. После применения алгоритма оптимизации количество рёбер сократилось до 5, а точек стало 6 (рис. 7).

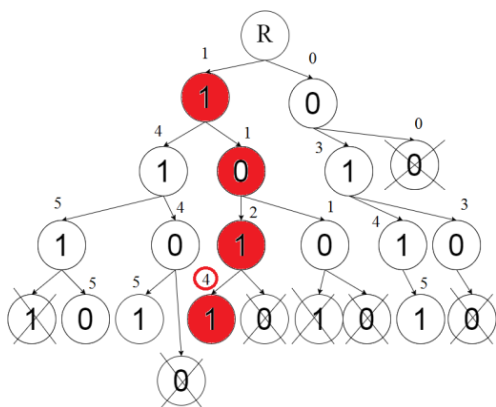


Рис. 6. Пример решения методом ветвей и границ (поиск с возвратом)



Рис. 7. Оптимизированный скелет

Литература

1. *Котович Н. В., Славин О. А.* Распознавание скелетных образов: Методы и средства работы с документами // Сборник трудов Института системного анализа РАН. М., 2000.

2. *Проскурин А. Е., Токарева И. В.* Использование модели «неокогнитрона» в методах распознавания объектов // ИТ-технологии: развитие и приложения: XV Ежегодная Международная научно-техническая конференция (12–14 декабря 2018 г.): Сборник докладов. Владикавказ: Северо-Кавказский горно-металлургический институт (государственный технологический университет), 2018. С. 266–270.

3. Задача распознавания образов [Электронный ресурс]. <http://www.ampersant.ru/glaz/>

4. *Проскурин А. Е., Дзарасов Д. А.* Поиск оптимального количества точек в алгоритме выделения скелета растрового изображения // ИТ-технологии: развитие и приложения: Материалы семинара. Владикавказ: СКГМИ (ГТУ), 2016. 94 с.

ТЕХНОЛОГИИ ЗАЩИТЫ ИНФОРМАЦИИ

УДК 004.942

РАЗРАБОТКА ФОРМАЛЬНОЙ МОДЕЛИ ИССЛЕДОВАНИЯ ТЕХНИЧЕСКИХ СРЕДСТВ ЗАЩИТЫ ИНФОРМАЦИИ В ГОСУДАРСТВЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

Бабенко А. А., канд. пед. наук, доцент, ba_benko@mail.ru,
Волгоградский государственный университет,
г. Волгоград, Россия

Аннотация. Рассмотрена проблема определения эффективных технических средств защиты информации в государственных информационных системах. Проанализированы угрозы безопасности информации в государственных информационных системах. Определены критерии оценки технических средств защиты информации в государственных информационных системах. Разработана формальная модель выбора наиболее эффективных технических средств защиты информации в государственных информационных системах.

Ключевые слова: государственная информационная система, информационная безопасность, технические средства защиты, система защиты информации.

DEVELOPMENT OF A FORMAL MODEL OF RESEARCH OF TECHNICAL MEANS OF INFORMATION PROTECTION IN PUBLIC INFORMATION SYSTEMS

Babenko A. A.

Annotation. *The problem of determining effective technical means of information protection in state information systems is considered. Analyzed threats to the security of information in state information systems. Criteria for evaluating technical information protection tools in state information systems are defined. A formal model has been developed for choosing the most effective technical means of protecting information in state information systems.*

Keywords: *state information system, information security, technical means of protection, information protection system.*

Введение

Актуальность проблемы обеспечения информационной безопасности (ИБ) информации в государственных информационных системах (ГИС) обоснована высоким спросом на системы такого класса. Эффективность ГИС в значительной степени зависит от уровня ее безопасности. Защита информации в ГИС обусловлена требованиями: ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации» [1–3].

В [4] отмечается роль ГИС – «для устойчивого функционирования информационной инфраструктуры Российской Федерации устанавливается необходимость обеспечения единства государственного регулирования, централизованный мониторинг и управление функционированием информационной инфраструктуры РФ на уровне информационных систем и центров обработки данных».

Исходя из этого, сформулируем цель данного исследования: выбор наиболее эффективных технических средств защиты информации в ГИС.

Анализ угроз безопасности информации в государственных информационных системах

Особенностями ГИС, влияющих на защищенность информации являются:

- сложный состав программно-аппаратных платформ и систем защиты информации (СЗИ);
- деление информационного потока на внутренний и внешний;
- территориальная распределенность компонентов;
- взаимодействие с открытыми сетями передачи данных;
- различные виды обрабатываемой информации, определяющие её ценность;
- требования правовых, технических и иных норм эксплуатации ГИС на различных этапах жизненного цикла.

По статистическим данным компании Infowatch, в I-м квартале 2017 года больше половины всех атак было направлено на автоматизированные рабочие места (АРМ) (табл. 1). Внутренний нарушитель является наиболее частой причиной утечек информации и составляет 64,5 %, на внешние атаки приходится 35,5 % [5].

Наиболее ценной информацией для злоумышленников являются персональные данные (ПДн) и платежная информация. Наиболее ве-

роятными каналами утечки этих видов информации являются сеть и бумажные документы.

Таблица 1

Угрозы АРМ ГИС

Угроза	Техника воздействия
Подбор паролей	Попытки подбора аутентификационной информации для доступа к сервисам и ресурсам контролируемых организаций – RDP, SSH, SMB, DB, Web
Нарушение политик ИБ	Нарушение пользователями/администраторами контролируемых ресурсов требований политик ИБ в части использования устаревших версий или недоверенного ПО, которое может быть использовано злоумышленником для атаки путём эксплуатации уязвимости. Также использование ресурсов компании для получения собственной выгоды (майнинг bitcoin/ethereum). Использование торрент-трекеров
Вредоносное ПО	Заражение конечной системы, распространение вируса по локальной сети, отключение/блокировка служб, препятствующих распространению вируса, попытки проведения иных атак внутри сети для получения критичной информации и передачи на командные серверы
Попытки эксплуатации уязвимостей	Использование недостатков в системе для нарушения контроля целостности данных и воздействие на правильную работу системы. Уязвимость может быть результатом ошибок программирования; недостатков, допущенных при проектировании системы; ошибок конфигурации; отсутствия обновлений. Некоторые уязвимости известны только теоретически, другие же активно используются и имеют известные эксплойты

Отметим, что наряду с традиционными угрозами нарушения конфиденциальности, целостности и доступности для информации, обрабатываемой в ГИС, характерны:

- угроза несанкционированного доступа (НСД);
- угроза отказа сетевого оборудования;
- вредоносное программное обеспечение;
- хищение данных;
- промышленный шпионаж;
- компрометация учетных данных;

- подмена исполнительных модулей;
- человеческий фактор.

Наибольший ущерб для ГИС наносят угрозы нарушения целостности и угрозы, присущие центрам обработки данных (ЦОД), серверам и баз данных [6, 7]. Среднему ущербу могут подвергнуть угрозы получения НСД к информации в ГИС, АРМ пользователей или файловому серверу [8]. Анализ угроз информации в ГИС позволяет соотнести угрозы нарушения ИБ с техническими мерами и средствами ее предотвращения [1, 2, 9].

Определение критериев оценки технических средств защиты информации в государственных информационных системах

Состав мер защиты информации в ГИС представлен в [1, 2]. Если организация подключена к ГИС, то [1] обязывает аттестовать систему, а для защиты информации должны применяться только сертифицированные средства защиты информации, имеющие действующие сертификаты ФСТЭК или ФСБ. Государственный реестр сертифицированных СЗИ и перечень СЗИ, сертифицированных ФСБ России представлен в [10], а перечень ФСТЭК в [11].

Мера защиты информации считается выполненной, если:

- 1) в организационно-распорядительных документах по обеспечению безопасности информации определена процедура установки средств безопасности в ГИС;
- 2) принятые организационные и технические меры, а также используемые СЗИ исключают несанкционированную установку неразрешенных средств или их компонент;
- 3) в ГИС отсутствуют запрещенные к использованию программные и технические средства или их компоненты, целостность установленного СЗИ не нарушена;
- 4) обеспечивается периодический контроль установленного в ГИС средства защиты на предмет его соответствия перечню разрешенных к установке средств защиты в ГИС.

Для СЗИ, применяемых в ГИС, регулятор предъявляет требования. Так в ГОСТ Р ИСО / МЭК 15408 определяется профиль защиты (ПЗ) – совокупность требований безопасности в отношении определенной категории изделий ИТ, независящую от реализации.

Для ГИС, обрабатывающих информацию, не содержащую сведения, отнесенные к ГТ, актуальным ПЗ является четвертый класс.

В результате анализа технических СЗИ в ГИС установлены классы используемых СЗИ: межсетевые экраны (МЭ), средства обнаружения вторжений (СОВ), средства антивирусной защиты информации (САВЗ), средства доверенной загрузки (СДЗ), средств контроля съемных носителей (СКН). Эти классы были проанализированы по следующей схеме: 1) выявление нормативно-правовых актов, содержащих требования к классу СЗИ; 2) определение класса СЗИ, типа и области его применения; 3) определение соответствия классов защищенности ГИС и классов защиты СЗИ; 4) определение соответствия СЗИ с типом и классом защиты ГИС. Данная процедура является универсальной и может быть использована для выбора не только технических средств защиты информации, но и программных и применяется для всех классов информационных систем.

Учитывая требования нормативно-правовых актов, сформулируем критерии оценки для анализа технических средств защиты информации в ГИС (табл. 2).

Таблица 2

Критерии оценки для анализа технических СЗИ в ГИС

Обозначение	Название
К1	Срок действия сертификата ФСТЭК/ФСБ
К2	Многофункциональность, выполнение СЗИ по нескольким требованиям, например к САВЗ и МЭ
К3	Уровень контроля на отсутствие НДВ в ПО СЗИ
К4	Стоимость СЗИ

Перечисленные критерии будут использоваться для определения наиболее эффективных СЗИ ГИС.

Формальная модель выбора наиболее эффективных технических средств защиты информации в государственных информационных системах

Формализовано процедуру оценки эффективности средств защиты информации в ГИС можно представить вектором критериев.

$$K = (K_1, K_2, K_3, K_4), \quad (1)$$

где K_1 – срок действия сертификата ФСТЭК/ФСБ:

$$K_1 = \begin{cases} 0, & \text{заканчивается в 2019} \\ 0,5 & \text{заканчивается в 2020} \\ 1, & \text{заканчивается в 2021–2023} \end{cases} \quad (2)$$

K_2 – многофункциональность – выполнение средством защиты нескольких функций защиты (например, программный комплекс Электронный замок «Витязь» версия 2.2, на соответствие требованиям СДЗ (ИТ.СДЗ.УБ2.ПЗ) и САВЗ (ИТ.САВЗ.Г2.ПЗ)):

$$K_2 = \begin{cases} 0 - \text{нет} \\ 1 - \text{да;} \end{cases} \quad (3)$$

K_3 – соответствие заявленного в программном обеспечении средства защиты информации уровня контроля на отсутствие недеklarированных возможностей (НДВ):

$$K_3 = \begin{cases} 0 - \text{нет} \\ 1 - \text{да;} \end{cases} \quad (4)$$

K_4 – стоимость СЗИ ГИС;

$$K_4 = \begin{cases} 0, \text{ высокая, } C_i > C_{cp} \\ 1, \text{ низкая, } C_i < C_{cp}, \end{cases} \quad (5)$$

где C_i – стоимость i -го СЗИ ГИС;

C_{cp} – среднее значение стоимости в группе СЗИ ГИС;

n – количество в группе СЗИ ГИС.

$$C_{cp} = \frac{\sum_{i=1}^n C_i}{n}. \quad (6)$$

Идеальному СЗ соответствует вектор K^* , в котором все значения критериев равны единице. Для оценки эффективности СЗИ вводится скалярная величина, равная Евклидову расстоянию между наилучшим вектором и вектором критериев, полученному для i -го оцениваемого СЗИ:

$$K^i = (K_1^i, K_2^i, K_3^i, K_4^i). \quad (7)$$

Евклидово расстояние для i -ой группы СЗИ ГИС рассчитывается по формуле:

$$E^i = \sqrt{\sum_{j=1}^4 (K_j^* - K_j^i)^2}. \quad (8)$$

СЗИ, для которой расстояние до наилучшего вектора окажется наименьшим, можно считать наиболее эффективным для защиты информации в ГИС.

Предложенная формальная модель позволяет выбрать наиболее эффективные СЗИ для защиты информации в ГИС.

Заключение

В результате проведенных экспериментов были установлены эффективные технические средства для защиты информации в ГИС:

1) МЭ «Универсальный шлюз безопасности «UserGate UTM», выполняющий требования к МЭ ИТ.МЭ.А4.ПЗ, ИТ.МЭ.Б4.ПЗ и требования к СОВ ИТ.СОВ.С4.ПЗ;

2) Маршрутизаторы ESR-1000, ESR-200, ESR-100, выполняющие требования к МЭ ИТ.МЭ.А5.ПЗ;

3) МЭ Huawei Eudemon (модель Eudemon 8000E-X3) версии V500, маршрутизаторы серии Huawei AR (модели: AR2220E, AR2240, AR161FG-L) версии V200, МЭ серии Cisco ASA 5500-X (модели: ASA 5506-X, ASA 5508-X, ASA 5516-X) с установленным ПО Cisco ASA версии 9.x, выполняющие требования к МЭ ИТ.МЭ.А6.ПЗ и ИТ.МЭ.Б6.ПЗ;

4) ПАК «Горизонт-ВС», выполняющий требования к ИТ.СДЗ.ПР4.ПЗ.

Перечисленные технические СЗИ рекомендуются для защиты информации в ГИС, не обрабатывающих сведения, отнесенные к государственной тайне, содержащих в своём составе ИСПДн и имеющие подключения к сетям общего пользования 2 класса.

Литература

1. Об утверждении «Требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах»: Приказ ФСТЭК России от 11 февраля 2013 г. № 17 (ред. от 15.02.2017). URL: <http://fstec.ru/component/attachments/download/567> (дата обращения: 05.03.2019).

2. Методический документ «Меры защиты информации в государственных информационных системах», утверждён ФСТЭК России от 11 февраля 2014 г. [Электронный ресурс.] URL: <https://fstec.ru/component/attachments/download/675> (дата обращения: 05.03.2019).

3. О безопасности критической информационной инфраструктуры Российской Федерации: Федеральный закон от 26.07.2017 № 187-ФЗ // Собрание законодательства РФ. 2017. № 31 (Часть I). Ст. 4736.

4. О Стратегии развития информационного общества в Российской Федерации на 2017–2030 годы: Указ Президента РФ от 09.05.2017 № 203 // Собрание законодательства РФ. 2017. № 20. Ст. 2901.

5. Глобальное исследование утечек конфиденциальной информации в I полугодии 2018 года. URL: https://www.infowatch.ru/report2018_half (дата обращения 05.03.2019).

6. *Козунова С. С., Бабенко А. А.* Автоматизация управления инвестициями в информационную безопасность предприятия // Вестник компьютерных и информационных технологий. 2015. № 3 (127). С. 38–44.

7. *Нестеровский И. П., Язов Ю. К.* Возможный подход к оценке ущерба от реализации угроз безопасности информации, обрабатываемой в государственных информационных системах // Вопросы кибербезопасности. 2015. 2 (10). С. 20–25

8. *Козунова С. С., Бабенко А. А.* Система оптимизации рисков инвестирования информационной безопасности промышленных предприятий // Вестник компьютерных и информационных технологий. 2016. № 7 (145). С. 22–29.

9. Методика определения актуальных угроз безопасности персональных данных при их обработке в информационных системах персональных данных (утв. зам. директора ФСТЭК России 14 февраля 2008 г.) URL: <https://fstec.ru/component/attachments/download/290> (дата обращения: 05.03.2019).

10. Перечень средств защиты информации, сертифицированных ФСБ России. // URL: <http://clsz.fsb.ru/certification.htm> (дата обращения: 05.03.2019).

11. Государственный реестр сертифицированных средств защиты информации // URL: <https://fstec.ru/tekhnicheskaya-zaschita-informatsii/dokumenty-po-sertifickatsii/153-sistemasertifikatsii/591> (дата обращения: 05.03.2019).

АНАЛИЗ МЕТОДОВ КРИПТОЗАЩИТЫ ИНФОРМАЦИИ В МНОГОКАНАЛЬНЫХ ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ

Ибрагимов Б.Г., д-р техн. наук, профессор
Азербайджанский технический университет,
г. Баку, Республика Азербайджан, i.bayram@mail.ru

Мамедов Т.Г., адъюнкт
Военная академия Вооруженных сил,
г. Баку, Республика Азербайджан

Аннотация. Представлен аналитический обзор работ, посвященных вопросам передачи и криптозащиты информации в многоканальных телекоммуникационных системах. Проанализированы методы повышения криптозащиты сообщений от несанкционированного доступа к каналам связи при передаче информации смешанного типа трафика. Предложен новый подход к защите информации от несанкционированного доступа в многоканальных телекоммуникационных системах, учитывающий особенности криптографического метода шифрования и кодирования передаваемых сообщений по каналам связи.

Ключевые слова: многоканальная телекоммуникационная система, защита информации, криптография, кодирование, несанкционированный доступ, криптозащита.

ANALYSIS OF THE METHODS INFORMATION CRYPTO PROTECTION IN MULTICHANNEL TELECOMMUNICATION SYSTEMS

Ibrahimov B.G., Mamedov T.G.

Abstract. An analytical review works on the issues transmission and cryptographic information in multichannel telecommunication systems is presented. The methods improving the cryptographic protection messages from unauthorized access to communication channels during the transmission information of a mixed type traffic are analyzed. A new approach to protecting information from unauthorized access in multichannel telecommunication systems is proposed, taking into account

the peculiarities of the cryptographic method encrypting and encoding transmitted messages over communication channels.

Keywords: *multichannel telecommunications system, information protection, encryption, cryptography, unauthorized access, cryptographic protection.*

Введение

Интенсивное развитие многоканальных телекоммуникационных систем (МТС) с использованием архитектурной концепции NGN требует нового подхода для эффективной защиты информации, которая является одной из главных необходимых условий развития современных инфокоммуникационных технологий и систем связи.

Известно [1, 2], что в течение многих лет во всех странах мира защита информации являлась прерогативой государства, а на развитие и использование технологий защиты информации устанавливалась государственная монополия. Средства и алгоритмы защиты информации использовались в основном в военной, правительственной и дипломатической связи под жестким правительственным контролем. В этих условиях происходило развитие науки о защите информации – криптологии [3, 4].

На основе системно-технического анализа установлено [2, 4, 5–7], что криптология диалектически объединяет две науки:

1. Криптография – новая наука, которая изучает методы защиты информации от несанкционированного доступа;
2. Криптоанализ является наукой о методах несанкционированного доступа к защищаемой информации.

Следовательно, развитие банковских технологий, деятельность коммерческих структур, избирательные компании конкурирующих политических партий, изобретательская и другие виды деятельности, требующие защиты информации, обусловили переживаемый в настоящее время нашей страной спрос на криптографические технологии и специалистов в области криптологии [1, 7].

С учетом вышеизложенного, возникает важный вопрос – изучение описания наиболее часто используемых алгоритмов, методов и протоколов криптографической защиты информации и особенности их использования в технике электросвязи.

В данной работе рассматривается решение вышесформулированной задачи – создание нового подхода к защите информации, позволяющее анализировать методы передачи и криптозащиты информации в многоканальных телекоммуникационных системах.

Общая постановка задачи

Проблема качества передачи и защиты информации в сети исследуется при определённой модели передачи сообщений и при определённом сценарии действий злоумышленника.

С целью анализа и обзора при защите информации предлагается новый подход, который учитывает алгоритмы работы когерентного сетевого кодирования, когда на узлах сети связи выполняются определённые линейные операции над входящими сообщениями и сценарий работы злоумышленника.

В сценарии работы злоумышленника имеются подмножества МТС связи, доступных злоумышленнику, то есть он может подслушивать или записывать передаваемую по ним информацию с намерением определить сообщения источников. Такая проблема впервые рассматривалась [1, 4] авторским коллективом в первой основополагающей работе по сетевому кодированию. В последующие годы эти исследования продолжены, дополнены и обобщены в работах Юнга в соавторстве с другими учёными [3–5].

В МТС при передаче информации в открытых каналах связи (КС) широко используются криптографические методы защиты информации от несанкционированного доступа к абонентской линии: метод шифрования и метод кодирования.

Для достижения поставленной задачи предлагается новый подход к криптографической защите информации от несанкционированного доступа в МТС с учетом особенности криптографического метода шифрования и кодирования передаваемых сообщений по каналам связи.

Анализ нового подхода защиты информации от несанкционированного доступа

С целью анализа методов передачи и криптозащиты информации рассмотрим описание наиболее часто используемых алгоритмов и протоколов, достаточных для реализации действующих криптосистем.

Для создания нового подхода на рис. 1 представлена структурная схема, которая описывает функциональные процессы криптографической защиты информации от несанкционированного доступа в МТС.

Из рис. 1 следует, что система защиты информации от несанкционированного доступа состоит из двух важных системных функциональных блоков – системы передачи и шифрования исходных сооб-

щений и обратное. В обобщенном виде задача защиты информации от несанкционированного доступа на основе рис. 1 может быть сформулирована следующим образом. Абонент-А, являющийся источником и инициатором передачи сообщения X , намерен передать это сообщение к абоненту-Б, называемому получателем сообщения, при этом абонент-А хочет быть уверенным в том, что в процессе передачи сообщения никто не сможет прочесть его, т. е. раскрыть содержание, изменить, или создать другое похожее сообщение и передать его абоненту-Б от имени абонента-А. В свою очередь абонент-Б, получив сообщение, хочет быть уверенным в возможности в любой момент установить и доказать его получение именно от абонента-А, а также тот факт, что данное сообщение не является повтором более раннего.



Рис. 1. Структурно-функциональная схема защиты информации от несанкционированного доступа

В данном случае передаваемое сообщение является информацией смешанного типа трафика (информационный файл, речь, документальное сообщение и др.). В схеме сообщение X обычно называется исходным или открытым сообщением или текстом.

В качестве сообщения X может выступать, например, оцифрованная речь телефонного разговора, текстовый или другой информационный файл, факсимильное сообщение и др. виды информации, подлежащие защите от несанкционированного доступа. Для защиты сообщения X от несанкционированного доступа применяется некоторое преобразование исходного сообщения, называемое процессом шифрования.

Данное преобразование предполагает сопоставление исходному сообщению X некоторой функции $E(X)$ превращающей его в нечитаемое без специальных средств зашифрованное сообщение Y :

$$Y = E(X), \quad X \in (x_1, x_2, \dots, x_i), \quad (1)$$

Однако зашифрованное сообщение Y обычно имеет родственную исходному X структуру, т. е. длина сообщения Y обычно равна или превышает длину сообщения X :

$$L_{dc}(Y) \geq L_{dc}(X). \quad (2)$$

Следует отметить, что в ряде случаев в процессе шифрования используется сжатие информации в сообщении X , с целью сокращения избыточности сообщения, при этом сообщение Y может оказаться короче сообщения X , однако это не является следствием шифрования как такового [4, 6].

Из рис. 1, выражений (1) и (2) следует, что исследуемый подход криптографической защиты информации учитывает также два вида несанкционированного доступа [6, 7]:

- 1) к каналу связи;
- 2) к информации, передаваемой по каналу связи.

На выходе канала связи для восстановления исходного сообщения X из зашифрованного сообщения Y применяется операция дешифрования с использованием функции $D(Y)$:

$$X = D[E(X)] \Rightarrow D(Y) = X. \quad (3)$$

Выражение (3) характеризует совокупность алгоритмов процессов шифрования и дешифрования, которая является алгоритмом криптозащиты информации в целом.

Исследования и расчет параметров криптографических методов защиты информации в МТС

Известно [1, 4, 7], что основными современными криптографическими методами защиты сообщения являются методы шифрования (замена, перестановка, аналитическое преобразование, гаммирование и комбинированные) и методы кодирования (смысловое и символьные). Здесь под методом шифрования подразумевается процесс, в котором криптографическому преобразованию подвергается каждый символ открытого текста. В данном методе в шифре всегда различают два элемента: алгоритм и ключ.

В схеме рис. 1 видно, что алгоритм, реализующий функцию шифрования $E(X)$, называется алгоритмом шифрования. Соответственно,

алгоритмом дешифрования называется алгоритм, реализующий функцию $D(Y)$. Совокупность алгоритмов шифрования и дешифрования будем называть алгоритмом криптозащиты системы телекоммуникации.

Стойкость этих алгоритмов базируется на вычислительной сложности решения некоторых задач. Если защита информации с помощью какого-то алгоритма криптозащиты обеспечивается секретностью, т. е. недоступностью структуры самого алгоритма, то такой алгоритм называется ограниченным. Ограниченные алгоритмы не могут эксплуатироваться широким кругом пользователей, поскольку утечка информации о структуре алгоритма приводит к необратимой потере секретности сразу для всех пользователей, лишая смысла всю систему защиты в целом [3, 5].

Рассматриваемый алгоритм для методов шифрования позволяет использовать сравнительно короткий ключ для шифрования сколь угодно большого текста. Для защиты сообщения в многоканальных телекоммуникационных системах в основном используются метод шифры.

В системе передачи и защиты информации (рис. 1) используемый ключ обычно является внешним по отношению к алгоритмам шифрования и дешифрования информации, определяющей конкретный вид зашифрованного сообщения Y при данном исходном сообщении X .

На основе нового подхода общая формула при использовании методов шифрования для i -го символа алфавита выражается следующим образом [6]:

$$Y_i = k_1 x_i + k_2 \pmod{n}, \quad (4)$$

где k_1 и k_2 – являются константы;

x_i – i -й символ открытого текста, т. е. номер буквы в алфавите;

n – длина используемого алфавита.

Выражение (4) определяет в криптографии широко используемый метод замены. При замене каждой буквы алфавита открытого текста ставится в соответствие одна буква шифр текста из этого же алфавита.

На основе исследования установлено [4, 8, 9], что подход к защите информации в многоканальных телекоммуникационных системах от несанкционированного доступа должна обеспечить:

- конфиденциальность – информация должна быть защищена от несанкционированного прочтения как при хранении, так и при передаче сообщения в МТС;

- контроль доступа – информация должна быть доступна только для того, для кого она предназначена;
- аутентификацию – возможность однозначно подтвердить подлинность информации и идентифицировать отправителя в МТС;
- целостность – информация должна быть защищена от несанкционированной модификации при хранении и передаче в МТС;
- неотречаемость – отправитель не может отказаться от факта отправки данного сообщения в МТС.

Учитывая особенности нового подхода, криптографическая защита информации описывается следующей формулой:

$$Y_i(x_i, k_i) = x_i + k_i \pmod{n}, \quad (5)$$

где k_i – i -я буква ключа, в качестве которого используется слово или фраза.

На основе исследуемого метода и алгоритма криптозащиты информации определим открытый текст слова «ЗАМЕНА».

С учетом методов подстановки и выражения (5) определяются показатели зашифрованного сообщения $y_i, i = \overline{1, 6}$ следующим образом:

$$y_2 = 1 + 12 \pmod{33} = 13 \rightarrow M,$$

$$y_3 = 13 + 31 \pmod{33} = 11 \rightarrow K,$$

$$y_4 = 6 + 24 \pmod{33} = 30 \rightarrow \text{Э},$$

$$y_5 = 14 + 11 \pmod{33} = 25 \rightarrow \text{Ш},$$

$$y_6 = 1 + 12 \pmod{33} = 13 \rightarrow M.$$

Аналогично задача для азербайджанского и английского языка решается, используя таблицы кодирования букв азербайджанского и английского алфавита, соответственно.

Анализ защиты информации в МТС с помощью эффективных методов кодирования

Одним из важных криптографических методов и алгоритмов защиты информации в МТС является метод сетевого кодирования. Ко-

дированием называется процесс преобразования сообщений в комбинации из дискретных сигналов, т. е. процесс замены элементов открытого текста (символов, комбинации кодов и слов) кодами [1, 2, 5–8]. В этой работе авторы рассматривают одноадресную передачу в сети и применяют случайное сетевое кодирование на базе корректирующих кодов.

Учитывая вышеизложенные алгоритмы, сущность предлагаемого подхода заключается в том, что при передаче сообщения конструкция сетевого кодирования состоит из нескольких операций [8–10]:

- выбор подходящих элементов и параметров кода:

$$N_k(n) = G(n, m, d, r),$$

где r, m – число проверочных и информационных символов в кодовых комбинациях;

n – общая длина кодовой комбинации;

d – кодовое расстояние.

- анализ и выбор последовательности символов сетевого кодирования над полем:

$$Q_{ck}(n, q, r) = GF^{(n-r)}(q, m), \quad m = (n-r) \in M. \quad (6)$$

С учетом параметров конструкция сетевого кода передаваемого сообщения описывается следующим образом:

$$X = D[E(X)] = N_k(n) + GF^{(n-r)}(q, m). \quad (7)$$

Из (7) следует, что источник генерирует сообщение m из общего множества M , представляющее собой последовательность символов над полем $GF^{(n-r)}(q, m)$ и состоит из двух слагаемых. Источник генерирует также сеансовый ключ k из множества K .

Заключение

В результате исследования предложен новый подход к криптографической защите информации в МТС от несанкционированного доступа, учитывающий методы сетевого кодирования и шифрования.

Разработана структурно-функциональная схема защиты информации от несанкционированного доступа абонентской и сетевой линий связи, обеспечивающая совершенную секретность – статистическую независимость передаваемого сообщения от исходного сообщения. Получены аналитические выражения, позволяющие оценить показатели криптостойкости системы защиты информации в телекоммуникационных системах.

Литература

1. *Коржик В. И.* Основы криптографии. СПб.: ИЦ Интермедия, 2016. 296 с.

2. *Ибрагимов Б. Г., Ибрагимов Г. Г.* Исследование криптографического метода и средств защиты передаваемой информации в системах телекоммуникаций // Научно-технический журнал. Ученые записки. Азербайджанский технический университет. 2018. № 1. С. 40–45.

3. *Peeters E.* Advanced DPA Theory and Practice: Towards the Security Limits of Secure Embedded Circuits / E. Peeters. New York : Springer-Verlag New York, 2013. 139 p.

4. *Петраков А. Б.* Основы практической защиты информации. М.: СОЛОН-Пресс, 2005. 384 с.

5. *Cai N., Yeung R. W.* Secure Network Coding on a Wiretap Network // IEEE Trans. on Inf. Theory. 2011. V. 57. № 1. P. 424–435.

6. *Ибрагимов Б. Г., Гаджиев Т. М.* Эффективность криптографических методов активной защиты информации в системе телекоммуникаций // Сборник докладов XV Международной НТК «ИТ-технологии: Развитие и приложения». 12–14 декабря 2018. Владикавказ, 2018. С. 107–114.

7. *Баричев С. Г., Гончаров В. В., Серов Р. Е.* Основы современной криптографии. М.: Горячая линия. Телеком, 2001. 120 с.

8. *Шеннон К.* Работы по теории информации и кибернетике / Пер. с англ. под ред. Н. А. Железнова. М.: ИЛ, 1963. 829 с.

9. *Оков И. Н.* Криптографические системы защиты информации. СПб.: ВУС, 2001. 236 с.

10. *Ibrahimov B. G., Humbatov R. T., Ibrahimov R. F.* Cryptographic Methods and Means Protection Transmitted Information in Telecommunication Systems // The Proceedings 18-th Conference on Technology, Culture and International Stability. IEEE Explore. Elsevier. IFAC-Papers OnLine, (Scopus). 2018. Volume 51. Issue 30. Pp. 821–824.

ТЕХНОЛОГИИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

УДК 004.4

ЯЗЫКОВЫЕ И ПРОГРАММНЫЕ СРЕДСТВА ОБЛАЧНОГО ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ

Касьянов В. Н., д-р физ.-мат. наук, профессор, kvn@iis.nsk.su
Касьянова Е. В., канд. физ.-мат. наук, доцент, kev@iis.nsk.su
Институт систем информатики им. А. П. Ершова Сибирского
отделения РАН, Новосибирский государственный университет,
г. Новосибирск, Россия

Аннотация. Работа посвящена языковым и программным средствам поддержки облачного параллельного программирования, создаваемых в Институте систем информатики СО РАН при поддержке Российского научного фонда.

Ключевые слова: аннотированное программирование, иерархический граф, параллельное программирование, язык функционального программирования.

LANGUAGE AND SOFTWARE MEANS OF CLOUD PARALLEL PROGRAMMING

Kasyanov V. N., Kasyanova E. V.

Abstract. *The work is devoted to language and software means of supporting cloud parallel programming being under development at Institute of Informatics System with support of Russian Science Foundation.*

Keywords: *annotated programming, hierarchical graph, parallel programming, functional programming language.*

Введение

Параллельные вычисления являются одной из главных парадигм современного программирования и охватывают чрезвычайно широкий круг вопросов разработки программ. Ввиду значительно более слож-

ной природы параллельных вычислений, по сравнению с последовательными, большое значение приобретают методы автоматизации разработки параллельного программного обеспечения, основанные на применении техники формальных моделей, функциональных спецификаций и преобразований параллельных программ.

Первым языком функционального программирования был язык Лисп, разработанный в 1961 г. американским ученым Дж. Маккарти. Хотя язык и получил широкую известность, благодаря его большей выразительности и элегантности по сравнению с традиционными языками, его применимость ограничивалась в основном задачами искусственного интеллекта. Новый период функционального программирования начался с Тьюринговой лекции 1978 г. изобретателя Фортрана Дж. Бекуса «Может ли программирование освободиться от бремени фон-неймановского стиля? Функциональный стиль и его алгебра программ». Это новое понимание и более широкое принятие функционального программирования определилось в первую очередь начатым в эти годы процессом по переходу к рассмотрению задачи программирования в ее полном контексте, начиная со спецификации задачи и логического анализа ее разрешимости, побочным продуктом которого является сама программа. Разработка функциональных методов параллельного программирования успешно продолжилась в конце 70-х в языках VAL и БАРС, а также в ряде более современных проектов DCF, Пифагор, COLAMO и др., среди которых следует выделить язык SISAL (аббревиатура с английского выражения Streams and Iterations in a Single Assignment Language) [1], первая версия которого относится к 1983 г. SISAL разрабатывался как язык функционального программирования, специально ориентированный на параллельную обработку и на замену языка Фортран на суперкомпьютерах в научных вычислениях. О реальном вытеснении говорить еще рано, но SISAL как язык параллельного программирования достаточно интересен сам по себе и уже нашел свое применение в десятках организаций разных стран мира. Существует несколько реализаций языка SISAL (версии 1.2) для суперЭВМ, в частности, на Denelcor HEP, Vax 11-780, Cray-1, Cray-X/MP выполнены работы по созданию прототипа оптимизирующего компилятора с языка SISAL 1.2 в распределенные программы для вычислителей, аппаратно или программно поддерживающих многонитевые вычисления, такие как, например, TERA, *T, TAM и MIDC. Ливерморская национальная лаборатория и Манчестерский университет разработали усовершенствованную версию языка SISAL-90, которая пока еще нигде не была реализована.

Доклад посвящен языковым и программным средствам облачного параллельного программирования CPPS, разрабатываемым в лаборатории конструирования и оптимизации программ Института систем информатики им. А.П. Ершова СО РАН за счет гранта Российского научного фонда (проект 18-11-00118).

1. Система параллельного программирования CPPS

Современные подходы к разработке параллельных программ в основном являются архитектурно-ориентированными, когда для достижения эффективной работы создаваемые программы тесно связаны с архитектурами параллельных вычислительных систем, на которых они выполняются и, как правило, разрабатываются. Поэтому требования к квалификации разработчиков параллельных программ весьма высоки, тем более что протестировать и отладить параллельную программу намного сложнее, чем последовательную, а проблема верификации параллельных программ весьма далека от решения не только практически, но и теоретически. Причем лишь у узкого круга отечественных пользователей есть доступ к высокопроизводительной вычислительной технике.

Более того, в современной вычислительной технике идет постоянная смена архитектурных парадигм, что, в свою очередь, ведет к проблеме переносимости уже разработанных параллельных программ. Приходится постоянно адаптировать уже созданный продукт под изменившиеся аппаратные средства. Это обуславливается тем, что различные параллельные вычислительные системы имеют свойственные только им ресурсные ограничения, которые необходимо учитывать во время разработки программы. Проведение таких адаптаций является весьма интеллектуальной задачей, требующей существенного переписывания параллельных программ и выполнения практически заново их верификации и отладки. В результате адаптированные параллельные программы зачастую содержат новые ошибки и не являются столь эффективными, как хотелось и могло бы быть.

Поэтому представляется весьма перспективным выполняемый в ИСИ СО РАН проект по разработке языковых и программных средств, поддерживающих создание, верификацию и отладку архитектурно-независимых параллельных программ и их корректное преобразование в эффективный код параллельных вычислительных систем различных архитектур с помощью семантических преобразований. Будут разработаны методы и создана экспериментальная версия облачной

расширяемой интегрированной визуальной системы параллельного программирования CPPS на языке Cloud Sisal [2], который продолжает традицию предыдущих версий языка SISAL, оставаясь функциональным потоковым языком, ориентированным на написание больших научных программ, и расширяет их возможности средствами поддержки облачных вычислений.

Функциональная семантика языка Cloud Sisal гарантирует детерминированные результаты для параллельной и последовательной реализации – то, что невозможно гарантировать для традиционных императивных языков, подобных языкам Фортран или С. Более того, неявный параллелизм языка снимает необходимость переписывания исходного кода при переносе его с одного вычислителя на другой. Гарантировано, что Cloud-Sisal-программа, правильно исполняющаяся на персональном компьютере, будет правильно исполняться на любом высокоскоростном параллельном или распределенном вычислителе. При этом методы аннотированного программирования [3] и конкретизирующих преобразований [4], поддерживаемые системой CPPS, позволят в рамках декларативного стиля программирования настраивать процессы адаптации переносимых параллельных программ на классы задач и архитектуру вычислителя при сохранении их корректности, а также получать более эффективный параллельный код за счет использования при адаптации знаний пользователя о задаче, программе и вычислителе, выраженных в аннотациях.

Система CPPS разрабатывается как интегрированная облачная среда программирования на языке Cloud Sisal, которая содержит как интерпретатор, поддерживающий диалоговое взаимодействие с пользователем при построении и отладке программы, так и оптимизирующий кросс-компилятор, осуществляющий построение параллельной программы по ее функциональной спецификации.

В рамках CPPS прикладной программист будет иметь возможность через браузер создавать, верифицировать и отлаживать Cloud-Sisal-программу в визуальном стиле и без учета целевого вычислителя, а затем с помощью оптимизирующего кросс-компилятора производить настройку отлаженной программы на тот или другой супервычислитель, доступный ему по сети, с целью достижения высокой эффективности исполнения получаемой параллельной программы, а также передавать построенную программу супервычислителю на счет и получать результаты.

Система CPPS использует внутреннее теоретико-графовое представление функциональных программ, которое ориентировано на их

визуальную обработку и основано на атрибутированных иерархических графах [5], и осуществляет построение наглядных изображений графовых внутренних представлений обрабатываемых Cloud-Sisal-программ [6]. Она будет также поддерживать как создание визуальных представлений внутренних структур данных, возникающих в компиляторе при построении параллельных программ, и динамических процессов, возникающих при исполнении построенных параллельных программ, так и их использование для управляемой оптимизации с целью повышения рабочих характеристик параллельных программ, получаемых с помощью компилятора по их функциональным спецификациям.

2. Язык Cloud Sisal

Язык Cloud Sisal обладает обычными преимуществами языков функционального программирования, такими как, например, однократное присваивание (т. е. каждая переменная в программе определяется только один раз), но содержит массивы и циклы, которые не присущи функциональным языкам. Рассмотрим следующий фрагмент Cloud Sisal программы:

```
type OneDim = array [..] of integer;
type TwoDim = array of OneDim;
function generate( N : integer returns TwoDim, OneDim )
for i in 1, N cross j in 1, N do A := i * j; B := i + j
returns array [.., ..] of A; array of B
end for
end function
```

Первые две строчки определяют имена типов для массивов. Видно, что размеры в них не указаны, все экземпляры описанных составных типов данных должны динамически создаваться, изменяться и удаляться во время выполнения программы. Только форма и типы элементов содержатся в данных спецификациях типов массивов. Во второй строчке (в определении типа TwoDim) форма опущена и по умолчанию полагается равной [..]. Заголовок функции «generate» показывает, что ожидается один целочисленный аргумент, «N», и вычисляются (возвращаются) два неименованных значения. Каждое возвращаемое значение представляет собой массив целых чисел, но опять же указываются только формы массивов, а не их размеры. Имена мо-

гут быть привязаны к этим возвращенным значениям в месте вызова функции, если это потребуется программисту. Вызов функции семантически эквивалентен воспроизведению кода функции на месте вызова с соответствующей заменой параметров. Эта эквивалентность, часто называемая «ссылочной прозрачностью», является фундаментальным свойством функциональных языков и представляет собой одно из достоинств языка Cloud Sisal. Данное свойство, в частности, упрощает процессы анализа, выполняемые оптимизирующим компилятором, поскольку функции не имеют побочных эффектов и являются детерминированными. Другими словами, две любые функции могут выполняться параллельно, если между функциями не существует зависимости по данным, а одна и та же функция с одинаковыми фактическими параметрами всегда возвращает одинаковые значения.

Все выражения Cloud-Sisal-программы, включая функции «целиком», вычисляют наборы значений. В приведенном выше случае функция «generate» вычисляет двумерный и одномерный массивы, которые являются значениями выражения, содержащегося в определении функции. Указанное выражение представляет собой конструкцию цикла, который говорит компилятору для Cloud Sisal о потенциальном параллелизме. Этот цикл имеет индексный диапазон, определяемый как декартово произведение двух более простых диапазонов. Это означает, что тело цикла будет выполняться столько раз, сколько есть значений в диапазоне индексов, в данном случае $N \cdot N$, и все экземпляры тела будут независимыми, поскольку между ними нет зависимостей по данным. Какие наборы независимых тел цикла будут выполняться параллельно, а какие нет, будет выбираться на основе осуществляемого компилятором и системой выполнения анализа связанных с этим затрат, а также по опциям, заданным программистом.

Нахождение имен «А» и «В» внутри тела цикла не следует рассматривать как повторное использование этих имен в смысле присваивания переменной в императивной программе. Здесь эти имена используются для обозначения значений в теле цикла, и на самом деле они, вероятнее всего, не будут реально существовать в исполняемой программе. Важным моментом здесь является то, что каждый экземпляр тела цикла, содержащий конкретные значения для i и j , будет независимо вычислять конкретные экземпляры целых значений, определенных как $i \cdot j$ и $i + j$; затем все эти отдельные значения будут собраны вместе в пару массивов и возвращены. Язык предлагает пользователю богатый набор разнообразных стандартных редукций, а также допускает определение и использование своих собственных редукций.

Использование редуций для вычисления результатов цикла хорошо тем, что их реализация может зависеть от целевой вычислительной системы. Когда программа выполняется в однопоточной среде, редукцию можно выполнять последовательно, но при выполнении в нескольких потоках она может выполняться параллельно.

Механизм Try-catch весьма популярен сегодня для обработки ошибок, но этот подход имеет конфликты с параллельным выполнением программы. Когда возникает исключение, все потоки выполнения должны быть остановлены, конвейер очищен и т. д. Также возникают сложности с сохранением программного детерминизма в случае параллельного выполнения и возникновения исключений. Для языка Cloud Sisal таких проблем не существует, поскольку в нем используется семантика «всегда завершаемых вычислений», что означает, что поток выполнения Cloud Sisal программы никогда не останавливается и всегда возвращает результирующее значение (возможно, содержащее значение «ошибка»), даже если возникают какие-либо ошибочные ситуации. Для этого в каждом типе существует выделенное ошибочное значение, например, булевский тип состоит из значений истины (true), лжи (false) и ошибочного значения (error[boolean]). Если не оговорено обратное и ошибочны какие-нибудь аргументы операций над встроенными типами или предопределённых функций, то результаты их тоже будут ошибочными значениями. Узнать, ошибочно ли значение выражения, всегда можно с помощью специальной операции.

Язык поддерживает аннотированное программирование [3] и конкретизирующие преобразования [4], позволяя пользователю описывать известные ему семантические свойства программы в виде формализованных комментариев. Комментарий, который начинается с символа доллара «\$», называется прагмой и задаёт свойства конструкции, идущей следом (одной конструкции можно сопоставлять несколько прагм). Результат унарного выражения в прагме обозначается через одиночный символ подчеркивания «_», а арности n -арного ($n > 1$) выражения обозначаются как «_[1]», ..., «_[n]». Прагма может иметь вид «имя» или «имя = список выражений», где в выражениях списка могут принимать участие имена, видимые в месте расположения прагмы. Например, расположенная перед выражением прагма «assert = расширенное булевское условие» содержит условие, которое должно быть истинным сразу после вычисления выражения. Прагма «assert» может располагаться в объявлениях функций как перед ключевым словом «returns» и накладывать условия на возвращаемые значения, так и пе-

ред первым формальным параметром и задавать условия на указанные в ней имена формальных параметров. Например, расширенное булевское условие ($\text{all } i : i > 2 : A[i] = 0$) истинно, если равны нулю в массиве или потоке A все те его элементы, у которых индекс больше двух, а условие ($\text{is } i : i > 2 : A[i] = 0$) истинно, если в A существует хотя бы один нулевой элемент с индексом больше двух.

Перед каждым выражением может также располагаться прагма «non_used = список значений», которые сразу после вычисления выражения становятся ненужными (в дальнейшем при исполнении программы никак не используются) и могут быть удалены из программы. Например, если, как это указано в аннотации, приведенной ниже, второй результат функции «generate» никогда не будет использоваться, то его вычисление в теле функции может быть удалено:

```
function generate (N : integer // $ non_used = _[2]
                  returns TwoDim, OneDim )
  for i in 1, N cross j in 1, N do A := i * j
  returns array [..., ..] of A
end for
end function
```

Заключение

Создаваемая технология открывает для широкого круга прикладных программистов реальный мир супервычислений, не требуя больших инвестиций в новые компьютерные системы. Ее применение повышает надежность создаваемых параллельных программ, поскольку позволяет формулировать решение задач на абстрактном уровне в декларативном стиле и без привязки к конкретным вычислительным ресурсам, а также проверять создаваемые параллельные программы с помощью формальных методов. Оно также повышает эффективность использования супервычислителей за счет переноса работ программистов по конструированию и отладке программ с дорогих супервычислителей на более дешевые и привычные персональные компьютеры, а также за счет снятия необходимости выполнять построение, верификацию и отладку программы для решения одной и той же задачи каждый раз заново при переходе с одного супервычислителя на другой.

Исследование выполнено за счет гранта Российского научного фонда (проект 18-11-00118). Авторы благодарны всем коллегам, которые принимали участие в работах, рассмотренных в докладе.

Литература

1. *Gaudiot J.-L.* The Sisal project: real world functional programming / J.-L. Gaudiot, T. DeBoni, J. X. Feo, et al. // Lecture Notices in Computer Science. 2013. Vol. 1808. Pp. 84–72.

2. *Касьянов В. Н.* Язык программирования Cloud Sisal / В. Н. Касьянов, Е. В. Касьянова. Новосибирск, 2018. 45 с. (Препринт/РАН, Сиб. отд-ние, ИСИ; N181).

3. *Касьянов В. Н.* Аннотирование программ и их преобразование // Программирование. 1989. № 4. С. 3–16.

4. *Касьянов В. Н.* Трансформационный подход к конкретизации программ // Кибернетика. 1989. № 6. С. 28–32.

5. *Касьянов В. Н.* Иерархические графы и графовые модели: вопросы визуальной обработки // Проблемы систем информатики и программирования. Новосибирск: ИСИ СО РАН, 1999. С. 7–32.

6. *Касьянов В. Н.* Визуализация графовых представлений потоковых программ / В. Н. Касьянов Т. А. Золотухин // GraphiCon 2018: Труды 28-й Международной конференции по компьютерной графике и машинному зрению. Томск: Национальные исследования. Томский политехнический университет, 2018. С. 134–137.

УДК 681.343.001

МОДЕЛИРОВАНИЕ ПРОЦЕССА ПЕРКОЛЯЦИИ НА МНОГОПРОЦЕССОРНЫХ СИСТЕМАХ И ВОЗМОЖНОСТЬ ПРИМЕНЕНИЯ ТЕОРИИ ПЕРКОЛЯЦИИ ДЛЯ АНАЛИЗА КОМПЬЮТЕРНЫХ СЕТЕЙ

Калиниченко А. В., канд. техн. наук

Габанова М. М., магистр

Северо-Кавказский горно-металлургический институт
(государственный технологический университет),

г. Владикавказ, Россия

Аннотация. В настоящей работе рассматривается практическая реализация процесса перколяции на многопроцессорных системах с использованием библиотеки Task Parallel Library. Рассмотрена возможность применения теории перколяции для моделирования компьютерных сетей, имеющих случайную структуру.

Ключевые слова: перколяция, параллельные вычисления, компьютерные сети.

***PERCOLATION PROCESS MODELING USING
MULTIPROCESSOR SYSTEMS AND PERCOLATION THEORY
APPLICATION FOR COMPUTER NETWORKS ANALYSIS***

Kalinichenko A. V., Gabanova M. M.

Abstract. *The paper presents a practical implementation of a percolation process using multiprocessor systems and Task Parallel Library. Possibilities of applying the percolation theory for modeling computer networks with random structure are given.*

Keywords: *percolation, parallel computing, computer networks*

Введение

Перколяция является удобной моделью для описания широкого класса явлений. Методы перколяции применяются для изучения таких процессов, как образование гелей, прыжковая проводимость в полупроводниках, распространение эпидемий, ядерные реакции, образование галактических структур, управление информационными потоками. Несмотря на то что в теории перколяции получен ряд строгих результатов, большинство результатов получено путем применения компьютерного моделирования. При этом приходится проводить многие тысячи компьютерных испытаний на больших объектах, что требует разработки эффективных алгоритмов, в том числе алгоритмов моделирования процессов перколяции на многопроцессорных системах.

Постановка задачи

Рассмотрим квадратную решетку размером $N \times N$, каждый из узлов которой может быть занятым или свободным. Пусть вероятность того, что узел занят, равна p , и, соответственно, вероятность того, что узел свободен, равна $(1 - p)$. Занятые узлы либо изолированы друг от друга, либо образуют группы, состоящие из ближайших соседей. Кластер определяется как группа занятых узлов решетки, связанных с ближайшим соседом по стороне узла. Два занятых узла принадлежат одному кластеру, если их можно соединить непрерывной цепочкой соседних занятых узлов. Стягивающий кластер (перколяционный кла-

стер) – такой кластер, который начинается на одной границе и заканчивается на противоположной границе решетки. Образование стягивающего кластера называется перколяцией. Теория перколяции изучает строение систем вблизи порога перколяции (порога протекания), т. е. при концентрациях p , при которых стабильно начинает появляться перколяционный кластер [1, 2]. Вероятность его образования в конкретной решетке $P_c(p)$ является предметом исследования.

На рисунке 1 показаны варианты заполнения решетки при $p = 0,05$, $p = 0,248$, $p = 0,347$ и $p = 0,504$. Перколяционный кластер – при $p = 0,504$. На рисунке занятые позиции обозначены черным цветом, незанятые – белым, перколяционный кластер – серым цветом.

Для ряда двумерных решеток критическая концентрация может быть рассчитана аналитически, однако для решеток произвольной структуры строгие результаты не найдены и задача решается путем компьютерного моделирования.

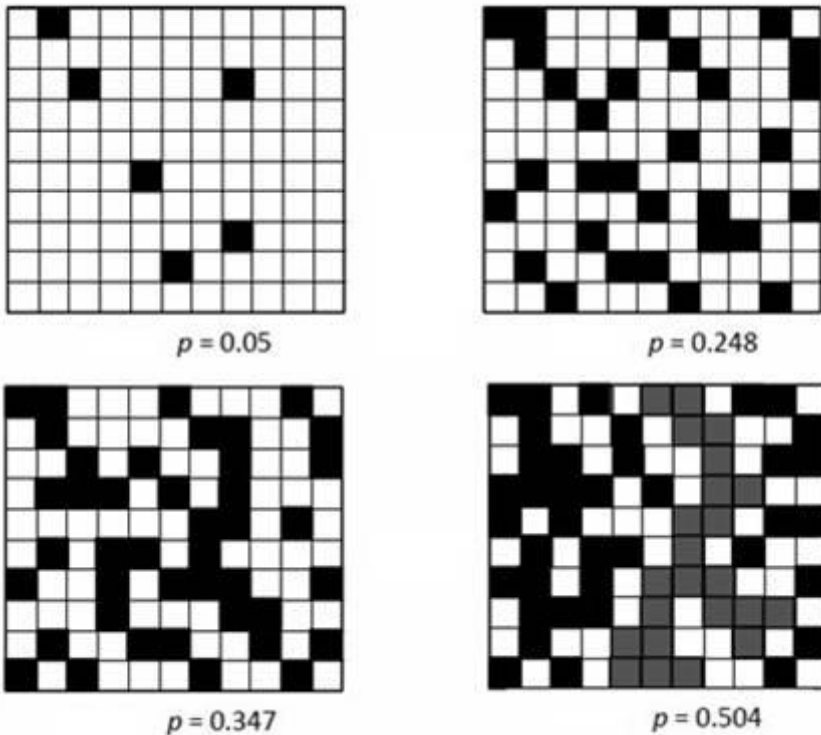


Рис. 1. Варианты заполнения решетки при различных значениях p

Алгоритм определения порога перколяции

Для реализации алгоритма нахождения вероятности образования перколяционного кластера будем использовать метод Монте-Карло [1–3]. Алгоритм содержит следующие шаги:

1. Задать значение вероятности p того, что узел занят.
2. Установить счетчик числа перколяционных кластеров M_c , равным нулю.
3. Задать количество испытаний M .
4. Установить состояние каждого узла (занят/свободен) при заданном значении p , тем самым построив конкретную реализацию решетки.
5. Определить, есть ли в этой реализации хотя бы один стягивающий кластер. Если есть, то счетчик числа перколяционных кластеров M_c увеличиваем на единицу.
6. Пункты 4–5 повторяем M раз.
7. Получаем оценку вероятности образования стягивающего кластера $P_c = M_c / M$.
8. Повторяем пункты 1–7 для ряда значений вероятности p в интервале от 0 до 1.

Для поиска и маркировки кластеров используется алгоритм Хошена – Копельмана, который позволяет идентифицировать все кластеры за один проход по решетке. Входные данные – размерность решетки n , матрица A размера $N \times N$, в которой единицы соответствуют занятым узлам, нули – свободным. Рассмотрим основные шаги алгоритма [1]:

1. Создаем дополнительный массив l для хранения меток кластеров узлов. Первоначально значения элементов этого массива совпадают с их номерами. Кластерные метки присваиваются узлам, начиная с единицы.

2. Просматриваем последовательно все элементы матрицы A . Если значение текущего элемента массива равно нулю, то переходим к следующему элементу. Если текущее значение элемента равно единице, то выполняем следующую проверку:

- a. Если сосед слева $A[i, j-1]$ и сосед сверху $A[i-1, j]$ равны 0, то полагаем, что данный элемент входит в новый кластер, и присваиваем текущему элементу номер очередной кластерной метки.
- b. Если сосед сверху имеет значение 0, а сосед слева не равен нулю, то текущий элемент и элемент слева принадлежат одному и тому же кластеру. Текущему элементу присваиваем номер кластерной метки элемента, находящегося слева.

с. Если сосед сверху имеет значение, отличное от нуля, а сосед слева – нулевое значение, то текущий элемент и его сосед сверху принадлежат одному и тому же кластеру. Однако у соседа сверху кластерная метка может быть неправильной, поскольку в результате последующих проверок могло выясниться, что кластер, к которому принадлежит этот элемент, слился с другим кластером. Поэтому текущему элементу необходимо присвоить номер правильной кластерной метки соседа сверху.

d. Если и сосед сверху, и сосед слева имеют значения, отличные от нуля, то все три ячейки принадлежат одному кластеру. Текущему элементу присваиваем наименьший из номеров правильных кластерных меток соседа слева и соседа сверху. Далее необходимо откорректировать массив кластерных меток. Для этого в элемент массива c_1 присваивается номер правильной кластерной метки, соответствующий большей из кластерных меток.

Программная реализация

Для реализации алгоритмов разработана программа в среде Microsoft Visual Studio на языке #. На рисунке 2 показана зависимость времени работы программы от размера решетки. При моделировании процессов методом Монте-Карло необходимо проведение большого числа статистических испытаний. С увеличением числа испытаний время работы программы растет. Следовательно, целесообразно произвести распараллеливание программы по испытаниям [4].

Для распараллеливания использована встроенная в C# библиотека TaskParallelLibrary, которая предоставляет поддержку распараллеливания циклов посредством явных методов, очень близких своим языковым эквивалентам, а именно Parallel.For(). Метод Parallel.For приспособливается к специфике выполнения отдельных итераций, учитывая количество задач, выполняющихся в каждый момент, и исключает вероятность дробления диапазона на слишком мелкие фрагменты, производя его деление динамически. Распределение итераций по рабочим потокам осуществляется либо по равным диапазонам индекса (range-partitioning), либо по блокам (chunk-partitioning). По умолчанию осуществляется разделение по диапазону (статическая декомпозиция). Планировщик до начала выполнения цикла разделяет итерации по свободным рабочим потокам. В процессе обработки цикла нет необходимости в синхронизации доступа. Часто обработка i -элемента зависит от номера индекса. Разная вычислительная «на-

грузка» при обработке элементов приводит к несбалансированности параллельной обработки: некоторые потоки быстро выполняют свою работу, а какие-то потоки будут продолжать работать. При статической декомпозиции освободившиеся потоки не помогают загруженным, однако их может использовать планировщик для обработки других задач. При блочной (динамической) декомпозиции распределение элементов (итераций цикла) по рабочим потокам выполняется динамически в течение всей обработки цикла. Блочная декомпозиция позволяет получить более сбалансированное разбиение, но требует затрат на синхронизацию доступа к элементам структуры.

Для изучения эффективности разработанных алгоритмов был проведен ряд экспериментов, выявляющих зависимость времени счета от размера решетки. Эксперимент проводился на компьютере со следующими характеристиками: CPU: IntelCore i5, 2,3 ГГц; ОП: 8192 Мб; ОС: Windows 10, x64. Эксперименты проводились на оптимальном числе потоков. Для сравнения на рисунке 2 представлена работа программы с разными размерами решетки с шагом 0,05 и количеством испытаний 100 без и с распараллеливанием.

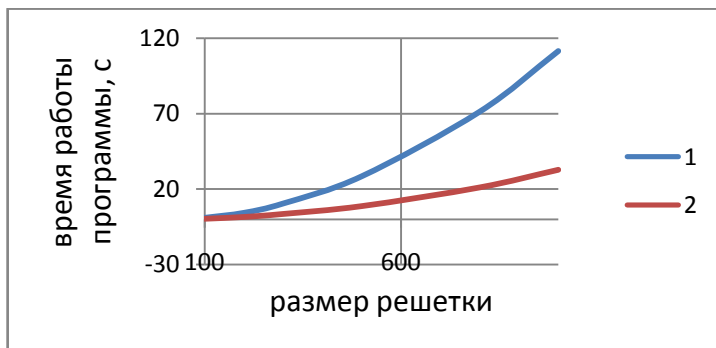


Рис. 2. Динамика роста времени расчета с увеличением размера решетки без распараллеливания (1) и с распараллеливанием (2)

Применение теории перколяции для моделирования компьютерных сетей

Существующие методы моделирования работы и управления информационно-вычислительными сетями часто строятся на использовании целевых функций управления при заранее известной структуре сети и числе узлов. При переходе к региональным и глобальным сетям

возникает проблема динамического изменения характеристик сети. Однако существующие методы недостаточно хорошо позволяют моделировать сеть, имеющую случайную структуру. Таким образом, возникает необходимость поиска новых подходов к моделированию и управлению передачей и обработкой данных. Одним из подходов, который может помочь преодолеть существующие трудности, является использование теории перколяции.

В работе [5] рассмотрена возможность применения теории перколяции для моделирования компьютерных сетей на примере беспроводной сети проекта Loon, анонсированного Google в 2013 году. Проект Loon нацелен на предоставление интернет-услуг в отдаленных регионах Земли, с использованием беспроводных сетей с воздушными шарами (WBN). В WBN заданное число воздушных шаров пространственно распределено по большой площади, что позволяет описать WBN как крупномасштабную случайную сеть, а также представить в виде графа, в котором воздушные шары представляют узлы, а беспроводные связи между воздушными шарами – ребра графа. Предполагается, что два узла соединены, если расстояние между узлами меньше предопределенного критического расстояния. Теория перколяции выступает важным математическим инструментом для анализа связности крупномасштабного случайного графа, то есть крупномасштабной случайной беспроводной сети [6]. Выводы, полученные в работе, могут быть использованы для оценки необходимого минимального количества узлов в целевой области, для того чтобы поддерживать сетевое соединение WBN.

В работе [7] рассмотрена методика статистической оценки порога пропускной способности компьютерных сетей посредством компьютерного моделирования на основе модели перколяции, выяснены условия кластеризации узлов в случайной среде, ее пороговые свойства.

Заключение

В настоящей работе рассмотрен вопрос моделирования процесса перколяции на многопроцессорных системах с использованием библиотеки TaskParallelLibrary. Реализация алгоритма на многопроцессорных системах с распараллеливанием по числу статистических испытаний позволяет успешно применять описанный метод для решения задач разных областей. Показана возможность применения теории перколяции для моделирования компьютерных сетей, имеющих случайную структуру.

Литература

1. *Тарасевич Ю. Ю.* Перколяция: теория, приложения, алгоритмы: Учебное пособие. М.: Едиториал УРСС, 2002.
2. *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике. Ч. 2. М.: Мир, 1990.
3. *Москалев П. В.* Анализ структуры перколяционного кластера // Журнал технической физики. 2009. Т. 79. Вып. 6.
4. *Яппарова А. А., Маякова С. А.* Распараллеливание алгоритмов численного моделирования процессов перколяции с вытеснением. Уфа : УГАТУ, 2010.
5. *NashidAnjum M. D., Honggang Wang, Hua Fang.* Percolation analysis of large-scale wireless balloon networks // Digital Communications and Networks. 2018.
6. *Franceschetti M., Meester R.* Random Networks for Communication: from Statistical Physics to Information Systems. Vol. 24. Cambridge University Press, 2008.
7. *Немов Д. А., Герасимов И. В., Кузьмин С. А., Лу А. В.* Методика статистической оценки пропускной способности компьютерных сетей на основе модели перколяции // Известия СПбГЭТУ «ЛЭТИ». № 1. 2018.

УДК 004.4'242::519.717.3::004.272.2

ПАРАЛЛЕЛЬНОЕ РЕШЕНИЕ ЗАДАЧИ АВТОМАТИЧЕСКОГО ДОСТРАИВАНИЯ ПОРОЖДАЮЩЕЙ МОДЕЛИ ПРОГРАММЫ НА БАЗЕ КОНСТРУИРУЮЩИХ XPath-ЗАПРОСОВ

Пекунов В. В., д-р техн. наук, rekunov@mail.ru
ОАО «Информатика»
г. Иваново, Россия

Аннотация. Рассмотрена проблема восполнения и реконструкции постановки и плана решения задачи в системах порождения программ. Предлагается упростить правила восполнения и реконструкции путем применения синтаксиса XPath-запросов в конструирующем смысле. Предложены соответствующие правила и алгоритм вывода постановки/плана (XML-документа) на их основе. Предложен подход к распараллеливанию. Показано хорошее ускорение в системе с общей памятью.

Ключевые слова: порождение программ, конструирующий запрос, XPath, параллельный алгоритм, восполнение модели, вывод модели, план решения.

PARALLELIZATION OF AUTOMATIC COMPLETION OF MODEL OF PROGRAM GENERATION USING CONSTRUCTING XPath-QUERIES

Pekunov V. V.

Abstract. The problem of completion and reconstruction of the task formulation and of the solving plan in the systems of program's generation is analyzed. The simplification of the completing and reconstructing rules using the syntax of XPath in the constructing sense is proposed. An appropriate rules and an algorithm of the inference of formulation/plan based on such rules is proposed. The parallelizing algorithm is formulated. The good speed-up on the shared-memory system is shown.

Keywords: program's generation, constructing query, XPath, parallel algorithm, model completion, model inference, solving plan

Введение

Любая система порождения программ с необходимостью требует наличия постановки решаемой задачи А. Независимо от того, какова первичная форма такой постановки (текстовое описание на естественном или искусственном языке, или визуальная диаграмма, или, например, набор формул и/или иных символических утверждений) [1, 2], система порождения программ транслирует ее в некий набор четко формализованных фактов В о содержании и, возможно, о путях решения поставленной задачи. Полученный набор фактов расширяется и дополняется с помощью некоего механизма вывода на базе известных правил [3], превращаясь в план решения задачи С, по которому генерируется программа D, реализующая данный план.

Такой подход характерен для системы PGEN++ [1, 3], в которой постановка А может быть текстовой или блочно-визуальной, набор В и план С представляют собой объектно-событийные модели [1], а правила вывода – предикаты GNU Prolog.

Актуальна задача упрощения процесса составления правил трансляции в PGEN++ путем устранения элементов программирования вообще, с переходом к чисто декларативной постановке. Это возможно,

если воспользоваться представлением В и С в виде XML-документов. Тогда правила трансляции В в С сводятся к правилам достраивания XML-документа. Такой процесс может быть весьма трудоемким, поэтому актуальна задача распараллеливания вывода С из В.

Итак, *целью* данной работы является упрощение процесса формализации правил, дополняющих фактологическую XML-постановку задачи в системе PGEN++ до плана решения такой задачи. Для достижения данной цели поставим следующие *задачи*: а) предложить простые декларативные правила дополнения и реконструкции XML-документов; б) предложить основные механизмы применения таких правил; в) предложить параллельные алгоритмы для указанных механизмов.

Алгоритм достраивания XML-документа (модели задачи)

Формально возможности достраивания XML-модели полностью описываются грамматикой соответствующего документа, представленной в виде его DTD. Поэтому, требуемые *правила достраивания* будут являться, скорее, *слабыми ограничениями* достраивания. Полученные варианты XML-модели проверяются на *заданных тестах* – двойках (I, O), где I – строка, представляющая содержимое входного файла программы (построенной системой PGEN++ для решения задачи, описываемой текущим вариантом XML-модели), а O – строка, представляющая требуемое содержимое выходного файла.

Разделим правила достраивания на *проверочные* (проверяющие некий факт) и *конструирующие* (являющиеся таким расширением проверочных правил, которое способно к частичному достраиванию текущего документа). Каждое правило может быть применено к текущей версии XML-документа *в проверочном режиме*, причем результат применения может быть одним из следующих:

а) *строгое противоречие* – правило не выполняется и никакие достраивания документа не смогут привести к его выполнению;

б) *нестрогое противоречие* – правило не выполняется, но его запуск *в конструирующем режиме* способен породить документ, в котором оно выполнится;

в) *соответствие* – правило выполняется.

Правила имеют *веса*, по умолчанию равные единице. Веса используются при вероятностном определении очередности выбора правила из набора активных правил. Вероятностная модель обучается на успешных (ранее приведших к корректному результирующему XML-документу) последовательностях применения правил.

Достраивание документа описывается *рекурсивным алгоритмом* ALG (G, E), выполняющим последовательный перебор всех возможных вариантов достраивания E из G с отсечением тех вариантов, которые некорректны (имеют структуру, невозможную с точки зрения текущего набора классов объектно-событийной модели, соответствующей текущему XML-документу) или входят в строгое противоречие с одним или несколькими правилами. Если все правила сообщили о соответствии, то вариант решения E отправляется в систему PGEN++ на порождение соответствующей программы с прохождением набора тестов. Правила, сообщившие о нестрогом противоречии, запускаются в конструирующем режиме, последовательность их применения определяется с помощью упомянутой выше вероятностной модели.

Определим правила. Поскольку стандартом доступа к XML-документу является язык запросов XPath, представляется целесообразным взять его за основу как в проверочном, так и в конструирующем случаях. Заметим, что XML-документ модели имеет фиксированный DTD и описывает *сетевой граф постановки задачи*, вершинами которого являются факты-объекты, а связи отражают отношения между объектами.

Введем понятие *конструирующего XPath-запроса* P. Такой запрос синтаксически эквивалентен обычному XPath-запросу Z, с той разницей, что применение P к документу K_1 производит в нем минимально необходимый набор операций достраивания до K_2 , таких, чтобы исполнение обычного эквивалентного запроса Z к полученному документу K_2 было успешным.

Конструирующий запрос P интерпретируется последовательно, слева направо. Если его очередным элементом является имя тега без предиката, то запрос проверяет наличие соответствующего тега в указанных местах документа и, если его там нет, создает (с пустыми набором атрибутов и содержанием). Если же очередной элемент – имя тега Q с предикатом T, то предикат T используется для генерации множества возможных наборов $\{S_i\}$ атрибутов тега, после чего в указанных местах документа создаются теги Q, каждый из которых отвечает следующим требованиям: а) набор атрибутов $W \in (\{S_i\} \setminus M)$, где M – множество наборов атрибутов, соответствующих уже присутствующим (в указанных местах документа) тегах Q; б) на наборе атрибутов W предикат T успешен при его интерпретации по обычным правилам XPath.

Постулируем, что *производящими операциями* в предикатах конструирующих XPath-запросов могут быть только «=», «AND» и «OR».

Результатом каждой такой операции является множество значений атрибутов в формате $\{A_i=V_i\}$.

Результатом операции «@ID=const» является $\{\text{@ID=const}\}$. Результатом операции «A OR B» является $A \cup B$. Результатом операции «A AND B» является:

$$\begin{cases} A \times B, & \text{если } A \neq \emptyset, B \neq \emptyset; \\ A, & \text{если } B = \emptyset; \\ B, & \text{если } A = \emptyset. \end{cases}$$

Результатом всех прочих операций является \emptyset .

Например, если имеем конструирующий запрос вида:

```
/OBJS/clsSimpleBlock[(@ID=«A» OR @ID=«B») AND (@NAME=«C»)],
```

то результирующий документ будет обязательно содержать следующий фрагмент:

```
<OBJS>...<clsSimpleBlock ID=«A» NAME=«C»></clsSimpleBlock>  
<clsSimpleBlock ID=«B» NAME=«C»></clsSimpleBlock>...</OBJS>
```

Выделим 4 вида правил, не приводя описание их синтаксиса:

1. Конструирующее *правило единственности*. В проверочном режиме определяет, возвращает ли указанное в нем XPath-выражение единственный элемент, а в конструирующем режиме достраивает документ соответствующим образом.

2. Проверочное *правило следования*. Содержит два XPath-выражения. Вычисляет наборы объектов J_1 и J_2 текущего графа-модели (представленной XML-документом), соответствующих узлам документа, выделяемым указанными XPath-выражениями. Правило возвращает соответствие, если в графе из любой вершины, входящей в J_1 , существует путь в какую-либо вершину, входящую в J_2 .

3. Проверочное *правило количественного отношения*. Содержит два XPath-выражения, которые вычисляются с результатами K_1 и K_2 соответственно. Правило возвращает соответствие, если K_1 находится в указанном отношении («>», «<», «>=», «<=», «=», «!=») с K_2 , смысл отношения зависит от типов K_1 и K_2 .

4. Конструирующее *правило восполнения*. Содержит два XPath-выражения, одно из которых (A) *ссылается* на другое (B). В прове-

рочном режиме правило вычисляет выражение В, после чего определяет, существует ли набор узлов, соответствующий выражению А. Результатом проверки является соответствие (существует) или нестрогое противоречие (не существует). В конструирующем режиме правило достраивает документ, воспринимая XPath-выражение А как конструирующее. Это правило является основным, определяющим фактически условное достраивание.

Распараллеливание алгоритма. Апробация

Базовый алгоритм достраивания документа ALG является рекурсивным, с порождением множества вариантов на каждом уровне рекурсии, и, как следствие, хорошо распараллеливаемым в системе с общей памятью. Был введен *пул потоков*, количество которых выбиралось в диапазоне $N..1,5N$, где N – количество имеющихся в системе ядер. На каждом очередном этапе рекурсивного спуска порождаемые варианты по мере возможности распределялись по свободным потокам. Подмножество вариантов, которые не удавалось «отправить» в параллельные потоки по причине их исчерпания, исполнялось в породившем варианты потоке. Такой подход обеспечил минимальное количество порожденных потоков и, соответственно, минимальные затраты на их обслуживание. Дополнительно велся контроль множества уже обработанных вариантов, для сокращения объема перебора. Использовалась мемоизация, позволившая обрабатывать каждый уникальный вариант только один раз.

Предложенный подход был применен для восполнения постановки задач простой обработки векторных данных. Постановка формулировалась в виде текста на русском языке, система PGEN++ преобразовывала ее в первичную XML-модель, которая дополнялась по предложенной в работе методике. Как показали замеры среднего времени работы, в режиме «4 ядра – 4 потока» показано суперлинейное ускорение в 4,33 раза, что может объясняться как более эффективным использованием кэш-памяти ядер при разделении задачи на фрагменты, так и случайными факторами, связанными с вероятностной природой использованного алгоритма.

Выводы

В данной работе предложен новый подход к восполнению и реконструкции постановки задачи (для системы порождения программ

PGEN++), записанной в форме XML-документа. Показано, что восполнение и реконструкция такого документа могут быть выполнены с помощью декларативных правил, базирующихся на конструирующих XPath-запросах. Предложена семантика таких правил, а также обозначен алгоритм вывода документной XML-модели на их основе. Предложен подход к распараллеливанию вывода на системах с общей памятью. Показаны высокие результаты по ускорению в параллельном режиме.

Литература

1. *Пекунов В. В.* Автоматизация параллельного программирования при моделировании многофазных сред. Оптимальное распараллеливание // Автоматика и телемеханика. 2008. № 7. С. 170–180.

2. *Зубков В. П., Назаретский С. П.* IPGS – интеллектуальная система автоматизированного программирования // Информационная среда вуза: Сборник статей. Иваново: ИГАСА, 2000. С. 213–215.

3. *Пекунов В. В.* Автоматическое распараллеливание C-программ с применением директив Cilk++ на базе распознающих объектно-событийных моделей // Программные системы и вычислительные методы. 2018. № 4. С. 124–133. DOI: 10.7256/2454-0714.2018.4.28086. URL: http://e-notabene.ru/ppsvm/article_28086.html

УДК 378:004.9

**ОРГАНИЗАЦИЯ ПРОЦЕССА ИНТЕЛЛЕКТУАЛЬНОГО
АНАЛИЗА ОБРАЗОВАТЕЛЬНЫХ ДАННЫХ**

Оськин Д. А., d.a.oskin@pdu.by,

Белорусский государственный экономический университет,

г. Минск, Беларусь, www.bseu.by

Оськин А. Ф., канд. техн. наук, a.f.oskin@pdu.by,

Полоцкий государственный университет,

г. Новополоцк, Беларусь, www.psu.by

Аннотация. В статье рассмотрены предпосылки для организации в вузах процесса интеллектуального анализа образовательных данных (ИАОД). Определены основные цели и задачи ИАОД и заинтересованные лица. Описана методология внедрения проектов интеллектуального анализа – стандарт CRISP-DM – и её применение для внедрения ИАОД в вузе.

Ключевые слова: интеллектуальный анализ, интеллектуальный анализ образовательных данных, системы управления обучением, Crisp-Dm.

**ORGANIZATION OF EDUCATIONAL
DATA MINING PROCESS**

Os'kin D. A., Os'kin A. F.

Annotation. *The article describes the prerequisites for the organization of the process of educational data mining (EDM) in Universities. The main goals and objectives of EDM and stakeholders are defined. The methodology of implementation of intellectual analysis projects – CRISP-DM standard – and its application for the implementation of EDM at the University is described.*

Keywords: *data mining, educational data mining, learning management system, CRISP-DM.*

Введение

В последнее десятилетие в высших учебных заведениях Беларуси широко применяются системы управления обучением (LMS). Наибольшее распространение получили системы на базе свободно распространяемого программного обеспечения Moodle. В некоторых университетах республики для управления обучением используются бесплатные облачные сервисы платформы G Suite for Education от компании Google. Помимо своей основной функции – поддержки и управления обучением, системы сохраняют информацию о взаимодействии участников образовательного процесса в файлах регистрации, в которых фиксируется вся информация о действиях пользователей в системе. Созданный на основе таких файлов массив информации пригоден для дальнейшего анализа и поиска неявных закономерностей, с целью улучшения образовательного процесса, оптимизации учебных планов и прочих задач.

Изучением скрытых закономерностей в массивах данных занимается раздел науки Data mining [1] или интеллектуальный анализ данных. Если говорить об образовательных данных, то их изучением и поиском закономерностей занимаются относительно новые разделы Data mining: Educational data mining и Learning Analytics. Первые публикации по данной тематике появляются в начале 2000-х годов. В русскоязычном научном сообществе эти направления обобщенно называют: интеллектуальный анализ образовательных данных (ИАОД). Главной целью ИАОД является повышение качества подготовки специалистов.

По мнению ученых К. Ромеро и С. Вентура [2], потребителям результатов ИАОД являются: обучающиеся, преподаватели, администраторы и исследователи. В соответствии с потребителем перед ИАОД ставятся различные цели. Так, **для обучающихся** основными целями являются получение рекомендаций по индивидуализации образовательной траектории, более качественной обратной связи с преподавателем и улучшение усвоения знаний. **Для преподавателей** основными целями являются применение технологий и методов обучения, наиболее подходящих для конкретной группы обучающихся, улучшение понимания социальных, поведенческих и когнитивных аспектов обучения. **Для администраторов** – обоснованные управленческие решения и оптимизация распределения ресурсов учебного заведения. **Для исследователей** – развитие и сравнение между собой различных методов и алгоритмов ИАОД, оценка эффективности и результативности учебного процесса.

На основе этих целей формируются следующие задачи:

для обучающихся:

- осознанное формирование индивидуальной образовательной траектории;
- правильный выбор факультативных дисциплин и дисциплин по выбору;
- профессиональная ориентация и точный выбор сферы будущей профессиональной деятельности;

для преподавателей:

- разделение студентов на кластеры и подбор для каждого кластера оптимальной технологии и наиболее эффективных методов обучения;
- оптимизация структуры и содержания лекционного курса;
- прогнозирование успешности учебной деятельности;

для исследователей:

- разработка методов объективной оценки эффективности и результативности учебного процесса;
- разработка новых технологий и методов обучения;
- совершенствование существующих и разработка новых методов и алгоритмов ИАОД;

для администраторов:

- поддержка принятия научно обоснованных управленческих решений.

Однако для успешной реализации всех положительных аспектов, связанных с применением ИАОД, необходимо организовать внедрение ИАОД в университетские процессы.

Организация процесса ИАОД в вузе

На сегодняшний день одной из наиболее популярных методологий внедрения проектов интеллектуального анализа является стандарт CRISP-DM (CRoss Industry Standard Process for Data Mining), разработанный в рамках программы Европейского Союза ESPRIT (European Strategic Program on Research in Information Technology) четырьмя компаниями: ISL, NCR Corporation, Daimler-Benz и OHRA. Первая версия стандарта выпущена в 1999 году, начиная с 2006 года и по настоящее время используется вторая версия [3]. CRISP-DM разбивает процесс анализа данных на шесть основных этапов (рисунок 1) [4].

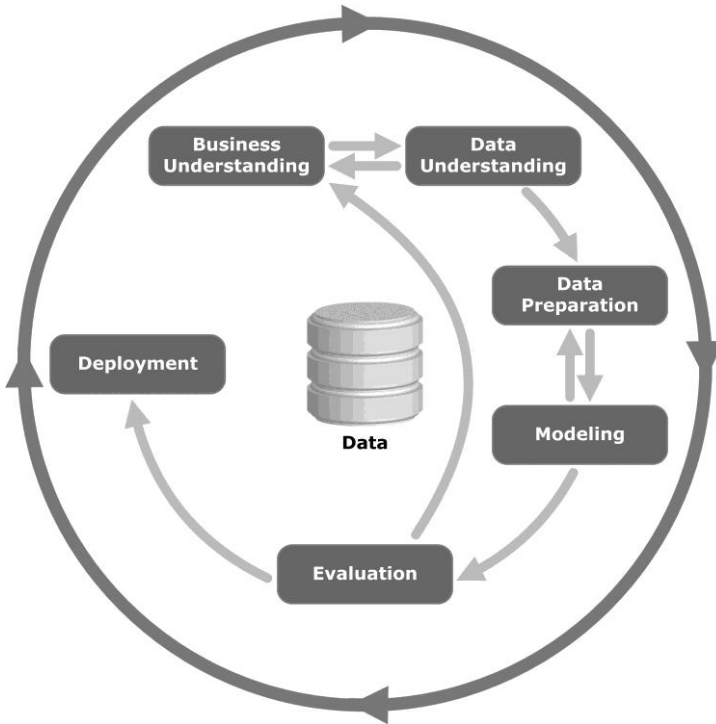


Рис. 1. Процесс анализа данных по CRISP-DM [3]

1. Понимание бизнеса (Business Understanding)

Первая фаза процесса направлена на определение целей проекта и требований со стороны бизнеса. Затем эти знания конвертируются в постановку задачи интеллектуального анализа данных и предварительный план достижения целей проекта.

2. Понимание данных (Data Understanding)

Вторая фаза начинается со сбора данных и ставит целью познакомиться с данными как можно ближе. Для этого необходимо выявить проблемы с качеством данных, такие как ошибки или пропуски, понять, что за данные имеются в наличии, попробовать отыскать интересные наборы данных или сформировать гипотезы о наличии скрытых закономерностей в данных.

3. Подготовка данных (Data Preparation)

Фаза подготовки данных ставит целью получить итоговый набор данных, которые будут использоваться при моделировании, из исходных разнородных и разноформатных данных. Задачи подготовки дан-

ных могут выполняться много раз. Они включают в себя отбор таблиц, записей и атрибутов, а также конвертацию и очистку данных для моделирования.

4. Моделирование (Modeling)

В этой фазе к данным применяются разнообразные методики моделирования, строятся модели, и их параметры настраиваются на оптимальные значения. Обычно для решения любой задачи анализа данных существует несколько различных подходов. Некоторые подходы накладывают особые требования на представление данных. Таким образом, часто бывает нужен возврат на шаг назад к фазе подготовки данных.

5. Оценка (Evaluation)

На этом этапе проекта уже построена модель и получены количественные оценки её качества. Перед тем как внедрять эту модель, необходимо убедиться в достижении всех поставленных бизнес-целей. Основной целью этапа является поиск важных бизнес-задач, которым не было уделено должного внимания.

6. Развертывание (Deployment)

В зависимости от требований фаза развертывания может быть простой, например, составление финального отчета, или сложной, например, автоматизация процесса анализа данных для решения бизнес-задач.

Рассмотрим внедрение ИАОД в университете с точки зрения стандарта CRISP-DM.

Этап 1. Основываясь на практике университетов США, Великобритании и Австралии, на этом этапе разрабатывают стратегию внедрения ИАОД и определяют цели и требования к проекту [5]. Согласно исследованиям [5], цели проекта, требования к нему и будущий результат напрямую зависят от этапа развитости ИАОД в конкретном университете. По мнению авторов [5], модель этапов развитости можно представить, как показано на рисунке 2. Помимо этого успех разработки стратегии напрямую зависит от степени заинтересованности и вовлеченности в процесс руководства вуза, а также наличия в организации специалистов в области интеллектуального анализа данных.

Этап 2. Для образовательных данных всегда имеется набор вариантов источников. Такими источниками могут быть: информационные системы для поддержки студентов, системы управления обучением, социальные сети, данные пропускной системы университета и т. д. Укрупненно источники наборов данных делятся на те, что находятся внутри учреждения образования, и те, что находятся за его пределами.

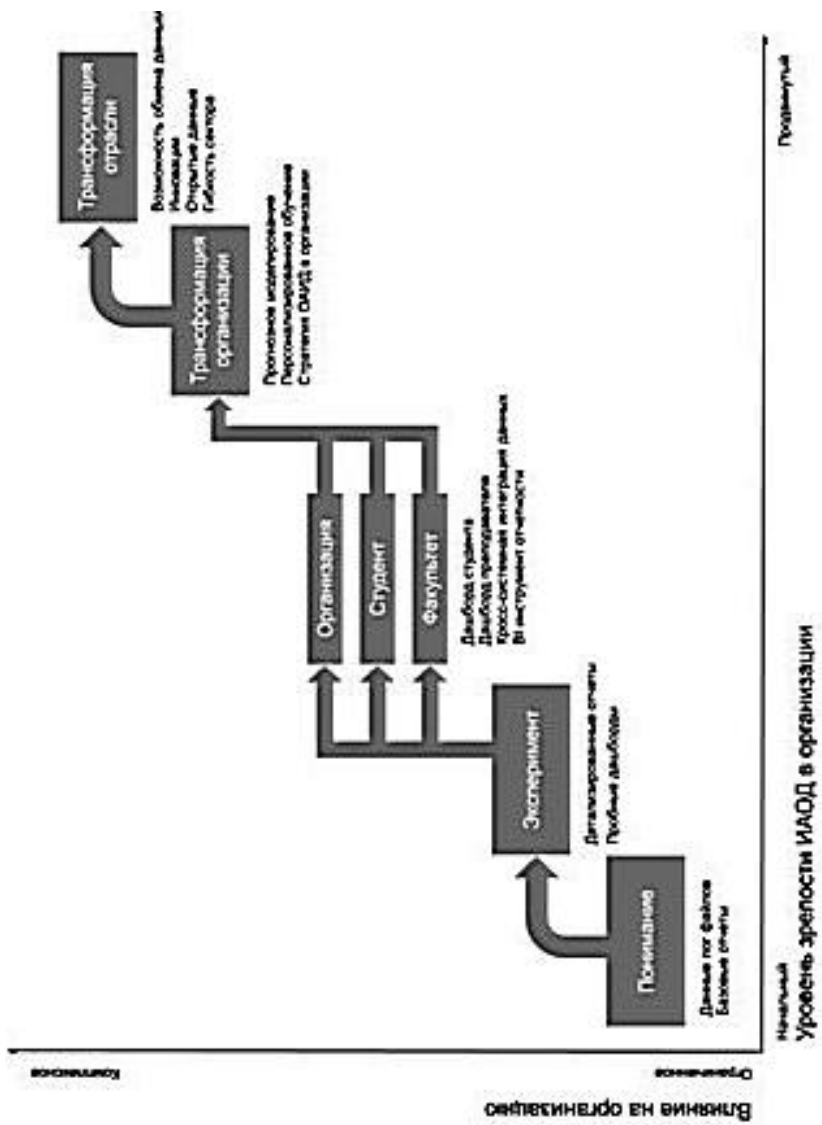


Рис. 2. Модель этапов зрелости ИАОД [5]

На выбор источников влияют заданные на первом этапе цели и задачи интеллектуального анализа. Так, для улучшения маркетинга учреждения образования будут использоваться одни источники, а для создания определенной модели педагогики или оценки уже существующей педагогической модели, основанной на определенных атрибутах этого ученика или на определенных практиках педагога, – совершенно другие.

Поэтому лишь после определения того, что необходимо достичь конкретным аналитическим действием, можно понять, какие данные и из каких источников нужно собрать.

Однако, как правило, как уже упоминалось выше, основным источником для ИАОД является массив данных системы управления обучением.

Этап 3. Если основой для интеллектуального анализа являются данные, поступившие из LMS, то, как правило, не требуется дополнительных действий по их очистке и подготовке. Однако в этом случае необходимо, как минимум, согласовать эти данные с данными информационной системы учащегося, например, результаты сессии с активностью студентов по курсу, размещенному в LMS вуза. Полученная в результате такого объединения таблица пригодна для поиска скрытых закономерностей между активностью студента по курсу и его академической успеваемостью.

В случае если данные для анализа поступают из социальных систем или различных разрозненных по смыслу систем, то, скорее всего, придется пройти процесс как очистки, так и последующей интеграции этих наборов данных.

Этап 4. По мнению авторов [6], базовыми методами, применяемыми для решения задач ИАОД, являются: классификация и кластеризация, статистический анализ, ассоциативный и причинный анализ, ассоциативная классификация. Например, в статье авторов [7] для поиска закономерностей между результатами первой экзаменационной сессии и окончанием учебы в отведенный срок был использован метод кластеризации.

Этап 5. На данном этапе происходит критический анализ полученных результатов и оценивается соответствие с заявленными на первом шаге целями проекта. При необходимости вносятся коррективы и цикл повторяется заново.

Этап 6. Последний этап подразумевает внедрение полученных результатов в учебный процесс университета. Конкретные результаты приведены на рисунке 2 и напрямую зависят от уровня зрелости

процесса ИАОД. Это могут быть базовые отчеты, детализированные отчеты, пробные дашборды, дашборды студента, преподавателя, факультета и т. д. по уровням зрелости процесса ИАОД.

Заключение

Таким образом, организация процесса ИАОД в вузе – комплексная задача, затрагивающая различные структурные подразделения, поскольку требует не только наличия развитой информационной инфраструктуры, но и вовлечения в процесс руководства сотрудников, а также специалистов по интеллектуальному анализу данных. Как показывает зарубежный опыт [5], ИАОД способен решать задачи по улучшению качества подготовки специалистов, снижать количество отчисленных студентов и оптимизировать ресурсы, затрачиваемые на профориентационную работу вуза и др. В связи с этим, несмотря на сложность, следует уделять внимание методам и стратегии ИАОД и внедрять их в отечественную систему высшего образования, т. к. это обеспечит подготовку конкурентоспособных специалистов для нужд новой экономики.

Литература

1. *Fayyad U., Piatetsky-Shapiro G., Smyth P.* From Data Mining to Knowledge Discovery: an Overview // *Advances in Knowledge Discovery and Data Mining*. AAAI / MIT Press. 1996. P. 37–54.
2. *Romero C., Ventura S.* Data mining in education // *Wiley interdisciplinary reviews. Data mining and knowledge discovery*. 3(1). 2013. P. 12–27.
3. CRoss Industry Standard Process for Data Mining [Электронный ресурс] / The CRISP-DM consortium. London, 2009. Режим доступа: <http://crisp-dm.eu/home/crisp-dm-methodology/> / Дата доступа: 25.04.2019.
4. CRISP-DM [Электронный ресурс] / *MachineLearning.ru* Профессиональный информационно-аналитический ресурс. М., 2008. Режим доступа: <http://www.machinelearning.ru/wiki/index.php?title=CRISP-DM/> / Дата доступа: 20.04.2019.
5. *Improving the Quality and Productivity of the Higher Education Sector* [Электронный ресурс] / *Society for Learning Analytics Research*. 2013. Режим доступа: https://solaresearch.org/wp-content/uploads/2017/06/SoLAR_Report_2014.pdf. Дата доступа: 20.04.2019.

6. Handbook of Educational Data Mining /С. Romero [et al.]; Series editor V. Kumar. Minnesota: CRC Press, 2011. 503 p.

7. *Оськин А. Ф., Оськин Д. А.* Применение интеллектуального анализа данных для прогнозирования успешности учебной деятельности // Вестник Полоцкого государственного университета. Серия С. Фундаментальные науки. ПГУ Новополоцк, 2016. С. 8–12.

УДК 37.02

К ВОПРОСУ О ПОВЫШЕНИИ ЭФФЕКТИВНОСТИ ПРЕПОДАВАТЕЛЬСКОЙ ДЕЯТЕЛЬНОСТИ

Даурова А. А., канд. техн. наук, albina_daurova@mail.ru
Астахова Л. Г., ст. преподаватель, astahovalg@mail.ru
Северо-Кавказский горно-металлургический институт
(государственный технологический университет),
г. Владикавказ, Россия

Аннотация. Предлагается новый подход для повышения эффективности преподавательской деятельности. В работе рассматривается задача формирования у студентов понятий, причем в распоряжении преподавателя имеются методы подачи материала. Предполагается, что преподаватель может прогнозировать время усвоения каждым студентом каждого понятия при использовании каждого метода подачи материала. Требуется так выбрать методы подачи материала, чтобы суммарное время их усвоения было минимальным.

Ключевые слова: стратегия, обучение, формирование, эффективность.

THE ISSUE OF IMPROVING THE TEACHING EFFICIENCY

Daurova A. A., Astakhova L. G.

Annotation. A new approach to improve efficiency is proposed-news of teaching activities. The paper deals with the problem of forming the concepts of students, and at the disposal of the teacher there are methods of presenting the material. It is assumed that the teacher can predict the time of mastering each concept by each student using each method of presenting the material. It is required to choose the methods of material supply, so that the total time of their assimilation was minimal.

Key words: strategy, training, formation, efficiency.

Введение

Процесс обучения является результатом творческой работы преподавателя, с одной стороны, и желанием студента получить искомые знания, с другой. Успех преподавательской деятельности зависит от многих факторов. Обеспеченность научно-обоснованной учебной программой, набором методик преподавания, материалов, возможность прикладного использования полученных знаний – все это составляющие процесса обучения.

В связи с этим рассмотрим задачу. В ходе занятия n студентов должны усвоить K понятий, причем в распоряжении преподавателя имеется m методов подачи материала, относящегося к каждому понятию. Предполагается, что преподаватель может прогнозировать время усвоения каждым студентом каждого понятия при использовании каждого метода подачи материала. Требуется так выбрать методы подачи материала, чтобы суммарное время их усвоения было минимальным.

Введем следующие обозначения и определения:

- $z(i, j, k)$ – булева переменная, равна единице, если для формирования k -го понятия у j -го студента преподаватель пользуется i -м методом, и равна нулю – в противном случае.

- $t(i, j, k)$ – прогнозируемое время формирования k -го понятия у j -го студента i -м методом.

- m – число используемых преподавателем методов обучения;
- n – число студентов (учеников);
- K – число формируемых понятий.

Тогда рассматриваемая задача формально будет выглядеть следующим образом:

$$\left\{ \begin{array}{l} \sum_k \max_i \max_j z(i, j, k) \cdot t(i, j, k) \rightarrow \min; \\ \forall k, \forall j : \sum_{i=1}^m z(i, j, k) = 1; \\ \forall k, \exists i : \sum_{j=1}^n z(i, j, k) = n; \\ \forall i, \forall j, \forall k : z(i, j, k) = 1, 0. \end{array} \right. \quad (1)$$

Форма представления исходных данных следующая:

Для каждого k -го понятия формируется матрица Mk , строки которой отвечают различным методам обучения, которые могут быть использованы преподавателем при формировании k -го понятия, а столб-

цы – ученикам. Тогда в ячейке $Mk(i, j)$ содержится прогнозируемое время формирования k -го понятия i -м методом у j -го студента.

Усложним задачу. Пусть новый материал должен быть усвоен на уровне не ниже заданного. В этом случае ставится следующая задача: пусть в ходе занятия n студентов должны усвоить K понятий, причем в распоряжении преподавателя имеется m методов подачи материала, относящегося к каждому понятию. Предполагается, что преподаватель может прогнозировать время усвоения каждым студентом каждого понятия при использовании каждого метода подачи материала применительно к оценкам «3», «4» и «5». Требуется так выбрать методы подачи материала, чтобы суммарное время их усвоения было минимально, а средний балл был не ниже заданной величины S [3].

Введем следующие обозначения и определения:

- $z(i, j, k, q)$ – булева переменная; равна единице, если для формирования k -го понятия у j -го студента преподаватель пользуется i -м методом, позволяющим прогнозировать оценку, равную q , и равна нулю – в противном случае.

- $t(i, j, k, q)$ – прогнозируемое время формирования k -го понятия у j -го студента i -м методом на оценку, равную q .

- m – число используемых преподавателем методов обучения;
- n – число студентов (учеников);
- K – число формируемых понятий.

Тогда рассматриваемая задача формально будет выглядеть следующим образом:

$$\left\{ \begin{array}{l} \sum_k \max_i \max_j \sum_q z(i, j, k, q) \cdot t(i, j, k, q) \rightarrow \min; \\ \forall k, \forall j : \sum_i \sum_q z(i, j, k, q) = 1; \\ \forall k, \exists i : \sum_{j=1}^n \sum_{q=3}^5 z(i, j, k, q) = n; \\ \sum_i \sum_j \sum_k \sum_q q \cdot z(i, j, k, q) \geq S \cdot n \cdot K; \\ \forall i, \forall j, \forall k, \forall q : z(i, j, k, q) = 1, 0. \end{array} \right. \quad (2)$$

Пусть в ходе занятия n студентов должны усвоить K понятий, причем в распоряжении преподавателя имеется m методов подачи материала, относящегося к каждому понятию. Предполагается, что пре-

подаватель может прогнозировать времена усвоения каждым студентом каждого понятия при использовании каждого метода подачи материала применительно к оценкам «3», «4» и «5». Будем выбирать так стратегию подачи материала, чтобы суммарное прогнозируемое время его усвоения аудиторией не превышало величины T , а средний балл был бы максимальным.

Введем следующие обозначения и определения:

- $z(i, j, k, q)$ – булева переменная; равна единице, если для формирования k -го понятия у j -го студента преподаватель пользуется i -м методом, позволяющим прогнозировать оценку, равную q , и равна нулю – в противном случае.

- $t(i, j, k, q)$ – прогнозируемое время формирования k -го понятия у j -го студента i -м методом на оценку, равную q .

- m – число используемых преподавателем методов обучения;

- n – число студентов (учеников);

- K – число формируемых понятий.

Тогда рассматриваемая задача формально будет выглядеть следующим образом:

$$\left\{ \begin{array}{l} \sum_i \sum_j \sum_k \sum_q q \cdot z(i, j, k, q) \rightarrow \max; \\ \sum_k \max_i \max_j \sum_q z(i, j, k, q) \cdot t(i, j, k, q) \leq T; \\ \forall k, \forall j: \sum_{i=1}^m \sum_{q=3}^5 z(i, j, k, q) = 1; \\ \forall k, \exists i: \sum_{j=1}^n \sum_{q=3}^5 z(i, j, k, q) = n; \\ \sum_i \sum_j \sum_k \sum_q q \cdot z(i, j, k, q) \geq S \cdot n \cdot K; \\ \forall i, \forall j, \forall k, \forall q: z(i, j, k, q) = 1, 0. \end{array} \right. \quad (3)$$

Рассмотрим задачу на примере 1.

				T	S	
$Mk =$	5, 7, 9	7, 8, 11	1, 3, 5	12 , 13, 16	12	4,50
	4, 5, 6	11 , 13, 16	3, 4, 8	7, 9 , 12	11	4,25
	6, 8, 10	9, 10 , 11	5, 8 , 11	10 , 13, 15	10	4,00
	7, 9, 14	14 , 15, 16	4, 7, 10	5, 10 , 15	14	4,25
	8, 11, 14	16, 19 , 22	2, 5, 8	20 , 22, 24	20	4,25

Исходные данные к примеру 1

Для каждого k -го понятия формируется матрица Mk , строки которой отвечают различным методам обучения, которые могут быть использованы преподавателем при формировании k -го понятия, а столбцы – ученикам. В каждой ячейке $Mk(i, j)$ содержится три числа: первое отражает прогнозируемое время формирования k -го понятия i -м методом у j -го студента на тройку, второе – на четверку, третье – на пятерку.

При использовании преподавателем гарантирующих стратегий для получения нижней границы времени изучения k -го понятия, нижняя граница прогнозируемого времени усвоения всеми студентами k -го понятия равна **10**, соответствующий средний балл равен **4,0**, верхняя граница – **24**, соответствующий средний балл равен **5,0**.

Таким образом, содержательное описание алгоритма поиска решения данной задачи следующее:

- Шаг 0. Ввод величины T , величина $S = 0$.
- Шаг 1. $T1$ = нижняя граница времени формирования k -го понятия, полученная с помощью гарантирующей стратегии.
- Шаг 2. Подставляя в (3) вместо T величину $T1$, выделяем значения $z(i, j, k, q)$, удовлетворяющие системе (3) и, одновременно, максимизирующие сумму баллов $S1$.
- Шаг 3. Если $S1 > S$, то перейти к шагу 4, в противном случае – к шагу 5.
- Шаг 4. $S = S1$.
- Шаг 5. Если T меньше верхней границы, то перейти к шагу 6, в противном случае – к шагу 7.
- Шаг 6. $T = T+1$, перейти к шагу 2.
- Шаг 7. Печать вектора Z , « T равно»; T ; « S =»; $S1$.
- Шаг 8. Конец алгоритма.

Заключение

Рассмотренный подход может быть распространен на другие более сложные задачи.

Литература

1. *Бордовская Н. В.* Методика оценки качества деятельности преподавателей вуза: Методические рекомендации / Н. В. Бордовская, Е. В. Титова. СПб.; Архангельск, 2011.

2. *Граничина О. А.* Изучение значимости показателей качества образовательного процесса, используемых при определении рейтингов высших учебных заведений. М.: Изд-во «Логос», 2011.

3. *Захаревич В. Г., Обуховец В. А.* Оценка качества работы преподавателей вуза // Высшее образование сегодня. 2003. № 5. С. 12–15.

4. *Немов Р. С.* Психология: Учебник для студентов высш. пед. учеб. заведений. В 3 кн. Кн. 2. Психология образования. 2-е изд. М.: Просвещение: ВЛАДОС, 1995. 496 с.

5. *Евланов Л. Г.* Теория и практика принятия решений. М.: Прогресс, 2009.

ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ДЛЯ РЕШЕНИЯ АКТУАЛЬНЫХ ТЕХНИЧЕСКИХ ЗАДАЧ

УДК 681.518.5

ПРИМЕНЕНИЕ АГРЕГИРОВАННЫХ КЛАССИФИКАТОРОВ ДЛЯ ДИАГНОСТИКИ ФУНКЦИОНИРОВАНИЯ ТЕХНИЧЕСКОГО ОБЪЕКТА

Клячкин В. Н., д-р техн. наук, профессор, v_kl@mail.ru

Жуков Д. А., студент

Ульяновский государственный технический университет,
г. Ульяновск, Россия

Аннотация. Задачи диагностики и прогнозирования часто сводятся к разделению состояний объекта на заданном горизонте на исправные, то есть способные выполнять заданные функции, или неисправные. В этом случае целесообразно использовать методы машинного обучения по прецедентам (с учителем). Для диагностики функционирования технического объекта предложена методика, основанная на использовании агрегированных классификаторов – комбинации нескольких базовых методов классификации. При этом предусмотрен выбор объема контрольной выборки и отбор значимых показателей функционирования из условия максимума F -меры.

Ключевые слова: машинное обучение, бинарная классификация, значимые показатели функционирования, F -мера.

APPLICATION OF AGGREGATED CLASSIFIERS FOR THE DIAGNOSIS OF THE FUNCTIONING OF THE TECHNICAL OBJECT

Klyachkin V. N., Zhukov D. A.

Abstract: *Diagnosis and prediction tasks often boil down to the separation conditions on a given object on horizon healthy, that is capable of performing the specified function, or defective. In this case, it is advisable*

to use machine learning techniques for precedents (the teacher). To diagnose the functioning of the technical object the technique based on the use of aggregated classifiers-combination of several basic methods of classification. When this is enabled by selecting the volume control sampling and selection of relevant indicators of the functioning conditions of maximum F-measure.

Keywords: *machine learning, a binary classification, meaningful performance indicators, F-measure.*

Введение

Диагностика и прогнозирование состояния технического объекта по результатам мониторинга показателей его функционирования при эксплуатации могут быть проведены с применением различных инструментов. В ряде ситуаций эффективно использование систем временных рядов, моделирующих процесс эксплуатации объекта [1–2]. Для мониторинга стабильности функционирования часто целесообразно применение методов статистического управления процессами, используемых при контроле технологических процессов [3–4]. Контроль независимых показателей функционирования проводится с помощью карт Шухарта, а для коррелированных показателей используют многомерные методы: алгоритм Хотеллинга для контроля среднего уровня процесса и алгоритм обобщенной дисперсии для мониторинга многомерного рассеяния [4–5].

Часто задачи диагностики и прогнозирования сводятся к разделению состояний объекта на заданном горизонте на исправные, то есть способные выполнять заданные функции, или неисправные. В этом случае целесообразно использовать методы машинного обучения по прецедентам (с учителем). Частным случаем этой задачи является бинарная классификация, т. е. разделение состояний объекта на два класса [6–7]. Используется множество различных подходов к решению задачи классификации. Это статистические методы (байесовские классификаторы, дискриминантный анализ, логистическая регрессия), методы, специально ориентированные на машинное обучение (метод опорных векторов, нейронные сети), композиционные подходы (бэггинг, бустинг) и другие. В работе [8] для повышения качества прогнозирования предложен агрегированный подход – использование комбинации нескольких методов классификации. Эти результаты были подтверждены экспериментально и для задач технической диагностики [9–10].

Цель работы – разработка методики построения классификатора, обеспечивающего необходимую точность при диагностике функционирования технического объекта.

Агрегированные классификаторы

В качестве исходных данных примем матрицу X показателей функционирования системы, элементы которой x_{ij} – результат i -го наблюдения по j -му показателю; $i = 1, \dots, l$, $j = 1, \dots, p$ (l – количество строк или число наблюдений, p – количество столбцов или число показателей) и вектор-столбец ответов Y , состоящий из 1 (для тех опытов, в которых объект исправен) и 0 при неисправном объекте. Каждой строке x_i матрицы X соответствует определённое значение y_i вектора Y . Совокупность пар (x_i, y_i) образует выборку исходных данных – прецедентов.

Задача состоит в построении функции (модели алгоритма) $a : X \rightarrow Y$, которая спрогнозирует ответ Y для любого заданного X [11–12]. При числовых признаках x_i часто используют линейные модели с вектором параметров $w = (w_0 \ w_1 \ \dots \ w_p)$:

$$a(x, w) = w_0 + w_1 x_1 + \dots + w_p x_p, \quad (1)$$

при этом в задачах бинарной классификации обычно вместо нуля и единицы используют множество ответов $Y = \{-1; +1\}$. В этом случае модель алгоритма примет вид:

$$a(x, w) = \text{sign} \sum_{j=0}^p w_j x_j \quad (x_0 = 1). \quad (2)$$

Параметры w_j подбираются по исходным данным; процесс подбора оптимальных параметров называется обучением алгоритма. Найденные параметры должны обеспечить оптимальное значение функционала качества. В рассматриваемой задаче минимизируется функционал ошибок – это среднее количество несовпадений, где $L(a, x_i)$ называют функцией потерь:

$$Q(a, X) = \frac{1}{l} \sum_{i=1}^l L(a, x_i) = \frac{1}{l} \sum_{i=1}^l [a(x_i) - y_i] \rightarrow \min. \quad (3)$$

Алгоритм a , который минимизирует функционал (3), может не обеспечивать хорошее прогнозирование исправности объекта. Ситуация, когда качество работы алгоритма на новых объектах значительно хуже, чем на исходной выборке, свидетельствует о переобучении: алгоритм слишком хорошо подогнан под обучающую выборку и не способен к обобщению на другие выборки. Таким образом, построенный алгоритм не сможет предсказывать состояние исследуемого объекта при новых параметрах функционирования.

Для оценки качества модели с точки зрения возможности прогнозирования исходную выборку из l опытов разбивают (чаще всего случайным образом) на два непересекающихся подмножества: собственно обучающую выборку объёма l_o , с помощью которой и решается задача обучения (3), и контрольную (или тестовую) объёма $l_k = l - l_o$, не используемую для обучения. При использовании кросс-валидации выборка разбивается на N частей. $(N - 1)$ часть используется для обучения, а оставшаяся – для контроля. Последовательно перебираются все варианты. Для каждого разбиения решается задача обучения по выборке l_o и вычисляется функция ошибок $Q(a, X)$ на контрольной выборке l_k . Среднее значение этой функции по всем вариантам разбиения и характеризует обобщающую способность алгоритма.

Качество классификации можно оценить и по дисперсии ошибок – среднему квадрату отклонения истинной вероятности $P(Y_i)$ исправности объекта для i -го объекта контрольной выборки от её прогнозируемого значения $\tilde{P}(Y_i)$:

$$\sigma^2 = \frac{1}{l} \sum_{i=1}^l [P(Y_i) - \tilde{P}(Y_i)]^2. \quad (4)$$

Для исправного объекта $P(Y_i) = 1$, для неисправного $P(Y_i) = 0$.

При несбалансированных классах (когда исправных состояний объекта значительно больше, чем неисправных) доля ошибок не может объективно оценивать качество классификации [12–13]. Более информативны точность

$$P = \frac{tp}{tp + fp}$$

и полнота

$$R = \frac{tp}{tp + fn},$$

где tp – количество правильно классифицированных исправных состояний;

fp – количество неправильно классифицированных исправных состояний;

fn – количество неправильно классифицированных неисправных состояний.

На основе этих двух показателей может быть сформирован единый критерий

$$F = \frac{2PR}{P + R} \quad (5)$$

– это гармоническое среднее точности и полноты (F -мера): чем ближе значение F к единице, тем качество классификации выше.

Функционалом качества также может быть выбрана площадь под ROC-кривой (receiver operating characteristics): AUC (area under the curve) [13–16]. ROC-кривая образуется, если по оси абсцисс брать значения $fp(c)$, а по оси ординат $tp(c)$, где c – порог. Площадь под ROC-кривой позволяет оценить модель в целом, не привязываясь к конкретному порогу. Критерий AUC-ROC устойчив к несбалансированным классам и может быть интерпретирован как вероятность того, что случайно выбранный объект из класса 1 будет иметь значение вероятности ближе к 1, чем случайно выбранный объект из класса 0.

Используется множество различных подходов к бинарной классификации. Практическая реализация расчетов возможна, например, на базе библиотеки инструментов Statistics and Machine Learning Toolbox в пакете Matlab. Часто эффективен агрегированный подход. Он был предложен для решения задач кредитного скоринга [8] и позднее применен для диагностики технического состояния систем. В композиционных подходах (бэггинг, бустинг) для построения ансамбля используется один и тот же метод классификации, построенный или на разных подмножествах выборки, или ориентированный на компенсацию ошибки предыдущей итерации. Представляет интерес совместное использование различных методов классификации, построенных на обучающей выборке. При этом для достижения наилучшего результата надо решить вопросы о том, какие методы обучения использовать, как их объединить, и как принять единое решение об исправности объекта на основе решений отдельных методов.

Воспользуемся полным перебором наборов из H базовых методов (под базовыми будем понимать 11 методов, реализованных в па-

кете Matlab). Тогда, например, при $H = 2$ получим три набора: два базовых и один агрегированный; при $H = 3$ наборов уже 7: три базовых, три агрегированных по два базовых и один агрегированный из всех трех базовых методов. Нетрудно видеть, что в общем случае число наборов равно $2^H - 1$. Для формирования единого решения об исправности объекта на основе решений отдельных методов классификации, рассмотрим агрегирование результатов по среднему значению, по медиане, и с помощью процедуры голосования.

Пусть $\hat{P}_K(X_r)$ – вероятность того, что r -й объект исправен, найденная с помощью K -го базового метода, $K = 1, \dots, H$. Тогда при агрегировании по среднему значению:

$$\hat{P}_{AKcp}(X_r) = \frac{\sum_{K=1}^H \hat{P}_K(X_r)}{H},$$

где $\hat{P}_{AKcp}(X_r)$ – вероятность того, что r -й объект исправен.

При агрегировании по медиане вначале следует ранжировать ряд, содержащий результаты базовых методов в наборе. При нечетном числе базовых методов вероятность того, что r -й объект исправен:

$$\hat{P}_{AKmed}(X_r) = \frac{\hat{P}_{H+1}(X_r)}{2}.$$

В случае четного числа базовых методов, соответствующая вероятность находится как полусумма результатов срединных значений.

Результат агрегированного метода классификации по голосованию представляет собой среднее значение результатов базовых методов, которые определили исправность объекта с вероятностью, например, не ниже, чем 0,1 (вероятность исправности объекта при исходных показателях функционирования объекта X_r $\hat{P}_K(X_r) \geq 0,1$). В противном случае вероятность того, что r -й объект исправен, считается нулевой. Таким образом, значения вероятностей классификации, оказавшиеся ниже 0,1, приравниваются к 0, а оставшиеся – к 1, и по этим значениям строятся агрегированные модели классификации.

При этом, поскольку, как уже отмечалось, разбивка исходных данных на обучающую и контрольную выборки производится случай-

ным образом, структуры агрегированных классификаторов оказываются различными, и возникает вопрос, какую структуру предпочесть для принятия окончательного решения об исправности объекта.

Алгоритм диагностики функционирования объекта

Предлагается следующий алгоритм построения классификатора для диагностики функционирования технического объекта.

1. Ввод исходных данных: матрицы X показателей функционирования и вектора-столбца ответов Y .

2. Ввод объема контрольной выборки (начальное значение 25 %) и формирование обучающей и контрольной выборок случайным образом.

3. Расчет F -меры по всем базовым классификаторам на всех показателях функционирования объекта с использованием кросс-валидации (в табл. 1 показан результат выполнения этого этапа на конкретном примере).

4. Выбор двух классификаторов, для которых значение F -меры максимально.

Таблица 1

Значения F -меры для базовых классификаторов

Классификатор	F -мера
Нейронная сеть (НС)	0,838
Метод опорных векторов (МОВ)	0,824
Логистическая регрессия (ЛР)	0,831
Дискриминантный анализ (ДА)	0,819
Наивный байесовский классификатор (БК)	0,825
Бэггинг деревьев решений (БДР)	0,873
Градиентный бустинг (ГБ)	0,851
AdaBoost (AB)	0,846
LogitBoost (LB)	0,841
GentleBoost (GB)	0,832
RUSBoost (RB)	0,835

5. Снижение объема контрольной выборки на 5 %, формирование обучающей и контрольной выборок, расчет F -меры по двум выбранным в п. 4 классификаторам с использованием кросс-валидации.

6. Повторение п. 5 с последовательным снижением объема контрольной выборки до 5 %.

7. Определение объема контрольной выборки, обеспечившего максимум F -меры по двум выбранным в п. 4 классификаторам (табл. 2; наилучший объем выборки 15 %).

Таблица 2

Значения F -меры при различных объемах контрольной выборки

Классификатор	F -мера при объеме контрольной выборки				
	25 %	20 %	15 %	10 %	5 %
Бэггинг деревьев решений (БДР)	0,873	0,878	0,882	0,872	0,869
Градиентный бустинг (ГБ)	0,851	0,851	0,856	0,854	0,849

8. Отбор показателей функционирования (столбцов матрицы X), значимо коррелированных с вектором Y , и расчет F -меры при объеме контрольной выборки, найденном в п. 7, по двум выбранным в п. 4 классификаторам.

9. Отбор показателей функционирования (столбцов матрицы X), значимых в линейной регрессионной модели $Y(X)$, и расчет F -меры при объеме контрольной выборки, найденном в п. 7, по двум выбранным в п. 4 классификаторам.

10. Выбор подмножества столбцов матрицы X , обеспечившего максимум F -меры (один из трех вариантов: 1) все показатели; 2) показатели, значимо коррелированные с вектором Y , или 3) показатели, значимые по регрессионной модели) (табл. 3 рассчитана при объеме контрольной выборки 15 %, наилучшее подмножество – 3)).

11. Построение агрегированных классификаторов по среднему, по медиане и по голосованию, путем перебора всех пар, включающих наилучший из базовых классификаторов, и всех остальных базовых классификаторов (проведенное ранее исследование показало, что увеличение в агрегате числа базовых классификаторов больше двух не дает значимого повышения значения F -меры).

Таблица 3

Значения F -меры при различных подмножествах показателей

Классификатор	F -мера при подмножествах показателей		
	1	2	3
Бэггинг деревьев решений (БДР)	0,882	0,880	0,887
Градиентный бустинг (ГБ)	0,856	0,852	0,859

12. Выбор наилучшего агрегированного классификатора (обеспечившего максимум F -меры) для дальнейшего прогнозирования исправности объекта (табл. 4: при агрегировании по среднему и по медиане результаты оказались одинаковыми, F -мера = 0,894 при структуре агрегата БДР+ГБ; этот вариант и обеспечит наилучшие результаты при прогнозировании исправности объекта).

Таблица 4

**Значения F -меры
при различных агрегированных классификаторах**

Метод агрегирования	Структура классификатора	F -мера
По среднему	БДР + ГБ	0,894
По медиане	БДР + ГБ	0,894
По голосованию	БДР + АВ	0,891

Заключение

Для диагностики функционирования технического объекта предложена методика, основанная на использовании агрегированных классификаторов при машинном обучении по прецедентам. При этом предусмотрен выбор объема контрольной выборки и отбор значимых показателей функционирования из условия максимума F -меры.

Литература

1. Клячкин В. Н., Кувайскова Ю. Е., Бубырь Д. С. Прогнозирование состояния объекта с использованием систем временных рядов // Радиотехника. 2015. № 6. С. 45–47.
2. Krashennnikov V. R., Klyachkin V. N., Kuvayskova Yu. E. Models updating for technical objects state forecasting // 3rd Russian-Pacific Conference on Computer Technology and Applications. IEEE Xplore. 2018. P. 1–4.
3. Уиллер Д., Чамберс Д. Статистическое управление процессами. Оптимизация бизнеса с использованием контрольных карт Шухарта / Пер. с англ. М.: Альпина Бизнес Букс, 2009. 409 с.

4. *Montgomery D. C.* Introduction to statistical quality control // New York, John Wiley and Sons, 2009. 754 p.

5. *Клячкин В. Н.* Модели и методы статистического контроля многопараметрического технологического процесса. М.: ФИЗМАТ-ЛИТ, 2011. 196 с.

6. *Witten I. H., Frank E.* Data mining: practical machine learning tools and techniques. 2nd ed. San Francisco: Morgan Kaufmann Publishers, 2005. 525 p.

7. Теория и практика машинного обучения: Учебное пособие / В. В. Воронина, А. В. Михеев, Н. Г. Ярушкина, К. В. Святков. Ульяновск: УлГТУ, 2017. 290 с.

8. *Клячкин В. Н., Шунина Ю. С.* Система оценки кредитоспособности заемщиков и прогнозирования возврата кредитов // Вестник компьютерных и информационных технологий. 2015. № 11. С. 45–51.

9. *Klyachkin V. N., Kuvayskova Yu. E., Zhukov D. A.* The use of aggregate classifiers in technical diagnostics, based on machine learning // CEUR Workshop Proceeding, vol. 1903. Data Science. Information Technology and Nanotechnology. 2017. P. 32–35.

10. *Клячкин В. Н., Кувайскова Ю. Е., Жуков Д. А.* Диагностика технического состояния аппаратуры с использованием агрегированных классификаторов // Радиотехника. 2018. № 6. С. 46–49.

11. *Воронцов К. В.* Машинное обучение. Композиция классификаторов. URL: <https://yadi.sk/i/FItIu6V0beVmF> (01.04.2019).

12. *Соколов Е. А.* ФКН ВШЭ. Лекция 4. Линейная классификация // <https://github.com/esokolov/ml-course-hse/blob/master/2018-fall/lecture-notes/lecture04-linclass.pdf> (01.04.2019).

13. *Davis J., Goadrich M.* The relationship between Precision-Recall and ROC curves // Proceedings of the 23rd international conference on Machine learning. Pittsburgh. 2006. Pp. 233–240.

14. *Дьяконов А. М.* AUC ROC (площадь под кривой ошибок) // <https://dyakonov.org/2017/07/28/auc-roc-площадь-под-кривой-ошибок/#more-5362> (01.04.2019).

15. *Neykov M.* On the Characterization of a Class of Fisher-Consistent Loss Functions and its Application to Boosting / M. Neykov, Jun S. Liu, Tianxi Cai // Journal of Machine Learning Research. 2016. № 17 (70). P. 1–32.

16. *Wyner A. J.* Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers / Abraham J. Wyner, Matthew Olson, Justin Bleich, David Mease // Journal of Machine Learning Research. 2017. № 18 (48). P. 1–33.

АЛГОРИТМ ПАРАМЕТРИЗАЦИИ ПОРОГОВОЙ КРИПТОСХЕМЫ РАЗДЕЛЕНИЯ СЕКРЕТА С ПРИМЕНЕНИЕМ МИНИМАЛЬНО ИЗБЫТОЧНОЙ МОДУЛЯРНОЙ СИСТЕМЫ СЧИСЛЕНИЯ

Коляда А. А., д-р физ.-мат. наук, доцент, razan@tut.by,

Коляда Н. А., науч. сотрудник, razan@tut.by,

Протасеня С. Ю., мл. науч. сотрудник, estellita@mail.ru

Научно-исследовательское учреждение «Институт прикладных физических проблем имени А. Н. Севченко» Белорусского государственного университета,

Минск, Республика Беларусь, <http://niipfp.bsu.by>

Аннотация. Приведены базовые теоретические положения пороговых криптосхем разделения секрета, основу которых составляет минимально избыточная модулярная арифметика (МИМА). Применение МИМА обеспечивает значительное сокращение (в сравнении с аналогами) реализационных затрат на выполнение декодирующей процедуры. Впервые дана математическая формализация проблемы корректности порогового принципа модулярного разделения секретной информации, сформулирован алгоритм параметризации криптосхем исследуемого класса.

Ключевые слова: разделение секрета, минимально избыточная модулярная арифметика, пороговая схема модулярного разделения секрета, псевдослучайная маскирующая функция, распределенные вычисления.

ALGORITHM OF PARAMETERIZATION OF THRESHOLD CRYPTOScheme OF THE SECRET DIVISION USING THE MINIMALLY REDUNDANT MODULAR NUMBER SYSTEM

Kolyada A. A., Kolyada N. A., Protasenya S. Yu.

Abstract. *The basic theoretical positions of the threshold cryptographic schemes of secret separation, which are based on minimum redundant modular arithmetic (MRMA), are given. The use of MRMA provides a significant reduction (in comparison with analogues) of the implementation costs of the decoding procedure. The mathematical formalization of the*

problem of correctness of the threshold principle of modular separation of secret information is given for the first time. The algorithm of parameterization of cryptosystems of the investigated class is formulated.

Keywords: *division of secret, minimally redundant modular arithmetic, modular threshold scheme of secret sharing, pseudo-random masking function, distributed computing.*

Неотъемлемой составляющей современного процесса развития распределенных компьютерных и инфокоммуникационных систем является обеспечение информационной безопасности на всех этапах хранения, обработки и передачи данных [1, 2]. Эффективное решение обозначенной задачи дает так называемая технология активной безопасности, предусматривающая периодическое обновление криптографических ключей, а также их пространственное разделение. На практике разделение секретной информации обычно осуществляется в рамках пороговых схем [1–6]. Реализуемое пороговой (t, n) -системой решающее правило обеспечивает разделение секрета (криптографического ключа) n абонентами с возможностью восстановления его по компонентам, принадлежащим любым l участникам сеанса связи ($2 \leq t \leq l \leq n$), причем группы абонентов числом k меньше порогового значения t , реконструировать искомый секрет по соответствующим компонентам не могут.

Перспективный инструментарий для построения пороговых криптосхем разделения секрета представляет арифметика модулярных систем счисления (МСС) [1, 2, 5, 6]. С одной стороны, модулярное кодирование служит простым средством декомпозиции (разделения) секрета на составные части, а с другой – позволяет минимизировать реализационные затраты при оперировании в диапазонах больших чисел. В криптографических приложениях фактор производительности занимает центральное место. Поэтому для решения проблем создания пороговых систем разделения секретной информации отмеченное обстоятельство имеет особую важность.

Представляемая разработка посвящена построению и оптимизации параметрической базы криптосхем исследуемого класса, базирующихся на арифметике МСС. Фундаментальные преимущества модулярных вычислительных структур наиболее полно удастся реализовать в рамках так называемого минимально избыточного кодирования [1, 7]. Исходя из этого, главное внимание в настоящем сообщении уделяется пороговым криптосхемам разделения секрета, компьютерно-арифметическую основу которых составляет минимально избы-

точная модулярная арифметика (МИМА) – арифметика минимально избыточных МСС [1].

Введем обозначения:

- $\lfloor a \rfloor$ и $\lceil a \rceil$ – наибольшее и наименьшее целые числа (ЦЧ) соответственно не большее и не меньшее вещественной величины a ;
- $\mathbf{Z}_m = \{0, 1, \dots, m-1\}$ – множество наименьших неотрицательных вычетов по натуральному модулю m ;
- $|a|_m = A(\bmod m)$ – элемент множества \mathbf{Z}_m , сравнимый с ЦЧ A по модулю m ;
- $|A/B|_m$ – элемент χ множества \mathbf{Z}_m , удовлетворяющий сравнению $B\chi \equiv A(\bmod m)$ (A и B – ЦЧ, $|B|_m \neq 0$);
- $(\chi_1, \chi_2, \dots, \chi_s) = (|X|_{m_1}, |X|_{m_2}, \dots, |X|_{m_s})$ – представление ЦЧ X (модулярный код) в МСС с основаниями m_1, m_2, \dots, m_s , составляющими её базис $\{m_1, m_2, \dots, m_s\}$ ($s > 1$).

Пусть p_1, p_2, \dots, p_n – упорядоченные по возрастанию попарно простые большие натуральные числа ($n > 1$);

$$P_i = \prod_{s=1}^i p_s \quad (i = \overline{1, n}); \quad _P_j = \prod_{s=0}^{j-1} p_{n-s} = P_n / P_{n-j} \quad (j = \overline{1, n});$$

$$\mathbf{P} = \{p_1, p_2, \dots, p_n\};$$

$$\mathbf{P}(i_1, i_2, \dots, i_l) = \{p_{i_1}, p_{i_2}, \dots, p_{i_l}\} (1 \leq i_1 < i_2 < \dots < i_l \leq n; t \leq l \leq n);$$

$$\mathbf{P}(j_1, j_2, \dots, j_k) = \{p_{j_1}, p_{j_2}, \dots, p_{j_k}\} (1 \leq j_1 < j_2 < \dots < j_k \leq n; 2 \leq k < t);$$

p – большое натуральное число, взаимно простое с p_1, p_2, \dots, p_n .

Построение модулярной пороговой (t, n) -схемы разделения секрета с базисом \mathbf{P} , которая рассчитана на полное число n и пороговое число t разделяющих секрет сторон (абонентов), осуществляется в рамках следующих определяющих условий.

А. Исходный секрет, разделяемый n сторонами, представляет собой целое число $S \in \mathbf{Z}_p$.

Б. Секрет S разделяется путем его модулярной декомпозиции, то есть по правилу $\sigma_i = |S|_{p_i}$ ($i = \overline{1, n}$). При этом i -я сторона имеет часть σ_i секрета S .

В. Над S выполняется маскирующее преобразование $\tilde{S} = C + Cp$, где C – псевдослучайная целочисленная величина и результат маскирования подвергается модулярной декомпозиции:

$$\tilde{\sigma}_i = |\tilde{S}|_{p_i} = |\sigma_i + |Cp|_{p_i}|_{p_i} \quad (i = \overline{1, n}).$$

Цифры модулярного кода $(\tilde{\sigma}_1, \tilde{\sigma}_2, \dots, \tilde{\sigma}_n)$ рассматриваются как частичные маскирующие секреты, принадлежащие соответствующим абонентам.

Г. Любые t абонентов могут восстановить секрет S по принадлежащим им маскирующим частичным секретам. Но никакая группа абонентов числом меньше t сделать этого не может.

Теоретико-методологическую базу пороговых МИМА-криптосхем разделения секрета, синтезируемых согласно условиям А-Г, составляют нижеследующие утверждения.

Теорема 1. Пусть основания базиса $P = \{p_1, p_2, \dots, p_n\}$ модулярной (t, n) -схемы разделения секрета $S \in Z_p$ упорядочены по возрастанию и взаимно просты с p . Для того чтобы диапазоны $\{0, 1, \dots, \prod_{i=1}^k p_{j_i} - 1\}$ изменения вычетов $\tilde{S}(\text{mod } \prod_{i=1}^k p_{j_i}) = (\tilde{\sigma}_{j_1}, \tilde{\sigma}_{j_2}, \dots, \tilde{\sigma}_{j_k})$ в МСС с базисами $P(j_1, j_2, \dots, j_k) (2 \leq k < t)$ не пересекались с множеством $\tilde{S} = \{\tilde{S}_{nn}, \tilde{S}_{nn} + 1, \dots, \tilde{S}_{en}\}$ значений маскирующего секрета \tilde{S} (результата маскирования S), имеющего в МСС с базисами $P(i_1, i_2, \dots, i_l) \cdot (t \leq l \leq n)$ коды $(\tilde{\sigma}_{i_1}, \tilde{\sigma}_{i_2}, \dots, \tilde{\sigma}_{i_l})$, достаточно выполнения условия $\tilde{S}_{nn} = p_{t-1}, \tilde{S}_{en} = p_t - 1$, то есть $\tilde{S} \in \tilde{S} = \{-p_{t-1}, -p_{t-1} + 1, \dots, p_t - 1\}$.

Замечание 1. Диапазоны $\{0, 1, \dots, \prod_{j=1}^l p_{j_i} - 1\}$ всех МСС с базисом $P(i_1, i_2, \dots, i_l)$ включают фигурирующее в теореме 1 множество \tilde{S} , вследствие чего его правомерно квалифицировать как диапазон модулярной пороговой (t, n) -криптосхемы разделения секрета.

Теорема 2. Пусть $m_1 = p_{i_1}, m_2 = p_{i_2}, \dots, m_l = p_{i_l}; M_{l-1} = \prod_{j=1}^{l-1} m_j; M_{j,l-1} = \frac{M_{l-1}}{m_j}, \tilde{\sigma}_j = |\tilde{S}|_{m_j} (j = \overline{1, l}); I_l(\tilde{S})$ – интервальный индекс (ИИ) числа $\tilde{S} = (\tilde{\sigma}_1, \tilde{\sigma}_2, \dots, \tilde{\sigma}_l)$ в МСС с базисом $\{m_1, m_2, \dots, m_l\}$, определяемый равенством:

$$\tilde{S} = \sum_{j=1}^{l-1} M_{j,l-1} |M_{l-1}^{-1} \tilde{\sigma}_j|_{m_j} + M_{l-1} I_l(\tilde{S}). \quad (1)$$

Для того чтобы в МСС с основаниями m_1, m_2, \dots, m_l ИИ $I_l(\tilde{S})$ каждого элемента \tilde{S} диапазона $\{-p_{t-1}, -p_{t-1} + 1, \dots, p_0 p_{t-1} - 1\} (p_0 - \text{вспомогательный модуль})$ полностью определялся компьютерным ИИ – вычетом $\hat{I}_l(\tilde{S}) = |I_l(\tilde{S})|_{m_l}$, необходимо и достаточно, чтобы l -е основание удовлетворяло условию: $m_l \geq p_0 + l - 2 (p_0 \geq l - 2)$.

При этом для $I_l(\tilde{S})$ верны расчетные соотношения:

$$I_l(\tilde{S}) = \begin{cases} \hat{I}_l(\tilde{S}), & \text{если } \hat{I}_l(\tilde{S}) < p_0, \\ \hat{I}_l(\tilde{S}) - m_l, & \text{если } \hat{I}_l(\tilde{S}) \geq p_0; \end{cases} \quad (2)$$

$$\hat{I}_l(\tilde{S}) = \left| \sum_{j=1}^l R_{j,l}(\tilde{\sigma}_j) \right|_{m_l}; \quad (3)$$

$$R_{j,l}(\tilde{\sigma}_j) = \left| -m_j^{-1} \left| M_{j,l-1}^{-1} \tilde{\sigma}_j \right|_{m_j} \right|_{m_l} \quad (j \neq l), \quad R_{l,l}(\tilde{\sigma}_l) = \left| M_{l-1}^{-1} \tilde{\sigma}_l \right|_{m_l}. \quad (4)$$

Замечание 2. Теорема 2 остается в силе на любых диапазонах $\tilde{S} \subseteq \{_{-}P_{t-1}, _{-}P_{t-1} + 1, \dots, p_0 P_{t-1} - 1\}$. Семейство таких диапазонов можно описать в виде:

$$\tilde{S} = \{\tilde{S}_{НП}, \tilde{S}_{НП} + 1, \dots, \tilde{S}_{ВП}\} = \{[_{-}P_{t-1}pq], [_{-}P_{t-1}pq] + 1, \dots, [qP_t] - 1\} \\ (q \geq p^{-1}; 0 < _{-}q \leq p_0/p_t).$$

При использовании указанных диапазонов вспомогательный модуль p_0 должен удовлетворять ограничению:

$$_{-}qp_t \leq p_0 \leq p_t - t + 2.$$

Теорема 3. Пусть p_1, p_2, \dots, p_n – упорядоченные по возрастанию попарно простые натуральные числа, составляющие базис \mathbf{P} МИМА-схемы разделения секрета S , p – взаимно простой с p_1, p_2, \dots, p_n модуль кольца \mathbf{Z}_p принадлежности секрета S , который разделяется между n абонентами путем надления их частичными маскирующими секретами $\tilde{\sigma}_i = \left| \tilde{S} \right|_{p_i}$ ($i = \overline{1, n}$), получаемыми в результате модулярной декомпозиции применяемой функции маскирования: $\tilde{S} = S + Cp$ (C – псевдослучайный целочисленный параметр). Для того чтобы любые l абонентов ($2 \leq t \leq l \leq n$), за которыми закреплены основания $p_{i_1}, p_{i_2}, \dots, p_{i_l}$ ($1 \leq i_1 < i_2 < \dots < i_l \leq n$), могли восстановить секрет S по набору принадлежащих им частичных секретов – модулярному коду $(\tilde{\sigma}_{i_1}, \tilde{\sigma}_{i_2}, \dots, \tilde{\sigma}_{i_l})$ маскирующего секрета \tilde{S} , – никакая группа,

включающая $k < t$ абонентов, которым отвечают основания $p_{j_1}, p_{j_2}, \dots, p_{j_k}$ ($1 \leq j_1 < j_2 < \dots < j_k \leq n$), не имела возможности восстановления S по коду $(\tilde{\sigma}_{j_1}, \tilde{\sigma}_{j_2}, \dots, \tilde{\sigma}_{j_k})$, достаточно выполнения системы условий:

$$\left\{ \begin{array}{l} \tilde{S} \in \tilde{S} = \{\tilde{S}_{nn}, \tilde{S}_{nn} + 1, \dots, \tilde{S}_{nn}\} = \{[-P_{t-1}pq], [-P_{t-1}pq] + 1, \dots, [-qP_t] - 1\}, \\ C \in (C \setminus C_p), \end{array} \right.$$

где

$$q \geq p^{-1}; \quad 0 < -q \leq p_0/p_t; \quad -qp_t \leq p_0 \leq p_t - t + 2;$$

$$-P_{t-1} = \prod_{s=0}^{t-2} p_{n-s}; \quad P_t = \prod_{s=1}^t p_s;$$

$$C = \{C_{H\Pi}, C_{H\Pi} + 1, \dots, C_{B\Pi}\} \left(C_{H\Pi} = \left\lfloor \frac{\tilde{S}_{H\Pi}}{p} \right\rfloor; \quad C_{B\Pi} = \left\lfloor \frac{\tilde{S}_{B\Pi}}{p} \right\rfloor \right);$$

$$C_p = \{C \in C \mid \tilde{S} = S + Cp \in \tilde{S};$$

$$p - \text{делитель ЦЧ } Q = Q(\tilde{S}; j_1, j_2, \dots, j_k) = \left\lfloor \frac{\tilde{S}}{\prod_{i=1}^k p_{j_i}} \right\rfloor$$

$$(1 \leq j_1 < j_2 < \dots < j_k \leq n; 2 \leq k < t);$$

$$Q(\tilde{S}; j_1, j_2, \dots, j_k) \in \{Q_{H\Pi}, \quad Q_{H\Pi} + 1, Q_{B\Pi}\} =$$

$$= \left\{ \left\lfloor \frac{\tilde{S}_{H\Pi}}{\prod_{i=1}^k p_{j_i}} \right\rfloor, \left\lfloor \frac{\tilde{S}_{H\Pi}}{\prod_{i=1}^k p_{j_i}} \right\rfloor + 1, \dots, \left\lfloor \frac{\tilde{S}_{B\Pi}}{\prod_{i=1}^k p_{j_i}} \right\rfloor \right\}.$$

На базе приведенных теоретических приложений синтезирован алгоритм параметризации порогового МИМА-криптомодуля разделения секрета, состоящий в нижеследующем.

Входные данные алгоритма:

- полное число n и пороговое число t абонентов распределенной системы обработки данных ($2 \leq t \leq n$);
- разрядности оснований базиса $\mathbf{P} = \{p_1, p_2, \dots, p_n\}$ криптомодуля и модуля p кольца Z_p , содержащего секрет-оригинал S ;

- параметры q и $_q$ ($q \geq p^{-1}$; $0 < _q < 1$), позволяющие регулировать соответственно нижнюю и верхнюю границы рабочего диапазона криптомодуля.

Выходные данные:

- рабочий диапазон криптомодуля – множество $\tilde{S} = \{\tilde{S}_{НП}, \tilde{S}_{НП} + 1, \dots, \tilde{S}_{ВП}\}$ значений маскирующего секрета $\tilde{S} = S + Cp$ ($\tilde{S}_{НП}$ и $\tilde{S}_{ВП}$ – соответственно нижний и верхний пороги для \tilde{S} ; C – псевдослучайный целочисленный параметр).

- модуль p , оптимизирующий по мощности и структуре диапазон изменения параметра C ;

- множество C_p нейтрализуемых значений и реальный диапазон \tilde{C} изменения параметра C .

Тело алгоритма параметризации криптомодуля:

ПКМ. 1. В соответствии с установленным ограничением на разрядность сформировать модулярный базис P из упорядоченных по возрастанию попарно простых оснований p_1, p_2, \dots, p_n криптомодуля.

ПКМ. 2. Следуя теореме 2, выбрать вспомогательный модуль p_0 , удовлетворяющий условию:

$$_q p_t \leq p_0 \leq p_t - t + 2.$$

ПКМ. 3. В качестве рабочего диапазона криптомодуля принять множество:

$$\tilde{S} = \{\tilde{S}_{НП}, \tilde{S}_{НП} + 1, \dots, \tilde{S}_{ВП}\} = \{[-P_{t-1}pq], [-P_{t-1}pq] + 1, \dots, [-qp_0P_{t-1}] - 1\}.$$

ПКМ. 4. Положить $_p = 0$, а также активировать массивы C_p, \tilde{C} и $_C_p, _C$.

ПКМ. 5. Присвоить модулю p очередное значение из анализируемого набора.

ПКМ. 6. Получить исходный (с максимально возможной мощностью) диапазон изменения псевдослучайного параметра C маскирующей функции:

$$C = \{C_{НП}, C_{НП} + 1, \dots, C_{ВП}\} = \left\{ \left\lfloor \frac{\tilde{S}_{НП}}{p} \right\rfloor, \left\lfloor \frac{\tilde{S}_{НП}}{p} \right\rfloor + 1, \dots, \left\lfloor \frac{\tilde{S}_{ВП}}{p} \right\rfloor \right\} = \\ = \{ \lfloor [-P_{t-1}pq]/p \rfloor, \lfloor [-P_{t-1}pq]/p \rfloor + 1, \dots, \lfloor (-qp_0P_{t-1} - 1)/p \rfloor \}.$$

ПКМ. 7. Применяя теоремы 3 и 4, определить подмножества:

$$\mathbf{P}(j_1, j_2, \dots, j_k) (1 \leq j_1 < j_2 < \dots < j_k \leq n; 2 \leq k < t)$$

базиса \mathbf{P} , которые порождают числа:

$$Q = Q(\tilde{S}; j_1, j_2, \dots, j_k) = \lfloor \tilde{S} / \prod_{i=1}^k p_{j_i} \rfloor,$$

кратные модулю p и отвечающие им $C \in \mathbf{C}$. С этой целью для всех (j_1, j_2, \dots, j_k) ($k = 2, t-1$) выполнить действия:

ПКМ. 7.А. Найти пороговые значения целого числа $Q = Q(\tilde{S}; j_1, j_2, \dots, j_k)$ по формулам:

$$Q_{НП}(j_1, j_2, \dots, j_k) = \left\lfloor \frac{\tilde{S}_{НП}}{\prod_{i=1}^k p_{j_i}} \right\rfloor$$

и

$$Q_{ВП}(j_1, j_2, \dots, j_k) = \lfloor \tilde{S}_{ВП} / \prod_{i=1}^k p_{j_i} \rfloor.$$

ПКМ. 7.Б. Выделить из множества:

$$\{Q_{НП}(j_1, j_2, \dots, j_k), Q_{НП}(j_1, j_2, \dots, j_k) + 1, \dots, Q_{ВП}(j_1, j_2, \dots, j_k)\}$$

элемент Q , кратный модулю p , то есть элемент вида $Q = Dp$, где $D = D(j_1, j_2, \dots, j_k)$ – натуральное число. При отсутствии указанного элемента перейти к ПКМ. 7.Д

ПКМ. 7.В. Рассчитать значения $C(j_1, j_2, \dots, j_k)$ псевдослучайного параметра C , отвечающие числу $Q = D(j_1, j_2, \dots, j_k)p$ и подмножеству $\mathbf{P}(j_1, j_2, \dots, j_k)$ оснований базиса \mathbf{P} по правилу:

$$C(j_1, j_2, \dots, j_k) = D(j_1, j_2, \dots, j_k) \prod_{i=1}^k p_{j_i} + d(j_1, j_2, \dots, j_k) \quad (d(j_1, j_2, \dots, j_k) = 0, 1, \dots, \lfloor (\prod_{i=1}^k p_{j_i}) / p \rfloor).$$

ПКМ. 7.Г. Значения $C(j_1, j_2, \dots, j_k)$ параметра C , попадающие в диапазон \mathbf{C} , поместить в массив \mathbf{C}_p при условии, что там они уже не находятся. При этом запись в \mathbf{C}_p производится с упорядочением элементов по возрастанию.

ПКМ. 7.Д. Если подмножество $\mathbf{P}(j_1, j_2, \dots, j_k)$ оснований базиса \mathbf{P} криптомодуля не является последним из числа подлежащих рассмотрению, то набор (j_1, j_2, \dots, j_k) заместить очередным и перейти к ПКМ. 7.А.

ПКМ. 8. При $_p = 0$ перейти к ПКМ. 10.

ПКМ. 9. Провести сравнительный анализ качественных характеристик определяемого теоремой 4 реального диапазона $\tilde{C} = C | C_p$ изменения псевдослучайного параметра C при текущем p и оптимального на данный момент диапазона $_C$, полученного для модуля $p = _p$. В случае, когда по применяемым реализационным критериям, в частности, по критериям максимума мощности и простоты практического использования множество $_C$ более приемлемо, чем \tilde{C} , перейти к ПКМ.11.

ПКМ. 10. Выполнить операцию присвоения $_p = p$ с заменой $_C_p$ и $_C$ на C_p и \tilde{C} .

ПКМ. 11. Если анализируемый набор модулей p не исчерпан, то перейти к ПКМ.5.

ПКМ. 12. Модуль $p = _p$ и множества $\tilde{C} = _C$, $C_p = _C_p$ зафиксировать в качестве искомых выходных данных и завершить работу алгоритма.

Использование формируемого алгоритмом ПКМ.1 – ПКМ.12 множества \tilde{C} при генерировании маскирующего секрета \tilde{S} исключает возможность восстановления секрета-оригинала S по модулярным кодам $(\tilde{\sigma}_{j_1}, \tilde{\sigma}_{j_2}, \dots, \tilde{\sigma}_{j_k})$, так как $\tilde{S} \notin \{0, 1, \dots, \prod_{i=1}^k p_{j_i} - 1\}$. Вместе с тем в МСС с базами $P(i_1, i_2, \dots, i_l)$ декодирующее преобразование $\tilde{S} \rightarrow S$ в виду $\tilde{S} \in \tilde{S} \subseteq \{0, 1, \dots, \prod_{j=1}^l p_{i_j} - 1\}$ осуществляется корректно, причем по упрощенным МИМА-процедурам [1, 7], базирующимся на теореме (1) – (4).

Литература

1. Червяков Н. И. Модулярная арифметика и ее приложения в инфокоммуникационных технологиях / Н. И. Червяков, А. А. Коляда, П. А. Ляхов [и др.]. М.: ФИЗМАТЛИТ, 2017. 400 с.

2. Червяков Н. И. Применение искусственных нейронных сетей и системы остаточных классов в криптографии / Н. И. Червяков [и др.]. М.: Физматлит, 2012. 280 с.

3. Червяков Н. И. Численный метод порогового разделения секрета над группой точек эллиптической кривой / Н. И. Червяков, М. Г. Бабенко, П. А. Ляхов, А. П. Гладков, М. С. Афонин // Инфокоммуникационные технологии. 2013. Т. 11. № 1. С. 4–11.

4. Bahramian Mojtaba. An efficient threshold verifiable multiset sharing scheme using generalized jacobian of elliptic curves / Mojtaba

Bahramian, Khadijeh Eslami // Journal of algebraic structures and their applications. 2017. Vol. 4, issue 2. P. 45–55.

5. *Jia Xingxing*. A new threshold changeable secret sharing scheme based on the Chinese remainder theorem / Xingxing Jia, Daoshun Wang, Daxin Nie, Xiangyang Luo, Jonathan Zheng Sun // Information sciences. 2019. Vol. 473. P. 13–30.

6. *Ananda Mohan P. V.* Residue number systems: Theory and applications / P. V. Ananda Mohan. Basel: Birkhauser, Mathematics, 2016. 351 p.

7. Червяков Н. И. (RU), Коляда А. А. (BY), Кучуков В. А. (RU), Бабенко М. Г. (RU). Патент на изобретение № 2652450 РФ. МПК: J06F7/57, H03K19/00. Устройство вычисления модулярного произведения Монтгомери; Заявка № 2017129526. Приоритеты; дата подачи заявки – 18.08.2017; опубл. 26.04.2018; бюл. № 12.

УДК 004.031.6

АДАПТАЦИЯ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ К ПОТРЕБНОСТЯМ КОНКРЕТНОГО ПРЕДПРИЯТИЯ

Рябова А. А., канд. техн. наук, доцент, alinaryabova@gmail.com

Курдов К. Р., студент, cvxwasd@yandex.ru

Российский экономический университет им. Г. В. Плеханова,
филиал в г. Пятигорске,
г. Пятигорск, Россия

Аннотация. Статья посвящена анализу адаптации корпоративной информационной системы к потребностям конкретного предприятия. В ходе исследования на основании изученной литературы выявлены особенности адаптации КИС на примере деятельности предприятия ИКЕА. Сделан вывод о преимуществе внедрения корпоративной информационной системы для рассматриваемого предприятия: произойдёт улучшение внутренней управляемости, гибкости и устойчивости к внешним воздействиям, увеличится конкурентоспособность и прибыльность корпорации.

Ключевые слова: адаптация; корпоративные информационные системы; ИКЕА, хранение данных, автоматизация.

ADAPTATION OF CORPORATE INFORMATION SYSTEMS TO REQUIREMENTS OF THE CONCRETE ENTERPRISE

Ryabova A. A., Kurdov K. R.

Abstract. Article is devoted to the analysis of adaptation of a corporate information system to requirements of the concrete enterprise. During the research on the basis of the studied literature features of adaptation of corporate information systems on the example of activity of the IKEA enterprise are revealed. The conclusion is drawn on advantage of introduction to a corporate information system for the considered enterprise: there will be an improvement of internal controllability, flexibility and resistance to external influences, the competitiveness and profitability of corporation will increase.

Keywords: adaptation; corporate information systems; IKEA; data storage; automation.

Введение

Получение положительного эффекта от внедрения автоматизированной системы зависит от области охвата всех направлений деятельности компании. В настоящее время именно поэтому становится актуальной идея реализации корпоративных информационных систем (КИС).

Под корпоративной информационной системой здесь понимается система, реализующая информационные технологии для применения эффективных методов управления предприятием масштаба корпорации. Характерными чертами КИС являются: открытая архитектура построения; распределенная система обработки данных; развитая коммуникационная подсеть; многоплатформенность приложений и баз данных; новые информационные технологии корпоративного типа [3, с. 87].

КИС должны отвечать целому набору обязательных требований, представленных на рисунке 1 [2, с. 46].

Представленные на сегодняшний день системы управления деловыми процессами обеспечивают взаимодействие различного рода программного обеспечения, тем самым создавая единую систему. В результате находится решение проблемы координации работы сотрудников и подразделений, предоставления им достоверной информации и контролирования исполнительской дисциплины, а дирекция пред-

приятия получает оперативный доступ к актуальным данным о процессе производства и имеет возможность быстрого принятия управленческих решений. И, что самое главное, полученный автоматизированный комплекс представляет собой гибкую открытую структуру, которую можно перестраивать на лету и дополнять новыми модулями или внешним программным обеспечением [2, стр. 6].



Рис. 1. Требования к корпоративным информационным системам

Рассмотрим адаптацию КИС на примере предприятия ИКЕА. Выбор данной организации осуществлен путем сравнения характеристик и потребностей компании ИКЕА с основными характеристиками классической корпорации: масштабы и распределенная структура, большое число бизнес-направлений деятельности предприятия, нуждающихся в автоматизации, организационно-управленческая структура корпорации, разнообразие вычислительных средств, сетевого оборудования, приложений специального назначения.

Адаптацией информационных систем называется процесс настройки системы в соответствии с постоянно изменяющимися условиями эксплуатации, требованиями пользователей бизнес-процессов при проектировании систем, созданных с нуля, и при сопровождении существующих.

При проектировании корпоративной информационной системы для такого крупного предприятия, как ИКЕА, важно на этапе разработки проекта автоматизации руководствоваться стандартными средствами и методами документирования. Это позволит по истечении времени в будущем вносить доработки в существующие модули системы и создавать дополнительные. Документирование всех этапов жизнен-

ного цикла, являющееся обязательным требованием при разработке корпоративной информационной системы, позволяет вносить изменения не только разработчикам данной системы, но и новым IT-специалистам, услугами которых, возможно, необходимо будет воспользоваться по каким-либо причинам. Стоит отметить, что этап сопровождения системы все-таки лучше доверять самим разработчикам данной системы, так как это сократит сроки выполнения запланированных изменений системы.

По истечении времени любая система морально устаревает, это будет относиться и к информационной системе предприятия ИКЕА. Так как рассматриваемая организация постоянно развивается, расширяет масштабы своего производства и, соответственно, штат сотрудников, вовлекает в свою деятельность новых поставщиков, разрабатывает новые бизнес-направления работы предприятия, использует постоянно улучшающееся при производстве мебели материалы, то адаптировать систему необходимо будет постоянно.

Именно корпоративные информационные системы являются наиболее гибкими, способными адаптироваться под изменяющиеся условия, по сравнению с другими видами существующих на сегодняшний день систем. Данное утверждение подтверждается полученной экономической эффективностью от внедрения аналогичных корпоративных информационных систем на подобного рода крупных предприятиях по изготовлению мебели и в других сферах производства не только в России, но и по всему миру.

Корпоративная информационная система для предприятия ИКЕА должна адаптироваться под изменения внешних факторов, таких как изменения в законодательстве по отношению к рассматриваемой корпорации, а также под изменения бизнес-функций предприятия. С этой точки зрения в разрабатываемом проекте должно быть предусмотрено наличие инструментальных средств адаптации и сопровождения системы:

- управление организационной структурой корпорации и функциями бизнес-процессов;
- изменение информационного пространства (редактирование базы данных, модификация структуры, полей таблиц, связей, индексов);
- модификация интерфейсов ввода, просмотра и корректировки информации;
- изменение функционального наполнения рабочего места пользователя;
- генерация отчетов, сложных хозяйственных операций [5, с. 11].

Для предприятия ИКЕА, работающего уже не первый год, важно обеспечить надежное хранение всех данных по производству, продаже мебели и других товаров для дома. Для этого корпоративная информационная система должна предусматривать авторизацию информации, учитывать дату и время ввода и редактирования данных пользователем, содержать протоколы по удалению информации из базы данных.

Так как на крупном предприятии ИКЕА уже имеются различные системы автоматизации, то корпоративная информационная система благодаря своей функциональности, адаптации и гибкости позволит обеспечить интеграцию между уже существующими в организации системами с самой КИС.

Руководству организации ИКЕА и пользователям корпоративной информационной системы важно предусмотреть возможность консолидации информации:

- на уровне самого предприятия для интеграции всей информации филиалов, расположенных по всему миру, и дочерних компаний в России;
- на уровне определенных задач направлений организации;
- на уровне временных периодов, необходимых для обработки полученных данных по изменению показателей за текущий период по сравнению с предыдущими отчетными периодами.

Рассмотрим, какие виды корпоративных информационных систем являются наиболее адаптируемыми под потребности мебельной корпорации ИКЕА.

Корпоративные информационные системы условно подразделяются на две большие категории: финансово-управленческие и производственные.

Финансово-управленческие системы содержат в себе небольшие интегрированные системы, предназначенные для учета по одному или нескольким направлениям организации. Они подойдут к нуждам предприятия ИКЕА, так как являются значительно более гибкими в адаптации по сравнению с производственными. Такие системы имеют возможность изменить исходную систему посредством определения связей между таблицами баз данных или отдельными модулями.

Системы производственного управления содержат средние и крупные интегрированные системы, использующиеся в управлении и планировании производства. Системы данного класса возможно внедрить на предприятии ИКЕА после того, как будет проведен сравни-

тельный анализ корпоративных информационных систем. Такого рода система является наиболее сложной в установке для такой крупной сети, как корпорация ИКЕА, и займет от нескольких месяцев до полутора лет. Но зато корпоративная система производственного управления охватит все потребности корпорации.

Производственные системы часто ориентированы на одну или несколько отраслей производства. Эффект от внедрения системы будет заметен на основных направлениях деятельности мебельной корпорации, таких как планирование, закупки, производство, сбыт, запасы и финансы.

Обширное множество функций организации ИКЕА и постоянное увеличение их сложности требуют уделить особое внимание технической инфраструктуре и программно-технической платформе.

Так как предприятие ИКЕА, его филиалы и дочерние компании разбросаны по всем точкам земного шара, то важно в системе автоматизации исключить возможность двойного ввода данных. В корпоративных информационных системах такая проблема решается путем использования электронных интерфейсов, что значительно упрощает взаимодействие между системами.

Существуют корпоративные информационные системы заказные и тиражируемые. Заказные системы предназначены для управления деятельностью конкретного предприятия, не имеющего аналогов, и не подлежат в дальнейшем тиражированию.

Суть проблемы адаптации тиражируемых систем на предприятии ИКЕА заключается в том, что в конечном итоге такая система будет уникальной, но она все равно будет содержать стандартные свойства. Требования к адаптации и сложность их реализации существенно зависят от проблемной области, масштабов системы.

Корпоративную информационную систему для корпорации ИКЕА возможно разрабатывать с помощью использования референционной модели, содержащей характеристику облика системы, функций, организованных структур и процессов.

Референционная модель в начале работы по автоматизации предприятия может представлять собой описание существующей системы (как есть) и служит точкой отсчета, с которой начинаются работы по совершенствованию КИС [1, с. 48].

Адаптации и референционные модели включены во многие системы класса MRP II / ERP, что позволит уменьшить временные затраты на введение в эксплуатацию автоматизированной системы на предприятии ИКЕА.

Заключение

В данный момент времени на рынке имеется широкий выбор корпоративных информационных систем, позволяющих автоматизировать деятельность корпорации ИКЕА. Для получения наибольшего экономического эффекта от внедрения системы необходимо тщательно провести анализ существующих на рынке систем, особое внимание уделить возможности адаптации КИС под постоянно изменяющиеся условия как внутри организации, так и за ее пределами. Улучшение внутренней управляемости, гибкости и устойчивости к внешним воздействиям увеличит конкурентоспособность и прибыльность корпорации. В результате введения в эксплуатацию корпоративной информационной системы возрастет количество продаж, сократятся издержки выпускаемой продукции, снизятся складские запасы, улучшится сотрудничество с поставщиками.

Литература

1. Баранов Р. Д., Иноземцева С. А., Рябова А. А. Информационные ресурсы и системы. Учебное пособие для студентов высших учебных заведений, обучающихся по направлениям подготовки 230400 «Информационные системы и технологии», 230700 «Прикладная информатика». Пятигорск, РИА-КМВ, 2014. 160 с.
2. Карасева О. А. Корпоративные информационные системы: конспект лекций. Екатеринбург, 2015. 65 с.
3. Матяш С. А. Корпоративные информационные системы: учебное пособие. М.: Берлин: Директ-Медиа, 2015. 471 с.
4. Одинцов Б. Е. Информационные системы управления эффективною бизнеса: учебник и практикум для бакалавриата и магистратуры. М.: Юрайт, 2016. 206 с.
5. Яковлев В. П. Корпоративные информационные системы: конспект лекций. СПб., 2015. 117 с.

УДК 519.87:66.096.5

МОДЕЛИРОВАНИЕ РЕЖИМОВ ВИБРОКИПЕНИЯ ТОЛСТЫХ СЛОЕВ

Орлова Н. С., канд. техн. наук, norlova.umi.vnc@gmail.com
Южный математический институт ВНЦ РАН,
г. Владикавказ, Россия, www.smath.ru

Аннотация. Рассматривалось виброкипение относительно толстых слоев. Исследование режимов виброкипения толстых слоев осуществлялось с использованием усовершенствованной двухжидкостной модели. В результате исследования получено, что при виброкипении относительно толстых слоев (с начальной толщиной в диапазоне от 100 мм до 200 мм) наблюдаются такие же режимы, как и при виброкипении относительно тонких слоев, но при разных сочетаниях значений амплитуды и частоты колебаний. Кроме того, некоторые режимы протекают практически без изменения степени расширения виброкипящего слоя.

Ключевые слова: математическое моделирование, режимы виброкипения, толстый слой, двухжидкостная модель, OpenFOAM, UniHUB.

MODELLING OF MODES OF THICK LAYERS VIBROFLUIDIZATION

Orlova N. S.

***Annotation.** Vibrofluidization of relatively thick layers was considered. The study of the vibrofluidization modes of thick layers was carried out using an improved two-fluid model. As a result of the study, it was found that in vibrofluidization of relatively thick layers (with an initial thickness in the range from 100 mm to 200 mm), the same modes are observed as in vibrofluidization of relatively thin layers, but with different combinations of amplitude and vibration frequency values. In addition, some modes were passing with almost no change in the degree of expansion of the vibrofluidized layer.*

***Keywords:** mathematical modelling, vibrofluidization modes, thick layer, two-fluid model, OpenFOAM, UniHUB.*

Введение

Виброкипящий слой используется во многих технологических процессах: очистка газов, сушка гранулированных материалов, синтез порошковых материалов и т. д. Для конструирования технологического оборудования, использующего виброкипящий слой, применяются различные способы, в том числе методы математического и компьютерного моделирования. Существует много математических моделей виброкипящего слоя, основанных на

различных подходах [1–16]. Например, известны модели, основанные на дискретном подходе [1–6], и модели, основанные на континуальном подходе [5–11].

В предыдущих работах [8, 10–14] было проведено исследование двухжидкостной модели на основе континуального подхода. Были исследованы режимы виброкипения относительно тонких, средних и толстых слоев [11–14]. В настоящей работе представлены результаты моделирования динамики еще более толстых слоев в процессе виброкипения при значениях толщины засыпки в диапазоне от 100 мм до 200 мм. Проведено исследование влияния начальной толщины слоя (толщины засыпки) на поведение гранулированного материала в процессе виброкипения.

Как и в предыдущих работах [10], для выполнения расчетов использовался решатель `myTwoPhaseEulerFoam` свободно распространяемого пакета для численного моделирования задач гидро- и аэромеханики `OpenFOAM` в рамках программы «Университетский кластер». Осуществлялся удаленный доступ к консоли на управляющем узле вычислительного кластера JSCC и вычислительного кластера BL2 × 220 web-лаборатории UniHUB [17, 18]. В основе решателя лежит усовершенствованная двухжидкостная модель [19–21].

Размеры расчетной области: 0,4 м × 0,4 м × 0,4 м. Шаг по времени был равен 1×10^{-4} с, шаг по пространству во всех направлениях составлял 0,005 м. Проводилось распараллеливание расчетов на восьми ядрах (в случае использования вычислительного кластера JSCC) и двенадцати ядрах (в случае использования вычислительного кластера BL2 × 220). В расчетах использовались частицы стекла диаметром 0,3 мм. Более подробное описание модели, решателя и условий вычислений представлено в работах [8, 10–14].

Результаты моделирования. Для исследования влияния начальной толщины слоя (толщины засыпки) на поведение гранулированного материала в процессе виброкипения были проведены трехмерные вычислительные эксперименты при значениях толщины засыпки $H = 100$ мм, $H = 125$ мм, $H = 150$ мм, $H = 175$ мм и $H = 200$ мм. На рис. 1 приведены результаты моделирования для относительно толстых слоев (в момент времени $t = 1$ с): a – при $H = 100$ мм; b – при $H = 150$ мм; v – при $H = 200$ мм. Амплитуда колебаний была равна 1,5 мм, частота колебаний – 10 Гц.

Из рис. 1 видно, что при малых значениях амплитуды и частоты колебаний виброкипения (виброожижения) не происходит, как и в случае относительно тонкого слоя ($H = 50$ мм). Волны на поверхности

слоя практически не наблюдаются, и степень расширения (отношение максимальной высоты виброкипящего слоя к его начальной высоте) мала. Как видно из рис. 1, на поверхности слоя наблюдаются структуры в виде кругов. С увеличением толщины засыпки H их количество увеличивается. При этом размеры круговых структур, как правило, уменьшаются.

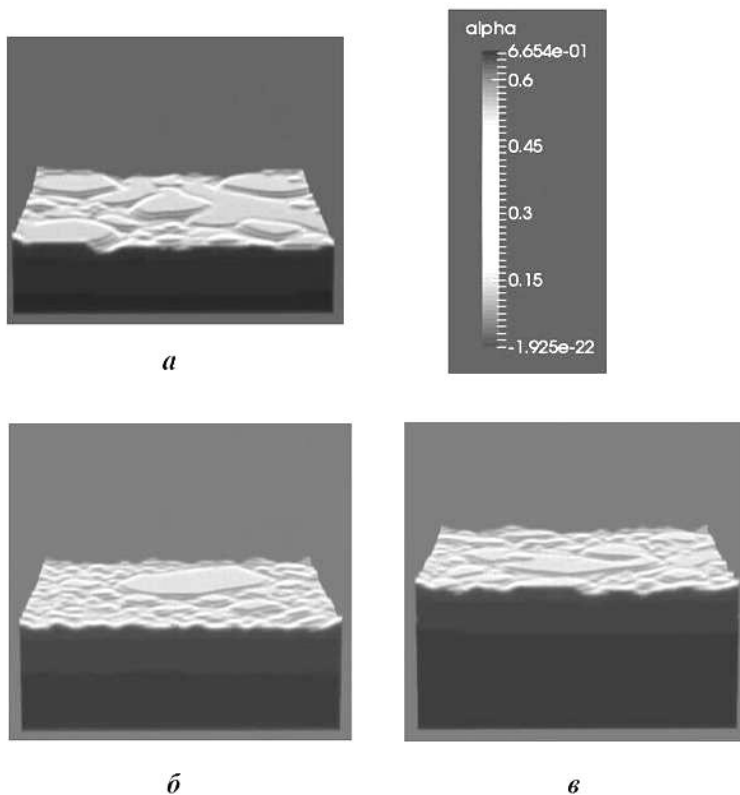


Рис. 1. Структура виброкипящего слоя:
 a – при $H = 100$ мм; b – при $H = 150$ мм; c – при $H = 200$ мм

Как и в случае, когда толщина засыпки равна 50 мм, с увеличением частоты колебаний (> 20 Гц) поверхность виброкипящего слоя становится волнообразной. Если увеличивать при этом амплитуду колебаний, то амплитуда волн гранулированного материала возрастает, что способствует расширению виброкипящего слоя.

На рис. 2 и 3 представлены результаты моделирования при значениях толщины слоя засыпки $H = 50$ мм, $H = 100$ мм, $H = 150$ мм и $H = 200$ мм. Частота колебаний равна 30 Гц, амплитуда колебаний равна 1,5 мм (*а*, *д*), 3 мм (*б*, *е*), 6 мм (*в*, *ж*) и 9 мм (*з*, *з*).

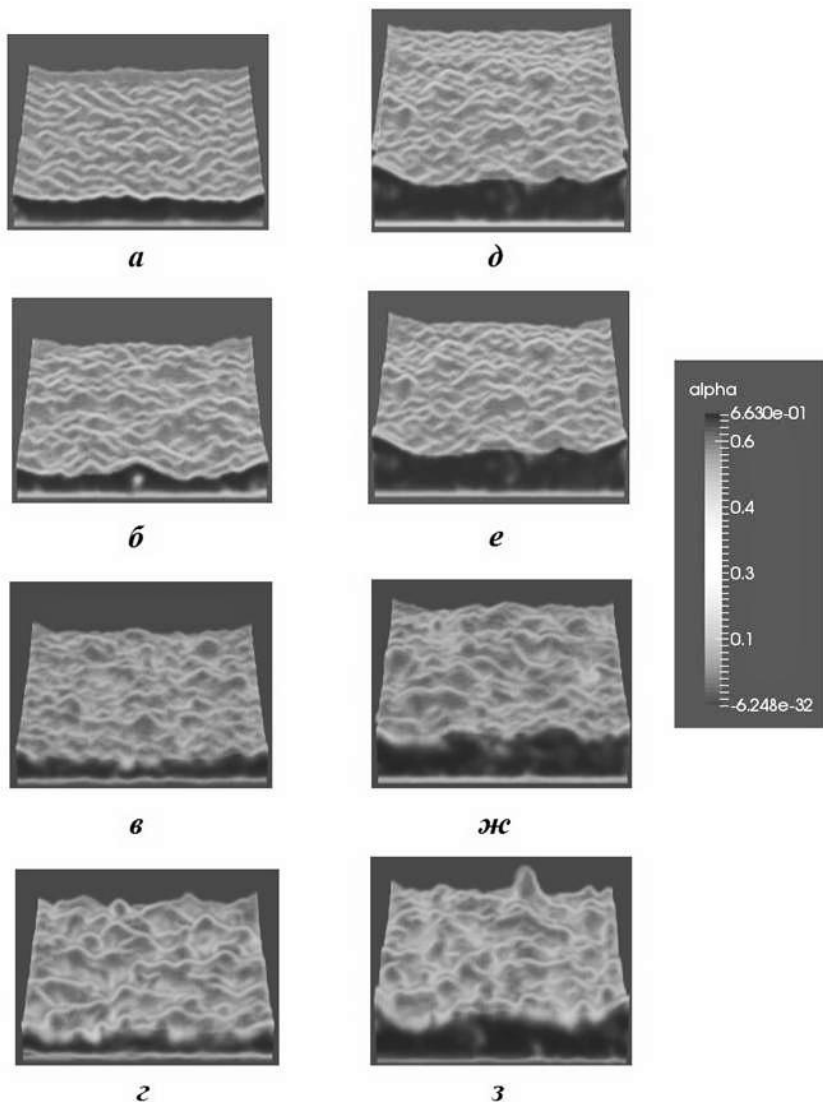


Рис. 2. Структура вибрирующего слоя при $H = 50$ мм (*а–з*) и $H = 100$ мм (*д–з*)

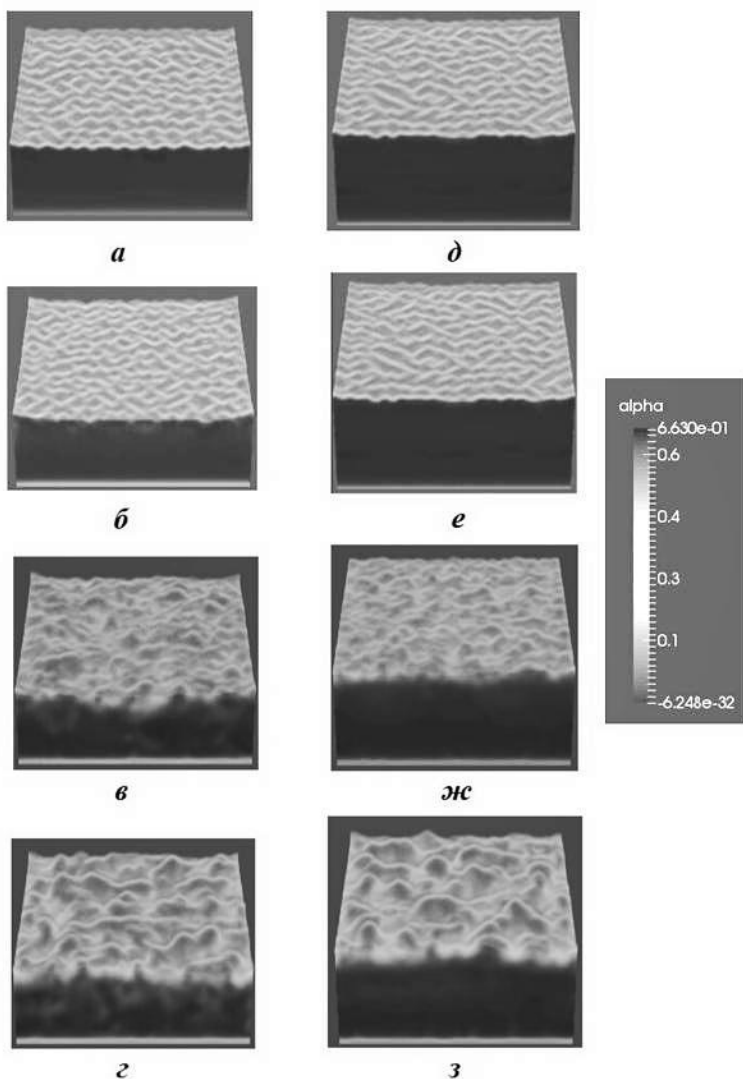


Рис. 3. Структура виброкипящего слоя при $H = 150$ мм (а–г) и $H = 200$ мм (д–з)

Из рис. 2 и 3 видно, что при значениях толщины засыпки $H = 150$ мм и $H = 200$ мм амплитуда волн на поверхности гранулированного материала увеличивается при амплитуде колебаний 6 мм, при меньших значениях амплитуды колебаний наблюдается волнообраз-

ная поверхность слоя с незначительной амплитудой волн, при этом степень расширения слоя крайне мала (высота слоя под воздействием колебаний практически не отличается от начальной толщины слоя). Такой режим наблюдается при значениях толщины слоя засыпки больше 100 мм, в частности, уже при $H = 125$ мм поведение слоя значительно отличается от случая, когда $H = 100$ мм.

При значениях $H = 50$ мм и $H = 100$ мм волны и всплески наблюдаются уже при амплитуде 3 мм. При значении амплитуды колебаний 9 мм во всех случаях наблюдаются всплески, больше похожие на фонтанирующие каналы. При этом высота всплесков больше для более тонких слоев засыпки.

Следует отметить, что чем меньше начальная высота слоя и больше амплитуда колебаний, тем больше высота зазора, который образуется между нижней границей слоя и колеблющейся полкой. В случае относительно толстых слоев ($H > 100$ мм) при малых значениях амплитуды колебаний (≤ 3 мм) зазор может не возникать совсем.

При амплитуде колебаний 9 мм, частоте 60 Гц и значении толщины засыпки $H = 100$ мм структура виброкипящего слоя более неоднородная за счет явно выраженных всплесков над поверхностью слоя, кроме того образуется зазор между нижней границей слоя и колеблющейся полкой, который увеличивается с ростом амплитуды колебаний. В случае более толстого слоя ($H = 150$ мм) структура внутренней части слоя (размером в высоту около 100 мм) остается практически однородной и плотно упакованной. Зазор между нижней границей слоя и колеблющейся полкой гораздо меньше, чем в случае относительно тонкого слоя.

На рис. 4 представлены результаты моделирования, полученные при частоте колебаний 80 Гц и амплитуде 6 мм ($a, в$) и 9 мм ($б, з$). Результаты представлены для двух случаев: когда толщина слоя равна $H = 100$ мм ($a, б$) и $H = 150$ мм ($в, з$). Видно, что в случае, когда толщина слоя $H = 100$ мм, образуется зазор между полкой и нижней границей слоя. В случае, когда $H = 150$ мм, зазор практически не наблюдается. В обоих случаях средняя часть слоя достаточно сильно уплотнена, при больших значениях амплитуды поверхность слоя теряет устойчивость, наблюдаются более крупные волны (с большей поверхностью) на поверхности материала. Следует отметить, что амплитуда волн больше во втором случае, когда толщина слоя равна 150 мм.

На рис. 5 представлены кривые изменения объемной доли частиц по высоте слоя в момент времени $t = 1,56$ с при частоте колебаний 60 Гц и амплитуде 6 мм (при таких значениях амплитуды и частоты наблюдается

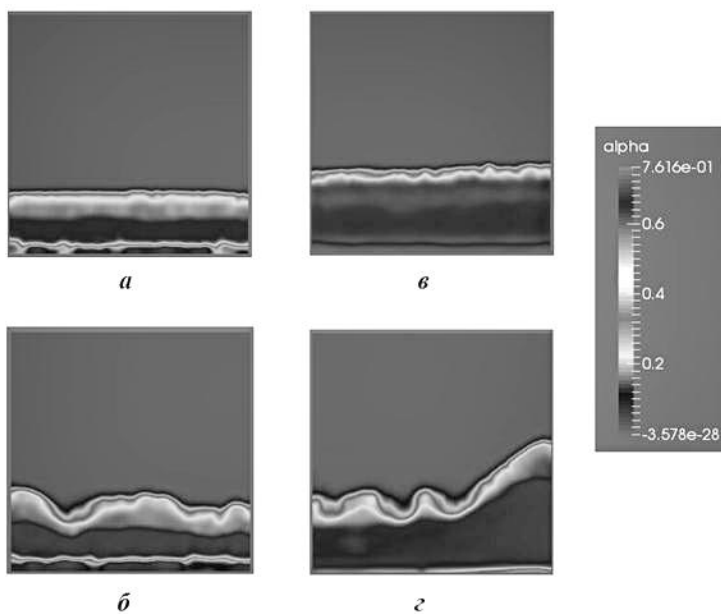


Рис. 4. Структура виброкипящего слоя при $H = 100$ мм (*а, б*) и $H = 150$ мм (*в, г*). Частота колебаний 80 Гц

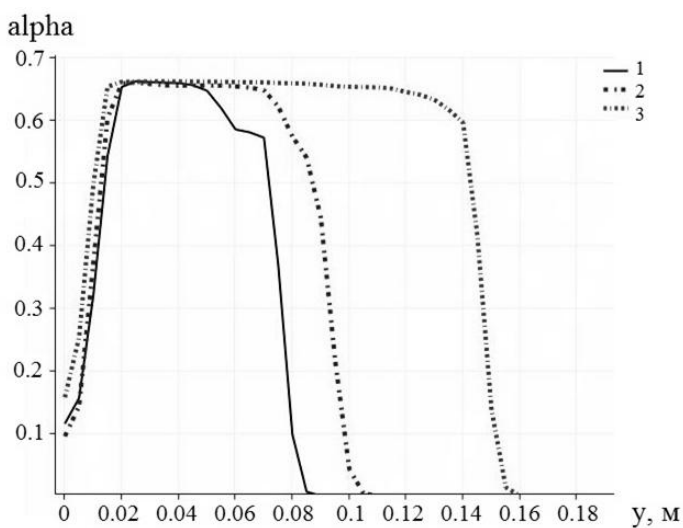


Рис. 5. Распределение объемной доли частиц по высоте слоя. Частота колебаний 60 Гц, амплитуда 6 мм

полный отрыв слоя от колеблющейся полки). Кривые построены в продольном сечении вблизи передней стенки на расстоянии $x = 0,18$ м. Кривая 1 соответствует случаю, когда толщина слоя засыпки равна $H = 50$ мм, кривая 2 – когда $H = 100$ мм, кривая 3 – когда $H = 150$ мм.

С увеличением толщины слоя засыпки уменьшается высота зазора между нижней границей слоя и полкой. При значениях толщины засыпки в диапазоне от 50 мм до 100 мм зазор достигает примерно 20 мм, а при значениях больше 100 мм (100–200 мм) зазор равен примерно 15 мм.

Для сравнения на рис. 6 представлены аналогичные результаты при частоте колебаний 60 Гц и амплитуде 3 мм. Обозначения кривых такие же, как и на рис. 5. При амплитуде колебаний 3 мм зазор во всех случаях равен примерно 15 мм (меньше, чем при амплитуде 6 мм, в случаях, когда толщина слоя засыпки лежит в пределах от 50 мм до 100 мм включительно). При амплитуде колебаний 9 мм зазор может достигать 35 мм (при $H = 75$ мм) и 40 мм (при $H = 100$ мм), в остальных случаях зазор равен примерно 20 мм. При амплитуде колебаний 1,5 мм зазор практически не возникает, т. к. амплитуда колебаний слишком мала.

Из рис. 5 и рис. 6 также видно, что степень расширения слоя уменьшается с увеличением толщины слоя засыпки.

На рис. 7 представлена зависимость степени расширения виброкипящего слоя при частоте колебаний 60 Гц и разных значениях амплитуды колебаний. Кривая 1 соответствует значению амплитуды колебаний 1,5 мм, кривая 2 – значению 3 мм, кривая 3 – значению 6 мм и кривая 4 – значению 9 мм.

Следует отметить, что в предыдущих работах [11–14] по исследованию режимов виброкипения было установлено, что степень расширения виброкипящего слоя больше зависит от амплитуды колебаний. Чем больше амплитуда, тем больше степень расширения. В свою очередь, частота колебаний влияет на количество всплесков. Чем больше частота колебаний, тем больше всплесков над поверхностью слоя. В случае высоких частот (> 50 Гц) наблюдается другой режим, при котором образуются более крупные волны на поверхности материала, слой значительно уплотняется, и может наблюдаться скошенная поверхность, а также наблюдаются отдельные области слоя со скошенной поверхностью.

Из рис. 7 видно, что чем больше амплитуда колебаний, тем больше степень расширения слоя. При больших значениях толщины слоя засыпки (> 150 мм) степень расширения с увеличением амплитуды

колебаний меняется незначительно. Также видно, что с увеличением толщины слоя засыпки значение степени расширения стремится к единице, т. е. расширение слоя практически не происходит даже при больших значениях амплитуды колебаний.

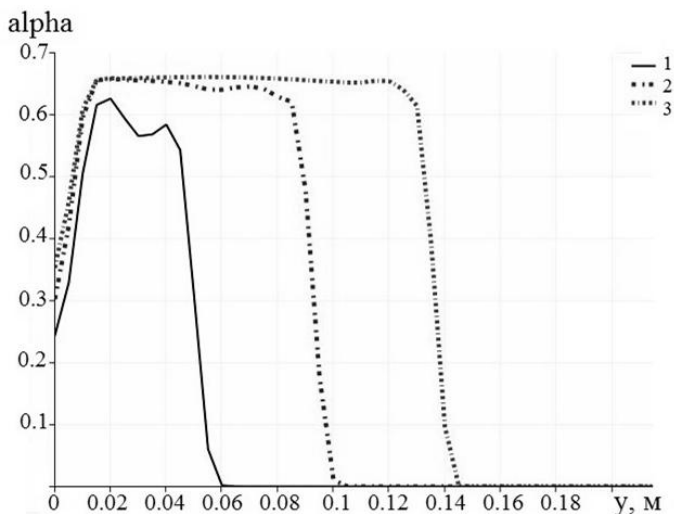


Рис. 6. Распределение объемной доли частиц по высоте слоя.
Частота колебаний 60 Гц, амплитуда 3 мм

Важно отметить, что с увеличением частоты колебаний в случае относительно толстых слоев (как и в случае относительно тонких слоев) степень расширения не растет. С увеличением частоты колебаний при больших значениях амплитуды колебаний меняется структура виброкипящего слоя. При малом значении частоты колебаний (10–20 Гц) на поверхности слоя наблюдаются волны незначительной амплитуды, в продольном сечении эти волны едва заметны, нижняя часть слоя практически не отрывается от полки. При значениях частоты колебаний 30–40 Гц поверхность слоя теряет устойчивость, наблюдаются всплески, напоминающие фонтанирующие каналы. Происходит отрыв слоя от полки, высота зазора может достигать 10 мм. Такой эффект наблюдается и в случае относительно тонких слоев при частоте 20–50 Гц.

При частоте колебаний 50–80 Гц слой уплотняется и отрывается от полки, зазор между полкой и нижней границей слоя может достигать 20–30 мм. Количество всплесков увеличивается, всплески менее

выражены и имеют высоту и размеры меньше, чем всплески, образующиеся при меньшей частоте колебаний. Чем больше частота колебаний, тем плотнее упакован слой. При этом максимальная высота слоя практически не меняется с увеличением частоты колебаний.

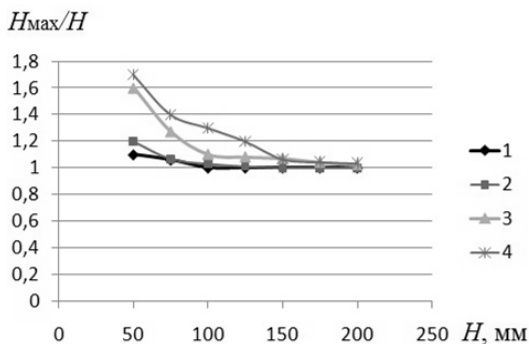


Рис. 7. Зависимость степени расширения слоя от начальной высоты.
Частота колебаний 60 Гц

Заключение. Таким образом, при виброкипении относительно толстых слоев с начальной толщиной в диапазоне от 100 мм до 200 мм наблюдаются такие же режимы, как и при виброкипении относительно тонких слоев, но при разных сочетаниях значений амплитуды и частоты колебаний. Кроме того, некоторые режимы протекают практически без изменения степени расширения виброкипящего слоя, особенно при максимальном рассмотренном значении толщины слоя засыпки (200 мм).

Литература

1. *Tatemoto Y., Mawatari Y., Yasukawa T., Noda K.* Numerical simulation of particle motion in vibrated fluidized bed // *Chemical Engineering Science*. 2004. V. 59. P. 437–447.
2. *Xiang L., Shuyan W., Huilin L., Goudong L., Juhui Ch., Yikun L.* Numerical simulation of particle motion in vibrated fluidized beds // *Powder Technology*, 2010. V. 197. P. 25–35.
3. *Lu L., Xu Yu., Li T., Benyahia S.* Assesment of different coarse graining strategies to simulate polydisperse gas-solids flow // *Chemical Engineering Science*. 2018. V. 179. P. 53–63.
4. *Каменецкий Е. С., Орлова Н. С., Волик М. В., Минасян Д. Г.* Тестирование модели виброкипящего слоя, использующей метод дис-

кретного элемента // Известия вузов. Северо-Кавказский регион. Естественные науки. 2017. № 4-1. С. 18–23.

5. Орлова Н. С. Разработка и исследование математических моделей виброкипящего слоя. Дис. ... канд. техн. наук. Таганрог, 2013.

6. Орлова Н. С. Разработка и исследование математических моделей виброкипящего слоя. Deutschland, Saarbrucken: LAP LAMBERT Academic Publishing, 2013. 173 с.

7. Орлова Н. С. Сравнение результатов экспериментального исследования виброкипящего слоя с расчетами по гидродинамической модели гранулярного газа // Инженерно-физический журнал. 2014. Т. 87. № 2. С. 429–435.

8. Орлова Н. С. Сравнение расчетов по двухжидкостной модели виброоживленного слоя с экспериментальными данными // Инженерно-физический журнал. 2012. Том 85. №6. С. 1202–1207.

9. Орлова Н. С. Тестирование двух моделей виброоживленного слоя // Известия вузов. Северо-Кавказский регион. Технические науки. 2012. № 2. С. 42–45.

10. Каменецкий Е. С., Орлова Н. С., Тагиров А. М., Волик М. В. Трехмерное моделирование виброкипящего слоя с использованием двухжидкостной модели гранулярного газа // Инженерно-физический журнал. 2016. Т. 89. № 6. С. 1480–1486.

11. Орлова Н. С., Качалкина Я. Н. Исследование режимов виброкипящего слоя с использованием пакета OpenFOAM // Труды Института системного программирования РАН. 2014. Выпуск 5. Том 26. С. 143–154.

12. Орлова Н. С., Коханова В. И. Классификация режимов виброкипящего слоя с использованием параметра перегрузки // Материалы VII Международной научно-практической конференции «Молодые ученые в решении актуальных проблем науки». 24–26 июня 2017 г. Владикавказ: Веста, 2017. С. 20–24.

13. Орлова Н. С., Коханова В. И. Классификация режимов виброкипания на основе результатов моделирования с учетом параметра толщины слоя // Современная техника и технологии: проблемы, состояние и перспективы: Материалы VII Всероссийской научно-практической конференции с международным участием 27–28 октября 2017 г. Рубцовский индустриальный институт. Рубцовск, 2017. С. 58–65.

14. Orlova N. S. Application of OpenFoam for Studying the Vibrofluidized Bed Dynamics // Fluxes and structures in fluids: Proceedings of International Conference. June 23-26, 2015. Kaliningrad. M.: MAKS Press, 2015. P. 161–165.

15. *Свердлик Г. И., Рево А. А., Каменецкий Е. С., Орлова Н. С.* Сравнение результатов экспериментов и математического моделирования виброоживленного слоя // Известия вузов. Северо-Кавказский регион. Технические науки. 2011. № 1. С. 24–27.
16. *Martin T. W., Huntley J. M., and Wildman R. D.* Hydrodynamic model for a vibrofluidized granular bed. *J. FluidMech.* 2005. Vol. 535. P. 325–345.
17. Консоль на управляющий узел кластера «JSCC Cluster Console». URL: <https://unihub.ru/resources/jscc> (дата обращения: 10.02.2016).
18. Консоль на управляющий узел кластера «BL2x220 Cluster Console». URL: <http://desktop.weblab.cloud.unihub.ru/> (дата обращения: 24.10.2018).
19. *Rusche H.* Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions // Thesis submitted for the degree of Doctor of Philosophy of the University of London and Diploma of Imperial College. 2002. 343 p.
20. *B. van Wachem.* Derivation, Implementation, and Validation of Computer Simulation Models for Gas-Solid Fluidized Beds // Dissertation at Delft University of Technology. 2000. 222 p.
21. *Каменецкий Е. С., Орлова Н. С., Волик М. В., Минасян Д. Г.* Исследование динамики кипящего гранулированного слоя с использованием пакета OpenFoam // Известия вузов. Северо-Кавказский регион. Естественные науки. 2014. № 5. С. 37–42.

УДК 517.9

ТОЧНЫЕ РЕШЕНИЯ МАТЕМАТИЧЕСКОЙ МОДЕЛИ ОДНОКАМЕРНОГО СЕРДЦА И УСЛОВИЯ РИТМИЧНОЙ РАБОТЫ СЕРДЕЧНОЙ КАМЕРЫ

Черняев А. П., д-р физ.-мат. наук, профессор, chernyaev40@yandex.ru
Московский физико-технический институт
(государственный университет),
г. Москва, Россия

Аннотация. Найдены точные решения упрощенной математической модели однокамерного сердца (имеющегося у простейших организмов), основанной на вынужденных колебаниях. На основании этих точных решений найдены условия ритмичной работы сердечной камеры.

Ключевые слова: гемодинамика, свободные и вынужденные колебания, колеблющиеся решения.

***THE EXACT SOLUTION OF THE MATHEMATICAL MODEL
SINGLE CHAMBER OF THE HEART AND CONDITIONS
THE RHYTHMIC WORK OF THE HEART CHAMBER***

Chernyaev A. P.

Abstract. Exact solutions of a simplified mathematical model of a single-chamber heart (of the simplest organisms) based on forced oscillations are found. Based on these exact solutions, the conditions for the rhythmic operation of the cardiac chamber are found.

Keywords: hemodynamics, free and forced oscillations, oscillating solutions.

Введение

Для описания работы сердца и сосудистой части кровеносной системы существует множество моделей: модель четырехкамерного сердца Холодова-Симакова [1–6], модель Калябина [7], модель Лебеденко-Новоселовой-Ракитянской-Ефимцевой [8], Волобуева [9]. Существует великое множество других моделей.

Однако до сих пор отсутствует единая математическая теория, описывающая процессы трансформации ритма в проводящей системе сердца, позволяющая прогнозировать нарушения ритма сердца при стрессовых воздействиях различных видов, позволяющая рассчитывать количественные характеристики амплитудно-временной упорядоченности сердечного ритма при различных аритмиях [10], [11]. Несмотря на большое разнообразие математических моделей сердечной деятельности, при моделировании сердечно-сосудистой системы возникает целый ряд проблем: отсутствие преемственности в разработке моделей, многочисленность несвязанных подходов, несравнимость моделей между собой [12–15].

Несмотря на большое разнообразие математических моделей сердечно-сосудистой системы, у некоторых групп моделей есть общие элементы. Так, модели Холодова-Симакова, Калябина и Лебеденко-Новоселовой-Ракитянской-Ефимцевой объединяет то, что в них имеются одинаковые линейные обыкновенные дифференциальные уравнения второго порядка с правыми частями, которые можно назвать уравнениями вынужденных колебаний.

1. Уравнения вынужденных колебаний

Эти уравнения описывают кровоток в камерах сердца, независимой переменной этих уравнений является время. Коэффициенты каждого из этих уравнений зависят от инерционности и растяжимости стенки соответствующей камеры сердца, а также от гидродинамического сопротивления этой камеры [1–8]:

$$I_j \frac{d^2 V_j}{dt^2} + r_j \frac{dV_j}{dt} + \frac{V_j}{C_j} = p_j(t) - p_j^{ext}(t), \quad j = 1, 2, 3, 4. \quad (1.1)$$

Здесь j – индекс камеры;

I_j и C_j – инерционность и растяжимость стенок камеры;

p_j – давление в камере;

$V_j = V_j(t)$ – объемы сердечных камер;

$p_j^{ext} = p_j^{ext}(t)$ – задаваемая извне функция, определяющая внешнее давление, обусловленное сокращением мышц сердца;

r_j – гидродинамическое сопротивление в межкамерных каналах.

В (1.1) искомыми функциями являются $V_j = V_j(t)$, а правые части – суть задаваемые извне функции, определяющие внешнее давление, обусловленное сокращением мышц сердца, которые описывают систолу и диастолу, а также давления перед входами и за выходами из сердца. Далее, к (1.1) добавляются законы сохранения массы для каждой из камер, а также общие граничные условия для задач о движении крови в сердце и в сосудистой сети и условия совместности.

2. Упрощенная математическая модель для однокамерного сердца

Наиболее обстоятельно из этих трех моделей рассмотрена гемодинамическая модель Калябина [8], потому что она включает в себя и упрощенную модель однокамерного сердца, которая состоит из одной активной и одной пассивной камеры. Эта модель описывает гемодинамическую систему простейших.

$$I_1 \frac{d^2 V_1(t)}{dt^2} + R_1 \frac{dV_1(t)}{dt} + \frac{V_1(t)}{C_1} = P_1(t) - F(t), \quad (2.1)$$

$$I_2 \frac{d^2 V_2(t)}{dt^2} + R_2 \frac{dV_2(t)}{dt} + \frac{V_2(t)}{C_2} = P_2(t), \quad (2.2)$$

$$R_{1,2} \frac{dV_1(t)}{dt} = P_2(t) - P_1(t), \quad (2.3)$$

$$R_{2,1} \frac{dV_2(t)}{dt} = P_1(t) - P_2(t). \quad (2.4)$$

В (2.1)–(2.4) $V_1(t), V_2(t)$ – подлежащие определению объемы первой и второй камер; $P_1(t), P_2(t)$ – давления в них; $F(t)$ – дополнительное внешнее давление, создаваемое активной стенкой первой камеры; I_1, I_2 – коэффициенты инерции камер; C_1, C_2 – коэффициенты растяжимости; R_1, R_2 – сопротивления камер, $R_{1,2}$ – коэффициент сопротивления межкамерного потока. Из (2.3) и (2.4) мы получаем:

$$\frac{dV_1(t)}{dt} = -\frac{dV_2(t)}{dt} \Rightarrow V_1(t) + V_2(t) = V_0 = \text{const}. \quad (2.5)$$

3. Сведение модели однокамерного сердца к одному уравнению вынужденных колебаний

Для удобства из (2.5) получим три равенства:

$$V_2(t) = V_0 - V_1(t), \quad \frac{dV_2(t)}{dt} = -\frac{dV_1(t)}{dt}, \quad \frac{d^2 V_2(t)}{dt^2} = -\frac{d^2 V_1(t)}{dt^2},$$

которые мы подставим в (2.2):

$$-I_2 \frac{d^2 V_1(t)}{dt^2} - R_2 \frac{dV_1(t)}{dt} + \frac{V_0 - V_1(t)}{C_2} = P_2(t).$$

Запишем последнее уравнение в виде:

$$I_2 \frac{d^2 V_1(t)}{dt^2} + R_2 \frac{dV_1(t)}{dt} + \frac{V_1(t)}{C_2} = \frac{V_0}{C_2} - P_2(t).$$

Складывая последнее с (2.1), получим:

$$(I_1 + I_2) \frac{d^2 V_1(t)}{dt^2} + (R_1 + R_2) \frac{dV_1(t)}{dt} + (C_1^{-1} + C_2^{-1}) V_1(t) = \frac{V_0}{C_2} + P_1(t) - P_2(t) - F(t).$$

Подставляя разность давлений в последнее уравнение из уравнения (2.3), имеем:

$$(I_1 + I_2) \frac{d^2 V_1(t)}{dt^2} + (R_1 + R_2 + R_{1,2}) \frac{dV_1(t)}{dt} + (C_1^{-1} + C_2^{-1}) V_1(t) = \frac{V_0}{C_2} - F(t). \quad (3.1)$$

Уравнение вынужденных колебаний (3.1) является линейным уравнением второго порядка с постоянными коэффициентами и правой частью [18]. Оно имеет три различных режима решений в зависимости от того, являются ли корни характеристического уравнения:

$$(I_1 + I_2) \lambda^2 + (R_1 + R_2 + R_{1,2}) \lambda + (C_1^{-1} + C_2^{-1}) = 0$$

различными действительными, совпадающими действительными, или комплексно сопряженными.

В качестве иллюстрации этого приведем три примера уравнений (3.1). В случае, когда $I_1 + I_2 = 1$, $R_1 + R_2 + R_{1,2} = 3$, $C_1^{-1} + C_2^{-1} = 2$, корни характеристического уравнения -2 и -1 , и в случае, когда правая часть (3.1) равна $10 + \sin(0,25t)$, решение выражается формулой:

$$V_1(t) = 5 + \frac{(2 - 0,25^2) \sin(0,25t) - 0,75 \cos(0,25t)}{0,25^4 + 5 \cdot 0,25^2 + 4} + \tilde{C}_1 e^{-2t} + \tilde{C}_2 e^{-t}, \quad (3.2)$$

где \tilde{C}_1 и \tilde{C}_2 – произвольные постоянные.

При $\tilde{C}_1 = \tilde{C}_2 = 0,125$ график (3.2), построенный в Wolfram Mathematica, изображен на рис. 1.

В случае, когда $I_1 + I_2 = 1$, $R_1 + R_2 + R_{1,2} = 4$, $C_1^{-1} + C_2^{-1} = 4$, корни характеристического уравнения совпадают и равны -2 , и в случае, когда правая часть (3.1) равна $20 + \sin(0,25t)$, решение выражается формулой:

$$V_1(t) = 5 + \frac{(2 - 0,25^2) \sin(0,25t) - \cos(0,25t)}{0,25^4 + 8 \cdot 0,25^2 + 16} + (\tilde{C}_1 + \tilde{C}_2 t) e^{-2t}, \quad (3.3)$$

где \tilde{C}_1 и \tilde{C}_2 – произвольные постоянные.

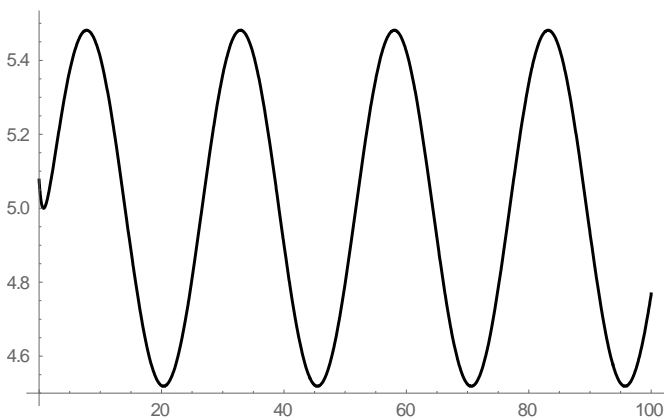


Рис. 1. График функции (3.2) в Wolfram Mathematica:
 $\text{Plot}[5 + ((2 - 0,25^2) \cdot \text{Sin}[0,25t] - 0,75 \cdot \text{Cos}[0,25t]) /$
 $/(0,25^4 + 5 \cdot (0,25)^2 + 4) + 0,125\text{Exp}[-2t] + 0,125\text{Exp}[-t],$
 $\{t, 0, 100\}, \text{PlotStyle} \rightarrow \text{Black}]$

При $\tilde{C}_1 = \tilde{C}_2 = 0,125$ график (3.3), построенный в Wolfram Mathematica, изображен на рис. 2.

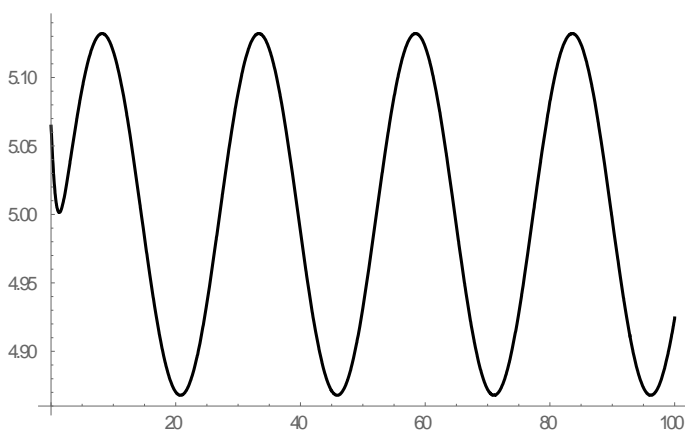


Рис. 2. График функции (3.3) в Wolfram Mathematica:
 $\text{Plot}[5 + ((2 - 0,25^2) \cdot \text{Sin}[0,25t] - \text{Cos}[0,25t]) /$
 $/(0,25^4 + 8 \cdot (0,25^2) + 16) + (0,125 + 0,125t)\text{Exp}[-2t], \{t, 0, 100\},$
 $\text{PlotStyle} \rightarrow \text{Black}]$

В случае, когда $I_1 + I_2 = 1$, $R_1 + R_2 + R_{1,2} = 2$, $C_1^{-1} + C_2^{-1} = 2$, корни характеристического уравнения комплексно сопряженные и равны $-1 \pm i$, и в случае, когда правая часть (3.1) равна $10 + \sin(0,25t)$, решение выражается формулой:

$$V_1(t) = 5 + \frac{(2 - 0,25^2)\sin(0,25t) - 0,5\cos(0,25t)}{0,25^4 + 4} + (\tilde{C}_1 \sin t + \tilde{C}_2 \cos t)e^{-t}, \quad (3.4)$$

где \tilde{C}_1 и \tilde{C}_2 произвольные постоянные.

При $\tilde{C}_1 = \tilde{C}_2 = 0,125$ график (3.4), построенный в Wolfram Mathematica, изображен на рис. 3.

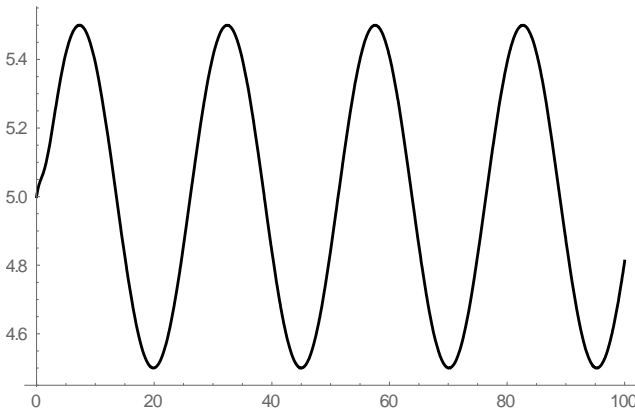


Рис. 3. График функции (3.4) в Wolfram Mathematica:
`Plot[5 + ((2 - 0,25^2) · Sin[0,25t] - 0,5 · Cos[0,25t]) / ((0,25^4 + 4) + 0,125Exp[-t] · (Sin[t] + Cos[t]), {t, 0,100}, PlotStyle → Black]`

Заключение

Итак, в результате рассмотрения указанных математических моделей работы сердечных камер можно заключить, что условия должны быть следующими: решение, описывающее работу сердечной камеры, должно быть ограниченным и таким, чтобы существовала горизонталь, относительно которой решение было бы колеблющимся.

Отметим, что при рассмотрении модели [9] полезны результаты [16–17], которые подтверждают сформулированные условия.

Литература

1. *Холодов А. С.* Некоторые динамические модели внешнего дыхания и кровообращения с учетом их связности и переноса веществ / Компьютерные модели и прогресс медицины. М.: Наука, 2001, 392. С. 127–163.

2. *Симаков С. С.* Численное исследование динамики системного кровотока при кровопотере // Информационные технологии моделирования и управления. 2006. Том 8. № 33. С. 931–938.

3. *Симаков С. С., Холодов А. С.* Численное исследование содержания кислорода в крови человека при низкочастотных воздействиях // Математическое моделирование. 2008. Том 20. № 4. С. 87–102.

4. *Василевский Ю. В., Саламатова В. Ю., Симаков С. С.* Об эластичности сосудов в одномерных моделях гемодинамики // Журнал вычислительной математики и математической физики. 2015. Том 55. № 9. С. 1599–1610.

5. *Btssonov N., Sequeira A., Simakov S., Vassilevskii Yu., Volpert V.* Methods of Blood Flow Modelling // Mathematical Modelling of Natural Phenomena. 2016. Vol. 11. № 1. Pp. 1–25.

6. *Симаков С. С.* Современные методы математического моделирования кровотока с помощью осредненных моделей // Компьютерные исследования и моделирование. 2018. Том 10. № 5. С. 581–604.

7. *Калябин Г. А.* Применение математического анализа к описанию процессов репарации инфаркта миокарда и прогнозированию кардиологических заболеваний: Учебное пособие. М.: РУДН. 2008. 144 с.

8. *Лебеденко И. С., Новоселова Е. С., Ракитянская Л. С., Ефимцева Ю. А.* Математическая модель сердца // Биотехносфера. 2009. № 3(3). С. 24–31.

9. *Волобуев А. Н.* Течение жидкости в трубках с эластичными стенками // Успехи физических наук. 1995. Том 165. № 2. С. 177–186.

10. *Мезенцева Л. В.* Теоретические основы нарушений сердечного ритма при экстремальных внешних воздействиях: Автореф. дис. ... д-ра биол. наук: 03.03.01 – Физиология, 03.01.02 – Биофизика. М., 2014. 36 с.

11. *Мезенцева Л. В.* Теоретические основы нарушений сердечного ритма при экстремальных внешних воздействиях: Дис. ... д-ра биол. наук: 03.03.01 – Физиология, 03.01.02 – Биофизика. М., 2014. 229 с.

12. Мезенцева Л. В. Анализ устойчивости различных режимов кардиодинамики методом компьютерного моделирования // Биофизика. 2014. Том 59 (1). С. 151–155.

13. Мезенцева Л. В. Параметры атриовентрикулярного проведения и устойчивость различных режимов кардиодинамики // Биофизика. 2014. Том 59 (1). С. 156–161.

14. Мезенцева Л. В., Перцов С. С. Устойчивость физиологических функций и методы ее оценки // Вестник новых медицинских технологий. 2014. Том 21 (1). С. 12–17.

15. Мезенцева Л. В. Компьютерное моделирование параметрической устойчивости кардиодинамики при мерцательной аритмии // Биофизика. 2014. Том 59 (6). С. 1180–1185.

16. Черняев А. П., Черняева С. А. Вырождение кноидальных волн в неограниченные решения для уравнения Кортевега–Де Фриза // Журнал радиоэлектроники [электронный журнал]. 2018. № 6. Режим доступа: <http://jre.cplire.ru/jre/jun18/5text.pdf> DOI 10/30898/1684–1719.2018.6.5

17. Черняев А. П., Черняева С. А. Постановка и численный анализ некоторых задач для кноидальной волны и солитона, как решений уравнения Кортевега–Де Фриза // IT-технологии: развитие и приложения: XV Ежегодная Международная научно–техническая конференция: Сборник докладов. Владикавказ: Северо–Кавказский горно–металлургический институт (государственный технологический университет), 2018. 340 с.

УДК 004

МЕТОДИКА ПОСТРОЕНИЯ СЕТИ ШИРОКОПОЛОСНОГО ДОСТУПА ПО ТЕХНОЛОГИИ PON

Юрошева Т. А., канд. техн. наук, trini-83@yandex.ru
Кцова А. Р., магистрант, ctsoveva@yandex.ru
Северо-Кавказский горно-металлургический институт
(государственный технологический университет),
г. Владикавказ, Россия

Аннотация. Рассмотрено построение структурной схемы и принцип действия сети широкополосного доступа по технологии PON,

а также выявлены конструкционные особенности волоконно-оптического кабеля.

Ключевые слова: оптическая сеть доступа, технология, модель, схема, трафик абонента.

TECHNIQUE OF BUILDING A NETWORK OF BROADBAND ACCESS TECHNOLOGY PON

Yurosheva T. A., Ktsoeva A. R.

Abstract. *The article discusses the construction of the structural scheme and the principle of operation of the network of broadcast access using the PON technology, as well as the structural features of the fiber-optic cable.*

Keywords: *optical access network, technology, model, scheme, subscriber traffic.*

Введение

Главными особенностями построения пассивных оптических сетей (PON) конфигурации P2M является использование всего одного оптического приемо-передающего модуля на стороне оператора – для организации связи со множеством абонентских устройств (до 128) и использование пассивного устройства – оптического сплиттера – для распределения сигнала. Таким образом, можно отметить следующие основные достоинства архитектуры PON:

- отсутствие промежуточных активных узлов, требующих организации бесперебойного питания и обслуживания;
- экономия оптических приемопередатчиков на центральном узле;
- экономия оптических волокон в магистральном и распределительном кабеле.

Структурная схема оптической сети доступа

Основными элементами PON P2MP сети являются:

- оптический линейный терминал OLT (Optical Line Terminal) – устройство, устанавливаемое в помещении оператора, принимающее данные со стороны мультисервисной сети связи и формирующее нисходящий поток к абонентским терминалам (прямой поток);

– оптический сетевой терминал узел ONT (Optical Network Terminal) – оборудование, устанавливаемое у абонента, и оборудованное оптическим интерфейсом для связи с OLT и абонентскими интерфейсами для предоставления услуги Triple Play:

– оптический сплиттер (разветвитель) – пассивное устройство, в одном направлении разделяющее оптическое излучение на несколько портов и объединяющее излучение с нескольких портов в обратном направлении.

В общем случае у разветвителя может быть M входных и N выходных портов. В сетях PON наиболее часто используют разветвители $1 \times N$ с одним входным портом. Разветвители $2 \times N$ могут использоваться в системе с резервированием по волокну [1, 2].

Количество ONT, подключенных к одному приемо-передающему модулю OLT, в основном определяется энергетическим потенциалом системы передачи и максимальной скоростью передачи оборудования.

Для организации двунаправленной передачи по одному ОВ используется принцип спектрального уплотнения: для передачи нисходящего потока от OLT к ONT, как правило, используются длины волн 1490/1550 нм, а восходящий поток от разных ONT к OLT передается на длине волны 1310 нм.

В приемо-передающих модулях OLT и ONT встроены WDM (волновые) мультиплексоры, разделяющие исходящие и входящие потоки. Реализация этого принципа показана на рисунке 1.

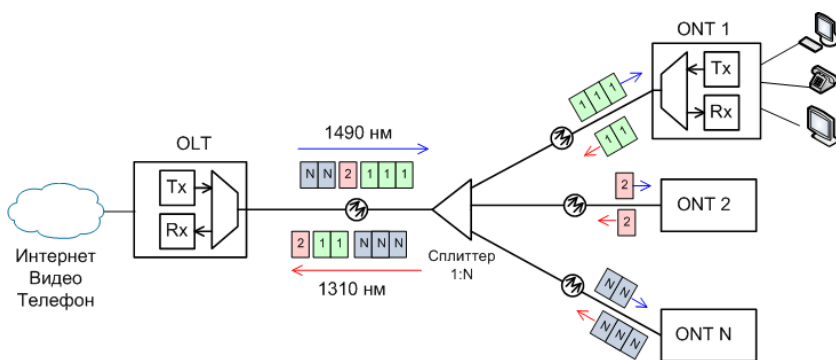


Рис. 1. Структура и принцип действия PON

Информация от OLT для всех абонентских ONT передается по принципу временного разделения каналов. На уровне оптических сигналов прямой поток является широкоэмитальным. Каждый

абонентский узел ONT, читая адресные поля, выделяет из общего потока предназначенную только ему часть информации.

Если передачу телевизионного сигнала предполагается производить на отдельной несущей (1550 нм) OLT, устанавливается оптический мультиплексор WDM для объединения передаваемых сигналов на длинах волн 1490 нм (голос, данные) и 1550 нм (видео).

В нисходящем потоке все абонентские узлы ONT ведут передачу на одной и той же длине волны – 1310 нм, используя концепцию множественного доступа с временным разделением TDMA (time division multiple access). Для того чтобы исключить возможность пересечения сигналов от разных ONT, для каждого из них устанавливается свое индивидуальное расписание по передаче данных с учетом поправки на задержку, связанную с удалением данного ONT от центрального узла OLT. Эту задачу решает протокол TDMA MAC [2, 3].

Модель обеспечения качества обслуживания МССС основана на архитектуре дифференцированных услуг (RFC2474, RFC2475, RFC3260). В основе архитектуры дифференцированных услуг (ДУ) лежат механизмы разделения трафика на классы и предоставления каждому классу соответствующего уровня обслуживания.

Каждому классу соответствует определенный код дифференцированной услуги. OLT для каждого кадра производит анализ значения идентификатора ДУ. На основании значения идентификатора ДУ OLT выбирает соответствующий механизм передачи пакета (механизм управления перегрузками). Управление пропускной способностью в случае перегрузок осуществляется с помощью механизма очередей. Пакеты помещаются в очереди, которые упорядоченно обрабатываются.

Классификация трафика абонента производится на OLT. Для осуществления классификации используется VLAN, в рамках которого выполняется передача трафика, поля IP Precedence в принятых от абонента пакетах.

Модель обеспечения качества обслуживания, применяемая на сети, использует 7 классов трафика:

- класс трафика управления сетевыми устройствами;
- класс трафика услуги IP телефонии;
- класс трафика услуги IPTV;
- класс трафика VPN уровень Premium;
- класс трафика VPN уровень Silver;
- класс трафика VPN уровень Bronze;

класс трафика услуги доступа к сети Интернет.

Классификация трафика управления производится на интерфейсе управления OLT.

Классификация трафика услуги IP-телефонии производится по признаку VLAN, в рамках которого передается голосовой трафик при предоставлении интегрированных услуг. Классификация трафика услуги IPTV производится по признаку VLAN, в рамках которого передается видеотрафик при предоставлении интегрированных услуг.

Классификация трафика услуги VPN различных уровней производится по полю IP-Precedence на порту абонента с подключенной услугой VPN.

Классификация трафика услуги доступа к сети Интернет производится по признаку VLAN, в рамках которого передается трафик абонента.

На границе сети GPON осуществляется перемаркировка классов трафика абонента, выделяемых на основе IP-Precedence (первые 3 бита в поле DSCP).

Предоставление услуг телефонии осуществляется с применением программного коммутатора оператора электросвязи. Используемый протокол – SIP. С завершением строительства IMS-ядра подключение абонентов будет осуществляться непосредственно к ядру сети IMS.

Предоставление услуги передачи данных позволяет предоставить клиенту возможность обеспечения связи компьютеров в его локальной сети (в сокращенном варианте это один компьютер) с ресурсами в публичной сети Интернет. Связь осуществляется по протоколу IP.

Предоставление IPTV (Передача видео по IP-сетям) – технология, предполагающая трансляцию видеосигнала по сетям передачи данных с использованием стека протоколов TCP/IP на телевизионный приемник абонента.

Общая схема организации связи, обеспечивающая предоставление услуг, приведена на рисунке 2.

Конструкционные особенности волоконно-оптического кабеля

Одним из важнейших компонентов ВОЛС является волоконно-оптический кабель (ВОК). Определяющими параметрами при производстве ВОК являются условия эксплуатации и пропускная способность линии связи.

По условиям эксплуатации кабели подразделяют на монтажные, станционные, зонные, магистральные. Первые два типа кабелей

предназначены для прокладки внутри зданий и сооружений. Они компактны, легки и, как правило, имеют небольшую строительную длину. Кабели последних двух типов предназначены для прокладки в колодцах кабельных коммуникаций, в грунте, на опорах вдоль ЛЭП, под водой. Эти кабели имеют защиту от внешних воздействий и строительную длину более двух километров.

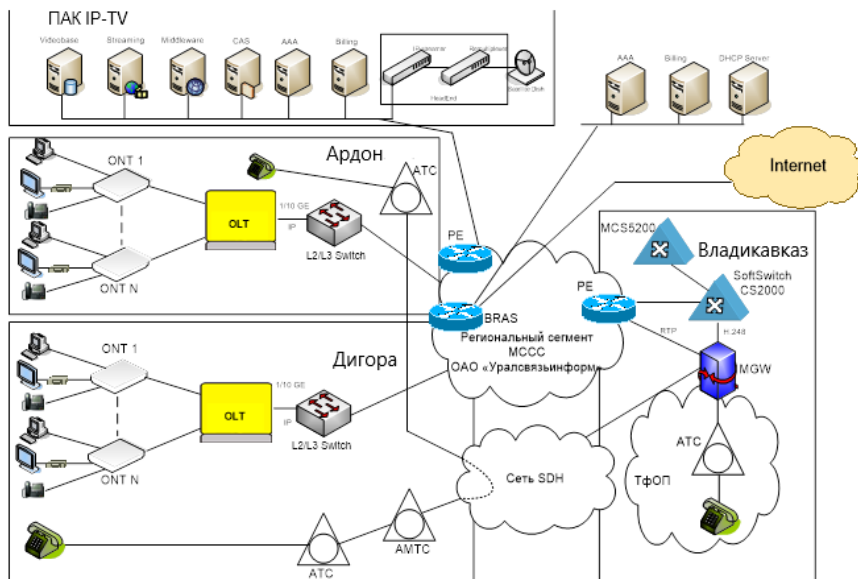


Рис. 2. Схема предоставления услуг

Приведенные выше особенности и требования определяют конструкции и типы оптических кабелей. В настоящее время условно можно выделить четыре типа конструкций ОК (условно потому, что по компоновке оптических волокон и по назначению они могут быть разделены на большее число типов и конструкций) [4, 5]:

- а) многоповивные, или кабели повивной скрутки;
- б) кабели пучковой скрутки;
- в) кабели с профильными несущими сердечниками;
- г) ленточные кабели.

GPON – лишь одна из десятков технологий, которые одновременно используются в современных сетях. Ее сильная сторона – устойчивость к проблемам электропитания в доме. Сеть

потенциально может обеспечивать высокие скорости для абонента, которые будут востребованными еще на протяжении пяти лет. В числе слабых сторон GPON, по сравнению с технологией ETTN, которую используют практически все городские провайдеры, – необходимость прокладывать избыточные линии связи при первоначальном строительстве, долгий срок окупаемости, использование относительно хрупких оптических кабелей, большой риск их повреждений в подъездах и долгий ремонт.

Заключение

GPON является наиболее перспективной технологией семейства пассивных оптических сетей. Основанная на принятых в последние годы стандартах, GPON предоставляет операторам эффективное решение высокоскоростной сети широкополосного доступа («последней мили»). GPON обеспечивает существенную экономию оптических волокон за счет древовидной архитектуры сети. Кроме того, использование технологии обеспечивает высокую надежность, благодаря пассивным элементам ветвлений.

Используя автоматизацию расчетов при проектировании, можно влиять на потенциальную аварийность линии и существенно повысить ее эксплуатационную надежность, а также осуществлять подбор оптического кабеля с оптимальными характеристиками под технические требования проекта, разрабатывать альтернативные технические решения, подбирать все комплектующие и сопутствующую арматуру. Кроме этого, разработанный модуль позволяет проводить:

- расчет электрического поля ВЛ (пересечений двух ВЛ), определение оптимальной точки подвеса ОКСН;
- расчет тяжений и стрел провеса ОКСН/ОКГТ;
- расчет дополнительной нагрузки на опору от ОКСН;
- расчет термического воздействия токов КЗ на ОКГТ.

Это создает культуру проектирования волоконно-оптических линий связи, которая учитывает потребности заказчиков в применении современных и эффективных решений.

Литература

1. *Фокин В. Г.* Оптические системы передачи и транспортные сети: Учебное пособие для вузов. М.: ЭКОТRENДЗ, 2008.

2. *Грачев А. Ф., Чернышевская Е. И., Пустова Г. Н.* Выпускная квалификационная работа: Методические указания. Новосибирск, 1999.

3. *Иванов А. Б.* Волоконная оптика: компоненты, системы передачи, измерения. М.: Компания САЙРУС СИСТЕМС, 1999.

4. СНиП 3.02.01-87. Земляные сооружения, основания и фундаменты.

5. СНиП 3.03.01-87. Несущие и ограждающие конструкции.

ИТ-ТЕХНОЛОГИИ: РАЗВИТИЕ И ПРИЛОЖЕНИЯ

Сборник докладов
Международной научно-технической конференции
(Владикавказ, декабрь 2019)

Редакторы: *Боциева Ф. А., Хадарцева Ф. С.*
Компьютерная верстка: *Кравчук Т. А.*

Подписано в печать 6.12.2019. Формат бумаги 60x84 ¹/₁₆.

Бумага офсетная. Гарнитура «Таймс».

Печать на ризографе. Усл. п.л. 10,46. Уч.-изд.л. 8,30. Тираж 60 экз. Заказ № _____.

Северо-Кавказский горно-металлургический институт (государственный технологический университет). Изд-во «Терек».

Отпечатано в отделе оперативной полиграфии СКГМИ (ГТУ).
362021. Владикавказ, ул. Николаева, 44.

Для заметок