

Министерство науки и высшего образования РФ
Смоленский государственный университет

Системы компьютерной математики и их приложения

Материалы XX Международной научной конференции

Выпуск 20
Часть 1

Смоленск
Издательство СмолГУ
2019

УДК 621.396.218
ББК 32.97
С 409

*Печатается по решению
редакционно-издательского
совета СмолГУ*

Редакционная коллегия: *К.М. Расулов*, д-р физ.-мат. наук, проф. (ответственный редактор); *Н.Г. Анищенкова*, канд. физ.-мат. наук, доц.; *С.А. Гомонов*, канд. физ.-мат. наук, доц.; *Г.С. Евдокимова*, д-р пед. наук, проф.; *Е.П. Емельченков*, канд. физ.-мат. наук, доц.; *В.И. Мунерман*, канд. техн. наук, доц.; *Г.Е. Сенькина*, д-р пед. наук, проф.; *Н.М. Тимофеева*, канд. пед. наук, доц.; *И.В. Тихонов*, д-р физ.-мат. наук, проф.

С 409 **Системы компьютерной математики и их приложения:** материалы XX Международной научной конференции. Смоленск: Изд-во СмолГУ, 2019. Вып. 20. Ч.1. 284 с.
ISBN 978-5-88018-445-3, *продолжающееся издание*

В сборнике публикуются тексты научных докладов и сообщений, представленных на XX Международной научной конференции «Системы компьютерной математики и их приложения», проходившей 17–19 мая 2019 года в г. Смоленске на базе физико-математического факультета Смоленского государственного университета. В работе конференции приняли участие научные работники и преподаватели вузов ряда стран СНГ и Прибалтики.

В материалах сборника рассматриваются вопросы применения систем компьютерной математики и их приложений в различных областях науки и техники, в математическом, техническом и гуманитарном образовании.

Сборник рекомендуется научным работникам, преподавателям вузов, аспирантам и студентам старших курсов университетов.

УДК 621.396.218
ББК 32.97

ISBN 978-5-88018-445-3,
продолжающееся издание

© Авторы, 2019
© Издательство СмолГУ, 2019

СЕКЦИЯ 1

Системы компьютерной математики

Ю.Г. Аветисов, В.Р. Кристалинский
Смоленский государственный университет

УДК 004.912

ИСПОЛЬЗОВАНИЕ НЕЧЕТКИХ МЕТОДОВ ПРИ РЕШЕНИИ ЗАДАЧ ОПТИМИЗАЦИИ РАСКРОЯ

Ключевые слова: прокат, нечеткие методы, оптимизация длины.

В статье предложена методика раскроя слитков стали для изготовления изделий оптимальной длины, учитывающая физические свойства стали при расчете длины раската и немерных длин заготовок с применением нечетких методов.

Актуальность исследования

Ресурсосберегающие технологии играют важную роль в металлургическом производстве, применение которых способствует приросту прибыли. Одной из таких технологий является оптимизация раскроя заготовки при производстве арматуры периодического профиля.

Формализация задачи

Пусть M_b – масса заготовки, M_l – масса погонного метра готового изделия, l – длина готового изделия в метрах. Далее вводятся переменные параметры: X_1 – количество выходных продуктов прокатного стана (в штуках), X_2 и X_3 – длины технологических потерь, связанных с особенностями технологического процесса. Тогда, если предположить, что максимальная длина конечного продукта проката будет равна $L_{max} = \frac{M_b}{M_l}$, целевая функция будет иметь вид:

$$F(x) = L_{max} - (l \cdot X_1 + X_2 + X_3) \rightarrow \min. \quad (1)$$

Для целевой функции вводятся следующие ограничения:

$$\begin{cases} L_{\max} - (l \cdot X_1 + X_2 + X_3) \geq 0 \\ X_1 \geq 1 \\ X_2 \geq X_{2\min} \\ X_3 \geq X_{3\min} \\ X_2, X_3 \geq 0 \end{cases}$$

Введение нечеткой величины

По технологическому процессу перед непосредственной прокаткой заготовка разогревается в подогревательной печи некоторое время. В процессе нагрева на всей поверхности заготовки образуется окалина, которая удаляется. Вес окалины зависит от температуры и длительности нагрева. Предлагается в модель (1) ввести нечеткую величину M_w , тогда целевая функция будет иметь следующий вид:

$$F(x) = L_{\max} - M_w - (l \cdot X_1 + X_2 + X_3) \rightarrow \min, \quad (2)$$

M_w является нечеткой величиной, содержащей набор из двух входных лингвистических переменных «температура» и «время» и выходной переменной «угар».

В свою очередь, переменная «угар» состоит из набора $\langle \beta, T, U, G, M \rangle$, где β – наименование лингвистической переменной, имеющей значение «угар»;

T – терм-множество {малый, средний, большой} со следующими функциями принадлежности:

$$\mu^{\text{малый}}(x, 0, 1.5); \mu^{\text{средний}}(x, 1, 2, 3); \mu^{\text{большой}}(x, 2.5, 4); x \in U.$$

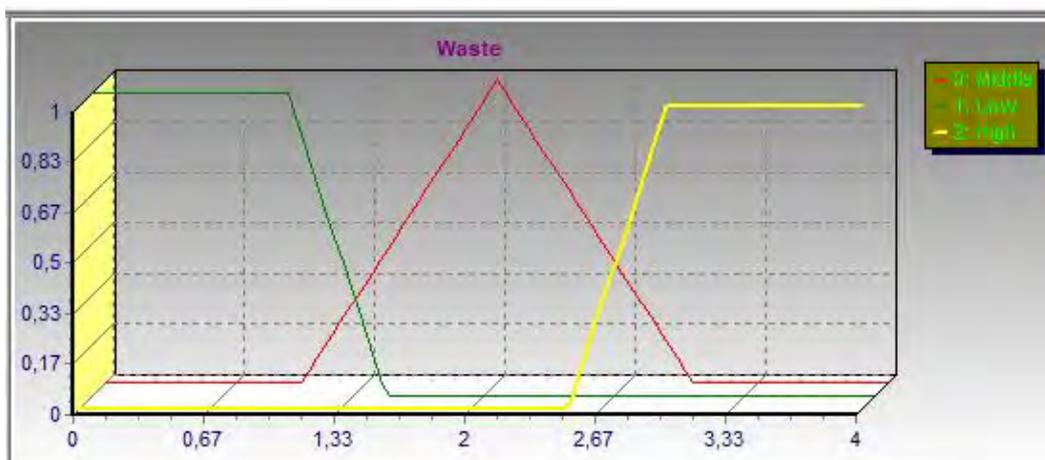


Рис. 1. Функции принадлежности для выходной переменной «угар»

U – универсальное множество $[0; 4]$;

G – синтаксические правила, порождающие новые термы с использованием квантификаторов «и», «или», «не» и т.д.;

M – процедура, ставящая каждому новому терму в соответствие нечеткое множество из U .

Входная переменная «температура» состоит из набора $\langle \beta, T, U, G, M \rangle$, где β – наименование лингвистической переменной, имеющей значение «температура»;

T – терм-множество {низкая, средняя, высокая} со следующими функциями принадлежности:

$$\mu^{\text{низкая}}(x, 0, 400); \mu^{\text{средняя}}(x, 350, 650, 950); \mu^{\text{высокая}}(x, 900, 1300); x \in U$$

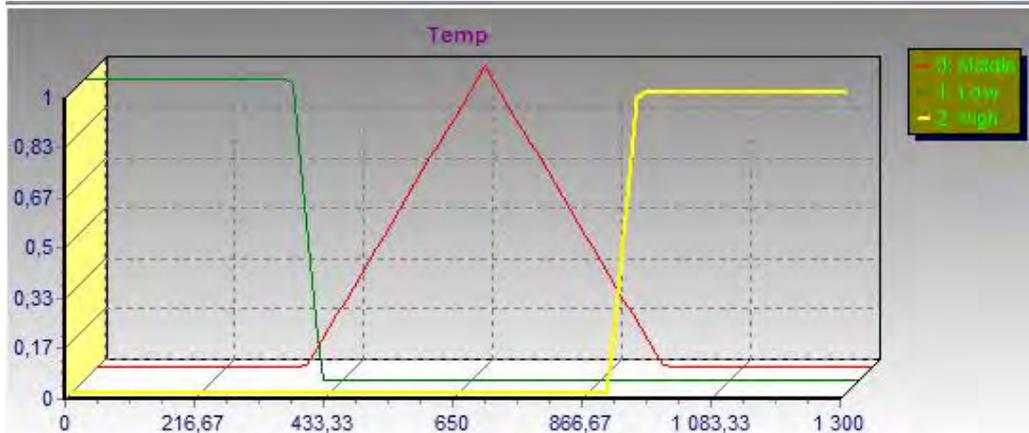


Рис. 2. Функции принадлежности для входной переменной «температура»

U – универсальное множество $[0; 1300]$;

G – синтаксические правила, порождающие новые термы с использованием квантификаторов «и», «или», «не» и т.д.;

M – процедура, ставящая каждому новому терму в соответствие нечеткое множество из U .

Входная переменная «время» состоит из набора $\langle \beta, T, U, G, M \rangle$, где β – наименование лингвистической переменной, имеющей значение «время»;

T – терм-множество {малое, среднее, большое} со следующими функциями принадлежности:

$$\mu^{\text{малое}}(x, 0, 30); \mu^{\text{среднее}}(x, 30, 60, 90); \mu^{\text{большое}}(x, 90, 120); x \in U$$

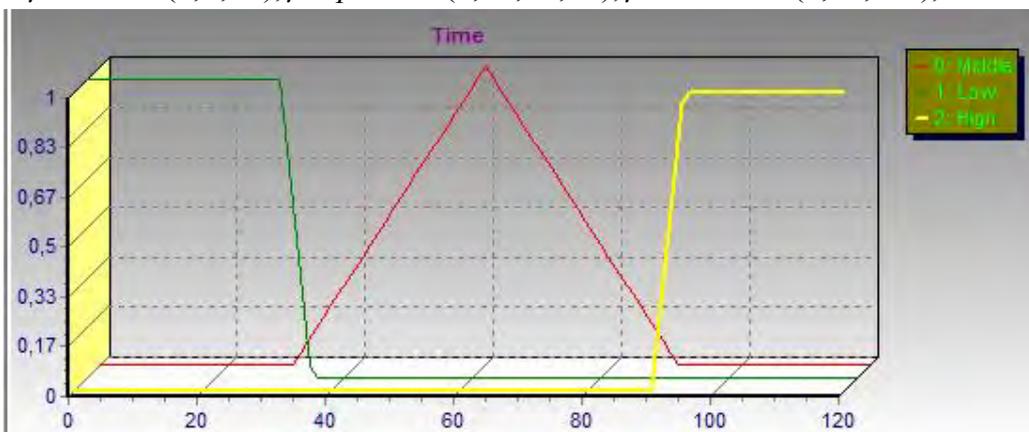


Рис. 3. Функции принадлежности для входной переменной «время»

U – универсальное множество $[0; 120]$;

G – синтаксические правила, порождающее новые термы с использованием квантификаторов «и», «или», «не» и т.д.;

M – процедура, ставящая каждому новому терму в соответствие нечеткое множество из U .

База правил нечеткой системы будет выглядеть следующим образом:

1. Если *температура низкая* и *время малое*, то *угар малый*;
2. Если *температура низкая* и *время среднее*, то *угар малый*;
3. Если *температура низкая* и *время большое*, то *угар средний*;
4. Если *температура средняя* и *время малое*, то *угар средний*;
5. Если *температура средняя* и *время среднее*, то *угар средний*;
6. Если *температура средняя* и *время большое*, то *угар большой*;
7. Если *температура высокая* и *время малое*, то *угар большой*;
8. Если *температура высокая* и *время среднее*, то *угар большой*;
9. Если *температура высокая* и *время большое*, то *угар большой*.

Практический эксперимент

Заготовка массой 1420 кг разогревалась в нагревательной печи 106 минут при температуре 1074 градуса. Тогда угар будет равняться 3,376% от длины заготовки, а именно 53,98 м.

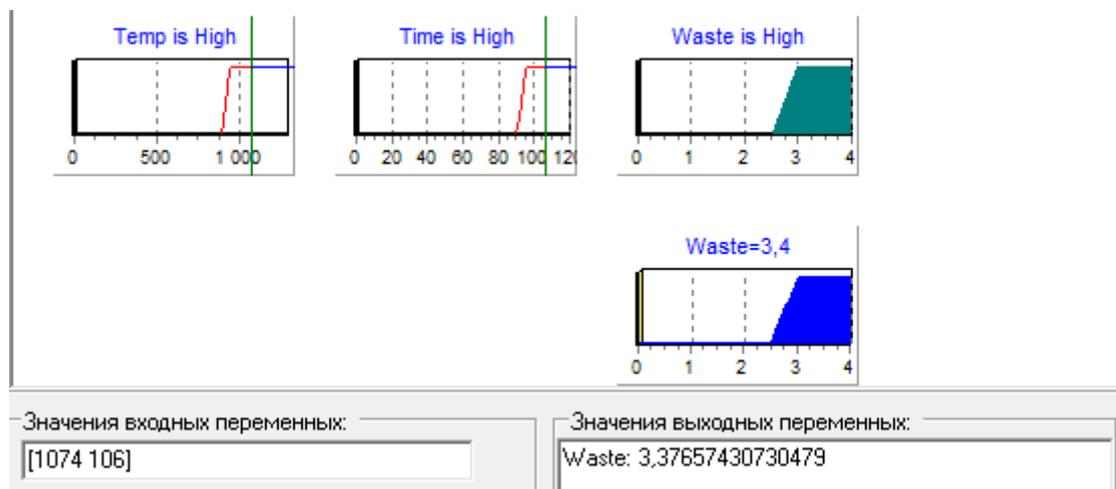


Рис. 4. Вычисление значения угара заготовки

Масса погонного метра готового изделия $M_l = 0,888$ кг, найдем максимальную длину $L_{\max} = \frac{1420}{0.888} = 1599,1$ м. Длина минимальных торцевых резов $X_{2\min} = 1,34$ м, $X_{3\min} = 0,87$ м. Длина готового изделия $l = 11,7$ м.

Подставим полученные значения в уравнение (2) и найдем оптимальное решение:

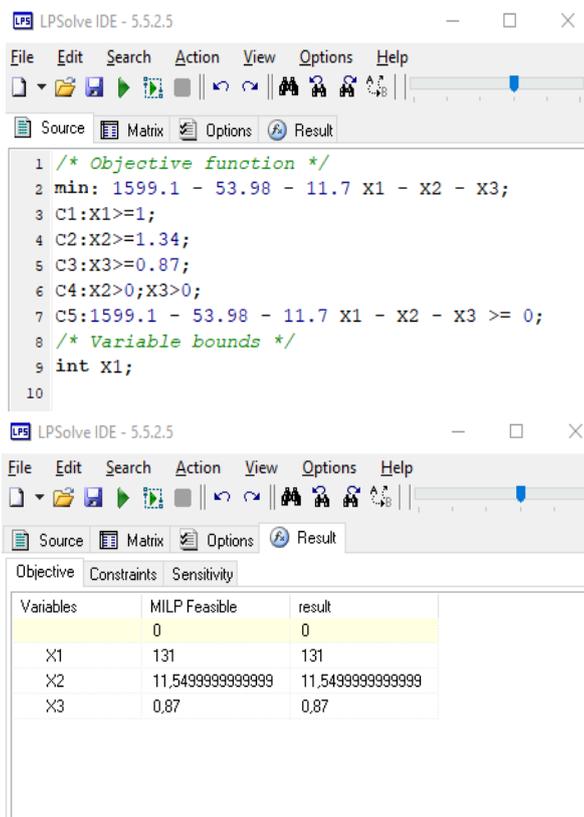


Рис. 5. Вычисление оптимального раскроя

По результатам эксперимента можно сделать вывод, что при оптимальном раскрое данной заготовки будет произведено 131 годное изделие, а технологические потери составят 11,55 м и 0,87 м соответственно.

Заключение

Применение нечетких методов в данной модели способствует не только более точной оценке потерь при нагреве заготовки, а, следовательно, и более точному вычислению значений целевой функции, но и может послужить для дальнейшего выбора оптимального времени и температуры нагрева, при котором угар будет минимальным.

Литература

1. Ашманов С. А. Линейное программирование. М: Наука, 1981.
2. Борисов А. Н., Крумберг О. А., Федоров И. П. Принятие решений на основе нечетких моделей: Примеры использования // Зинатне, 1990. Т. 184.
3. Раскин Л. Г., Серая О. В. Нечеткая математика. Харьков: Парус. 2008. С. 352.

Y.G. Avetisov, V.R. Kristalinskii
Smolensk State University

**USING FUZZY METHODS FOR SOLVING
OPTIMIZATION PROBLEMS CUTTING**

Keywords: *rental, fuzzy, optimization of length.*

Abstract. *The article proposes a method of cutting steel ingots for the manufacture of products of optimal length, taking into account the physical properties of steel when calculating the length of the roll and non-measuring lengths of blanks.*

Т.А. Агасиев, А.П. Карпенко
*Московский государственный технический
университет имени Н.Э. Баумана*

УДК 519.6

МЕТА-ОПТИМИЗАЦИЯ АЛГОРИТМОВ ГЛОБАЛЬНОЙ ПАРАМЕТРИЧЕСКОЙ ОПТИМИЗАЦИИ

Ключевые слова: *САПР, глобальная оптимизация, параметрическая оптимизация, метаоптимизация, суррогатное моделирование, ландшафтный анализ.*

Представляем постановку задачи глобальной условной оптимизации. Определяем такие сущности, как характерные признаки задачи, базовые задача и алгоритм оптимизации, метазадача и мета-алгоритм оптимизации, стратегия базовой задачи, индикатор эффективности стратегии. Приводим постановки мультииндикаторной, мультиклассовой и мультибюджетной задач мета-оптимизации. Даем анализ современных методов метаоптимизации базовых алгоритмов.

Введение. Современные прикладные задачи глобальной (параметрической) оптимизации характеризуются, прежде всего, высокой вычислительной сложностью целевой и ограничивающих функций, что не позволяет в процессе решения этих задач производить большое число испытаний (вычислений значений указанных функций). Поэтому в современном «промышленном» программном обеспечении для решения таких задач широко практикуют их предварительные анализ и обработку, включая анализ исходных данных, понижение размерности пространства поиска, ландшафтный анализ целевой функции и т.д. [1]. По той же причине широко применяют предварительный выбор (с помощью методов метаоптимизации) наиболее эффективного алгоритма оптимизации, учитывающего особенности решаемой задачи. Общей практикой является применение

прямых алгоритмов глобальной оптимизации, не использующих информацию о производных целевой функции. Для достижения высокой эффективности используют составные (гибридные) алгоритмы, комбинирующие различные «простые» алгоритмы оптимизации. С целью уменьшения числа испытаний применяют аппроксимирующие модели целевой функции (метамодел, суррогатные функции) [2].

Ограничиваемся анализом современных методов мета - оптимизации базовых алгоритмов.

1. Постановка задачи и основные определения. Рассматриваем задачу q глобальной условной оптимизации

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

где $X = (x_1, x_2, \dots, x_{|X|})$ – $|X|$ -мерный вектор варьируемых параметров (размерность задачи q); $D_X \subset \mathbb{R}^{|X|}$ – область поиска; $f(X)$ – целевая функция; X^*, f^* – искомые оптимальный вектор X и значение целевой функции.

Результатом предварительной оценки свойств задачи $q = q(C)$ является $|C|$ -мерный вектор *характерных признаков* (ХП) $C = (c_1, c_2, \dots, c_{|C|})$ этой задачи. Различаем априорные и апостериорные ХП.

Априорные ХП прямо вытекают из постановки задачи (1) и включают в себя: размерность $|X|$ пространства варьируемых параметров, признак наличия или отсутствия ограничивающих функций в определении области D_X , тип целевой и ограничивающих функций, прогнозируемое число локальных экстремумов и т.д.

Апостериорные ХП, в отличие от априорных, требуют вычислительных затрат на предварительные испытания целевой функции в области поиска D_X с целью последующей экспертной и/или автоматической оценки результатов испытаний. К числу апостериорных ХП задачи оптимизации относят, прежде всего, ХП целевой функции. Определение значений апостериорных ХП посредством экспертной оценки имеет ряд очевидных недостатков – низкая точность оценки, высокие временные затраты эксперта и т.д. Методы оценки значений апостериорных ХП целевой функции без использования экспертных оценок называют методами *ландшафтного анализа* (ЛА) [3].

Современные алгоритмы оптимизации имеют, как правило, значительное число свободных параметров $(b_1, \dots, b_{|B|}) = B$, от значений которых может существенно зависеть эффективность этих алгоритмов. Отсюда возникает задача *настройки алгоритма* – задача определения в

каком-то смысле наилучших («оптимальных») значений этих параметров. Вектор B свободных параметров алгоритма оптимизации $a = a(B)$ называют *стратегией* этого алгоритма (*Б-стратегий*). Множество D_B допустимых Б-стратегий определяет набор алгоритмов $A(B) = \{a(B), B \in D_B\}$.

Современным подходом к настройке алгоритмов оптимизации является подход на основе автоматизированного или автоматического решения этой проблемы, как *задачи метаоптимизации (М-задачи)*. Задача метаоптимизации заключается в отыскании «оптимальной» стратегии B^* алгоритма $a(B)$ при решении данной Б-задачи $q(C)$ или класса Б-задач $Q(D_C) = \{q(C) \mid C \in D_C\}$, где D_C – множество допустимых значений компонентов вектора ХП. Настраиваемый алгоритм оптимизации $a(B)$ называем *базовым алгоритмом (Б-алгоритмом)*, а настраивающий – *метаалгоритмом (М-алгоритмом)*.

Критерий эффективности Б-стратегии называем *индикатором эффективности* и обозначаем $e(q(C), a(B)) = e(C, B)$. М-задачу редко ставят и решают для одной Б-задачи, напротив, обычно эту задачу решают на некотором классе задач, который мы выше определили как $Q(D_C)$. Таким образом, *одно-индикаторную М-задачу* для класса Б-задач $Q(D_C)$ (*multi-instance problem tuning*) записываем в виде

$$\underset{B \in D_B}{\text{opt}} e(Q(D_C), B) = e(D_C, B^*) . \quad (2)$$

В зависимости от класса рассматриваемых М-алгоритмов искомой в задаче (2) является *статическая стратегия* B^* или *динамическая стратегия* $B^*(t)$, которая программно или адаптивно изменяется в процессе решения Б-задачи с ростом номера t итераций Б-алгоритма.

В качестве индикатора эффективности Б-стратегий на классе задач D_C обычно используют усредненные значения $\bar{e}(D_C, B)$ индикатора $e(q(C), B)$ на этом классе задач. Иногда рассматривают двумерный вектор индикаторов, первая компонента которого есть $\bar{e}(D_C, B)$, а вторая компонента – оценка стандартного отклонения $\sigma(D_C, B)$ индикатора $e(q(C), B)$ на том же классе задач. Другими словами, рассматривают М-задачу как мульти-задачу.

2. Мультизадачи метаоптимизации. Выделяем три следующих класса мультизадач метаоптимизации.

Мульти-индикаторные М-задачи. В этих задачах эффективность Б-стратегий оценивают с помощью вектора индикаторов $E = (e_1, e_2, \dots, e_{|E|}) = E(D_C, B)$. Аналогично задаче (2), мульти-индикаторную М-задачу рассматриваем в постановке

$$\underset{B \in D_B}{\text{opt}} \uparrow E(D_C, B) = E(D_C, \mathbf{B}^*) , \quad (3)$$

где индекс \uparrow указывает на многокритериальность постановки задачи; \mathbf{B}^* – набор «оптимальных» стратегий, принадлежащих множеству Парето задачи (3). В качестве компонентов вектора $E(D_C, B)$ аналогично задаче (2) могут использоваться их средние значения $\bar{e}_i(D_C, B)$ и стандартные отклонения $\sigma_i(D_C, B)$; $i \in [1:|E|]$.

Мульти-классовые M-задачи (multu-problem tuning). M-задачу (2) часто решают не на одном классе задач $Q(D_C)$, но на некотором наборе таких классов задач $Q(D_{C_1}), Q(D_{C_2}), \dots, Q(D_{C_{|D|}})$, то есть решают $|D|$ штук M-задач вида

$$\underset{B \in D_B}{\text{opt}} e(D_{C_i}, B) = e(D_{C_i}, B_i^*), \quad D_{C_i} \in D_C, \quad i \in [1:|D|]. \quad (4)$$

Относительно решений M-задачи (4) возникает вопрос: Насколько чувствительна каждая из полученных «оптимальных» B-стратегий $B_i^* = B^*(D_{C_i})$ к изменению (сужению или расширению) класса задач $Q(D_{C_i})$? Ответ на этот вопрос отыскивают путем решения многокритериальной мульти-классовой M-задачи

$$\underset{B \in D_B}{\text{opt}} \uparrow E(\mathbf{D}_C, B) = E(\mathbf{D}_C, \mathbf{B}^*) . \quad (5)$$

Здесь приняты следующие обозначения: $\mathbf{D}_C = \{D_{C_i} \in D_C, i \in [1:|D|]\}$ – набор, определяющий рассматриваемые классы задач; $E(\mathbf{D}_C, B)$ – векторный индикатор эффективности.

Мультибюджетные M-задачи возникают в связи с тем, что «оптимальная» B-стратегия может сильно зависеть от используемой конфигурации вычислительной системы, то есть, вообще говоря, для разных доступных вычислительных ресурсов нужно отыскивать разные «оптимальные» стратегии. Применительно к каждой данной B-задаче мощность этих ресурсов измеряют максимально допустимым числом испытаний целевой функции $f(X)$ – вычислительным *бюджетом* задачи. При небольшом бюджете, обусловленном использованием ЭВМ малой мощности, «оптимальная» B-стратегия обеспечивает преимущественную интенсификацию поиска. Напротив, мощные вычислительные системы позволяют использовать «оптимальные» стратегии, которые реализуют широко диверсифицированный поиск [4].

По общим правилам, при решении мульти-бюджетной M-задачи на классе задач $Q(D_C)$ в качестве индикатора эффективности B-стратегий можно использовать, например, среднее значение индикатора $e(C, B)$ на этом классе задач.

Как в одноиндикаторных М-задачах, искомой в рассмотренных мульти-задачах М-оптимизации, могут является статическая V^* или динамическая $V^*(t)$ стратегии.

3. Методы решения задачи мета-оптимизации. На верхнем уровне иерархии выделяют две группы методов М-оптимизации – методы *настройки* параметров и методы *управления* параметрами (рис. 1) [5]. При настройке параметров выбранную стратегию не меняют в ходе решения Б-задачи. Информацию об эффективности Б-алгоритма с данной стратегией получают после окончания решения этой задачи. В случае управления параметрами стратегию варьируют в процессе решения Б-задачи.

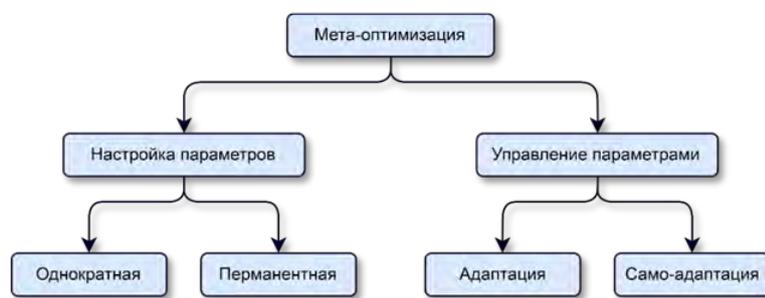


Рис. 3. Классификация методов мета -оптимизации

Методы настройки параметров являются более универсальными в сравнении с методами управления параметрами, поскольку не требуют доработки Б-алгоритма, который в этом случае рассматривается как «черный ящик». Все пользователи Б-алгоритма после окончания процесса настройки получают «оптимальную» стратегию для решения определенного класса Б-задач, что избавляет их от дальнейших вычислительных затрат на М-оптимизацию. Методы настройки позволяют дополнять накопленный опыт решения Б-задач, уточняя найденные «оптимальные» Б-стратегии.

Подходы к одноиндикаторной настройке параметров отличаются, прежде всего, соотношением интенсификационной и диверсификационной составляющих поиска «оптимальной» Б-стратегии. Выделяют следующие основные классы методов одноиндикаторной настройки параметров: *стохастические поисковые методы*, «заточенные» под быстрое отыскание «оптимальных» стратегий; *методы сэмплирования (sampling)*, позволяющие провести обширное исследование пространства стратегий; *скрининговые методы*, содержащие различные техники раннего выявления перспективных стратегий; *мета-модельные методы*, использующие суррогатную модель индикатора эффективности стратегий (*М-модель*) и

ориентированные на интенсификацию поиска «оптимальных» стратегий; *комбинированные методы* [6].

По общим правилам решения задач многокритериальной оптимизации двумя основными методами решения мультииндикаторной задачи (3) являются метод скалярной сверки частных индикаторов и метод, предполагающий построение тем или иным способом конечномерной аппроксимации фронта, а тем самым и множества Парето этой задачи. Эти аппроксимации дают возможность ЛПР не формальными либо формализованными методами выбрать «оптимальную» с его точки зрения стратегию, изучить влияние различных компонентов Б-стратегии на эффективность Б-алгоритма, исследовать устойчивость значений индикаторов эффективности к вариациям стратегий и т.д. Для построения конечномерных аппроксимаций множества и фронта Парето задачи (3) используют как хорошо известные в многокритериальной оптимизации методы (*NSGA-II*, *SPEA-2* и т.д.), так и методы, специально разработанные для решения мультииндикаторной М-задачи. Рассмотрим некоторые из них.

Метод M-FETA (Multi-Function Evolutionary Tuning Algorithm) [7] использует специализированный алгоритм мультииндикаторной настройки параметров. Основным преимуществом алгоритма является наличие специальных операторов, понижающих число требуемых запусков Б-алгоритма для исследования каждой из испытываемых Б-стратегией. С этой целью эффективность некоторых стратегий оценивают с помощью значений индикаторов эффективности «соседних» стратегий; в перспективных подобластях множества D_B генерируют большее число стратегий, что уменьшает расстояние между «соседними» перспективными стратегиями и позволяет уточнить оценки их эффективности.

Метод REVAC (Relevance Estimation and Value Calibration of Parameters) в исходном варианте относится к эвристическим итеративным стохастическим поисковым методам настройки [8], однако известна его мультииндикаторная модификация. В этой модификации фронт Парето Б-стратегий не строят, но используют оригинальную методику многоиндикаторного ранжирования стратегий на основании оценки их эффективности при решении различных Б-задач. Метод позволяет использовать «гонки» с целью уменьшения числа испытаний Б-стратегий.

Метод Bonesa [9] в своей основе является итеративным методом настройки параметров, использующим М-модель. Для уменьшения числа запусков стохастических Б-алгоритмов с каждой из испытываемых Б-стратегий используется фильтр Гаусса, который уменьшает уровень зашумленности индикатора эффективности. Сравнение эффективности стратегий выполняют с помощью «гонок».

Для построения фронта Парето метод *Bonesa* использует известный алгоритм *SPEA2*.

Метод *EMOPaT* (*Evolutionary Multi-Objective*) [10], является мульти-классовой модификацией метода *SEPaT*. В качестве М-алгоритма использован известный алгоритм *NSGA-II*. Метод включает в себя анализ Парето-оптимальных решений с помощью оригинального простого и эффективного подхода *innovation (innovation through optimization)*.

Метод *Flexible Budget* [11] относится к классу мульти-бюджетных методов М-оптимизации. Результатом настройки является набор Б-стратегий, эффективных при различных бюджетах.

Заключение. В целях сокращения вычислительных затрат, в программном обеспечении «промышленной» глобальной оптимизации возрастает поддержка предварительного анализа постановки задачи на основе использования известных и новых алгоритмов анализа данных, понижения размерности пространства поиска, ландшафтного анализа целевой функции. Имеет место тенденция преимущественного использования гибридных адаптивных алгоритмов оптимизации, объединяющих несколько стохастических алгоритмов глобальной оптимизации и алгоритмов локальной оптимизации (мультимемевые алгоритмы).

Повышается «интеллектуальность» программных комплексов глобальной оптимизации, в том числе за счет реализации в них методов метаоптимизации базовых алгоритмов оптимизации. Если в настоящее время, как правило, используются методы однократной настройки параметров этих алгоритмов, то перспективным является использование методов перманентной настройки, а также адаптивных и самоадаптивных методов управления параметрами. С другой стороны, сейчас, почти без исключений все указанные мультизадачи метаоптимизации решают только по отдельности. Актуальной является проблема совместного решения таких задач. При этом возникают сложные задачи многокритериального принятия решений, требующие одновременного анализа более одного многомерного фронта (множества) Парето. В целом современные методы многокритериального принятия решений еще недостаточно широко используются в процессе решения задач метаоптимизации. Например, перспективным, на наш взгляд является использование подходов, основанных на выявлении так называемой функции предпочтений ЛПР [12].

Для решения современных практически важных задач оптимизации все более широко используют параллельные вычислительные системы, имеющие разные архитектуры – вычислительные кластеры, системы с общей памятью, системы на основе графических процессорных

устройств, слабосвязанные вычислительные системы типа ГРИД-систем и т.д. В этой связи актуальными являются проблемы разработки методов, алгоритмов и соответствующего программного обеспечения для решения базовых задач оптимизации и задач метаоптимизации, ориентированных на указанные классы параллельных систем.

Литература

1. Shan S., Wang G. G. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions // Structural and Multidisciplinary Optimization. 2010. Vol. 41, No. 2. P. 219–241.
2. Агасиев Т.А., Карпенко А.П. Современные техники глобальной оптимизации. Обзор // Информационные технологии, 2018. № 6. С. 370–386.
3. Mersmann O. [et al.] Exploratory landscape analysis // Proceedings of the 13th annual conference on Genetic and evolutionary computation. ACM. 2011. P. 829–836.
4. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы вдохновленные природой. М.: Изд-ство МГТУ им. Н.Э. Баумана, 2014. 446 с..
5. Eiben A.E., Hinterding R., Michalewicz Z. Parameter control in evolutionary algorithms // IEEE Transactions on Evolutionary Computation. 1999. Vol. 3. Is. 2. P. 124 – 141.
6. Smit S.K. Parameter tuning and scientific testing in evolutionary algorithms. Vrije Universiteit, SIKS Dissertation Series. 2012. No.38. 195 p.
7. Smit S.K., Eiben A.E., Szilávik Z. An MOEA-based Method to Tune EA Parameters on Multiple Objective Functions, IJCCI (ICEC), 2010. P. 261–268.
8. Nannen V., Eiben A.E. A method for parameter calibration and relevance estimation in evolutionary algorithms // Proceedings of the 8th annual conference on Genetic and evolutionary computation. ACM. 2006. P. 183–190.
9. Smit S.K., Eiben A.E. Multi-problem parameter tuning using bonesa // Artificial Evolution. 2011. P. 222–233.
10. Ugolotti R. Meta-optimization of Bio-inspired Techniques for Object Recognition. Diss. Università di Parma. Dipartimento di Ingegneria dell'Informazione, 2015. 144 p.
11. Branke J., Elomari J.A. Meta-optimization for parameter tuning with a flexible computing budget // Proceedings of the 14th annual conference on Genetic and evolutionary computation. ACM. 2012. P. 1245–1252.
12. Карпенко А.П., Моор Д.А., Мухлисулина Д.Т. Многокритериальная оптимизация на основе нейро-нечеткой аппроксимации функции предпочтений лица, принимающего решения // Наука и образование. 2010. № 6.

T.A. Agasiev, A.P. Karpenko
Bauman Moscow State Technical University

META-OPTIMIZATION OF ALGORITHMS OF GLOBAL PARAMETRIC OPTIMIZATION

Keywords: *computer-aided design systems, global optimization, parametric optimization, meta-optimization, surrogate modeling, landscape analysis.*

Abstract. *We present the formulation of the global constrained optimization problem. We define such entities as the characteristic features of the task, the basic task and the optimization algorithm, the meta-task and the meta-optimization algorithm, the strategy of the basic task, the indicator of the effectiveness of the strategy. The paper presents the formulation of multi-indicator, multi-class and multi-budget meta-optimization problems. We also give an analysis of the modern methods of meta-optimization of basic algorithms.*

Е.А. Божинский, Г.С. Евдокимова
Смоленский государственный университет

УДК 339.13

СТАТИСТИЧЕСКИЙ АНАЛИЗ РЫНКА СМАРТФОНОВ ЗА 2018 ГОД

Ключевые слова: *статистический анализ; рынок смартфонов; прогноз цены; факторный анализ; нейронная сеть.*

В работе представлен статистический анализ рынка смартфонов, анонсированных в 2018 году, от 11 популярнейших производителей на территории Российской Федерации. Исследование, основанное на входных параметрах (характеристиках смартфона), было направлено на получение адекватного прогноза цены для произвольного устройства, а также выделение среди характеристик тех, которые оказывают наибольшее влияние на конечную стоимость устройства.

Статистическая база представляет собой список смартфонов, которые были анонсированы в 2018 году и уже поступили в продажу от 11 наиболее продаваемых брендов на территории России [2]. В нее вошли 178 устройств и их основные технические характеристики со стоимостью (переменные).

Взятый с ресурса market.yandex.ru с параметрами фильтрации (рис. 1) – только смартфоны с датой анонса 2018 и только наиболее популярных брендов, а именно: Apple, ASUS, Honor, HUAWEI, LG, Meizu, Nokia, OnePlus, Samsung, Sony, Xiaomi, ZTE, список смартфонов насчитывает 178 пунктов и содержит статистическую базу основных характеристик устройства: X_1 – количество SIM-карт, X_2 – тип SIM-карт, X_3 – режим работы SIM-карт, X_4 – вес, X_5 – ширина, X_6 – высота, X_7 – толщина, X_8 – технология производства матрицы, X_9 – диагональ экрана, X_{10} – высота экрана в пикселях, X_{11} – ширина экрана в пикселях, X_{12} –

плотность пикселей, X_{13} – соотношение сторон экрана, X_{14} – количество основных модулей камеры, X_{15} – суммарное разрешение основной камеры, X_{16} – диафрагма, X_{17} – разрешение фронтальной камеры, X_{18} – наличие разъема 3,5 мм, X_{19} – наличие LTE, X_{20} – стандарт Wi-Fi, X_{21} – поддержка технологии Wi-Fi Direct, X_{22} – версия Bluetooth, X_{23} – наличие разъема USB Type-C, X_{24} – наличие модуля NFC, X_{25} – частота CPU, X_{26} – количество ядер CPU, X_{27} – объем встроенной памяти, X_{28} – объем оперативной памяти, X_{29} – максимальная поддерживаемая емкость карт памяти, X_{30} – емкость аккумулятора, X_{31} – наличие датчика освещенности, X_{32} – наличие датчика приближения, X_{33} – наличие сканера отпечатка пальца, X_{34} – средняя цена в г. Смоленске на момент 10 февраля 2019 года.

Рис. 1. Параметры фильтрации для поиска данных

Для обработки статистических данных был выбран факторный анализ. Основными требованиями к его применению являются: минимальная потеря информации, которая содержится в исходных данных, и возможность интерпретации полученных факторов.

В начале проведения факторного анализа необходимо было определиться с количеством факторов. Это было сделано с помощью двух критериев. По графику каменистой осыпи определено, что в точке 4 осыпание существенно замедляется, исходя из этого можно ограничиться тремя, четырьмя или пятью факторами (рис. 2).

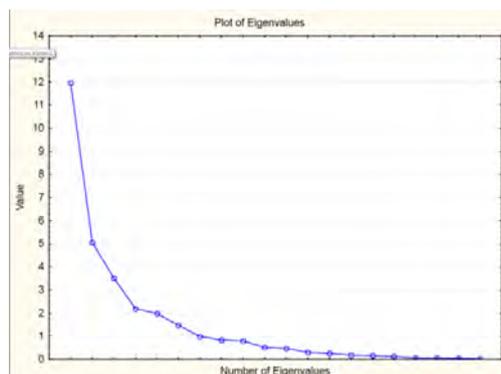


Рис. 2. График каменной осыпи

Критерий Кайзера показывает (рис. 3), что в исследовании стоит ограничиться четырьмя факторами (четыре собственных значения больше двойки). Объединяя результаты двух критериев, удалось выделить четыре фактора, которые объясняют 72,6% совокупной дисперсии.

Eigenvalues (MobileStatya)				
Extraction: Principal components				
Value	Eigenvalue	% Total variance	Cumulative Eigenvalue	Cumulative %
1	11,54301	38,47669	11,54301	38,47669
2	4,60793	15,35978	16,15094	53,83647
3	3,46927	11,56423	19,62021	65,40070
4	2,17095	7,23648	21,79116	72,63719
5	1,97061	6,56871	23,76177	79,20590
6	1,45508	4,85026	25,21685	84,05615

Рис. 3. Критерий Кайзера

В качестве метода факторного анализа был выбран анализ главных компонент, в основе которого лежит математический метод нахождения собственных значений и собственных векторов корреляционной матрицы. Для лучшей интерпретации данных произведен поворот осей с помощью варимакс-вращения [3, с. 171]. Результат факторных нагрузок после вращения представлен на рисунке 4.

Factor Loadings (Varimax raw) (MobileStatya)				
Extraction: Principal components				
(Marked loadings are > .700000)				
Variable	Factor 1	Factor 2	Factor 3	Factor 4
SIM type	0,041401	0,977458	0,042865	-0,037095
Wrk mode	0,041401	0,977458	0,042865	-0,037095
VES	0,122442	0,389794	0,854857	0,072818
SHIR	-0,011758	0,103874	0,905592	0,121551
VIS	0,121154	-0,105484	0,922365	0,110080
TOL	-0,096581	-0,041573	0,170956	-0,278370
Matrix type	0,384840	0,180336	-0,217319	0,682649
Diagonal	0,421604	0,117999	0,764960	0,268666
SHIR PPX	0,913162	0,001301	0,200248	0,234501
VIS PPX	0,881862	-0,053915	0,225778	0,245177
PPI	0,918701	-0,045749	0,023128	0,188795
SOOT STORON	0,617231	0,375420	0,004845	0,046262
CGG	0,420151	0,379847	-0,157706	0,688987
Numb Camera	0,331129	-0,249713	0,423264	0,595213
Size CAM	0,225861	-0,131242	0,226036	0,841612
Diaphragma	0,795275	0,084380	0,056975	0,439469
Frontalka	0,119286	-0,325411	0,336454	0,400465
JACK	-0,025852	-0,754170	-0,162762	-0,071639
WiFi acn	0,682339	0,104618	0,515393	0,322627
WiDirect	0,325636	-0,448182	0,315791	-0,449187
Bluetooth	0,346879	0,206542	0,457227	0,507274
UCB C	0,412696	0,261258	0,326964	0,596152
NFC	0,684104	0,302666	-0,120523	0,278753
Freq	0,709232	0,185009	0,117022	0,392057
CoreNumb	0,558804	-0,316723	0,507319	0,242292
ROM	0,340683	-0,093492	0,244419	0,802000
RAM	0,363904	-0,128706	0,285945	0,785452
MemoryCard	0,575766	-0,294046	-0,111669	-0,052543
Battery	-0,079943	-0,573136	0,767327	-0,040348
OTPE4ATOK	0,283052	0,071806	0,321621	0,261646
Expl.Var	7,001539	4,178756	5,369731	5,241130
Prp.Totl	0,233385	0,139292	0,178991	0,174704

Рис. 4. Факторные нагрузки после вращения

Красным цветом отмечены факторные нагрузки, которые по абсолютной величине больше 0,7. Очевидно, что в первый фактор вошли такие показатели, как разрешение дисплея, плотность пикселей, соотношение сторон экрана, размер диафрагмы камеры, частота ЦП, стандарты Wi-Fi-модуля, наличие NFC, количество ядер процессора, поддержка внешних карт памяти. Во второй – размер SIM-карты, режим работы радио модуля, наличие стандартного гнезда для наушников 3.5, от которого в последнее время многие стали отказываться. В третий – ширина и высота устройства, вес устройства, диагональ экрана, размер батареи, датчик отпечатка. В четвертый – разрешение камер, оперативная память, объем встроенной памяти, толщина корпуса, тип матрицы (AMOLED/IPS), наличие ударопрочного стекла GorillaGlass, количество модулей камеры, разрешение фронтальной камеры, версия Bluetooth и наличие Type-C разъема.

Наличие исходных признаков позволило определить фактор 1 как «технические особенности», фактор 2 – «второстепенные черты устройства», фактор 3 – «габариты», фактор 4 – «технологичность». Таким образом, удалось обнаружить скрытые факторы, влияющие на итоговую стоимость устройств, среди самых продаваемых брендов на территории Российской Федерации и сократить число первоначально изученных признаков.

Следующим шагом было изучение полученных результатов и выявление именно тех параметров, который больше всего влияют на итоговую цену смартфона. Для этого была построена матрица корреляций между факторными оценками всех четырех факторов и итоговой стоимостью. Результаты представлены на рисунке 5.

Correlations (Spreadsheet27)							
Marked correlations are significant at p < ,05000							
N=22 (Casewise deletion of missing data)							
Variable	Means	Std.Dev.	Var1	Var2	Var3	Var4	NewVar
Var1	0,00	1,00	1,000000	0,000000	0,000000	-0,000000	0,524090
Var2	0,00	1,00	0,000000	1,000000	-0,000000	-0,000000	0,726534
Var3	-0,00	1,00	0,000000	-0,000000	1,000000	0,000000	0,044005
Var4	-0,00	1,00	-0,000000	-0,000000	0,000000	1,000000	0,298309
NewVar	28549,14	20153,42	0,524090	0,726534	0,044005	0,298309	1,000000

Рис. 5. Матрица корреляций

Наибольшее влияние на итоговую стоимость устройства оказал второй фактор, который был охарактеризован как «второстепенные черты устройства». С одной стороны, на стоимость устройства должна влиять производительность и начинка аппарата, но с другой стороны, китайский рынок в виде Xiaomi, Meizu и HUAWEI предлагает те же самые характеристики у аппаратов, что и у лидирующих брендов, таких как Apple или Samsung, но только цена на данные аппараты, в 2-3 раза меньше. Да и по качеству китайские производители не уступают лидерам данного рынка. Соответственно можно сделать вывод, что на

итоговую цену устройства влияет не мощность аппаратной составляющей, а какие-то всего-навсего, мелкие детали, которые позволяют достичь наибольшего комфорта при использовании телефона. А эти параметры и попадают под второй фактор.

Осталось только выполнить прогноз стоимости аппарата, основываясь на произвольных его характеристиках. Для этого была построена искусственная нейронная сеть на исходных данных.

«Искусственная нейронная сеть представляет собой параллельно распределенную систему процессорных элементов (нейронов), способных выполнять простейшую обработку данных, которая может настраивать свои параметры в ходе обучения на эмпирических данных. Накопленные знания нейронной сети сосредоточены в весах межэлементных связей. Визуально искусственные нейронные сети представляются в виде графов, в которых нейроны узлы, а связи линии» [4].

Построение нейросетевых моделей производилось в программе STATISTICA в режиме регрессии. В результате получено несколько вариантов сетей (рис. 6), среди которых стоит выделить 43-9-1 и 43-22-1. В конечном итоге была выбрана модель 43-9-1 в качестве основной, так как она дала наилучшую точность на тренировочной выборке.

Index	Net. ID	Net. name	Training perf.	Test perf.	Validation p...	AI
12	12	MLP 43-16-1	0,929894	0,963633	0,999516	BI
13	13	MLP 43-12-1	0,954434	0,962672	0,999550	BI
14	14	MLP 43-20-1	0,966127	0,870222	0,998627	BI
15	15	MLP 43-20-1	0,907241	0,945287	0,999933	BI
16	16	MLP 43-7-1	0,945824	0,980488	0,998620	BI
17	17	MLP 43-10-1	0,939809	0,994677	0,999978	BI
18	18	MLP 43-22-1	0,940025	0,989082	0,999921	BI
19	19	MLP 43-9-1	0,999789	0,997678	0,998764	BI
20	20	MLP 43-9-1	0,955735	0,969842	0,999336	BI

Рис. 6. Результаты FA-ANN

С помощью этой нейросетевой модели проведен прогноз стоимости устройства, обладающего произвольными, и максимально возможными характеристиками на момент 2018 года, вобравшие в себя все максимально возможно аппаратные характеристики устройства, результаты которого представлены на рисунке 7.

Custom predictions spreadsheet (MobileStatya)								
Cases	19.PRICE_(t)	VES	SHIR	VIS	TOL	Diagonal	VIS PPX	PPI
1	70528,04	140,0000	70,00000	150,0000	7,000000	6,000000	2960,000	568,0000

Рис. 7. Результаты предсказания на основе NN-модели

В результате получили: цена – 70528,04 руб. Данная информация говорит нам о том, что каким-бы ни был технологичным и современным смартфон, все равно присутствует фактор, из-за которого он будет стоить дороже, имея более слабые характеристики, а именно – переплата за бренд и удобство использования. Также в результате исследования получилась небольшая нейросетевая модель, основываясь на которой можно предсказывать стоимость мобильных телефонов по характеристикам еще до официальной презентации устройства. Другими словами, такой подход позволяет экономить время и силы в случае изучения рынка, а также позволяет заглянуть на несколько шагов вперед. Выбор технологий в стандартном наборе пакета STATISTICA достаточно широк, что позволяет получить удобные и практичные решения. Например, благодаря инструментам экспорта нейросетевую модель можно будет встроить в уже имеющиеся в компаниях аналитические информационные системы в качестве отдельного дополнительного модуля.

Литература

1. Боровиков В.П. Популярное введение в современный анализ данных в системе Statistica. М.: Горячая линия – телеком, 2013. 288 с.
2. Издательский дом Коммерсантъ URL: <https://www.kommersant.ru/doc/3096151> (дата обращения: 15.04.2018).
3. Орлова И.В. Многомерный статистический анализ в экономических задачах: компьютерное моделирование в SPSS. М.: Вузовский учебник, 2013. 330 с.
4. Ресурс для IT-специалистов «Хабр» URL: <https://habr.com/post/312450/> (дата обращения: 13.04.2018).

E.A Bozhinskiy, G.S.Evdokimova
Smolensk State University

STATISTICAL ANALYSIS OF THE SMARTPHONE MARKET IN 2018

Keywords: *statistical analysis, smartphone market, price forecast, factor analysis, neural network.*

Abstract. *The paper presents a statistical analysis of the smartphone market, announced in 2018, from 11 popular manufacturers in the Russian*

Federation. The study, based on the input parameters (characteristics of the smartphone), was aimed at obtaining an adequate price forecast for an arbitrary device, as well as highlighting among the characteristics of those that have the greatest impact on the final cost of the device.

Р.Р. Вишняков

МИРЭА – Российский технологический университет

УДК 51-74

АЛГОРИТМ СТРУКТУРНО-ПАРАМЕТРИЧЕСКОГО СИНТЕЗА РАДИОТЕХНИЧЕСКОЙ СИСТЕМЫ

Ключевые слова: *оптимизация, алгоритм, программный комплекс, структурная схема радиотехнического тракта, элементная база.*

В статье рассматривается алгоритм построения программного комплекса синтеза радиотехнической системы. Предлагаемый алгоритм представляет собой четыре основных блока и реализован в виде программного комплекса. Каждый блок представляет собой последовательно выполняемый шаг алгоритма. Для каждого блока приведено описание принципа программно-математической реализации.

Современный уровень развития радиоэлектронной промышленности предоставляет большой выбор возможности построения линейных трактов приемников и передатчиков. В то же время количество нормируемых параметров, по которым формируются требования к приемникам и передатчикам, увеличивается. Провести сравнительный анализ структурного построения приемного или передающего тракта с большим количеством контролируемых параметров при значительном количестве как структурного, так и элементного вариантов построения достаточно сложно. С целью решить данную задачу в настоящей работе приведено описание алгоритма, на основе которого возможно построение программного комплекса синтеза радиотехнической системы.

Существующие САПР позволяют производить расчет тракта с заданным структурным построением. При этом параметры каждого из узлов тракта определяются пользователем, что затрудняет анализ при большом количестве элементов, которые можно использовать для построения схемы. Кроме того, узлы тракта в современных САПР для обеспечения большей универсальности проектирования представляют собой унифицированные блоки, что позволяет воспроизводить в виде модели большинство из применяемых элементов. Однако это влечет за

собой ряд сложностей. Первое заключается в том, что параметры, которые можно регулировать и задавать у отдельного узла, ограничены стандартным набором. Это обстоятельство не позволяет учесть отсутствующие в модели характеристики или более сложные математические зависимости характеристики узла, например, нелинейные зависимости, зависимости в виде функции многих переменных или многофункциональной зависимости. Как следствие, ряд характеристик линейного тракта в целом не поддается расчету. Второе ограничение связано с невозможностью в готовых САПР произвести структурную оптимизацию, допуская лишь параметрическую оптимизацию при заранее заданной структуре. Для решения вышеперечисленных задач в настоящей работе предлагается алгоритм, который позволил бы построить программный комплекс. Такой комплекс должен выполнять расчет требуемых характеристик тракта на основе данных о его структуре и параметрах его узлов, а также произвести структурно-параметрическую оптимизацию, опираясь на базу данных реальных элементов. Программный комплекс должен упростить принятие решения о построении тракта, опираясь на применение тех или иных элементов, автоматизировав поиск элементов, а также позволить произвести структурно-параметрическую оптимизацию, опираясь на результаты расчета характеристик структурной схемы.

Программный комплекс, предназначенный для синтеза радиотехнической системы, предлагается реализовать по следующему алгоритму.

1. Задание начальных условий и базы данных.
2. Формирование возможных структурных построений с учетом физических ограничений.
3. Предварительный расчет и оценка параметров выбранной структурной схемы.
4. Блок расчета требуемых характеристик и структурно-параметрической оптимизации.
5. Блок принятия решения.

Блок задания начальных условий и базы данных должен содержать в себе требования, предъявляемые к тракту. База данных должна представлять собой перечень реальных элементов и описание их характеристик. Перечень доступных элементов может формироваться на любой основе: на основе моделей проектируемых элементов, предлагаемых современными производителями, на основе моделей реализованных устройств и др. Наиболее полное описание характеристик узла (математическое описание варьируемых характеристик, функциональная взаимосвязь различных параметров элемента, статистическое отклонение от номинального значения)

позволит получить более точное описание характеристик тракта. Это в свою очередь позволит произвести оптимизацию схемы.

Следующим шагом алгоритма предполагается **формирование структурной схемы тракта** по заданным характеристикам. Поскольку при автоматизированном подборе элементов вариантов реализации тракта будет определяться выражением (1), поиск подходящего структурного построения может потребовать больших вычислительных и ресурсов памяти.

$$k = \sum_{i=2}^N N_i^M, \quad (1)$$

где k – количество вариантов реализации тракта;

N – количество узлов в линейном тракте;

M – количество доступных элементов для построения тракта.

Чтобы ускорить работу алгоритма, предлагается ввести физические ограничения на начальную структурную схему. Все физические ограничения выбираются из опыта разработки подобных трактов. Ограничения задаются в виде признаков, присваиваемых используемым элементам схемы, и алгоритмически реализуются в виде проверки следующих условий.

1. Количество элементов определенного класса (усилитель, фильтр, сумматор мощности и др.) в определенной структуре не может превышать количество, определяемое классом элемента. Например, количество аттенюаторов должно быть ограничено количеством в две-три единицы, количеством фильтров, усилителей – в три-четыре единицы и т.д.

2. Как в случае с вышеупомянутым правилом, при построении тракта, необходимого для усиления, в его составе должно быть не менее одного каскада усиления.

3. Позиционирование элементов определенных классов в тракте также определяется классом самих элементов. К примеру, усилитель промежуточной частоты, или фильтр нижних частот, не может располагаться в тракте перед малошумящим усилителем и т.д.

4. Элементы одного класса не могут следовать друг за другом в тракте более определенного количества раз, определяемого классом элементов. Например, размещение четырех и более каскадов фильтрации или аттенюации следом друг за другом в тракте недопустимо.

Кроме того, чтобы учесть грубые ошибки построения тракта (несовпадение рабочих диапазонов частот элементов, неудовлетворительные параметры выбранных элементов и т.д.) в алгоритме предполагается использование предварительного расчета характеристики, отражающей корректность начальных условий. При этом такой расчет не должен занимать большое количество ресурсов.

Это может быть оценка суммарного коэффициента усиления тракта или суммарный ток потребления составленной схемы.

Блок расчета характеристик представляет собой вычислительный алгоритм, представленный на рисунке 1.

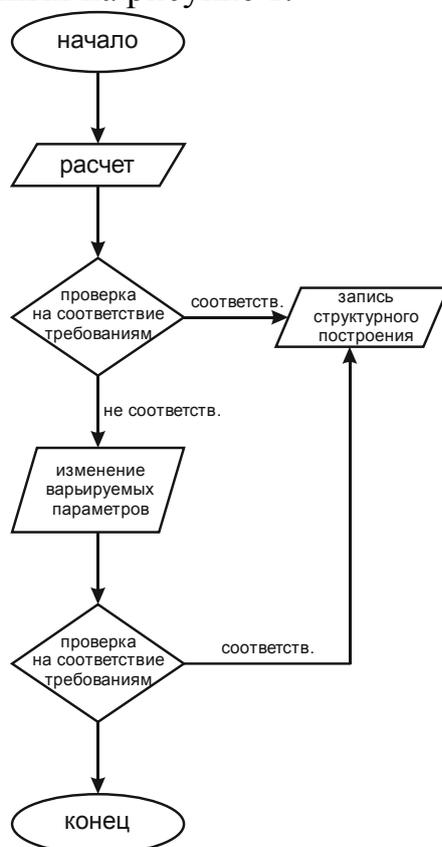


Рис. 1. Блок-схема алгоритма вычисления

Основная задача представленного алгоритма заключается в моделировании радиотехнической системы и параметрической оптимизации узлов. Блок расчета состоит из следующих последовательно выполняемых действий.

На этапе расчета по заданному структурному построению, определенному на этапе формирования структурной схемы тракта, по известным математическим соотношениям рассчитываются все требуемые параметры тракта.

На следующем этапе производится проверка характеристик тракта на соответствие требованиям, заданным в блоке задания начальных условий. Если выбранные элементы и структурное построение удовлетворяют требованиям, предъявляемым к радиотехнической системе, то программный комплекс производит запись всех требуемых параметров. Записи подлежит структурная реализация и выбранные элементы базы данных.

В случае если требования к проектируемому тракту не выполняются, то следующим шагом становится варьирование параметров узлов тракта. В комплексе с последующим этапом проверки

на соответствие требованиям данный алгоритм представляет собой параметрическую оптимизацию. Параметрическая оптимизация может быть реализована различными методами. Методы оптимизации могут быть реализованы аналитическими и алгоритмическими способами [1, 2]. Аналитические методы (дифференциальное исчисление, метод множителей Лагранжа, принцип максимума Понтрягина) требуют сложных преобразований, поскольку выражения для некоторых рассчитываемых параметров схемы имеют сложный вид. В то же время численные методы, требуя значительных вычислительных ресурсов, достаточно алгоритмичны, что позволяет автоматизировать расчеты. В случае если параметрическая оптимизация является одномерной, то есть в тракте существует всего один элемент с единственным варьируемым параметром, то такая задача одномерной оптимизации может быть решена любым из группы детерминированных (метод прямого перебора, метод линейного (нелинейного) программирования, градиентный метод) или случайных (метод статистического градиента, метод Монте-Карло и др.) методов поиска [3]. Чаще целевая функция зависит от многих варьируемых параметров. В таком случае многомерная задача оптимизации может быть решена одним из группы неградиентных методов (метод Гаусса-Зейделя, симплексный метод, случайный поиск) либо из группы градиентных методов (метод градиента, векторный поиск и др.) [4].

Результатом работы описанной части программного комплекса является ряд вариантов структурно-элементно-параметрической реализации приемного/передающего линейного тракта. Поскольку в данном блоке программы все варианты реализации удовлетворяют предъявляемым требованиям, следующим шагом необходимо выполнить поиск наиболее подходящего решения.

Блок принятия решения является конечным шагом алгоритма структурно-параметрической оптимизации. На данном этапе программным комплексом реализуется выбор наиболее подходящей схемы тракта. Поскольку результирующие характеристики схемотехнической реализации представляют собой компромиссное решение задачи выбора удовлетворительного значения всех параметров, то в описываемом блоке должен быть реализован один из методов принятия решения. Метод принятия решения может быть однокритериальным, если выбор осуществляется по одному параметру схемы, или многокритериальным, в случае если требуется сравнить схемы по двум и более параметрам. Однокритериальный метод может представлять собой сортировку по выбранному параметру. Многокритериальный метод необходим при комплексном поиске лучшей структурно-параметрической реализации. К таким методам можно отнести метод главного критерия, метод последовательных

уступок, метод свертки критериев, метод Парето. Параметры приведенных методов определяются эмпирически и определяются на основании опыта разработки аналогичных систем.

Заключение. В данной работе предложен алгоритм структурно параметрической оптимизации. Данный алгоритм позволяет реализовать программный комплекс синтеза радиотехнической системы, в частности линейного тракта приемника или передатчика. Программный комплекс должен решать следующие задачи:

1. на основе заданных пользователем требований к радиотехнической системе и опираясь на доступную элементную базу синтезировать структурную схему тракта с учетом критерия физической реализуемости и технической целесообразности.

2. далее по известным математическим выражениям рассчитать конечные характеристики синтезированной схемы и, при необходимости, провести параметрическую оптимизацию.

3. блок принятия решения программного комплекса позволит произвести цикл структурно-параметрической оптимизации.

Отличительной особенностью описываемого комплекса от аналогичных САПР, позволяющих решать схожие задачи, является наличие структурно-элементно-параметрической связи. Наличие такой увязки также позволит в дальнейшем решать обратную задачу – определение требований к характеристикам для поиска подходящего элемента. При этом результирующие характеристики тракта не ограничиваются стандартным набором наиболее важных параметров. Возможно использование дополнительных математических выражений для описания физических параметров. Это позволяет проводить более глубокий анализ схемы на этапе выбора структурного построения и выбора элементов, например, с учетом статистического разброса параметров элементов, учетом многофункциональной, многопараметрической зависимости характеристик схемы. Данный программный комплекс может быть использован как самостоятельный инструмент построения схемы, подбора элементов, так и в качестве модуля сквозного проектирования радиотехнических систем. Кроме того, приведенный алгоритм может быть распространен для решения любой задачи от схемотехнического до системотехнического синтеза либо анализа построения радиотехнического устройства или системы. В этом случае потребуются корректировка начальных условий и блока расчета итоговых характеристик.

Литература

1. Черноруцкий, И. Г. Методы оптимизации. Компьютерные технологии. СПб.: БХВ-Петербург, 2011. 384 с.

2. Parkinson A.R., Balling R.J., Hedengren J.D. Optimization Methods for Engineering Design. Applications and Theory. Provo, Utah, USA: Brigham Young University, 2013. 208p.

3. Васильев Ф. П. Методы оптимизации. М.: Факториал Пресс, 2002. 824 с.

4. Пантелеев А.В., Летова Т.А. Методы оптимизации в примерах и задачах: учеб. пособие, 2-е изд., испр. – М.: Высш. шк., 2005. 544 с.

R. R. VISHNYAKOV

MIREA – Russian Technological University

STRUCTURAL AND PARAMETRIC SYNTHESIS ALGORITHM OF RADIO SYSTEM

Keywords: *optimization, algorithm, software application, radio system block scheme, hardware components*

Abstract. *In this paper algorithm of radio system synthesis software constructing is considered. The proposed algorithm consists of four main blocks. Each block is a sequentially executed step of the algorithm. For each block description of software and mathematical concept is presented.*

Т.Т. Гатауллин, А.М. Нигмедзянова

*Казанский (Приволжский) федеральный университет,
Институт математики и механики им. Н.И. Лобачевского*

УДК 004.94, 517.9

ИССЛЕДОВАНИЕ РЕШЕНИЙ И ПОСТРОЕНИЕ ФАЗОВЫХ ПОРТРЕТОВ ЛИНЕЙНЫХ АВТОНОМНЫХ СИСТЕМ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В СКМ Maple

Ключевые слова: *математическое моделирование, системы компьютерной математики, дифференциальное уравнение, программирование, качественный анализ систем дифференциальных уравнений, фазовый портрет, СКМ Maple.*

Статья посвящена разработке программы для автоматического вычисления линейного дифференциального уравнения и построение его фазового портрета. Работа представляет собой электронное приложение по решению задач математической физики в системе компьютерной математики Maple. Актуальность темы в том, что по нашей программе создание фазового портрета линейных автономных систем, определение вида, а также решение дифференциального уравнения заключается в одной процедуре, написанной в СКМ Maple.

Целью данной работы является создание программы по качественному анализу автономных систем дифференциальных уравнений произвольного вида в системе компьютерной математики Maple и визуализации их решений. Научная новизна данного проекта состоит в том, что данная программа позволяет определять вид, решать и строить фазовый портрет линейных дифференциальных уравнений в одной программе, это способствует быстрому и наглядному решению.

В статьях одного из авторов [1-11] построено немало моделей, помогающих наглядно визуализировать ряд непростых задач, связанных с преподаванием математики.

Исследование решений и построение фазовых портретов линейных автономных систем дифференциальных уравнений выполнено средствами компьютерных технологий. В качестве инструмента обучения используется система компьютерной математики Maple как современная инновационная технология, обладающая такими дидактическими возможностями, как наглядность, моделирование и динамика. В данной работе написана процедура, которая помогает визуализировать основные свойства линейных дифференциальных уравнений, заданных в произвольном виде.

Программа рассчитана для построения фазового портрета для линейных динамических систем. Для начала мы подключили пакеты для построения фазовых портретов и графиков:

```
with(plots);
with(plottools);
with(DETools);
with(LinearAlgebra).
```

Процедура рассматривает систему, которые состоят из линейных уравнений:

$$P := (x; y) \rightarrow a * x + b * y + j$$

$$Q := (x; y) \rightarrow c * x + d * y + l.$$

Эту систему мы исследуем и построим фазовые портреты решений. Далее проверяем, является ли система автономной («защита от дурака»).

if j = l = 0 then

```
print("Автономная система дифференциальных уравнений")
```

Если система является автономной, то решаем дифференциальное уравнение:

```
SODE4 := diff(x(t); t) = P(x(t); y(t)); diff(y(t); t) = Q(x(t); y(t));
```

```
print("Система дифференциальных уравнений:"SODE);
```

```
resh := dsolve([SODE4]);
```

```
print("Решение системы дифференциального уравнения:"resh).
```

С помощью уравнения

$$S := \text{solve}(\lambda^2 - (a + d) * \lambda + a * d - b * c = 0; \lambda)$$

мы определяем точки покоя (с помощью корней уравнения S).

```
if type(evalf(S[1])); numeric then
```

```

print("характеристическое уравнение имеет вещественные корни");
if S[1] <> S[2] then
print("невырожденный");
if evalf(S[1] * S[2]) > 0 then
print("невырожденный");
q := 0 :
p := 200 :
if evalf(S[1]) < 0 then
print("Устойчивый узел");
else
print("Неустойчивый узел");
fi;
else
print("седло");
q := 10 :
p := 50 :
fi;
else
print("Вырожденный");
q := 50 :
p := 100 :
if S[1] < 0 then
print("Устойчивый узел");
else
print("Неустойчивый узел");
fi;
fi;
izo := solve(k = (c + d * k)/(a + b * k); k);
print(izo) :

```

Далее с помощью пакета DEtools строим фазовый портрет. Если же система является неоднородной, то подбираем преобразование, приводим систему к автономной и решаем его.

```

Sis := solve([a * x + b * y = j; c * x + d * y = l]; x; y);
print(Sis"Положение равновесия");
x0 := Sis[1];
y0 := Sis[2];
P := (ξ; η) -> a * (ξ + rhs(x0)) + b * (η + rhs(y0)) + j;
Q := (ξ; η) -> c * (ξ + rhs(x0)) + d * (η + rhs(y0)) + l;
print(P)

```

Детерминант матрицы равен нулю, это случай, когда точек покоя нет и фазовым портретом являются только прямые линии. Выводим решение. DEtools[DEplot](SODE3; [x(t); y(t)]; t = 300::300; [[x(0) = m; y(0) = n]]; stepsize = 0:1; method = dverk78; arrows = SMALL; color = NAV Y; linecolor = RED)

В итоге для одного из уравнений мы получаем следующий результат (приводим для примера результат выполнения программы для

двух различных линейных автономных систем дифференциальных уравнений):

"Система дифференциальных уравнений:" $\left[\frac{d}{dt} x(t) = x(t) + y(t) + 2, \frac{d}{dt} y(t) = -x(t) + y(t) + 1 \right]$

"Неоднородная система дифференциальных уравнений"

"имеет положение равновесия"

$\left\{ x = -\frac{1}{2}, y = -\frac{3}{2} \right\}$ "Положение равновесия"

$(\xi, \eta) \rightarrow 1 (\xi + rhs(x0)) + 1 (\eta + rhs(y0)) + 2$

"Измененная система дифференциальных уравнений:" $\left[\frac{d}{dt} \xi(t) = \xi(t) + \eta(t), \frac{d}{dt} \eta(t) = -\xi(t) + \eta(t) \right]$

"Решение дифференциального уравнения" $\{ x(t) = e^t (-C1 \sin(t) + C2 \cos(t)), y(t) = e^t (\cos(t) C1 - \sin(t) C2) \}$

"Корень характеристического уравнения", $1 + 1 \cdot I = \lambda_1$

"Корень характеристического уравнения", $1 - 1 \cdot I = \lambda_2$

"Вертикальная изоклина:", $x + y + 2 = 0$

"Горизонтальная изоклина:", $-x + y + 1 = 0$

$1 + I$

"характеристическое уравнение имеет комплексные корни"

"Фокус"

"Неустойчивый фокус"

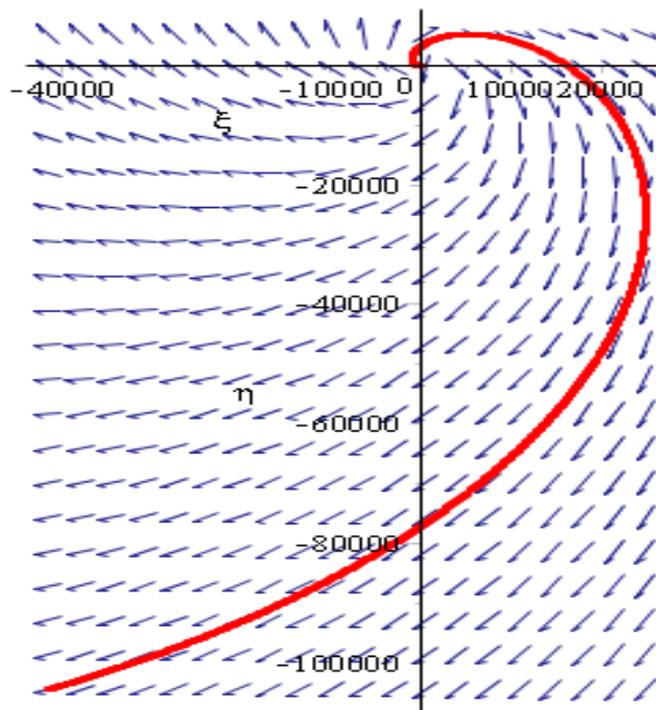


Рис. 1

"Система дифференциальных уравнений:" $\left[\frac{d}{dt} x(t) = x(t) - 4y(t), \frac{d}{dt} y(t) = x(t) - y(t) \right]$

"Решение системы дифференциального уравнения:" $\left\{ x(t) = C1 \sin(\sqrt{3} t) + C2 \cos(\sqrt{3} t), y(t) = -\frac{1}{4} C1 \cos(\sqrt{3} t) \sqrt{3} + \frac{1}{4} C2 \sin(\sqrt{3} t) \sqrt{3} + \frac{1}{4} C1 \sin(\sqrt{3} t) + \frac{1}{4} C2 \cos(\sqrt{3} t) \right\}$

"Однородная система дифференциальных уравнений"

"Корень характеристического уравнения", $1\sqrt{3} = \lambda_1$

"Корень характеристического уравнения", $-1\sqrt{3} = \lambda_2$

"Вертикальная изоклина:", $x - 4y = 0$

"Горизонтальная изоклина:", $x - y = 0$

"характеристическое уравнение имеет комплексные корни"

"Центр"

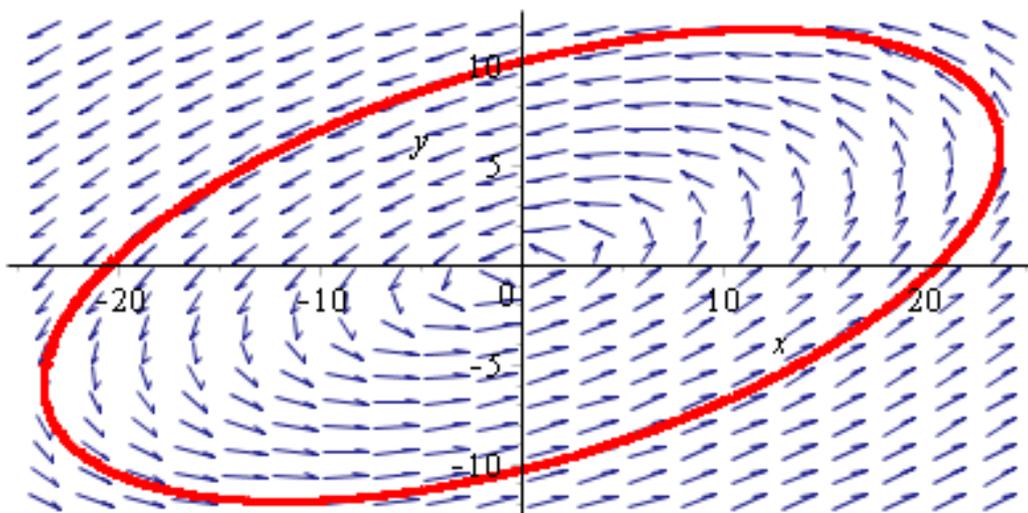


Рис. 2

Литература

1. Нигмедзянова А.М. Оснащенная динамическая визуализация задач математической физики // Информационные технологии в образовании и науке – ИТОН-2012: материалы конференции и труды семинара. 8-12 октября 2012 г. Казань, 2012. С. 127–121.

2. Нигмедзянова А.М. Оснащенная динамическая визуализация построений сечений многогранников // Международный научный семинар «Нелинейные поля в теории гравитации и космологии» и Российская школа «Математическое и компьютерное моделирование фундаментальных объектов и явлений». Казань, 21-26 октября 2013 г. Казань, 2013. С. 151–157.

3. Нигмедзянова А.М. Оснащенная динамическая визуализация построения точки по ее координатам на проективной прямой // Системы компьютерной математики и их приложения: материалы международной конференции. Смоленск: Изд-во СмолГУ, 2014. Вып. 15. С. 36–38.

4. Нигмедзянова А.М. Динамическая визуализация построения точки в пространстве по ее проективным координатам // Международная научно-практическая конференция ИТОН-2014. IV-й международный семинар и международная школа «Математическое и компьютерное моделирование фундаментальных объектов и явлений в системах компьютерной математики». Казань: Изд-во «Фолиант», 2014. С. 236–239.

5. Нигмедзянова А.М. Приведение кривых второго порядка к каноническому виду с помощью метода инвариантов в СКМ Maple // Системы компьютерной математики и их приложения: материалы XVI Международной научной конференции, посвященной 75-летию профессора В.П. Дьяконова. Смоленск: Изд-во СмолГУ, 2015. Вып. 16. С. 30–32.

6. Нигмедзянова А.М. Классификация поверхностей второго порядка к каноническому виду с помощью метода инвариантов в СКМ Maple // Международная научно-практическая конференция «Информационные технологии в образовании и науке – ИТОН-2015». Казань: КГУ, 2015. С. 160–162.

7. Нигмедзянова А.М. Приведение поверхностей второго порядка к каноническому виду с помощью метода инвариантов в СКМ Maple // Системы компьютерной математики и их приложения: материалы XVII Международной

научной конференции «Системы компьютерной математики и их приложения» (СКМП-2016). Смоленск: Изд-во СмолГУ, 2016. Вып. 17. С. 17–19.

8. Нигмедзянова А.М. Оснащенная визуализация приведения поверхностей второго порядка к каноническому виду с помощью метода инвариантов в СКМ MAPLE // Информационные технологии в образовании и науке (ИТОН-2016): сборник V-ой международной науч.-практической конф. Казань, 5-7 ноября 2016 г. Казань, 2016. С. 59-63.

9. Нигмедзянова А.М., Ахмадиева Л.Р. Визуализация построения графиков замечательных кривых в СКМ Maple // XVIII Международная научная конференция «Системы компьютерной математики и их приложения», посвященная 70-летию В.И. Мунермана, 19-21 мая 2017. Смоленск: Изд-во СмолГУ, 2017. С. 27-30.

10. Нигмедзянова А.М., Исмоилова А.Р. Замощение плоскости многоугольниками в системе GeoGebra // XIX Международная научная конференция «Системы компьютерной математики и их приложения», посвященная 100-летию Смоленского государственного университета, 18-20 мая 2018. - Смоленск: Изд-во СмолГУ, 2018. С. 70-79.

11. Нигмедзянова А.М., Уткина М.С. Графическое построение замечательных точек и линий произвольного треугольника в СКМ Maple // XIX Международная научная конференция «Системы компьютерной математики и их приложения», посвященная 100-летию Смоленского государственного университета, 18-20 мая 2018. - Смоленск: Изд-во СмолГУ, 2018. С. 79-85.

12. Понтрягин Л.С. Обыкновенные дифференциальные уравнения. М.: Наука, 1974.

13. Петровский И. Г. Лекции по теории обыкновенных дифференциальных уравнений. М.: Наука, 1970.

14. Федорюк М.В. Обыкновенные дифференциальные уравнения. М.: Наука, 1980.

A.M. Nigmedzianova, T.T. Gataullin
Kazan Federal University

RESEARCH SOLUTIONS AND BUILDING PHASE PORTRAITS OF LINEAR AUTONOMOUS SYSTEMS OF DIFFERENTIAL EQUATIONS IN Maple SCM

Keywords: *mathematical modeling, systems of computer mathematics, differential equation, programming, qualitative analysis of systems of differential equations, phase portrait, SCM Maple.*

Abstract. *The work is devoted to the development of a program for automatic calculation of the linear differential equation and the construction of its phase portrait. The work is an electronic application for solving problems of mathematical physics in the system of computer mathematics Maple. The relevance of the topic is that according to our program, the creation of a phase portrait of linear autonomous systems, the definition of the form, as well as the solution of the differential equation is one procedure written in SCM Maple.*

**ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПОГРЕШНОСТИ
ЭКСПЕРИМЕНТАЛЬНОГО ИССЛЕДОВАНИЯ
РАСПРЕДЕЛЕНИЯ ДЛИН ПУТЕЙ МЕЖДУ УЗЛАМИ
В ГЛОБАЛЬНОЙ СЕТИ**

Ключевые слова: *распределение длин путей между узлами, глобальные сети, traceroute, имитационное моделирование.*

В статье рассматривается экспериментальное исследование распределения длин путей между узлами в глобальной сети. Описывается методика измерения, используемые инструменты, полученные результаты. Анализируется корректность методики исследования, а также количественная оценка вносимой измерением погрешности при помощи имитационного моделирования.

1. Постановка задачи. Разработка высокоэффективных сетевых приложений невозможна без информации о структуре сети, в которой планируется их работа. При этом структура малых сетей обычно проста, регулярна и, соответственно, легко прогнозируема. Её можно описать как в виде графа, вершины которого будут узлами сети, а рёбра — связями (его матрицы инцидентности или матрицы смежности), так и через детерминированный алгоритм роста, а затем воспроизвести в натурной или компьютерной модели для тестирования приложения [1]. Структура сети среднего размера уже не может быть описана детерминированным алгоритмом роста, но при наличии физического доступа к существующей сети может быть исследована и представлена в виде графа. Тем не менее использовать для тестирования ограниченное количество предопределённых графов обычно нерационально. Соответственно, желательно разработать алгоритм роста, позволяющий получить большое количество правдоподобных моделей сети. Это можно сделать, внося в алгоритм элемент случайности.

Строение глобальной сети Интернет невозможно описать абсолютно точно. Этому препятствуют как её гигантский размер (по разным оценкам, в глобальную сеть Интернет входит от 10^8 до 10^{10} узлов), так и постоянное изменение структуры.

Соответственно, для имитации крупных сетевых структур необходимо использовать алгоритмы, включающие стохастические элементы. Полученные таким образом модели часто объединяются под названием случайных графов [2; 3] (хотя некоторые источники называют

так только модель Эрдёша–Реньи). Как правило, для моделирования сетей различной природы в настоящее время используется модель Барабаши–Альберт. Она, с одной стороны, позволяет получить связанное дерево (именно так выглядит на сетевом уровне любая сеть, использующая стек протоколов TCP/IP), а с другой — адекватно описывает множество социальных структур. Но действительно ли она адекватна сетевому уровню глобальной сети Интернет?

Из-за невозможности полного исследования глобальной сети на этот вопрос нет достоверного ответа. Но косвенно подтвердить адекватность или неадекватность модели может проведение выборочных измерений какой-либо характеристики и её сопоставление со значением, предсказанным моделью.

Наиболее доступной для измерения характеристикой двух узлов сети является длина пути между ними, то есть количество промежуточных узлов, требующихся для передачи данных. Соответственно, для сети в целом это распределение длин путей и его моменты [4].

2. Методика выборочного измерения длин путей. Исследовать маршрут передачи данных между двумя узлами на сетевом уровне возможно при помощи утилиты `tracert` [5], использующей протокол UDP и доступной во всех основных операционных системах, включая GNU/Linux. В стандартной поставке ОС Microsoft Windows утилита `tracert` отсутствует, а её ближайший аналог — `tracert` — использует протокол ICMP и иной формат вывода. Для установления маршрута передачи данных от узла i к узлу j необходимо иметь возможность запустить `tracert` на узле i и знать IP-адрес или доменное имя узла j . Анализируя вывод утилиты, можно узнать как длину пути $s(i,j)$ между узлами i и j , так и имена и адреса промежуточных узлов.

Таким образом, процедура измерения длины пути априори несимметрична и участвующие в исследовании узлы разделяются на те, на которых запускается утилита `tracert` (в дальнейшем они называются корневыми), и те, которые только отвечают на запросы (в дальнейшем — конечные узлы). Количество доступных исследователю корневых узлов, где доступен запуск программ, как правило, неизмеримо меньше, чем конечных, для которых достаточно наличия белого IP-адреса и поддержки стандартного протокола UDP. В данном исследовании использовались три корневых узла, обозначаемые далее А, М и Н.

Соответственно, измерение разделяется на три фазы:

- 1) получение одной или несколько выборок IP-адресов конечных узлов;
- 2) измерение мгновенных расстояний от корневого узла до всех конечных узлов заданной выборки;

3) статистическая обработка полученных данных.

Для получения IP-адресов конечных узлов были выбраны файлообменные пиринговые сети. Для этого при помощи торрент-клиента rtorrent, обеспечивающего вывод информации об активных загрузках в виде текста, удобного для анализа, были скачаны несколько популярных файлов. После этого список активных пиров, взаимодействующих с торрент-клиентом, периодически сохранялся в текстовый лог и затем обрабатывался специально написанным скриптом на языке оболочки bash. Необходимо отметить, что в том случае, когда реальный IP-адрес пира был серым [6], в списке отображался адрес шлюза его провайдера. Соответственно, дальнейшие измерения не учитывают часть пути, лежащую внутри подсети провайдера такого пира. Из соображений симметрии в дальнейшем логично рассмотреть как длины путей между корневым узлом и полученным IP-адресом конечного узла, так и длины путей между шлюзом провайдера корневого узла и IP-адресом конечного узла.

В результате исследования трёх крупных торрент-трекеров, различающихся целевой аудиторией, были получены три крупных выборки конечных узлов, обозначаемые в дальнейшем P, R и T. Так как одной из целей исследования было определение зависимости распределения длин путей между узлами от набора конечных узлов, выборки P, R и T исследовались по отдельности.

Для второй фазы — измерения длин путей — необходимо задать список IP-адресов конечных узлов. Корневым является узел, на котором запускается скрипт, выполняющий действия второй фазы. Именно для каждого адреса конечного узла из списка автоматически вызывается утилита traceroute с перенаправлением вывода в уникальный для каждого узла файл, после чего сформированный набор файлов анализируется при помощи средств оболочки bash, а также утилит GNU / Linux. Итогом второй фазы исследования является файл, имя которого включает обозначения используемой выборки конечных узлов, текущего корневого узла и номер измерения. Каждая строка этого файла содержит как общую длину пути от корневого узла до некоторого конечного (общее количество использованных для передачи узлов), так и длину пути от подсети провайдера корневого узла до этого конечного узла.

Третья фаза измерения — обработка полученных файлов, позволяющая рассчитать для всех возможных x количество $\tilde{\zeta}(x)$ путей между узлами, имеющих полную длину x и количество $\zeta(x)$ путей, имеющих длину x без учёта подсети провайдера корневого узла, а также анализ и визуализация полученных распределений $\tilde{\zeta}(x)$ и $\zeta(x)$ — выполняется в среде GNU Octave [7] и также автоматизирована.

3. Результаты измерения. Полученные распределения $\tilde{\zeta}(x)$ полных длин путей между узлами приведены на рисунке 1. Они отчётливо разделяются на группы с различным положением главного максимума, причём разные группы соответствуют различным корневым узлам.

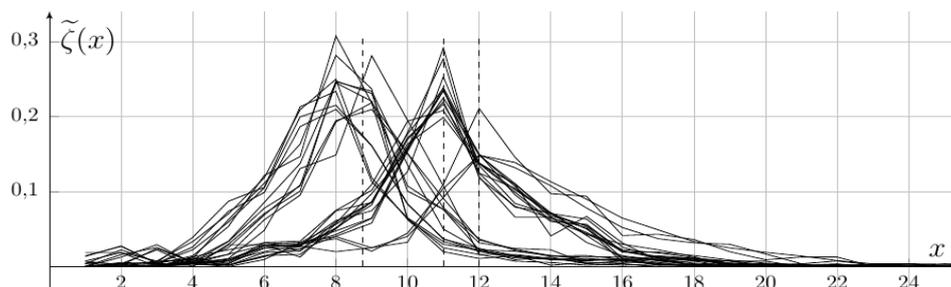


Рис. 1. Распределения полных длин путей между узлами

В таблице 1 показан разброс характеристик распределения $\tilde{\zeta}(x)$ для различных измерений с корневым узлом КУ и набором конечных узлов ОУ. Малый разброс характеристик измерений из корневого узла Н обусловлен ограниченным количеством и временем доступных измерений.

Таблица 1

Выборочные характеристики $\tilde{\zeta}(x)$ (полных длин путей)

КУ	ОУ	Характеристика				
		Мода	МО	СКО	Асимметрия	Экссесс
А	Р	8–9	8,2–9,4	3,4–3,9	5,4–6,0	38–50
А	R	8–9	8,1–9,3	3,4–3,8	3,1–3,7	21–26
А	Т	8	7,8–9,2	2,8–3,8	1,9–4,0	10–26
М	Р	11	11,1–11,7	3,5–4,4	3,6–4,3	26–28
М	R	11	11,4–11,8	3,5–3,7	2,9–3,5	19–23
М	Т	11	11,3–11,8	3,2–4,0	3,2–4,0	18–24
Н	Р	12	13,3	4,8	2,5	12
Н	R	12	13,3	4,8	1,4	7
Н	Т	12	12,6	4,3	0,7	5

Видно, что мода и математическое ожидание (МО) зависят от корневого узла, среднеквадратическое отклонение (СКО) принимает схожее значение для всех измерений, асимметрия и эксцесс имеют сильный разброс (что нормально для характеристик высокого порядка), но всегда положительны и достаточно велики.

При рассмотрении длин путей, отсчитываемых не от самого корневого узла, а от подсети провайдера корневого узла (то есть не включая в длину пути подсеть провайдера корневого узла), положение главного максимума приблизительно совпадает (рис. 2).

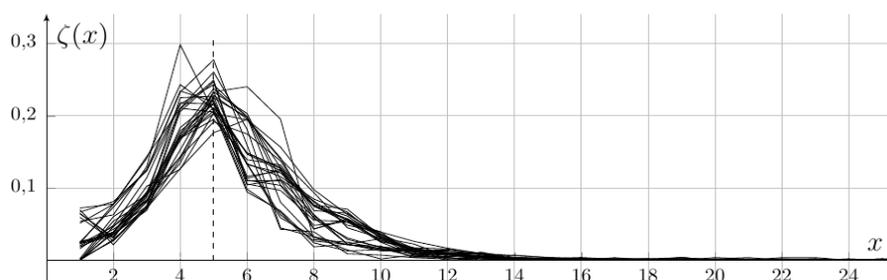


Рис. 2. Распределения длин путей между узлами без учёта подсети провайдера корневого узла

В таблице 2 показан разброс характеристик распределения $\zeta(x)$ без учёта подсети провайдера корневого узла для различных измерений с корневым узлом КУ и набором конечных узлов ОУ.

Таблица 2

Выборочные характеристики $\zeta(x)$ (длин путей без учёта подсети провайдера корневого узла)

КУ	ОУ	Характеристика				
		Мода	МО	СКО	Асимметрия	Экссесс
А	Р	5–6	5,0–6,1	3,3–3,9	5,8–6,2	42–56
А	Р	4–5	3,1–3,5	3,1–3,5	5,8–6,5	30–36
А	Т	4–5	4,8–6,0	2,5–3,5	2,5–4,5	13–31
М	Р	4–5	5,4–5,8	3,3–4,3	4,2–4,4	26–32
М	Р	5	4,9–6,0	3,3–3,6	4,0–4,7	21–26
М	Т	5	4,8–6,0	3,0–4,0	3,2–4,0	21–23
Н	Р	5	6,3	3,9	4,5	26
Н	Р	5	6,1	3,1	3,6	25
Н	Т	6	6,1	3,1	3,1	20

Исключение из рассмотрения подсети провайдера корневого узла привело к более или менее схожим характеристикам для всех измерений.

При этом с учётом того, что в качестве конечных узлов используются только белые IP-адреса, которые чаще всего являются шлюзами провайдеров, удаление из путей подсети провайдера корневого узла фактически соответствует рассмотрению не всей глобальной сети Интернет, а её подмножества — узлов с белыми IP-адресами. Это достаточно большая сеть, включающая более 10^9 узлов, и её структура должна быть в целом подобна глобальной.

4. Имитационное моделирование погрешности исследования.

Для сопоставления результатов экспериментального исследования с моделью Барабаши–Альберт была проведена также серия прогонов имитационной модели. В рамках каждого прогона выращивалось связное дерево G из заданного количества N вершин. Для него рассчитывались как полное (рассчитанное по всему дереву G) распределение длин $\zeta(x)$ путей, так и несколько его оценок, имитирующих экспериментальные измерения — частичные

распределения $\zeta_i(x)$, характеризующие расстояния от произвольно выбранного из G корневого узла до набора произвольно выбранных конечных.

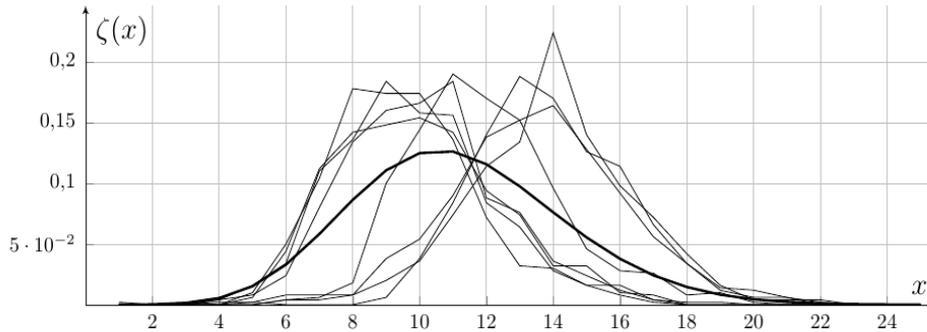


Рис. 3. Распределение $\zeta(x)$ длин путей между узлами (жирная линия) и его оценки

Как показала обработка результатов имитационного моделирования, форма полного распределения $\zeta(x)$ близка к дискретизации нормального, мода и математическое ожидание $\zeta(x)$ близки друг к другу (табл. 3), причём при увеличении размера дерева N математическое ожидание увеличивается пропорционально $\ln(N)$, что соответствует известным результатам [1]. Асимметрия слабоположительна, а эксцесс очень близок к нулю.

Таблица 3.

Характеристики полного распределения $\zeta(x)$ длин путей между узлами в деревьях Барабаши-Альберт

N	Характеристика				
	Мода	МО	СКО	Асимметрия	Эксцесс
10^3	6–7	6,3–7,6	1,9–2,5	0,1–0,5	–0,4–+0,2
10^4	8–9	8,1–9,6	2,4–3,1	0,1–0,5	–0,2–+0,1
10^5	10–12	10,4–11,8	2,8–3,2	0,2–0,4	–0,1–+0,2

Частичные распределения $\zeta_i(x)$, соответствующие различным корневым узлам, имеют разные математические ожидания μ_i , но они отклоняются от математического ожидания полного распределения μ как вверх, так и вниз, так что усреднённое значение $\text{avg}(\mu_i)$ является достаточно точной оценкой μ . В таблице 4 приведены некоторые из проведённых измерений для $N = 10^5$. Среднеквадратическое отклонение частных распределений σ_i практически всегда меньше среднеквадратического отклонения полного распределения σ , так что усреднённое значение $\text{avg}(\sigma_i)$ является оценкой σ , заниженной на 20–30%.

Таблица 4

Характеристики полного $\zeta(x)$ и частичных распределений $\zeta_i(x)$ длин путей между узлами в деревьях Барабаши-Альберт

$N = 10^5$	Характеристика				
	Мода	МО	СКО	Асимметрия	Эксцесс
$\zeta(x)$	10	10,4	2,8	0,4	0,2
Разброс $\zeta_i(x)$	7–13	8,1–14,0	1,9–2,1	–0,3–+0,6	0,2–2,7
Среднее $\zeta_i(x)$	10,1	10,6	2,0	0,3	0,8
$\zeta(x)$	10	10,4	2,9	0,3	0,1
Разброс $\zeta_i(x)$	5–13	6,3–13,6	2,1–2,4	–0,3–+0,7	–0,4–+1,2
Среднее $\zeta_i(x)$	10,0	10,6	2,2	0,3	0,1
$\zeta(x)$	11	11,4	3,1	0,3	0,2
Разброс $\zeta_i(x)$	7–14	7,3–14,3	2,1–2,4	–0,3–+0,6	0,1–0,8
Среднее $\zeta_i(x)$	10,9	11,1	2,3	0,1	0,4
$\zeta(x)$	11	11,4	3,0	0,2	0,0
Разброс $\zeta_i(x)$	8–14	7,8–14,5	2,0–2,4	–0,3–+0,4	0,0–1,7
Среднее $\zeta_i(x)$	10,5	10,4	2,2	0,1	0,3
$\zeta(x)$	11	11,6	3,2	0,2	–0,1
Разброс $\zeta_i(x)$	7–13	7,5–13,2	2,1–2,8	0,0–+0,3	–0,6–+1,7
Среднее $\zeta_i(x)$	10,8	11,0	2,4	0,1	–0,2

Разброс характеристик высоких порядков — асимметрии и эксцесса частичных распределений — достаточно велик. Тем не менее некоторые тренды, связывающие характеристики частичных распределений и характеристику полного, присутствуют и здесь. Асимметрия A_i частичных распределений для модели Барабаши–Альберт немного занижена по отношению к асимметрии полного A , и в среднем $\text{avg}(A_i)$ близка к нулю.

Эксцесс частичных распределений E_i так же, как и эксцесс полного E , близок к нулю. Тем не менее, в целом, если рассматривать $\text{avg}(E_i)$ как оценку E , есть тенденция небольшого завышения положительных значений эксцесса и занижения отрицательных.

В дальнейшем было также проведено имитационное моделирование деревьев разной структуры и измерений длин типа «корневой узел — конечные узлы». Обработка результатов подтвердила, что математическое ожидание таким способом оценивается более или менее корректно, среднеквадратическое отклонение — с занижением на 20–30%, систематическая погрешность асимметрии и эксцесса зависит от модели роста дерева. Тем не менее, существенно положительные для всех частичных распределений асимметрия и эксцесс получаются только при существенно положительных асимметрии и эксцессе полного распределения.

Сравнение экспериментально полученных данных и частичных распределений длин путей между узлами в деревьях Барабаши–Альберт позволяет сделать **выводы**:

— полученные в результате эксперимента значения характеристик не являются приближёнными значениями характеристик распределения длин путей между узлами в глобальной сети, но могут быть использованы для их оценки с учётом систематической погрешности методики измерения;

— различие между распределением длин путей между узлами на сетевом уровне в глобальной сети и в модели Барабаши–Альберт обусловлено не случайными отклонениями, а отличием роста реальной сети от модели Барабаши–Альберт;

— необходимо дополнительное исследование для построения адекватной модели роста глобальной сети.

Литература

1. Городилов А. В., Кононова А. И., Шаньгин В. Ф. Особенности передачи данных в децентрализованных пиринговых сетях // Известия вузов. Электроника (ВАК). 2012. No. 6(98). С. 95.
2. Райгородский А. М. Модели случайных графов. М.: МЦНМО, 2011. 136 с.
3. Евин И. А. Введение в теорию сложных сетей // Компьютерные исследования и моделирование. 2010. No. 2. С. 121–141.
4. Кобзарь А.И. Прикладная математическая статистика. М.: ФИЗМАТЛИТ, 2006. 816 с.
5. Команда traceroute Linux. <https://losst.ru/komanda-traceroute-linux>.
6. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. СПб.: Питер, 2001. 672 с.
7. GNU Octave. Сайт. <http://www.gnu.org/software/octave/>.

A. V. Gorodilov, A. I. Kononova

National Research University of Electronic Technology

SIMULATION AS TOOL OF ERROR ASSESSMENT OF EXPERIMENTAL MEASUREMENT OF PATH-LENGTHS DISTRIBUTION IN GLOBAL NETWORK

Keywords: *path length distribution, global networks, traceroute; simulation.*

Abstract. *In the article we present experiment that is aimed at establishing path-lengths distribution in global network by direct measurement. Method of measurement, tools and results are given. Method's applicability is accessed. Measurement error values have been evaluated by simulation.*

ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ПРИ АНАЛИЗЕ ВЫЖИВАЕМОСТИ БОЛЬНЫХ РАКОМ ЖЕЛУДКА

Ключевые слова: анализ выживаемости онкологических больных, прогнозирование времени жизни, метод Ньютона-Рафсона, модель Кокса, автоматизированные нейронные сети.

В данной статье проводится анализ выживаемости онкологических больных с диагнозом рак желудка. Целью данного исследования является прогнозирование с помощью автоматизированной нейронной сети выживаемости пациентов после постановки этого диагноза в зависимости от возраста и пола больного. С помощью модели Кокса выделены значимые переменные, которые влияют на функцию выживания.

«Обеспокоенность и повышенное внимание к онкологии обусловлены устойчивой тенденцией роста заболеваемости во всем мире» [3]. В настоящее время злокачественные новообразования – это вторая причина смерти в России после сердечно-сосудистой патологии. Раку подвержены все группы населения, независимо от социального статуса, пола, возраста.

«В 2017 г. в России впервые выявлено почти 541 тыс. онкологических больных, умерло от злокачественных новообразований 290,7 тыс. больных, что составляет 15,9% в общей структуре смертности. На конец 2017 г. в территориальных онкологических учреждениях России состояли на учете 3 630 567 пациентов» [2].

Врачи делают все возможное, чтобы спасти пациента, продлить ему жизнь после постановки страшного диагноза. В связи с этим возникает актуальный вопрос о выживаемости пациентов после диагностирования болезни.

Целью данного исследования является прогнозирование времени выживаемости больных раком желудка после постановки диагноза с помощью нейронной сети в зависимости от возраста и пола больного.

Материалом для исследования послужили статистические данные, содержащие информацию о 50 пациентах с диагнозом рак желудка Московского научно-исследовательского онкологического института имени П.А. Герцена. При формировании группы наблюдения были выполнены следующие условия:

- группа однородна по форме злокачественной опухоли (рак желудка);
- для всех больных, выбрана одинаковая точка отсчета (начальное время) – дата постановки диагноза.

Для работы со статистическим материалом был использован пакет прикладных программ Statistica 12.

Используя процедуру Regression model, удалось установить, что продолжительность жизни пациента зависит от возраста пациента на момент установления диагноза и его пола. На этом этапе было принято решение об использовании наиболее общей регрессионной модели (которая не предполагает никаких данных о природе или форме функции выживаемости) – пропорциональной модели интенсивностей Кокса. Диалоговое окно анализа представлено на рисунке 1.

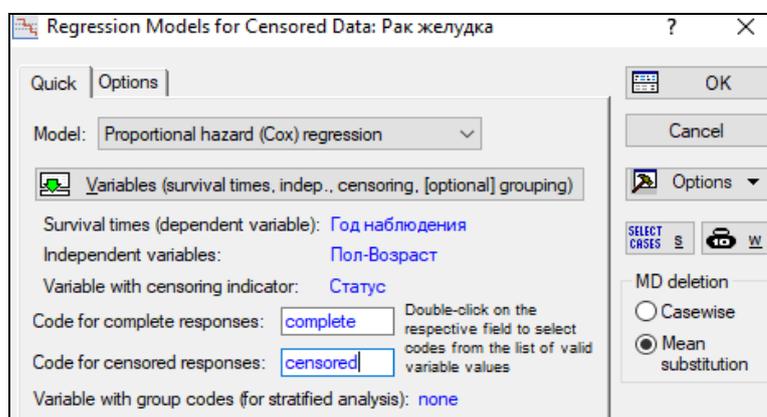


Рис. 1. Диалоговое окно регрессионной модели

После заполнения данного окна программа запускает итерационную процедуру оценивания параметров, которая максимизирует логарифмическую функцию правдоподобия регрессионной модели посредством метода Ньютона-Рафсона. После того как программа находит наиболее подходящие оценки параметров, появляется окно Regression Results (рис. 2).

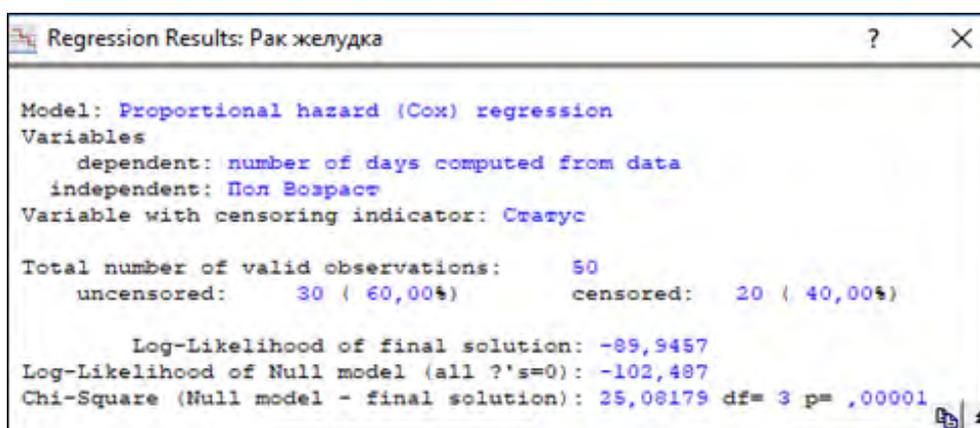


Рис. 2. Информационное окно регрессионной модели

Из информационной части окна видно, что значение (Chi-Square) статистически значимо (уровень значимости $p = 0,00001$, что существенно меньше 0,05), поэтому можно сделать вывод, что некоторые переменные связаны с выживаемостью больного.

На рисунке 3 изображена таблица оценок параметров. В первом столбце представлены коэффициенты параметров в регрессионном уравнении, во втором столбце – стандартные ошибки, в третьем столбце – значения t -критерия. В пятом и шестом столбцах приведены значения статистики Вальда (Wald Statist) и уровень значимости p . Так как в математической статистике принято считать статистически значимыми (при $p < 0,05$) оценки параметра, которые хотя бы в два раза превышают ошибку этого параметра ($t > 2,0$), то на основании данных таблицы приходим к выводу: все параметры являются значимыми предикторами функции выживаемости, потому что соответствующая величина t -критерия больше 2 и значение p существенно меньше 0,05.

Dependent Variable: Survival times in days (Пак желудка) Censoring var.: Статус Chi? = 25,0818 df = 3 p = ,00001								
N=50	Beta	Standard Error	t-value	Wald Statist.	p	Risk ratio	Risk ratio 95% lower	Risk ratio 95% upper
Пол	0.07585	0.481295	2.15759	0.02483	0.024781	1.078798	0.420011	2.770890
Возраст	0.00370	0.015344	2.24110	0.05813	0.009480	1.003706	0.973970	1.034350

Рис. 3. Информационное окно регрессионной модели

С помощью модели Кокса удалось выделить значимые переменные для функции выживаемости. Однако, к сожалению, данный анализ не обладает возможностью построить точный прогноз из-за малого количества факторов, а также наличия категориальной переменной. Поэтому было принято решение об использовании нейросетевых моделей.

Нейронные сети являются многофункциональным методом прогнозирования. Одно из главных преимуществ нейронных сетей – это возможность находить закономерности между переменными, выявлять сложные взаимосвязи. Искусственная нейронная сеть способствовала построению прогноза выживания онкологических больных после постановки диагноза – рак желудка.

Для дальнейшего проведения анализа применялся модуль Automated Neural Networks (автоматизированные нейронные сети) в программе Statistica 12.

В качестве выходной переменной была выбрана переменная Год наблюдения (количество лет, в течение которых пациент находился под наблюдением), в качестве непрерывной входной переменной – Возраст, в качестве категориальной переменной – Пол. Для обучения нейронной

сети использовали данные 45 пациентов, а значения оставшихся пациентов оставили для сравнения результатов прогноза нейронной сети с реальными данными.

Следующий шаг исследования, который представлен на рисунке 4, это процесс обучения нейронных сетей.

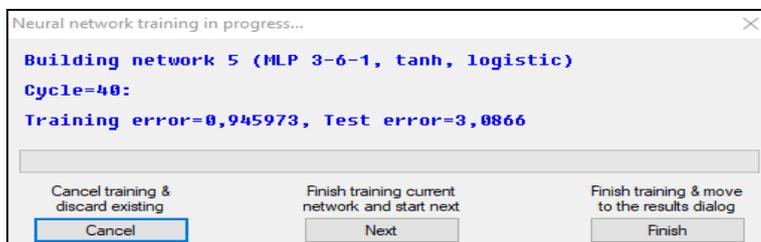


Рис. 4. Обучение нейронных сетей

После обучения нейронных сетей программа представляет данные (рис. 5), которые показывают наибольшую производительность на обучающей, контрольной и тестовой подвыборках.

Index	Net. ID	Net. name	Training perf.	Test perf.	Validation p...	Algori
1	1	MLP 3-6-1	0,882839	0,917946	0,929206	BFGS
2	2	MLP 3-8-1	0,872489	0,917008	0,935040	BFGS
3	3	MLP 3-7-1	0,865526	0,925601	0,927254	BFGS
4	4	MLP 3-9-1	0,882642	0,926337	0,935639	BFGS
5	5	MLP 3-6-1	0,887139	0,907323	0,938194	BFGS

Рис. 5. Результаты обучения нейронных сетей

Для выбора лучшей нейронной сети были использованы следующие критерии:

- 1) наибольшая производительность на всех подвыборках;
- 2) гистограмма распределения остатков должна быть близка к нормальному виду;
- 3) значения целевых и выходных переменных на диаграмме рассеивания находятся вблизи прямой.

Сравнивая модели по выше предложенным критериям, окончательно подобрали для анализа нейронную сеть № 1 MLP 3-6-1, имеющую самую высокую производительность на всех подвыборках и 6 нейронов на скрытом слое.

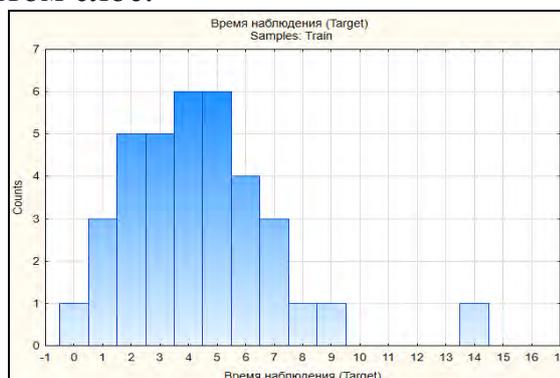


Рис. 6. Гистограмма остатков

Гистограмма остатков для нейронной сети № 1 приведена на рисунке 6, она близка к нормальному распределению.

Из диаграммы рассеивания целевых и выходных значений переменных сети на обучающей и контрольной подвыборках, представленной на рисунке 7, можно сделать вывод, что большинство точек расположено вблизи прямой, хотя присутствуют некоторые выбросы.

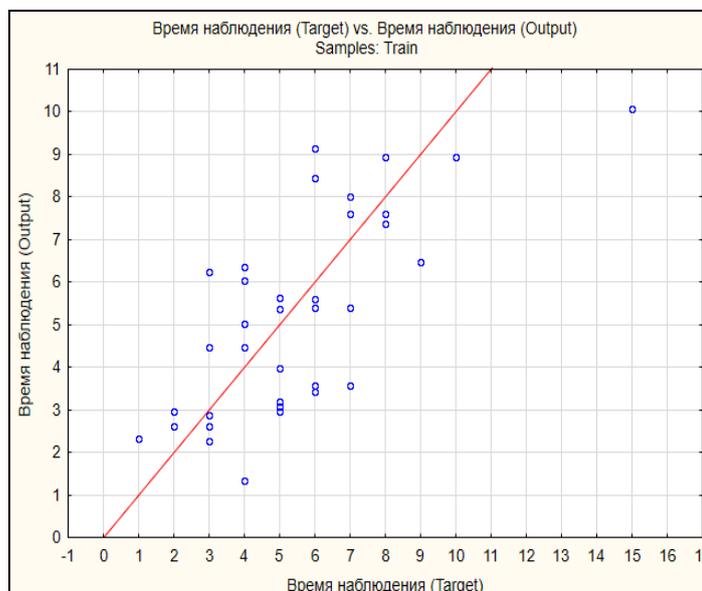


Рис. 7. Диаграмма рассеивания

	1 Пол	2 Возраст	Год наблуден
45	1	68	1
46	1	49	4
47	1	27	9
48	0	38	6
49	1	64	2
50	0	39	6

Рис. 8. Данные 5 больных

#	1.Год н...	Возраст	Пол
1	2,248910	68,0000...	1
2	4,435988	49,0000...	1
3	9,195158	27,0000...	1
4	7,528129	38,0000...	0
5	2,581653	64,0000...	1
6	7,116561	39,0000...	0

Рис. 9. Спрогнозированные данные

После выбора лучшей нейронной сети проведено сравнение предсказанных значений с реальными данными. На рисунке 8 представлены данные, которые не были включены в обучение нейронной сети, а на рисунке 9 – спрогнозированное время выживания этих больных.

Проанализировав данный результат, можно сделать вывод, что выбранная нейронная сеть дает неплохой прогноз.

Таким образом, исследуя статистическую базу данных с помощью анализа выживаемости, удалось подтвердить, что значимыми предикторами функции выживаемости больных раком желудка являются Пол и Возраст. Кроме того, построенная нейронная сеть

способна после учета этого диагноза спрогнозировать время выживания больного раком желудка, зная его возраст и пол.

Литература

1. Гашев С.Н., Бетляева Ф.Х. Математические методы в биологии: анализ биологических данных в системе STATISTICA: учебное пособие. Тюмень: Изд-во Тюменского государственного университета, 2014. 208 с.
2. Злокачественные новообразования в России в 2017 году (заболеваемость и смертность) / под ред. А.Д. Каприна, В.В. Старинского, Г.В. Петровой. М.: МНИОИ им. П.А. Герцена – филиал ФГБУ «НМИРЦ» Минздрава России, 2018. 250 с.
3. Организация онкологической службы России (методические рекомендации, пособия для врачей) / под ред. В.И. Чиссова, В.В. Старинского. М.: ФГУ МНИОИ им. П.А. Герцена Росмедтехнологий, 2007. – 663 с.
4. Халафян А.А. STATISTICA 6. Статистический анализ данных. ООО «Бином-Пресс», 2009. 512 с.

G. S. Evdokimova, A. K. Tarasevich
Smolensk State University

APPLICATION OF NEURAL NETWORKS IN THE ANALYSIS OF SURVIVAL OF PATIENTS WITH GASTRIC CANCER

Keywords: *analysis of cancer patient's survival, prediction of lifetime, Newton-Raphson method, Cox model, automated neural networks.*

Abstract. *This article analyzes the survival of cancer patients diagnosed with gastric cancer, the aim of this study is to predict the survival of patients after diagnosis using the automated neural network, depending on the age and sex of the patient. With the help of the Cox model, significant variables that affect the survival function are identified.*

В.Р. Кристалинский, А.А. Голубенцов
Смоленский государственный университет

УДК 330.46

НЕЧЕТКАЯ МОДЕЛЬ УПРАВЛЕНИЯ СИСТЕМОЙ УПРАВЛЕНИЯ В УМНОМ ДОМЕ

Ключевые слова: *нечеткая модель, нечеткое прогнозирование, умный дом, автоматизированные системы, нечеткая логика.*

Разработана модель, в которой методами нечеткой логики осуществляется процесс управления работой системы отопления в

умном доме. Представлены преимущества применения ее на практике. Описаны входные и выходные лингвистические переменные, а также составлены правила и функции принадлежности. Модель реализована в среде Visual Studio 2017, на языке C#, посредством библиотеки AForge.Fuzzy.

1. Постановка задачи. Нечеткая логика позволяет представить окружающий мир не только в черном и белом цвете, она охватывает более полный спектр красок, тем самым описывая явления на языке, более приближенному к естественному языку человека. Таким образом, системы, использующие нечеткую логику, легче в управлении и более гибкие в применении на практике. Модели, построенные на нечеткой логике, позволяют более полно описывать все аспекты неопределенности, присущие процессам и вещам в окружающем нас мире.

В данной статье рассмотрена нечеткая модель реализации Интернета вещей на примере системы управления отоплением в умном доме. Предложенная модель позволяет автономно без участия человека контролировать уровень тепла в доме. Система будет управлять устройством, изменяющим температуру, поддерживая ее комфортной для жизни.

Для реализации модели использована среда Visual Studio 2017 с помощью AForge.NET Framework и библиотеки AForge.Fuzzy.

AForge.NET Framework – это среда C#, созданная для разработчиков в области искусственного интеллекта, нейронных сетей, генетических алгоритмов и машинного обучения.

Фреймворк состоит из различных библиотек, предназначенных для решения разных задач. В частности, библиотека AForge.Fuzzy предназначена для работы с нечеткими вычислениями.

Составим следующие входные лингвистические переменные: «Температура в доме», «Температура на улице», «Скорость изменения температуры». Выходной переменной будет: «Температура отопления».

На основе данных переменных система будет принимать решения. Теперь для каждой из них рассмотрим множество значений, которые переменная может принимать.

В качестве терм-множества для первой лингвистической переменной будем использовать множество {«Очень холодная», «Холодная», «В пределах нормы», «Теплая», «Очень теплая»}, соответственно для других: {«Очень холодная», «Холодная», «В пределах нормы», «Теплая», «Очень теплая»}, {«Отрицательная», «Положительная»}. Для выходной переменной будем использовать множество {«Очень холодная», «Холодная», «В пределах нормы», «Теплая», «Очень теплая»}. Терм-множества для входных переменных

могут иметь одинаковые элементы, поскольку они используются для описания одной и той же физической величины – температуры.

Для функционирования системы необходимо задать функции принадлежности. В качестве базовой функции принадлежности выбрана трапецеидальная функция. Этот выбор был произведен на основе экспериментального анализа различных типов функций принадлежности. Задание функций принадлежности в системе AForge.Fuzzy показано в примере 1.

Пример 1. Выражение присваивания

```
FuzzySet fsVeryCold = new FuzzySet(«VeryCold», new  
TrapezoidalFunction(-50, -20, TrapezoidalFunction.EdgeType.Right));
```

задает функцию принадлежности для терма «Очень холодная». График функции fsVeryCold показан на рисунке 1.

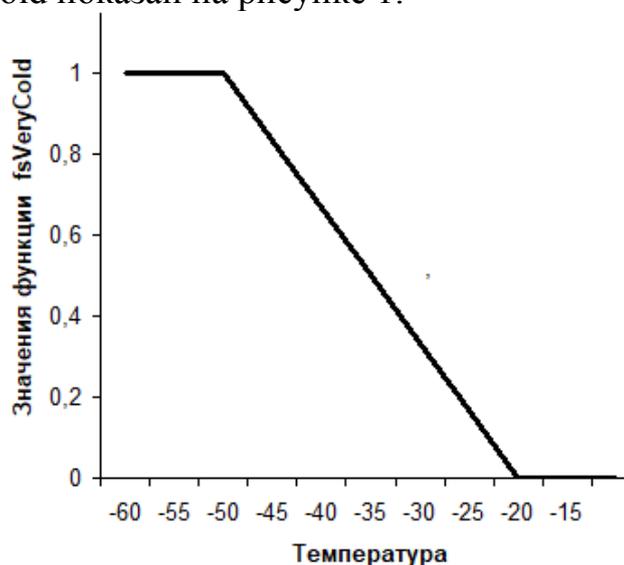


Рис. 1. График функции fsVeryCold

Далее будут заданы лингвистические переменные для температуры в доме и температуры на улице. Поскольку терм-множества этих переменных одинаковы, описание этих переменных аналогично. Это описание имеет следующий вид:

```
LinguisticVariable houseT = new  
LinguisticVariable("HouseTemperature", -50, 50);  
houseT.AddLabel(fsVeryCold);  
LinguisticVariable outdoorT = new  
LinguisticVariable("OutdoorTemperature", -50, 50);  
outdoorT.AddLabel(fsVeryCold);
```

Описание последней лингвистической переменной для скорости изменения температуры осуществляется с использованием функций fsNegative и fsPositive следующим образом.

```
LinguisticVariable temperatureChangeRate = new  
LinguisticVariable("TemperatureChangeRate", -10, 10);
```

```
temperatureChangeRate.AddLabel(fsNegative);
temperatureChangeRate.AddLabel(fsPositive);
```

Далее приводится описание выходной лингвистической переменной для температуры отопления.

```
LinguisticVariable heatingT = new
LinguisticVariable("HeatingTemperature", -50, 50);
heatingT.AddLabel(fsVeryCold);
```

На этом завершается описание лингвистических переменных.

Для работы системы все сведения о лингвистических переменных помещаются в общую базу данных fuzzyDB следующим образом.

```
Database fuzzyDB = new Database();
fuzzyDB.AddVariable(houseT);
fuzzyDB.AddVariable(outdoorT);
fuzzyDB.AddVariable(heatingT);
fuzzyDB.AddVariable(temperatureChangeRate);
```

Большинство величин в мире нельзя оценить измерительным устройством, но человек, исходя из своего опыта, может приблизительно дать оценку подобным величинам. Пользуясь такими оценками, человек может справиться с управлением или оптимизацией различных систем. Исходя из экспертных знаний и накопленного опыта, в нечетких моделях создаются правила нечеткого вывода, основанные на прошлых знаниях человека, описывающие те или иные ситуации.

Задание правил в системе AForge.Fuzzy показано в примере 2.

Пример 2

Процесс перехода от нечеткого представления выходной переменной к четкому значению (дефаззификация) в данной системе осуществляется методом центраида. Выбор метода также основан на экспериментальных данных. Для инициализации системы используется следующий оператор:

```
var IS = new InferenceSystem(fuzzyDB, new
CentroidDefuzzifier(1000));
```

Далее следует задание правил нечеткого вывода. Одно из правил имеет вид «ЕСЛИ температура в доме очень холодная И температура на улице очень холодная И изменение температуры отрицательное ТО температура отопления очень теплая». В системе AForge.Fuzzy задание этого правила имеет следующий вид:

```
IS.NewRule(«Rule 1», «IF HouseTemperature IS VeryCold AND
OutdoorTemperature IS VeryCold AND TemperatureChangeRate IS
Negative THEN HeatingTemperature IS VeryWarm»);
```

Аналогично описываются остальные правила, необходимые для работы системы.

Для получения результата необходимо ввести значения для входных переменных и получить выходной результат. Этот процесс реализован следующим набором процедур:

```
IS.SetInput("HouseTemperature", (float)houseTemperature);  
IS.SetInput("OutdoorTemperature", (float)outdoorTemperature);  
IS.SetInput("TemperatureChangeRate", (float)changeRateTemperature);  
  
var heatingTemperatureResult = IS.Evaluate("HeatingTemperature");  
  
return heatingTemperatureResult;
```

Первые три процедуры задают значения входных лингвистических переменных, четвертая процедура вычисляет значение выходной переменной.

На рисунке 2 (а, б) представлены окна ввода-вывода данных, демонстрирующих работу разработанной системы нечеткого вывода. На них же показана работа системы в различных условиях.

Входные данные		
Температура в доме	Температура на улице	Скорость изменения температуры
<input type="text" value="10"/>	<input type="text" value="0"/>	<input type="text" value="10"/>
Выходные данные		<input type="button" value="Считать"/>
Температура отопления	<input type="text" value="16.2663440704346"/>	

а)

Входные данные		
Температура в доме	Температура на улице	Скорость изменения температуры
<input type="text" value="25"/>	<input type="text" value="-10"/>	<input type="text" value="10"/>
Выходные данные		<input type="button" value="Считать"/>
Температура отопления	<input type="text" value="-5.84002590179443"/>	

б)

Рис. 2. Реакция системы на различные входные условия
а) повышение температуры, б) понижение температуры

Заключение. Разработанная нечеткая модель для управления системой отопления демонстрирует возможности использования нечеткого моделирования на практике. Соединение разработанной программной модели с датчиками оборудования системы IoT позволит решать различные задачи управления физическими объектами. Предложенный подход может быть использован в различных областях, таких как: системы управления освещением, видеонаблюдением, уборкой в доме и в промышленной области.

Литература

1. Хижняков Ю.Н. Алгоритмы нечеткого, нейронного и нечетконейронного управления в системах реального времени. Пермь: ПНИТУ. 2013.
2. Аверкин А.Н., Поспелова Д.А. [и др.] Нечеткие множества в моделях управления и искусственного интеллекта. М.: Наука. 1986.
3. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб.: БХВ-Петербург, 2005.

V.R. Kristalinskii, A.A. Golubencov
Smolensk State University

FUZZY MODEL OF CONTROL OF THE HEATING SYSTEM IN A SMART HOME

Keywords: *fuzzy model, fuzzy prediction, smart house, automated systems, fuzzy logic.*

Abstract. *A model has been developed in which fuzzy logic methods are used to control the operation of the heating system in a smart home. The advantages of using it in practice are presented. The input and output linguistic variables are described, and the rules and membership functions are compiled. The model was implemented in Visual Studio 2017 in C # using the AForge.Fuzzy library.*

В.Р. Кристалинский, А.Н. Колюсь
Смоленский государственный университет

УДК 519.688

О ВЕБ-СЕРВИСЕ ДЛЯ ВЫЧИСЛЕНИЯ ЗНАЧЕНИЙ ИСТИННОСТИ НЕЧЕТКИХ ВЫСКАЗЫВАНИЙ

Ключевые слова: *нечеткая логика, веб-сервис, степень истинности, нечеткие высказывания, экспертная база знаний, ассоциативные правила, LaTeX, Python.*

Django-Mathjax, диаграмма деятельности, кроссплатформенность, адаптивность, листинг.

В статье представлены теоретические основы нечеткой логики, необходимые для понимания важности создания веб-сервиса. Описан принцип работы веб-сервиса, приведен листинг важных частей кода. Продемонстрированы результаты, и сделаны выводы о значимости проведенной работы.

Сегодня в условиях бурного развития интеллектуальных технологий методы теории нечетких множеств являются удобным средством моделирования сложных динамических процессов в условиях многокритериальности, ограниченности и неопределенности. Процесс развития того или иного явления есть процесс труднопредсказуемый, практически непрогнозируемый, который зависит от многих условий и факторов, и в этих условиях целесообразно использовать нечеткие модели вывода.

Общеизвестно, что в 1965 году американским ученым Лотфи Заде были предложены обобщения классической теории множеств: математическая теория нечетких множеств и нечеткая логика.

Нечеткая логика предназначена для формализации человеческих возможностей к неточным или приближенным рассуждениям. Классическая логика игнорирует проблему неопределенности, так как высказывания и рассуждения в формальных логических системах могут иметь лишь значение «истина» (И, 1) или значение «ложь» (Л, 0). В отличие от этого в нечеткой логике истинность рассуждений оценивается в некоторой степени, которая может принимать и другие отличные от {И, Л} значения [1].

В нечеткой логике множество истинностных значений высказываний обобщается до интервала действительных значений $[0, 1]$, что позволяет высказыванию принимать любое значение истинности из данного интервала. Это численное значение является количественной оценкой степени истинности высказывания, относительно которого нельзя с полной уверенностью сделать вывод об его истинности или ложности. Использование в качестве множества истинностных значений интервала $[0, 1]$ позволяет построить логическую систему, в рамках которой оказалось возможным, используя лингвистические переменные [2], выполнять рассуждения с неопределенностью и оценивать истинность высказываний типа: «Температура в системе довольно высокая», «Птица летит весьма высоко» и др.

Нечеткая логика, прежде всего – логика, поэтому для нее используются те же логические операции, что и в классической логике, только их вычисление на порядок сложнее. Из математической логики известно, что каждая операция истинности задана таблицей истинности,

Нечеткая логика использует величину «степень истинности», вместо классических «истина» и «ложь». Значение «степени истинности» может быть любым из бесконечного множества от 0 до 1 включительно и имеет мощность континуум. Таким образом, в нечеткой логике нельзя пользоваться принятыми в классической логике таблицами истинности. В связи с этим для определения операций в нечеткой логике центральную роль играет отображение истинности (функция истинности).

Например, нечеткую конъюнкцию можно определить по формуле min-конъюнкции $T(A \wedge B) = \min(T(A), T(B))$, которую принято считать основной. Также степени истинности для нечетких операций, в зависимости от предметной области и решения эксперта, могут определяться по разным определениям, например, для нечеткой конъюнкции это могут быть алгебраическое, граничное или драстическое произведение степеней истинности.

Нечёткая логика широко применяется при построении экспертных систем, систем поддержки принятия решений, систем управления, основанных на экспертных знаниях. Нечёткие экспертные системы позволяют не только учитывать неопределенность, но и дают возможность моделировать рассуждения на основе опыта специалистов.

На основе знаний эксперта составляется база знаний – набор продукционных правил, написанных на естественном языке качественных понятий. Основой системы нечетких продукций является множество правил, описывающих взаимосвязь между входными и выходными переменными. С помощью этой системы правил на основе известных алгоритмов нечеткого вывода на основе степеней истинности входных переменных получают значения истинности выходных переменных, причем сами правила также имеют определенные степени истинности. Продукционные правила выглядят следующим образом:
ПРАВ₁: ЕСЛИ «Условие₁» И(ИЛИ) «Условие_k» ТО «Заключение₁» (F₁)
ПРАВ_n: ЕСЛИ «Условие_n» И(ИЛИ) «Условие_m» ТО «Заключение_n»(F_n)

После того, как система правил построена и в соответствии с выбранными алгоритмами нечеткой логики активировано ядро продукций, эксперт может оценить результат и выяснить, насколько он близок к результатам, которые он ожидал получить в соответствии со своими знаниями. Поскольку, как говорилось выше, определений операций нечеткой логики существует несколько, причем алгоритм вычисления значения истинности для каждого из них является непростым, а правил в базе может быть сколь угодно много, то для эксперта выбор определений, приводящих к наиболее адекватной модели, является весьма сложной задачей.

В силу данной проблемы нами было принято решение разработать программное решение, облегчающее труд эксперта, осуществляющее

вычисление степени истинности нечеткого высказывания с любым количеством нечетких операций любой сложности. На наш взгляд, данное решение должно быть кроссплатформенным и пользоваться им могли с любых устройств с доступом в Интернет, как нельзя лучше для этого решения подходит веб-сервис. Динамика использования веб-сервиса отображена на UML диаграмме деятельности (см. рис. 1).

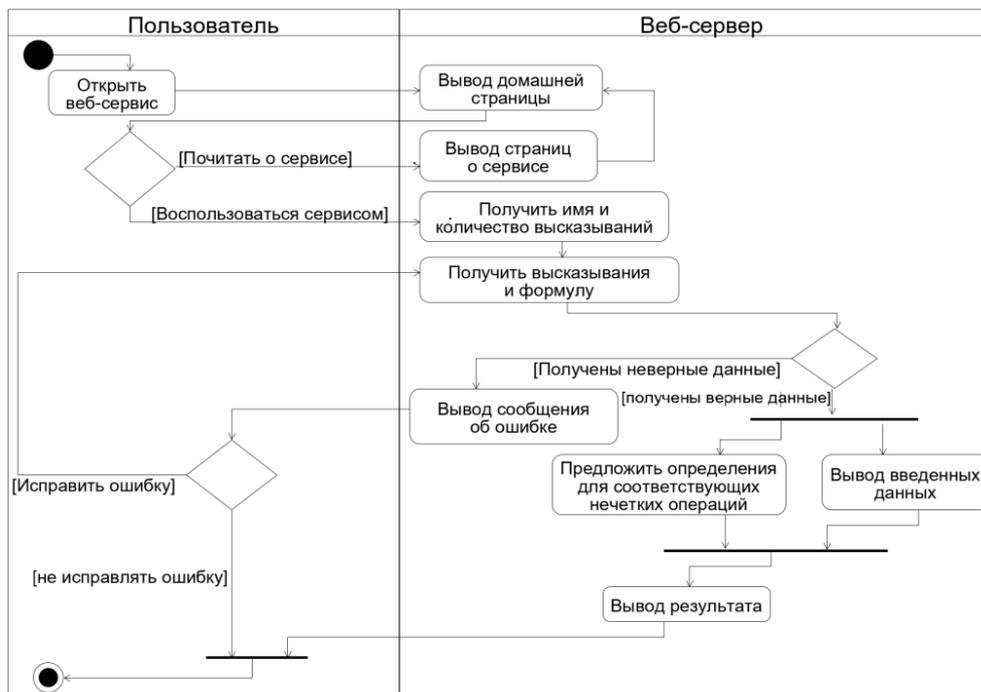


Рис. 1. UML диаграмма деятельности веб-сервиса

Веб-сервис разработали на языке программирования python, используя фреймворк django. Кроме основной задачи, в веб-сервисе, применяя технологии CSS и JavaScript, все изображения и текст адаптивны ко всем разрешениям экранов используемых устройств. А, используя пакет Django-Mathjax, математические выражения и символы отображаются с использованием набора макрорасширений системы компьютерной вёрстки TeX. Преобразование введенной пользователем формулы под формат LaTeX реализовано в функции TextEncodMathJax. В данной функции введенные пользователем операции конвертируются в соответствии с правилами Tex в команды.

Все введенные пользователем данные в формуле проверяются и перед вычислением отображаются на странице. Также, определяется, какие логические операции имеются в формуле и в соответствии с ними выводится выпадающий список для выбора определения. Формула обязательно должна состоять только из таких высказываний, которые введены пользователем, а количество открывающих и закрывающих скобок должно быть равно, иначе будет выдано сообщение об ошибке (см. рис. 2).

Если все данные успешно прошли проверку, то в соответствии с выбранными определениями можно получить результат.

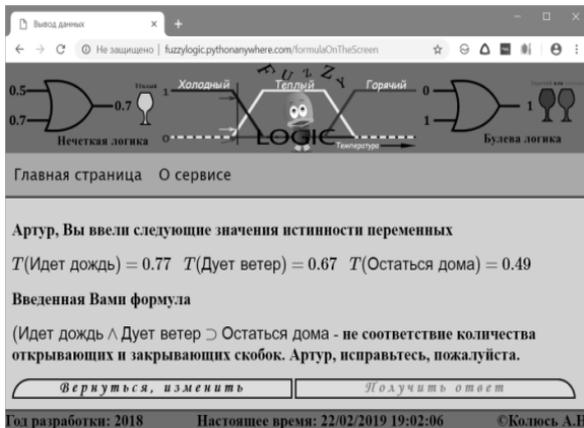


Рис. 2. Сообщение об ошибке

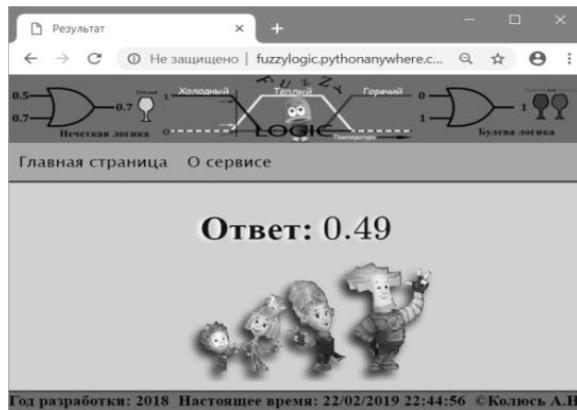


Рис. 3. Окно с результатом работы программы

Выбор данного фреймворка оправдан тем, что модификация кода не составляет особого труда. Например, чтобы добавить новое определение вычисления импликации, нужно добавить в классе формы значения в выпадающем списке и реализацию в функции вычисления импликации.

Поскольку известно, что ассоциативные правила есть не что иное, как другое представление нечетких продукционных правил [3], то основу данного веб-сервиса можно использовать для прогнозирования взаимосвязей между объектами.

Разработанный веб-сервис, расположен по адресу <http://fuzzylogic.pythonanywhere.com>.

Литература

1. Леоненков А.В. Нечеткое моделирование в среде MatLab и fuzzyTECH. СПб: БХВ-Петербург, 2003.
2. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976.
3. Кристалинский В. Р., Колюсь А.Н. Об использовании ассоциативных правил для нечеткого моделирования. Владикавказ: Северо-Кавказский горно-металлургический институт, 2018. С. 294-298.
4. Круглов В.В. Интеллектуальные информационные системы: компьютерная поддержка систем нечеткой логики и нечеткого вывода. М.: Физматлит, 2002. 198 с.
5. Lotfi Zadeh. From computing with numbers to computing with words — from manipulation of measurements to manipulation of perceptions // International Journal of Applied Math and Computer Science. 2002. Vol. 12, No. 3. P. 307-324.

V.R. Kristalinskiy, A. N. Kolyus
Smolensk State University

ABOUT WEB SERVICE TO CALCULATE THE VALUES OF TRUTH OF FUZZY STATEMENTS

Keywords: *fuzzy logic, web service, degree of truth, fuzzy expressions, expert knowledge base, association rules, LaTeX, Python. Django-Mathjax, activity diagram, cross-platform, adaptability, listing.*

Abstract. *The article presents the theoretical foundations of fuzzy logic necessary for understanding the importance of creating a web service. The principle of operation of the web service is described, the listing of important parts of the code is given. The results are demonstrated and are conclusions are made about the significance of the work.*

Р.Е. Кристалинский, С.А. Мартышин
ВА ВПВО ВС РФ

УДК 519.688

РЕШЕНИЕ НЕКОТОРЫХ ЗАДАЧ ОБ ОПТИМАЛЬНОМ РАСПРЕДЕЛЕНИИ РЕСУРСОВ

Ключевые слова: *распределение ресурсов, целераспределение.*

В статье рассматривается методика решения задач оптимального распределения ресурсов в системе Mathematica. В задачах военного характера весьма часто встает вопрос об оптимальном распределении ресурсов: средств поражения, материально-технических средств, людских ресурсов. Математические модели решения таких задач известны давно, однако во многих практически важных случаях их исследование сопряжено со значительными техническими трудностями. Система Wolfram Mathematica предоставляет удобные средства решения такого рода задач, что значительно облегчает использование этих математических моделей специалистами в области их приложений, в частности военного дела.

Многие задачи военного содержания допускают не один, а множество вариантов решения относительно искомым переменных x_1, x_2, \dots, x_n . Ответить на вопрос, какое решение из этого множества является наилучшим, без дополнительных условий невозможно. В качестве такого дополнительного условия выступает целевая функция. Под целевой функцией понимается возможный конечный результат, который желательно получить путём выбора и реализации переменных

x_1, x_2, \dots, x_n , оптимизирующих значение целевой функции и описывающих рассматриваемый процесс.

Рассмотрим некоторые примеры.

Пример 1. Задача целераспределения.

Пусть имеется S различных целей. Важность цели с номером i равна A_i . Имеется m типов средств поражения. Численность средств поражения типа j равна N_j . Средство поражения типа j поражает цель i с вероятностью $\omega_{i,j}$. Пусть $x_{j,i}$ – число средств поражения типа j , распределённых для поражения цели i . Требуется определить матрицу оптимального распределения $X = [x_{i,j}]$, для которой среднее ожидаемое значение ущерба, наносимого S , будет наибольшим.

Решение. Среднее ожидаемое значение ущерба, наносимое S объектам удара, находится из соотношения

$$F = \sum_{i=1}^S A_i \left(1 - \prod_{j=1}^m \varepsilon_{j,i}^{x_{j,i}} \right).$$

На параметры накладываются следующие ограничения:

$$x_{j,i} \geq 0, A_j > 0, 0 < \varepsilon_{j,i} = 1 - \omega_{j,i} < 1, \sum_{i=1}^S x_{j,i} = N_j.$$

Рассмотрим конкретный пример. Для удара по $S = 3$ целям с важностями

$$A_1 = 20; A_2 = 30; A_3 = 50$$

выделены три типа средств поражения в количествах

$$N_1 = 2; N_2 = 3; N_3 = 2.$$

Вероятности поражения каждым средством объекта i -го типа задаются матрицей

$$\begin{bmatrix} 0.40 & 0.16 & 0.30 \\ 0.20 & 0.38 & 0.50 \\ 0.52 & 0.26 & 0.18 \end{bmatrix}.$$

Находим решение поставленной задачи.

$$A = \{20, 30, 50\};$$

$$N1 = \{2, 3, 2\};$$

$$N2 = \text{Length} [N1];$$

$$\omega = \begin{bmatrix} 0.40 & 0.16 & 0.30 \\ 0.20 & 0.38 & 0.50 \\ 0.52 & 0.26 & 0.18 \end{bmatrix}.$$

```

      ε = 1 - ω;
      X = Table[xj,i, {j,1,3}, {i,1,3}];
      S=3;
      F = Sum[A[[i]](1-Prod[ε[[j,i]]X[[j,i]]],
              {i=1,S}, {j=1,N2}];
      T=True;
      Do[T=T & Sum[xj,i]==N1[[j]], {j,1,3}];
      Do[T=T & xj,i ≥ 0, {i,1,S}, {j,1,N2}];
      X1=Flatten[X];
      R=NMaximize[{F,T & X1 ∈ Integers},X1];
      X=X/.R[[2]];
      MatrixForm[X]
      
$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 1 & 1 & 0 \end{bmatrix}$$

      F[X]
      71.726

```

Пример 2. Имеется техническое устройство S , состоящее из m элементов. Безотказная работа каждого элемента безусловно необходима для работы всего устройства в целом.

Элементы могут отказывать (выходить из строя), причём независимо одно от другого. Надёжность (вероятность безотказной работы) всего устройства равна произведению надёжностей всех элементов:

$$P = \prod_{i=1}^m p_i,$$

где p_i – надёжность i -го элемента.

В нашем распоряжении имеются некоторые средства K , которые можно употребить на повышение надёжностей элементов.

Количество средств x_i , вложенное в приспособления, повышающие надёжность i -го элемента, доводит её до значения

$$p_i = f_i(x_i).$$

Требуется определить оптимальное распределение средств по элементам, приводящее к наибольшей надёжности устройства в целом.

Таким образом, мы должны найти наибольшее значение функции

$$F[x_1, x_2, \dots, x_m] = \prod_{i=1}^m f_i[x_i]$$

при следующих ограничениях

$$\sum_{i=1}^m x_i \leq K,$$

$$x_1 \geq 0, \dots, x_m \geq 0.$$

Пример 3. Имеющиеся боевые средства на каждом этапе боевых действий распределяются между двумя подразделениями. Предположим, что на очередном этапе боевых действий первому подразделению выделяется x единиц боевых средств, а второму подразделению y единиц боевых средств. Эффективность использования боевых средств для первого подразделения составляет $g(x)$, второго подразделения – $h(y)$. Вследствие потерь количество боевых средств в первом подразделении уменьшается до ax , второго подразделения – до by .

Предположим, что на первом этапе количество выделенных боевых средств равно 9, $a = 0.65$, $b = 0.73$, $g(x) = 0.65x^2$, $h(y) = 0.8y^2$, число этапов боевых действий равно 3. Требуется найти оптимальное распределение боевых средств на каждом этапе боевых действий и общую её эффективность.

Приведём программу решения рассматриваемой задачи.

$$g[x_] = 0.65 x^2; h[y_] = 0.8y^2;$$

$$a = 0.65; b = 0.73;$$

$$u = 9;$$

$$\omega[x_, y_] = h[y] + g[x];$$

$$x1 + y1 == u;$$

$$s1 = \omega[x1, y1];$$

$$x2 + y2 <= a x1 + b y1;$$

$$x2 + y2 <= 0.65 x1 + 0.73 y1$$

$$s2 = \omega[x2, y2];$$

$$x3 + y3 <= 0.65 x2 + 0.73 y2;$$

$$H[x1_, x2_, x3_, y1_, y2_, y3_] = s1 + s2 + s3;$$

$$\text{Maximize}[\{H[x1, x2, x3, y1, y2, y3], x1 + y1 == u, x2 + y2 <= a x1 + b y1, x3 + y3 <= a x2 + b$$

$$y2, \{x1, x2, x3, y1, y2, y3\} \in \text{Integers}, x1 >= 0, x2 >= 0, x3 >= 0, y1 >= 0, y2 >= 0, y3 >= 0\},$$

$$\{x1, x2, x3, y1, y2, y3\}]$$

$$\{79.85, \{x1 \rightarrow 9, x2 \rightarrow 0, x3 \rightarrow 0, y1 \rightarrow 0, y2 \rightarrow 5, y3 \rightarrow 3\}\}$$

$$x1 = 9, y1 = 0, x2 = 0, y2 = 5, x3 = 0, y3 = 3.$$

Общая оптимальная эффективность равна 79.85.

Литература

1. Вентцель Е.С. Исследование операций. М.: Советское радио, 1972. 552 с.
2. Березин Е.Ф. Оптимальное распределение ресурсов и элементы синтеза систем. М.: Советское радио, 1974. 304 с.
3. Фёдоров В.В. Курс методов оптимизации. М.: Физматлит, 2008. 808 с.
4. Ширяев В.И. Исследование операций и численные методы оптимизации. М.: Высшая школа, 2015. 977 с.

R.E. Kristalinskii, S.A. Martyshin

Military Academy of the anti-aircraft defense systems of the Armed Forces of the Russian Federation

SOLUTION OF SOME PROBLEMS OF THE OPTIMAL DISTRIBUTION OF RESOURCES

Keywords: *distribution of resources, distribution of aims.*

Abstract. *The article is devoted to the methods of solution of problems of the optimal distribution of resources using the system Mathematica. In problems of the military character the question of optimal distribution of resources: means of destruction, material and technical means, people resources often stands. Mathematical models of such problems are well-known long ago, but in many practically important cases their investigation is combined with technical difficulties. System Wolfram Mathematica provides convenient means of the solution of such problems and make sufficiently easier usage of such models by specialists in the field of application, in the military field in particular.*

Т.Б. Солодченкова, Р.Л. Гайчиков

Смоленский государственный университет

УДК 778.38:535.4

ВЫБОР КРИТЕРИЕВ АНАЛИЗА КАЧЕСТВА ИЗОБРАЖЕНИЯ В МЕТОДЕ ЦИФРОВОЙ ГОЛОГРАФИИ

Ключевые слова. *Цифровая голография, 2D голограмма, фотоматрица, пиксель, объектная волна, предметная волна, преобразование Фурье, метод свёртки, пиксель, разрешение матрицы.*

В статье представлены расчёты в среде MathCad четырёх двумерных изображений объектов разной формы методом цифровой

голографии, проведён анализ изображений в зависимости от длины волны опорного пучка. В качестве критериев, выбраны, наряду с визуальным наблюдением, величины отношения сигнал/шум и верность изображения. По результатам исследования предложено дополнительно оценивать качество цифровых 2D изображений средним значением RGB, позволяющим выбрать более оптимальный диапазон длин волн опорной волны.

1. Возможности цифровой голографии

Одним из направлений современной голографической интерферометрии является моделирование процессов с помощью средств вычислительной техники, или цифровая голография (ЦГ) – процесс записи и обработки голограмм с помощью светочувствительных матриц на основе ПЗС-камер в цифровой форме [1, с. 69-70; 2, с. 18]. Принцип получения и восстановления «искусственной» голограммы аналогичен традиционному «экспериментальному», и при этом ожидается расширение возможностей исследования путём изменения разных параметров. Для восстановления голограммы на плоскости используют интегралы Френеля–Кирхгофа, одним из упрощений вычисления которых служит сведение трёхмерной задачи к двумерной [2, с. 30]. Восстановление объектной волны 2D голограмм позволяет изменять масштабы изображения, для этого можно менять параметры схемы и длину волны [1, с. 21]. В ЦГ для исследования доступны и амплитудные, и фазовые характеристики волнового поля [2, с. 23], есть возможность устранять 0-й порядок дифракции и фильтровать фоновые шумы изображения. Процесс восстановления в ЦГ можно осуществлять двумя способами – на основе преобразования Френеля и методом свёртки, причём оба способа основаны на Фурье-преобразовании, но метод свёртки имеет ряд преимуществ [2, с. 37-40].

Недостатком ЦГ является малое разрешение цифровых матриц по сравнению с голографическими фотопластинками [2, с. 28]. Вследствие этого приходится выбирать малые углы между опорной и объектной волнами, что приводит к перекрыванию спектров разных дифракционных порядков. Цифровое восстановление голограмм требует разработку особых методов устранения нулевого порядка дифракции, подавления перекрывания дифракционных порядков и фильтрации спекл-шумов. Для контроля фокусировки на деталях объекта и оценки качества изображения предложено 14 величин [3, с. 31-32]. Однако отмечено, что вопрос количественной оценки, являющийся важным шагом в решении задачи оптимальных преобразований изображений, до конца не решён.

2. Постановка задачи

Осуществить метод цифровой голографии для тестовых 2D объектов разной формы. Численно смоделировать запись объектной волны в программной среде MathCad. Положить в качестве априорной информации значения неизменных параметров: пикселя фотоматрицы dx , угла падения опорной волны θ и расстояние d от объекта до голограммы. Варьировать: размер изображения (в pix), длину λ опорной волны излучения видимого диапазона (шаг 50 нм). Восстановить объектную волну в среде MathCad, применяя метод свёртки, что позволяет получить изображение такого же размера, как исходный объект. Критериями для оценки качества изображений тестовых объектов выбрать (из перечисленных в [3]): визуальное наблюдение, отношение сигнал / шум SNR , верность изображения IF . Вычислить SNR и IF при разных λ и d .

3. Получение голограммы с помощью MathCad

Применим алгоритм получения цифровой голограммы в среде MathCad, описанный в работе [3, с. 19-24], но с другими априорными данными. Он построен на основе метода свёртки, реализован как быстрое Фурье-преобразование (БФП) и включает в себя следующие шаги: 1) задание на рабочем листе MathCad параметров (длина волны опорного пучка $\lambda = 550 \text{ нм}$, размер пикселя фотоматрицы $dx = 0,007 \text{ мм}$, угол падения опорной волны $\theta = 0^\circ$, расстояние от тестового объекта до фотоматрицы $d = 45 \text{ мм}$, размер изображения 200 pix тестового объекта № 1 «точка»); 2) введение тестового объекта формата bmp с помощью встроенной функции (как массив амплитуд); 3) вычисление ядра свёртки посредством вложенных циклов и создание массива данных; 4) прямое Фурье-преобразование ядра и тестового изображения; 5) поэлементное умножение Фурье-преобразований ядра и тестового изображения; 6) обратное Фурье-преобразование результата поэлементного умножения; 7) вычисление суммы комплексных амплитуд объектной и опорной волн; 8) вычисление квадрата модуля суммы комплексных амплитуд объектной и опорной волн (для нахождения распределения интенсивности); 9) нормирование изображения и определение максимального значения массива; 10) центрирование изображения; 11) сохранение полученного изображения.

Аналогично проводилась запись голограмм для других тестовых образцов при тех значениях λ , dx , θ и d , но разных размеров (т. о. № 2 «автомобиль» $400*500 \text{ pix}$, т. о. № 3 «бабочка» $400*400 \text{ pix}$, т. о. № 4 «плакат-человек» $400*400 \text{ pix}$), а также голограмм всех тестовых объектов при изменении λ .

4. Восстановление изображения с помощью MathCad

Восстановление голограммы проводится по подобному алгоритму: 1) задаём априорную информацию (которая должна сохраняться прежней, как при записи); 2) вычисление ядра свёртки; 3) «освещение»

голограммы опорным пучком; 3) прямое Фурье-преобразование ядра и голограммы; 4) обратное Фурье-преобразование результата поэлементного умножения; 5) нормирование изображения; 6) центрирование изображения.

5. Результаты и выводы

Для всех четырёх тестовых образцов наилучшими по критериям визуального наблюдения, вычислений отношения сигнал – шум SNR и верности изображения IF оказались изображения при длине волны опорного пучка в средней части видимого диапазона $\lambda = 0,55$ мкм (табл. 1, рис. 1).

Таблица 1

<i>изображение тестового образца</i>	<i>№ 1 (точка)</i>	<i>№ 2 (бабочка)</i>	<i>№ 3 (автомобиль)</i>	<i>№ 4 (плакат-человек)</i>
<i>SNR</i>	6,89	7,42	12,01	16,02
<i>IF</i>	0,80	0,82	0,94	0,98

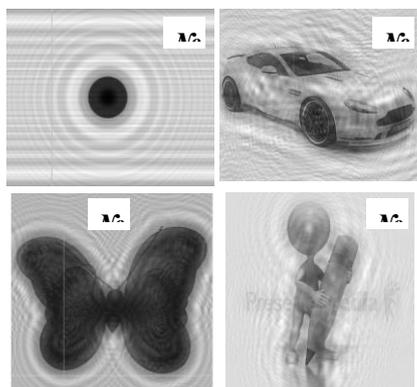


Рис. 1. Изображения тестовых объектов

Между значениями SNR и IF существует корреляция. Сравнение изображений между собой показывает, что лучшее качество имеет изображение тестового образца № 4 «плакат-человек». Однако все изображения объектов оказываются не совсем похожими на объект. Этот эффект зашумления обусловлен наложением 0-го порядка дифракции, (-1)-го порядка (формирует мнимое изображение) и (+1)-го порядка (формирует действительное изображение, всегда возникает в 2D голограммах [1, с. 30]) вследствие одноосного расположения источника опорной волны и объекта, то есть интерференционной схемы голограммы Габора. Полученный эффект наложения согласуется с результатами в работе [3, с. 25-26] и является неустранимым для 2D голограмм при $\theta = 0^0$. При этом существуют методы удаления действительного изображения специальными алгоритмами [2, с. 44].

Отношение сигнал-шум SNR при разных значениях длины волны λ опорного пучка для изображений четырёх тестовых образцов и фиксированном расстоянии $d = 45$ мм между изображением и голограммой представлено в таблице 2 (в скобках указаны

относительные погрешности по сравнению со значениями SNR при длине волны 550 нм) и на рисунке 2 (сплошная линия – это наиболее соответствующая линия тренда в виде полинома 2-го порядка).

Таблица 2

длина волны λ , мкм	0,40	0,45	0,50	0,55	0,60	0,65	0,70	0,75	0,77
SNR №1 (точка)	7,5 (9%)	6,9	6,9 (4%)	6,9	7,1 (2%)	7,5 (9%)	3,5 (49%)	7,2 (4%)	5,1 (26%)
SNR №2 (бабочка)	7,4	7,6 (3%)	7,7 (4%)	7,4	7,4	5,0 (33%)	4,4 (40%)	7,7 (4%)	4,0 (46%)
SNR №3 (автомобиль)	11,8 (2%)	12,0	12,3 (2%)	12,0	12,5 (4%)	9,6 (20%)	8,0 (33%)	11,9 (1%)	6,5 (46%)
SNR №4 (пл.-человек)	15,9 (1%)	16,0	16,0	16,0	16,2 (1%)	13,7 (14%)	12,3 (23%)	15,3 (5%)	12,5 (22%)

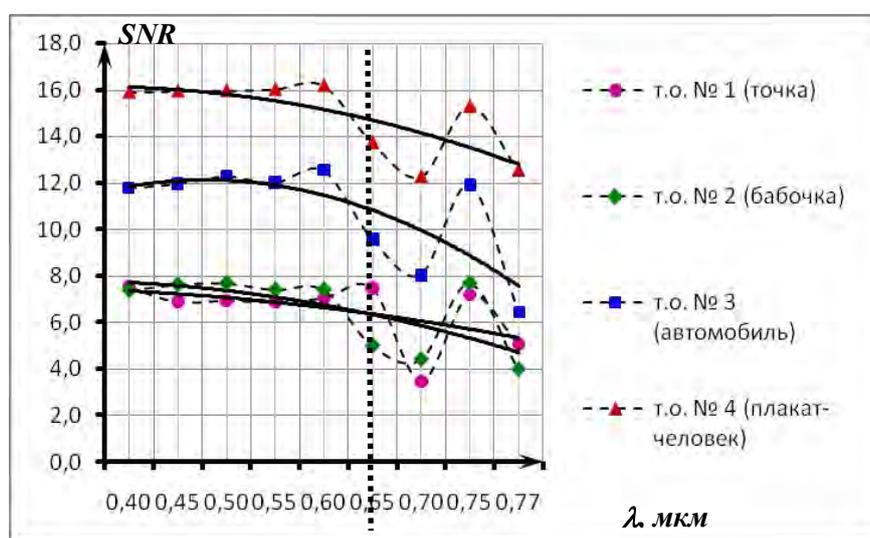


Рис. 2

Оценка качества по критерию «верность изображения» IF при разных λ опорного пучка и одинаковом $d = 45 \text{ мм}$ представлена в таблице 3 (в скобках – относительные погрешности по сравнению с IF при 550 нм) и на рисунке 3 (наиболее подходящая линия тренда – полином 2-го порядка).

Таблица 3

λ , мкм	0,40	0,45	0,50	0,55	0,60	0,65	0,70	0,75	0,77
IF №1	0,824 (3,6%)	0,796 (0,1%)	0,798 (0,4%)	0,795	0,803 (1,0%)	0,822 (3,4%)	0,553 (30,4%)	0,809 (1,8%)	0,691 (13,1%)
IF №2	0,818 (0,1%)	0,828 (1,1%)	0,828 (1,3%)	0,819	0,819 (0,0%)	0,684 (16,5%)	0,639 (22,0%)	0,830 (1,3%)	0,600 (26,7%)
IF №3	0,934 (0,3%)	0,936 (0,1%)	0,936 (0,4%)	0,937	0,944 (0,7%)	0,890 (5,0%)	0,843 (10,0%)	0,936 (0,1%)	0,775 (17,3%)
IF №4	0,974 (0,1%)	0,975 (0,0%)	0,975 (0,0%)	0,975	0,976 (0,1%)	0,958 (1,7%)	0,941 (3,5%)	0,970 (0,5%)	0,944 (3,2%)

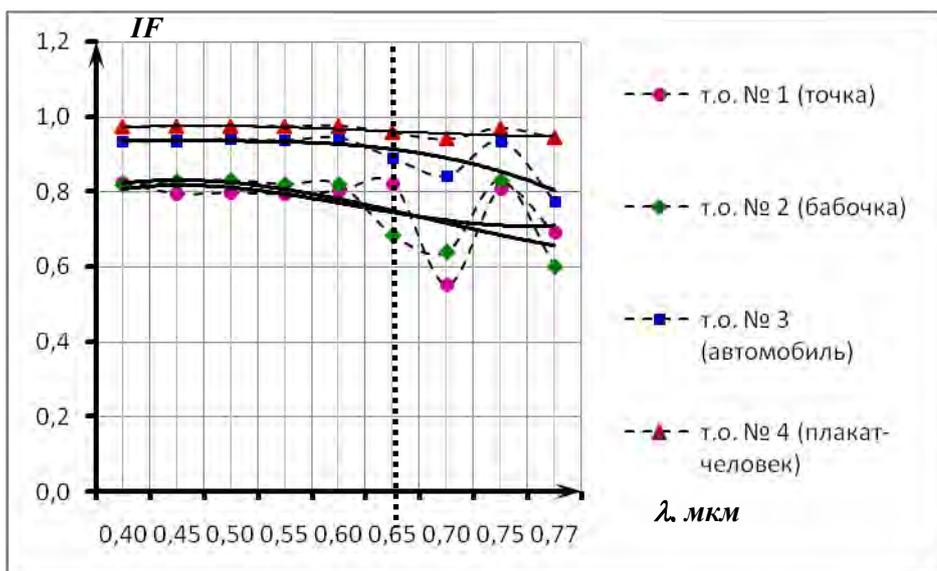


Рис. 3

Видно, что в диапазоне от 400 до 600 нм величины SNR и IF практически не зависят от λ . Этот результат ожидаемый, поскольку в методе свёртки размеры пикселей в голограмме и восстановленном изображении совпадают, то есть не зависят от λ опорного пучка.

Однако оказалось, что при длинноволновых (красных) опорных пучках возникли существенные флуктуации значений SNR и IF (на рис. 2 и 3 – области справа от пунктирной линии). Визуально ухудшение качества изображения проявляется в резком изменении его тона (яркости). Такое флуктуационное изменение изображения предлагаем дополнительно оценить величиной среднего значения RGB . Отношения среднего значения RGB при 550 нм к среднему значению RGB при 700, 750 и 770 нм представлены в таблице 4.

Видно, что наибольшие изменения RGB (или яркости), а также SNR и IF возникают для всех изображений на одних и тех же частотах (длинах волн) опорного пучка, то есть характерны для изображений любого размера, хотя для менее обтекаемой формы изображения (№ 4) их величина меньше, чем для обтекаемых фигур (№ 1, 2, 3).

Таблица 4

Отношение длин волн	RGB_{550}/RGB_{700}	RGB_{550}/RGB_{750}	RGB_{550}/RGB_{770}
т. о. № 1 (точка)	1,9	1,0	1,9
т. о. № 2 (бабочка)	1,6	1,0	1,80
т. о. № 3 (автомобиль)	1,6	1,0	1,9
т. о. № 4 (пл.-человек)	1,2	1,0	1,2

Причиной таких отклонений является, видимо, существенное перекрытие дифракционных максимумов 0-го и 1, -1 порядков. Возможным способом борьбы с перекрытием дифракционных порядков служит использование опорной волны с большим углом наклона θ . Другим возможным способом подавления влияния 0-го порядка может

быть применение частотного фильтра на основе фрагментирования голограммы [2, с. 63].

Литература

1. Перминов А.Ф., Файзрахманова И.С. Прикладная голография: Курс лекций. Учеб. пособие. Пермь: ПНИПУ, 2017.
2. Кузнецов Р. А. разработка системы неразрушающего контроля на основе методов цифровой голографической интерферометрии: дис. канд. техн. наук. Новосибирск: НГТУ, 2013.
3. Бульба А. В. Введение в цифровую голографию. Петрозаводск: ПГУ, 2009.

T.B. Solodchenkova, R.L. Gaychikov
Smolensk State University

SELECTION CRITERIA FOR IMAGE QUALITY ANALYSIS IN THE METHOD OF DIGITAL HOLOGRAPHY

Keywords: *digital holography, 2D hologram, fotometrica, the pixel of the object wave, object wave, Fourier transformation, convolution method, the pixel resolution of the matrix.*

Abstract. *The article presents calculations in MathCad of four two-dimensional images of objects of different shapes by digital holography, the analysis of images depending on the wavelength of the reference beam. The signal / noise ratio and image fidelity were selected as criteria along with visual observation. According to the results of the study, it is proposed to additionally assess the quality of digital 2D images by the average RGB value, allowing to choose a more optimal range of wavelengths of the reference wave.*

А.Г. Суханова
Смоленский государственный университет

УДК 519.86

МОДЕЛИРОВАНИЕ НЕЙРОННОЙ СЕТИ ДЛЯ ОСУЩЕСТВЛЕНИЯ КЛАССИФИКАЦИИ ИНВЕСТИЦИОННЫХ ПОРТФЕЛЕЙ В СИСТЕМЕ МАТНСАД

Ключевые слова: *нейронная сеть, персептрон, классификация, инвестиционный портфель, экспертная система нечеткого вывода.*

В работе использованы возможности системы Mathcad для моделирования двухслойной нейронной сети прямого распространения с целью осуществления классификации инвестиционных портфелей. Классификация осуществлена в соответствии с правилами нечеткого вывода. Показана работа нейронной сети на входном сигнале, соответствующем «неудовлетворительному» портфелю.

Инвестиционный портфель представляет собой [1]: совокупность всех акций, торгуемых на данном рынке, взвешенных по объему их рыночной капитализации, а также совокупность акций, входящих в портфель для расчета рыночного индекса.

Классификация портфелей была осуществлена с использованием нейро-сетевого подхода. Ранее автором [2] была разработана экспертная система нечеткого вывода для классификации инвестиционных портфелей. В настоящее время теория нечетких множеств широко используется в различных отраслях науки [3, 4].

Значения входных переменных для работы нейронной сети – нечетких лингвистических переменных (ЛП): «Доходность портфеля» и «Риск портфеля» выбраны в соответствии с таблицей 1.

Таблица 1

Множество значений ЛП «Доходность портфеля» и «Риск портфеля»

№ п/п	Название ЛП	Название термина ЛП	Множество значений ЛП
1	Доходность портфеля	Низкая	<0,4
		Средняя	0,4 – 0,8
		Высокая	>0,8
2	Риск портфеля	Минимальный	<0,5
		Повышенный	0,5 – 1
		Критический	>1

В качестве выходной переменной нейронной сети выбрана нечеткая лингвистическая переменная «Портфель ценных бумаг». Термножество значений выходной ЛП «Портфель ценных бумаг» задано в соответствии с правилами нечеткого вывода, представленными в [2].

Классификация инвестиционных портфелей была осуществлена в модуле Automated Neural Networks программы STATISTICA.

Среди всех конфигураций была выбрана сеть MLP 2-6-4, представляющая собой двухслойный персептрон, имеющий 2 нейрона на входном слое, 4 нейрона на выходном слое и 6 скрытых нейронов. Активационная функция скрытого слоя – гиперболический тангенс, выходного слоя – логистическая функция. Сеть имеет производительность по обучающей выборке 98,63% (рис. 1).

Summary of active networks (Spreadsheet11)								
Index	Net. name	Training perf.	Test perf.	Validation perf.	Training algorithm	Error function	Hidden activation	Output activation
54	MLP 2-6-4	98,63014	100,0000	100,0000	BFGS 21	SOS	Tanh	Logistic

Рис. 1. Характеристики нейронной сети MLP 2-6-4

За основу моделирования нейронной сети в системе Mathcad были взяты две матрицы весовых коэффициентов IW – для скрытого слоя, LW – для выходного слоя и два вектора смещений b_1 – для скрытого слоя нейронов, b_2 – для выходного слоя нейронов двухслойной нейросети MLP 2-6-4. Матрица полученных весовых коэффициентов для скрытого слоя в программе STATISTICA представлена на рисунке 2.

Network weights (Spreadsheet11)		
Weight ID	Connections	Weight values
	54.MLP 2-6-4	
1	Доходность --> hidden neuron 1	-13,4188
2	Риск --> hidden neuron 1	0,2443
3	Доходность --> hidden neuron 2	-1,6760
4	Риск --> hidden neuron 2	5,8552
5	Доходность --> hidden neuron 3	-12,1177
6	Риск --> hidden neuron 3	4,8688
7	Доходность --> hidden neuron 4	7,5376
8	Риск --> hidden neuron 4	4,7771
9	Доходность --> hidden neuron 5	-1,5392
10	Риск --> hidden neuron 5	-4,2881
11	Доходность --> hidden neuron 6	12,8396
12	Риск --> hidden neuron 6	-4,5785
13	input bias --> hidden neuron 1	8,7055

Рис. 2. Матрица весовых коэффициентов для скрытого слоя нейронной сети MLP 2-6-4

В системе Mathcad была построена математическая модель двухслойной нейронной сети прямого распространения MLP 2-6-4, в соответствии с (1) – (4) [5]:

$$V_1 = A \cdot IW + b_1, \quad (1)$$

где A – вектор входа из 2 элементов, IW – матрица весовых коэффициентов связей нейронов первого (скрытого) слоя со входами, b_1 – вектор смещений нейронов первого слоя.

$$B = f_1(V_1), \quad (2)$$

где $f_1(x)$ – функция активации (гиперболический тангенс) для нейронов первого слоя сети.

$$V_2 = B \cdot LW + b_2, \quad (3)$$

где LW – матрица весовых коэффициентов связей нейронов второго (выходного) слоя с нейронами первого слоя, b_2 – вектор смещений нейронов выходного слоя.

$$Y = f_2(V_2), \quad (4)$$

где $f_2(x)$ – логистическая функция активации для нейронов второго слоя сети, Y – выходной вектор сигналов нейронной сети из 4 элементов.

Математическая модель (1) – (4) может быть представлена в виде (5):

$$Y = f_2(f_1(A \cdot IW + b_1) \cdot LW + b_2), \quad (5)$$

Выходом сети могут быть выходные сигналы следующих 4 видов:
– (1 0 0 0) – «неудовлетворительный» портфель»;

- (0 1 0 0) – «удовлетворительный» портфель»;
- (0 0 1 0) – «хороший» портфель»;
- (0 0 0 1) – «очень хороший» портфель».

На рисунке 3 представлено задание матриц весовых коэффициентов IW и LW , векторов смещения b_1 и b_2 и функций активации f_1 и f_2 для двух слоев нейронной сети в системе Mathcad.

$$\begin{aligned}
 IW &:= \begin{pmatrix} -13.4188 & 0.2443 \\ -1.6760 & 5.8552 \\ -12.1177 & 4.8688 \\ 7.5376 & 4.7771 \\ -1.5392 & -4.2881 \\ 12.8396 & -4.5785 \end{pmatrix} & LW &:= \begin{pmatrix} 2.0192 & 5.0195 & 15.0894 & 7.6272 & -4.2462 & -10.3624 \\ 8.8982 & -1.7947 & -2.5325 & -9.1905 & -4.9240 & 9.9137 \\ -6.5841 & 3.9839 & 2.1371 & -2.9655 & 0.8212 & -1.0898 \\ -5.2673 & -5.2824 & -5.1428 & -1.8181 & 2.4440 & 2.7955 \end{pmatrix} \\
 b_1 &:= \begin{pmatrix} 8.7055 \\ -1.3642 \\ -5.5507 \\ -2.6360 \\ 4.3850 \\ -1.7741 \end{pmatrix} & b_2 &:= \begin{pmatrix} 9.7138 \\ -4.5760 \\ -3.9989 \\ -11.5662 \end{pmatrix} & f_1(x) &:= \frac{\exp(2 \cdot x) - 1}{\exp(2 \cdot x) + 1} & f_2(x) &:= \frac{1}{\exp(-x) + 1}
 \end{aligned}$$

Рис. 3. Задание матриц весовых коэффициентов, векторов смещения и функций активации для каждого слоя нейронной сети в системе Mathcad

На рисунке 4 представлено задание вектора аргументов для функций активации нейронов, а также расчет выходного сигнала первого (скрытого) слоя нейронов для заданного входного вектора, соответствующего «неудовлетворительному» портфелю.

$$\underline{A} := \begin{pmatrix} 0.1 \\ 0.79 \end{pmatrix} \text{ вектор входа } A \text{ из двух элементов, соответствующий "неудовлетворительному" портфелю}$$

$$v_1 := A^T \cdot IW^T + b_1^T = (7.557 \quad 3.094 \quad -2.916 \quad 1.892 \quad 0.843 \quad -4.107)$$

Расчет выходного сигнала B первого (скрытого) слоя нейронов:

$$B := \begin{pmatrix} f_1\left(\sum v_1^{(0)}\right) \\ f_1\left(\sum v_1^{(1)}\right) \\ f_1\left(\sum v_1^{(2)}\right) \\ f_1\left(\sum v_1^{(3)}\right) \\ f_1\left(\sum v_1^{(4)}\right) \\ f_1\left(\sum v_1^{(5)}\right) \end{pmatrix} \quad B = \begin{pmatrix} 1 \\ 0.996 \\ -0.994 \\ 0.956 \\ 0.688 \\ -0.999 \end{pmatrix}$$

Рис. 4. Расчет выходного сигнала первого (скрытого) слоя в системе Mathcad

Расчет выходного сигнала второго (выходного) слоя нейронов представлен на рисунке 5.

Расчет вектора аргументов для функций активации нейронов второго слоя:

$$v_2 := B^T \cdot LW^T + b_2^T = (16.456 \quad -17.023 \quad -9.92 \quad -19.832)$$

Расчет выходного сигнала Y второго (выходного) слоя нейронов:

$$Y := \begin{pmatrix} f_2\left(\sum v_2^{(0)}\right) \\ f_2\left(\sum v_2^{(1)}\right) \\ f_2\left(\sum v_2^{(2)}\right) \\ f_2\left(\sum v_2^{(3)}\right) \end{pmatrix} \quad Y = \begin{pmatrix} 1 \\ 4.044 \times 10^{-8} \\ 4.919 \times 10^{-5} \\ 2.438 \times 10^{-9} \end{pmatrix}$$

Рис. 5. Расчет выходного сигнала второго (выходного) слоя нейронов в системе Mathcad

Из рисунка 5 видно, что смоделированная средствами системы Mathcad двухслойная нейронная сеть при подаче на вход вектора, соответствующего «неудовлетворительному» портфелю, сгенерировала правильный выходной сигнал.

Литература

1. Ковалев В. В. Финансовый менеджмент: теория и практика. 3-е изд., перераб. и доп. М.: Проспект, 2013. – 1104 с.
2. Суханова А. Г. Нечетко-множественный подход к выбору инвестиционного портфеля в системе Mathcad // Системы компьютерной математики и их приложения: материалы XVI Международной научной конференции. Смоленск: Изд-во СмолГУ. 2015. Вып. 16. С. 38 – 41.
3. Седова Н.А., Седов В.А. Метод расхождения морских судов в зоне чрезмерного сближения на основе нейронечетких технологий // Известия Юго-Западного государственного университета. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение. 2018. Т. 8, № 4(29). С. 53–62.
4. Седова Н.А., Седов В.А. Управление умным домом с использованием нечеткой логики // ЭНЕРГЕТИКА, ИНФОРМАТИКА, ИННОВАЦИИ-2016 в 3 т. Национальный исследовательский университет «МЭИ», филиал в г. Смоленске. 2016. С. 336 – 339.
5. Леванов Д.Н., Феоктистов Н.А. Особенности использования многослойного персептрона при автоматизированном контроле знаний в электронных учебных курсах // Интернет-журнал «Науковедение». 2014. Вып. 2. С. 121.

A.G. Sukhanova
Smolensk State University

MODELING NEURAL NETWORKS FOR CLASSIFICATION OF INVESTMENT PORTFOLIOS IN MATHCAD

Keywords: *neural network, perceptron, classification, investment portfolio, expert fuzzy inference system.*

Abstract. *The paper uses the capabilities of the Mathcad system to simulate a two-layer neural network of direct distribution for the classification of investment portfolios. Classification is carried out in accordance with the rules of fuzzy inference. The work of the neural network on the input signal corresponding to the "unsatisfactory" portfolio is shown.*

М.И. Тихонова

*Национальный исследовательский университет
Высшая школа экономики, г. Москва*

УДК 519.685.1

ИЗВЛЕЧЕНИЕ ИНФОРМАЦИИ ИЗ РЕЗЮМЕ МЕТОДАМИ АВТОМАТИЧЕСКОЙ ОБРАБОТКИ ТЕКСТА

Ключевые слова: автоматическая обработка текстов, кластеризация, NLP-методы, машинное обучение, векторное представление слов, word embedding, метод k-средних, обучение без учителя, тематическое моделирование.

В статье рассматривается автоматическое извлечение информации из резюме кандидатов при приеме на работу, основанное на методах естественной обработки языка. Предлагается алгоритм, позволяющий трансформировать текстовую информацию в числовые признаки и кластеризовать кандидатов по типажу. Данная информация может быть использована для моделирования с целью построения прогноза об успешности кандидата. Помимо общего описания алгоритма дано краткое изложение экспериментов и результатов кластеризации резюме кандидатов.

1. Введение. В крупных компаниях, где поток новых сотрудников велик, а текучка кадров достаточно высока, актуален вопрос автоматической обработки резюме и прогнозирования пригодности потенциальных кандидатов на вакансию, а также вероятности их скорого оттока. В данном контексте возникает задача автоматической обработки резюме, извлечения информации из текстовых полей и трансформации данных полей в числовые признаки для дальнейшего использования в прогнозировании. В статье предлагается алгоритм, основанный на автоматических методах *обработки естественного языка* (сокращенно *NLP-методах* от английского *natural language processing*), по эффективному извлечению информации из полей резюме

кандидатов, таких как «*О себе*», «*Отрасль предыдущего места работы*», «*Предыдущая должность*» и других и их кластеризация. В результате работы алгоритма каждый кандидат на вакансию относится к одному из типов-кластеров, информация о его типе в дальнейшем может быть использована для моделирования и прогнозирования.

2. Описание алгоритма. Извлечение информации из текста резюме и кластеризация кандидатов осуществляется в несколько этапов.

Шаг 1. Построение векторного представления слов.

Шаг 2. Построение *tf-idf* индекса.

Шаг 3. Построение векторных представлений для полей из резюме.

Шаг 4. Подбор оптимального числа кластеров для кластеризации резюме .

Шаг 5. Итоговая кластеризация резюме.

Далее будет подробно рассмотрен каждый из этапов.

Шаг 1. Построение векторного представления слов

На первом этапе происходит векторное построение слов. То есть каждому слову, встречающемуся в резюме какого-либо кандидата, сопоставляется вектор в линейном пространстве R^n . При этом вектора сопоставляются так, чтобы близким словам соответствовали близкие векторы, а определенным семантическим и синтаксическим свойствам слов отвечали определенные направления в линейном пространстве. Это позволяет эффективно работать с векторами через стандартные операции векторного пространства. Например, производить над ними стандартные математические операции. Приведем пример, ставший уже классическим:

$$v(\text{король}) - v(\text{мужчина}) + v(\text{женщина}) = v(\text{королева})$$

где $v(\text{слово})$ означает вектор, соответствующий данному слову.

Важным свойством векторных представлений слов является то, что семантическую близость между словами можно установить с помощью измерения расстояния между векторами.

Методы векторных представлений слов на данный момент являются хорошо изученной областью. Существует ряд алгоритмов, которые позволяют строить векторные представления слов для коллекции текстов. К наиболее распространенным относятся *Word2Vec* [1-2], *FastText* [3-4], *GloVe* [5] и др. Более того, готовые векторные представления слов, построенные на больших коллекциях текстовых документов, находятся в открытом доступе.

Шаг 2. Построение *tf-idf* индекса

На следующем этапе происходит построение *tf-idf* индекса (от английского *term frequency – inverse document frequency*), отражающего меру важности слова в контексте документа, являющегося частью

коллекции документов. При этом каждому слову присваивается вес, пропорциональный его частоте вхождения в документ и обратно пропорциональный частоте его употребления в документах коллекции.

$tf(t, d)$ определяется как отношение числа вхождений слова t к общему числу слов документа d :

$$tf(t, d) = \frac{n_t}{\sum_k n_k},$$

где n_t – число вхождений слова t в документ.

$idf(t, D)$ определяется как инверсия частоты, с которой слово t встречается в документах коллекции D :

$$idf(t, d) = \log \frac{|D|}{|\{d_i \in D | t \in d_i\}|},$$

где $|M|$ обозначает число элементов в множестве M .

Итоговый вес $tf-idf(t, d, D)$ слова t в документе d , принадлежащего коллекции документов D рассчитывается как произведение $tf(t, d)$ на $idf(t, D)$:

$$tf-idf(t, d, D) = tf(t, d) \times idf(t, D).$$

В задаче извлечения информации из резюме в качестве документа выступает текстовое поле из конкретного резюме, а в качестве коллекции документов – множество всех резюме кандидатов. Таким образом, каждому слову t в каждом резюме d сопоставляется его $tf-idf(t, d)$ вес.

Шаг 3. Построение векторных представлений для полей из резюме

Далее происходит построение векторного представления для текстового поля из резюме (например, поле «О себе»). Для этого векторные представления слов, содержащихся в данном поле резюме для данного кандидата, суммируются с их $tf-idf$ весами:

$$v(d) = \sum_{w \in D} tf-idf(w, d) \times v(w)$$

где d – текстовое поле в резюме кандидата;

w – слово в резюме d ;

$v(d)$, $v(w)$ – векторные представления для резюме d и для слова w соответственно;

$tf-idf(w, d)$ – вес слова w в резюме d .

После такого преобразования каждому резюме сопоставляется вектор той же размерности, что и размерность исходного векторного представления слов.

Шаг 4. Подбор оптимального числа кластеров для кластеризации резюме

Полученные на предыдущем шаге векторные представления затем кластеризируются по типам. В качестве алгоритма кластеризации был выбран метод *k-средних* (также известный как *k-means*), предложенный

в 1950-х годах Гуго Штейнгаузом [6]. Работа алгоритма основана на минимизации отклонения точек от центроидов (центров кластеров). Таким образом алгоритм k-means решает следующую задачу минимизации *индекса инерции* $J(C)$:

$$J(C) = \sum_{k=1}^K \sum_{i \in C_k} \|x_i - \mu_k\|^2 \rightarrow \min_C,$$

где K – количество кластеров;

$J(C)$ – индекс инерции для кластеризации C с числом кластеров K ;

C_i – полученные кластеры;

μ_i – центроиды, центры масс векторов $x_i \in C$.

Метод k -средних хорошо работает в случае векторных представлений, поскольку он основан на кластеризации точек по расстоянию между ними. Однако особенностью данного алгоритма является то, что число кластеров k является входным параметром, которое необходимо определить заранее.

Для выбора оптимального числа кластеров в задаче распределения резюме по типам предлагается использовать следующий эвристический критерий. Рассмотреть кластеризации с числом кластеров из определенного диапазона. В качестве оптимального числа кластеров выбрать k , при котором скорость убывания функционала инерции $J(C)$ достигает минимума по k . Более формально это можно записать как минимизацию следующей величины:

$$D(k) = \frac{|J(C_k) - J(C_{k+1})|}{|J(C_{k-1}) - J(C_k)|} \rightarrow \min_k.$$

Шаг 5. Итоговая кластеризация резюме

После того как выбрано оптимальное число кластеров k , происходит итоговая кластеризация векторных представлений текстовых полей из резюме, полученных на шаге 3 методом k -средних. В результате каждый кандидат определяется к одному из k типов. Данный тип может быть в дальнейшем использован для моделирования. Например, он может быть применен в задачах предсказания успешности кандидата на данном месте работы или вероятности его скорого ухода с данной позиции.

3. Эксперименты. Для тестирования данного алгоритма была взята выборка из 5018 кандидатов на вакансию «Консультанта по банковским продуктам» в различные отделения Банка. В эксперименте рассматривалась кластеризация для поля «*Отрасль предыдущего места работы*» из резюме.

Для построения векторного представления слов были использованы готовые векторные представления, построенные алгоритмом *FastText* с использованием корпуса текстов русской Википедии, размерности 300 (векторные представления были взяты с <https://rusvectors.org/ru/models/>).

В эксперименте рассматривалось число кластеров в диапазоне от 2 до 30. Индекс инерции $J(C)$ представлен на графике 1, а соответствующее ему значение функционала $D(C_k)$ на графике 2.

В качестве оптимального числа кластеров было выбрано значение $k = 27$. Примеры кластеризации отраслей приведены в таблице.

По результатам эксперимента можно сделать вывод, что алгоритм позволяет отнести предыдущую отрасль работы к той или иной сфере. При этом поскольку тестирование проводилось для кандидатов на вакансию консультанта по банковским продуктам, кажется логичным, что банковская область выделилась в отдельный кластер.

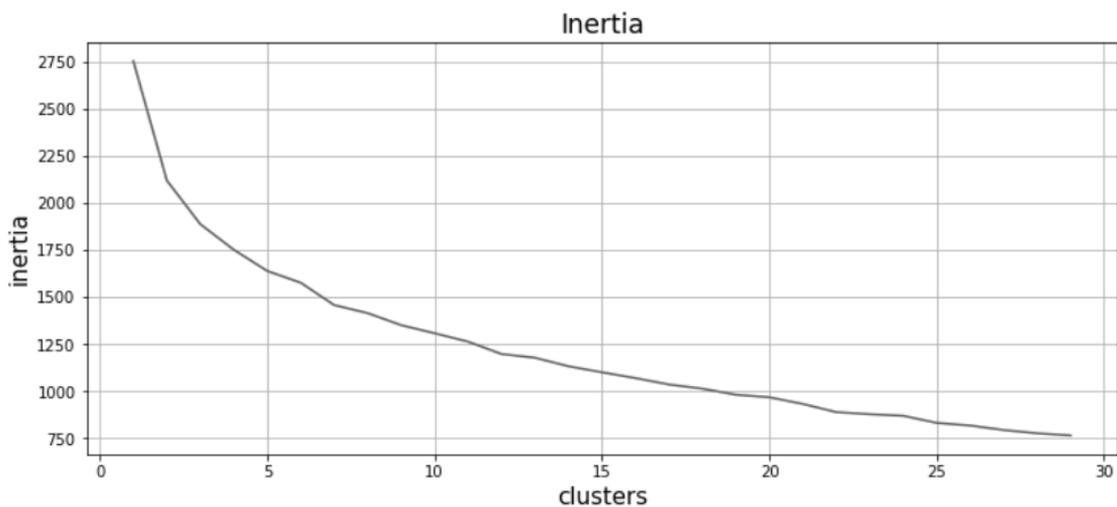


Рис. 1. Индекс инерции $J(C)$

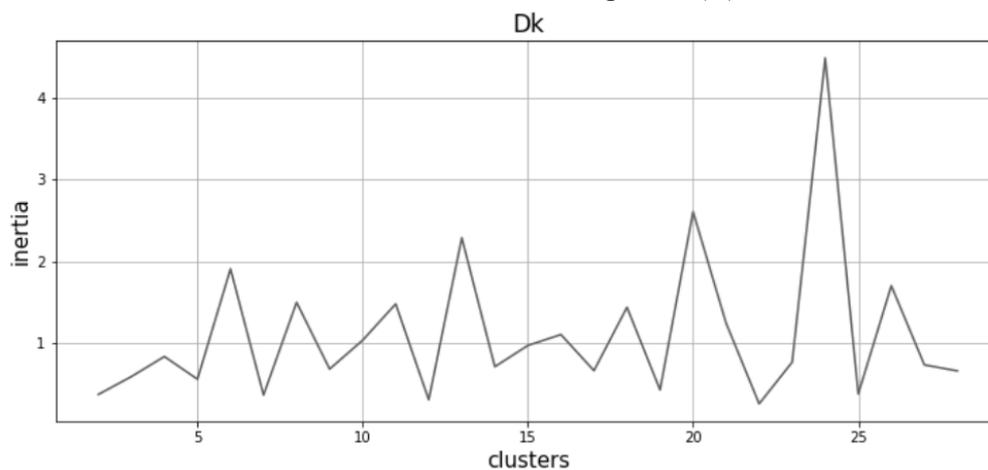


Рис. 2. Значение $D(C)$

Таблица 1
Примеры кластеров

Номер кластера	Примеры отраслей
0	одежда, обувь, аксессуары, текстиль (продвижение, оптовая торговля) одежда, обувь, аксессуары, текстиль (производство) пошив, ремонт одежды и обуви

1	банк
2	финансово-кредитное посредничество консалтинговые услуги услуги по ведению бухгалтерского и налогового учета
3	безалкогольные напитки (продвижение, оптовая торговля) алкогольные напитки (продвижение, оптовая торговля) кондитерские изделия (продвижение, оптовая торговля)
4	строительство жилищное архитектура, проектирование строительство дорожное и инфраструктурное строительство объектов нефтегазовой отрасли

4. Выводы. В статье рассмотрен метод автоматического извлечения информации из текстовых полей резюме с последующей кластеризацией по типам. Полученная кластеризация в дальнейшем может быть использована для моделирования и предсказания успешности кандидата.

Алгоритм был протестирован для кластеризации отраслей предыдущих мест работы для соискателей на вакансию консультанта по банковским продуктам. В данном эксперименте он позволил хорошо кластеризовать отрасли по сферам деятельности.

В дальнейшем планируется тестировать алгоритм для более широкого круга текстовых полей и других вакансий. Также возможно рассматривать другие векторные представления слов, например *GloVe*.

Литература

1. Mikolov T. et al. Efficient estimation of word representations in vector space // arXiv preprint arXiv:1301.3781. 2013.
2. Mikolov T., Yih W., Zweig G. Linguistic regularities in continuous space word representations // Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013. С. 746-751.
3. Enriching word vectors with subword information / P. Bojanowski [et al.] // Transactions of the Association for Computational Linguistics. 2017. Т. 5. С. 135-146.
4. Bag of tricks for efficient text classification / A. Joulin [et al.] // arXiv preprint arXiv:1607.01759. 2016.
5. Pennington J., Socher R., Manning C. Glove: Global vectors for word representation // Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014. С. 1532-1543.
6. Steinhaus H. Sur la division des corps materiels en parties. Bull. Acad. Polon. 1956.

M. I. Tikhonova

National Research University Higher School of Economics, Moscow

AUTOMATIC INFORMATION EXTRACTION FROM CV BASED ON NATURAL LANGUAGE PROCESSING METHODS

Keywords: *natural language processing, clusterization; NLP methods, machine learning, word embedding, k-means, unsupervised learning, topic modeling.*

Abstract. *The paper proposes a new method of automatic information extraction from candidate's CVs. The algorithm based on NLP methods which enables to transform text information to numerical features is proposed. Information obtained via the use of the algorithm may further be used to forecast the successfulness of the candidate. Besides the general method brief description of the experiments is given.*

О.В. Тихонова

МИРЭА - Российский технологический университет

УДК 519.876.5

МЕТОД ТЕСТИРОВАНИЯ ЦИФРОВОЙ СИСТЕМЫ СВЯЗИ С ПОМОЩЬЮ АЛГОРИТМИЧЕСКОГО ПОЛУНАТУРНОГО МОДЕЛИРОВАНИЯ

Ключевые слова: *цифровая система связи, тестирование, математическое моделирование, алгоритмическое полунатурное моделирование*

Для проверки правильности функционирования цифровой системы связи предлагается использовать метод алгоритмического полунатурного моделирования. Проведя серию численных экспериментов и сравнивая их результаты с результатами, зафиксированными в ходе работы реальной системы, можно не только сделать обоснованный вывод о штатной работе системы, но и указать конкретный элемент, ответственный за появление нештатной ситуации.

Введение

На всех этапах жизненного цикла системы цифровой связи от первоначального замысла до серийной эксплуатации важную роль занимает ее постоянное тестирование. На этапе создания цифровой системы связи под тестированием понимается не только проверка правильности функционирования в различных условиях, но и непосредственная отладка ее отдельных элементов. Отметим, что сложные радиотехнические системы функционально сочетают аналоговые элементы и чисто цифровые алгоритмы. Таким образом, задачи тестирования и отладки можно разделить на задачи проверки корректности работы программных средств и подтверждения правильности функционирования аналоговых подсистем.

Метод алгоритмического полунатурного моделирования

На этапе создания радиотехнической системы передачи данных можно проводить ряд экспериментов с элементами системы и непосредственно контролировать их работу. Для контроля используются методы сравнения результатов эксперимента с теоретическими расчетами и с результатами численных экспериментов на математической модели. Предлагается использовать для этого разработанный автором метод алгоритмического полунатурного моделирования, позволяющий сочетать одновременно как реальный эксперимент, так и математическое моделирование системы.

Суть алгоритмического полунатурного моделирования состоит в следующем. В ходе проведения натурального эксперимента фиксируются все процессы на входе и выходе блоков системы (заметим, что любая сложная система представляет собой совокупность отдельных блоков, формирующих ее иерархическую структуру). На последующих этапах проводится ряд численных экспериментов на математической модели (структура модели должна при этом повторять структуру реальной сложной системы). В каждом эксперименте ряд параметров моделирования заменяется на зафиксированные в ходе реального эксперимента данные. Таким образом, происходит как бы замена отдельных элементов математической модели их реальными аналогами. При сравнении результатов численных и реальных экспериментов выявляются те блоки системы, функционирование которых отличается от штатного (симулированного в модели).

В ходе эксплуатации системы роль алгоритмического полунатурного моделирования при тестировании еще выше: в некоторых случаях нет возможности непосредственно проверить все элементы системы из-за их недоступности (например, при связи со спутниковыми или авиационными системами). В этом случае зафиксированная первичная информация позволяет имитировать работу недоступных для наблюдения элементов и проводить оценку их работоспособности.

Многоуровневая модель цифровой системы связи

Для проведения тестирования системы цифровой связи необходима математическая модель, повторяющая структуру системы. Группа студентов и аспирантов кафедры радиосистем передачи информации МИРЭА создала универсальную многоуровневую модель системы передачи данных на основе стандартных представлений о реальном аналоге [1], схема модели представлена на рисунке 1.

В этой модели рассматриваются две подсистемы: подсистема передачи битовой информации и подсистема передачи сигнала; каждую из них можно заменить на статистический аналог.

Модель построена по принципам иерархического многоуровневого моделирования, в частности, ей присущи структурное подобие (при котором структура модели повторяет структуру самой системы), алгоритмическое соответствие (полное совпадение цифровых алгоритмов модели и реальной системы), сочетание имитационных и статистических методов моделирования, блочное построение. Универсальность модели обеспечивается возможностью использования разных типов кодирования, модуляции, способов синхронизации, алгоритмов перемежения. Возможно моделировать как жесткие, так и мягкие решения демодулятора, рассматривая различные типы простых (BPSK) и сложных (QPSK, DPSK) сигналов.

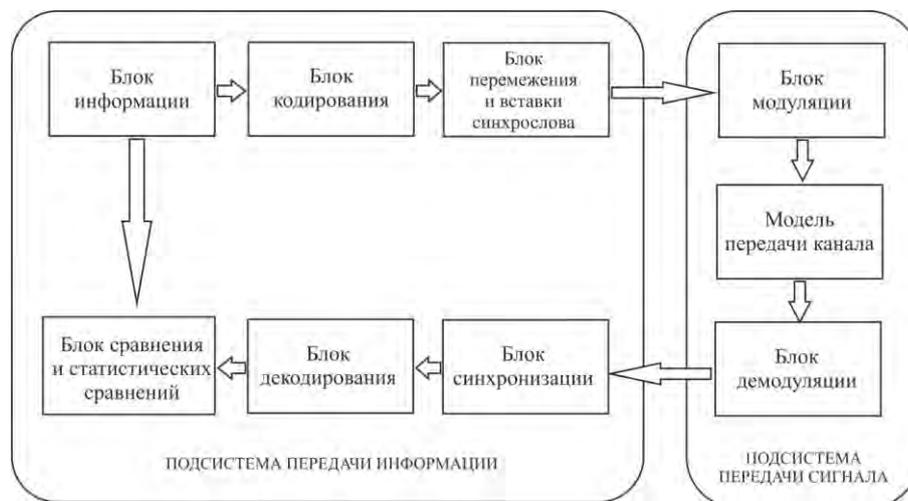


Рис. 1. Схема модели цифровой системы связи

Модель состоит из следующих блоков.

Блок информации предназначен для формирования случайных или наперед заданных информационных слов данных.

Блок кодирования воспроизводит алгоритм кодирования помехоустойчивым кодом, преобразовывая информационные слова в кодовые слова большей длины.

Блок перемежения производит перестановку символов кодовых слов с заданной глубиной перемежения.

Блок вставки последовательности предназначен для формирования маркеров, отмечающих начало и конец кодовых слов при непрерывной передаче данных по каналу связи.

Подсистема передачи отдельно имитирует работу модулятора, канала связи и демодулятора. Имитация канала связи (с учетом возмущающих воздействий) возможна лишь в статистических режимах.

Блок синхронизации выявляет в принимаемом потоке данных отдельные кодовые слова и восстанавливает исходную блочную структуру.

Блок дегермежения восстанавливает исходный порядок следования символов.

Блок декодирования исправляет ошибки передачи символов и восстанавливает исходные информационные слова.

При разработке радиотехнических систем иногда возникают дополнительные задачи, требующие отдельной аппаратуры. Их модельные аналоги также должны быть включены в модель.

Алгоритмическое полунатурное моделирование системы цифровой связи

Структура универсальной математической модели повторяет структуру реальной системы связи. Зафиксируем в ходе эксперимента следующие данные:

- информационную последовательность, предназначенную для тестовой передачи, что эквивалентно выходным параметрам блока информации;
- кодовую последовательность, соответствующую выходу блока кодирования;
- кодовую последовательность после перемежения и вставки синхрослова;
- модулированный высокочастотный сигнал, предназначенный для передачи по линии связи;
- модулированный высокочастотный сигнал на приемной стороне;
- выход алгоритма восстановления блоковой структуры данных, соответствующий выходу алгоритма синхронизации;
- блоки данных после дегермежения;
- декодированную информацию.

Заменяя выходные данные блоков модели их реальными аналогами, можно проверять функционирование отдельных подсистем и блоков, а также оценивать условия передачи сигнала. На рисунке 1 любая связь между блоками может быть заменена на данные, записанные в ходе натурального эксперимента. Например, заменяя выходные данные блока информации модели на реальные и сравнивая результаты на выходе блока кодирования и их реальные аналоги, можно сделать вывод о корректности работы канального кодера. Сравнивая выходные данные демодулятора из реального эксперимента и полученные с помощью моделирования при подаче на вход модели демодулятора записанного реального сигнала, можно проверить работу системы восстановления низкочастотного сигнала.

Здесь следует отметить, что совпадение и несовпадение реальных и модельных данных оценивается с помощью разных критериев в зависимости от физической природы исследуемого блока. Так, информация на выходе цифровых элементов системы может считаться совпадающей, если есть соответствие в каждом бите данных. Это

позволяет сделать вывод о корректной работе цифровых алгоритмов. Тогда как совпадения на входе приемной части системы и блока демодуляции должны проверяться на основе статистических характеристик: сигналы совпадают, если совпадают их статистические характеристики (во временной и спектральной областях), либо принимается решение, что оба варианта данных удовлетворяют гипотезе об их принадлежности к одной генеральной совокупности. Так, отличие в спектрах полученных в реальном эксперименте и модели сигналов может сигнализировать о методических неточностях в построении системы связи, например, в некорректном учете доплеровского смещения частоты принимаемого сигнала.

Отдельно следует проверять соответствие итоговых результатов работы реальной системы и ее математической модели. Основным критерием правильности работы системы цифровой передачи данных является вероятность передачи неверного символа, и после серии натурных экспериментов можно сделать обоснованную оценку этой величины и сравнить с модельной. Однако можно осуществить и более глубокую проверку. Так, если помехи в канале связи носят случайный характер и независимы, то события передачи неверного символа также являются независимыми и случайными. Интервалы времени между ними как случайные величины будут иметь экспоненциальное распределение. При наличии в канале связи регулярных воздействий это утверждение неверно. Однако если сравнить реальные результаты и результаты моделирования при имитации такого типа регулярных воздействий на предмет соответствия их законов распределения (речь идет о случайных величинах интервалов между нарушениями передачи), то можно определить правильность работы с большей точностью.

Последовательное тестирование элементов системы

Тестирование системы цифровой связи призвано подтвердить или опровергнуть вывод о правильном ее функционировании. Для получения оценки вероятности битовой ошибки при передаче данных (а эта вероятность может находиться в диапазоне очень малых чисел) требуется проанализировать большой объем информации.

Здесь можно воспользоваться автоматизированной системой диагностики работы системы. Суть ее состоит в сравнении результатов эксперимента и математической модели при последовательной замене выходных данных блоков модели записями реальных данных.

На первом этапе проверяется совпадение итоговых данных о работе системы связи путем сравнения вероятностей появления битовых ошибок и интервалов между ними с модельными. В случае расхождений важно определить, нет ли в записях о работе системы какого-либо особенного временного интервала, выпадающего из общей генеральной совокупности. При нахождении такого интервала следует

сосредоточиться на анализе работы системы именно в этот период времени, последовательно тестируя блоки подсистемы передачи информации (на передающей и приемной сторонах), подсистемы передачи сигнала (модуляцию, канал передачи и демодуляцию), а также оценить уровень помех в системе (как разность сигналов на входе приемника и на выходе передатчика). При наличии отклонений в работе блоков можно приступать к корректировке ошибок, а в случае наличия выявленных специфических помех следует ввести их имитацию в блок модели и провести повторное тестирование системы. Схема метода тестирования приведена на рисунке 2.



Рис. 2. Схема тестирования цифровой системы связи

В качестве примера использования метода алгоритмического полунатурного моделирования приведем результаты проверки алгоритмов определения вида сигнала [2]. В этом случае зафиксированные в ходе эксперимента сигналы на входе приемной части подавались в модель алгоритма определения вида сигнала и в реальную аппаратуру, выполняющую ту же функцию. Результаты проверки показали соответствие, что позволило сделать вывод о работоспособности разработанной аппаратуры.

Заключение

Автоматизация некоторых процессов оценки работы системы позволит сократить время на отладку и настройку системы в период ее создания, а также проводить периодическое тестирование системы во время ее эксплуатации. При этом даже не требуется останавливать работу системы и организовывать отдельный режим функционального контроля, достаточно только фиксировать параметры работы ее отдельных блоков.

Глубина и эффективность анализа зависит от объема записанной в ходе натурного эксперимента информации, но развитие элементов

памяти и аналого-цифровых преобразователей делает эту задачу решаемой. И конечно необходимо перед началом исследований провести полную верификацию математической модели системы цифровой связи. Отметим, что проверку модели также целесообразно проводить методом алгоритмического полунатурного моделирования.

Литература

1. Скляр Б. Цифровая связь. Теоретические основы и практическое применение. Изд. 2-е: пер. с англ. - М: Изд. дом «Вильямс», 2003. 1104 с.

2. Викулов В.М., Лопатко К.О., Тихонова О.В., Тронза С.Н. Алгоритм определения типа высокочастотного сигнала цифровой связи / В.М. Викулов [и др.] // Системы компьютерной математики и их приложения. материалы XIX Международной научной конференции. Смоленск, 2018. Вып. 19, С. 14-22

О. Tikhonova

MIREA- Russian Technological University

THE METHOD OF CONTROLLING DIGITAL COMMUNICATION SYSTEM WITH ALGORITHMIC SEMI-NATURAL SIMULATION

Keywords: *digital communication system, controlling, mathematical simulation, algorithmic semi-natural simulation.*

Abstract. *To verify the correct functioning of the digital communication system the method of algorithmic semi-natural simulation of the digital communication system is proposed. A series of the numerical experiments was conducted and their results were compared with the results recorded in the course of the real system. Thus a conclusion the regulation system is correct can be made. Moreover, a specific element responsible for the occurrence of an emergency can be showed.*

Хоанг Ван Зунг, А.А. Парамонов

МИРЭА – Российский технологический университет

УДК 621.391

ЭФФЕКТИВНОЕ ИСПОЛЬЗОВАНИЕ ЧАСТОТНО- ЭНЕРГЕТИЧЕСКОГО РЕСУРСА В СИСТЕМАХ ПЕРЕДАЧИ ИНФОРМАЦИИ С ПСЕВДОСЛУЧАЙНОЙ ПЕРЕСТРОЙКОЙ РАБОЧЕЙ ЧАСТОТЫ ПРИ НИЗКОЙ СКОРОСТИ ПЕРЕДАЧИ

Ключевые слова: *псевдослучайная перестройка рабочей частоты, преднамеренная помеха, помехозащищенность.*

Рассматриваются вопросы эффективного использования частотно-энергетического ресурса радиолинии в низкоскоростных системах передачи информации с псевдослучайной перестройкой рабочей частоты при воздействии шумовой помехи в части полосы частот.

Эффективным способом повышения помехозащищенности радиолинии в условиях воздействия преднамеренных помех признано применение псевдослучайной перестройки рабочей частоты (ППРЧ) по псевдослучайному, не известному противнику закону [1; 2]. В этом случае противник вынужден выставлять широкополосную заградительную помеху, что для него энергетически невыгодно. При низкой скорости передачи длительность передаваемого символа может оказаться столь большой, что противник успеет поставить на частоте передаваемого символа помеху. Чтобы избежать такой ситуации, целесообразно применение внутрисимвольной ППРЧ, при которой каждый передаваемый информационный символ делится по времени на L субсимволов, как это показано на рисунке 1 (на рисунке $L = 7$).

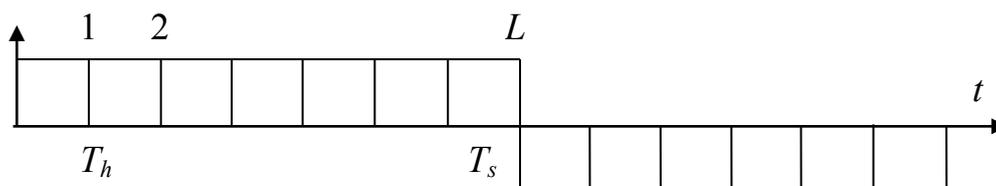


Рис. 1. Разбиение символа на субсимволы

Субсимволы передаются методом ППРЧ, при этом помехи на частотах передачи субсимволов оказываются независимыми. Если длительность символа составляет T_s , то длительность субсимвола (или длительность частотного скачка) составляет $T_h = T_s / L$.

Заметим, что рассматриваемый режим передачи обладает особенностью: группы из L субсимволов, относящиеся к одному передаваемому символу, переносят одну и ту же информацию о передаваемом символе. Здесь усматривается очевидное сходство с разнесенным приемом, поэтому при выборе способов принятия решения о символе по принятым субсимволам можно воспользоваться теорией разнесенного приема. В данной работе будем рассматривать мажоритарный алгоритм вынесения решения о двоичном символе по принятым решениям о субсимволах. Смысл алгоритма заключается в следующем: по всем L субсимволам, относящимся к одному передаваемому символу, принимаются решения, а окончательное решение о передававшемся символе выносится «по большинству голосов».

Для определенности примем, что передача информации в системе производится двоичными сигналами ЧМ, прием – оптимальный некогерентный. Будем считать, что энергия сигнала на длительности передаваемого символа T_s составляет E , а односторонняя спектральная плотность мощности теплового шума – T_0 . Противник, обладая ограниченным энергетическим ресурсом, выставляет помеху не на всех рабочих частотах, а только на ρ -ой их части. Односторонняя спектральная плотность мощности преднамеренной помехи – T_{II} .

Обозначим через M число используемых в системе с ППРЧ частот. Ширину спектра сигнала на каждой из M рабочих частот можно принять равной $2/T_s$. Таким образом, получаем связь между спектральной плотностью мощности преднамеренной помехи и мощностью преднамеренной помехи в точке приема:

$$N_{II} = P_{II} T_s / (2\rho M). \quad (1)$$

При вычислении средней вероятности ошибки приема символа необходимо учесть важное обстоятельство: вероятность ошибочного приема субсимвола существенным образом зависит от того, присутствует ли на частоте передачи субсимвола преднамеренная помеха.

Рассмотрим простой пример: $L = 3$.

Введем величины α_1, α_2 и α_3 – индикаторы наличия преднамеренной помехи на частотах передачи субсимволов. Если помеха присутствует на соответствующей частоте, индикатор равен единице, в противном случае – нулю. Эти индикаторы являются взаимно независимыми случайными величинами с распределениями

$$p(\alpha_i) = \begin{cases} \rho, & \text{если } \alpha_i = 1 \\ 1-\rho, & \text{если } \alpha_i = 0 \end{cases}. \quad (2)$$

Учитывая то, что в рассматриваемом случае энергия субсимвола составляет $E/3$, можно записать условную по α_i вероятность ошибочного приема i -го субсимвола:

$$p_{si} = \alpha_i \cdot \frac{1}{2} \exp\left[-\frac{E}{6(N_0 + N_{II})}\right] + (1 - \alpha_i) \cdot \frac{1}{2} \exp\left[-\frac{E}{6N_0}\right]. \quad (3)$$

При мажоритарном правиле вынесения решения о символе ошибка приема символа возникает в следующих четыре ситуациях:

- все три субсимвола приняты с ошибкой;
- первый субсимвол принят правильно, а второй и третий – с ошибками;
- второй субсимвол принят правильно, а первый и третий – с ошибками;
- третий субсимвол принят правильно, а первый и второй – с ошибками.

Таким образом, условная по α_1, α_2 и α_3 вероятность ошибки приема символа имеет следующий вид:

$$P_{\alpha_1, \alpha_2, \alpha_3} = p_{s1} p_{s2} p_{s3} + (1 - p_{s1}) p_{s2} p_{s3} + (1 - p_{s2}) p_{s1} p_{s3} + (1 - p_{s3}) p_{s1} p_{s2}$$

Сюда входят условные вероятности (3). Усреднение последнего выражения с учетом априорных вероятностей (2) приводит к средней вероятности ошибки при приеме символа:

$$P_e = \frac{1}{8} \left\{ \rho \exp \left[-\frac{E}{6(N_0 + N_{II})} \right] + (1 - \rho) \exp \left[-\frac{E}{6N_0} \right] \right\}^3 + \frac{3}{8} \left\{ \rho \exp \left[-\frac{E}{6(N_0 + N_{II})} \right] + (1 - \rho) \exp \left[-\frac{E}{6N_0} \right] \right\}^2 \times \left\{ 1 - \rho \exp \left[-\frac{E}{6(N_0 + N_{II})} \right] - (1 - \rho) \exp \left[-\frac{E}{6N_0} \right] \right\}.$$

Аналогичную формулу тем же способом можно получить для произвольных значений L :

$$P_e = \sum_{j=L-[L/2]}^L C_L^j \left\{ \frac{1}{2} \rho \exp \left[-\frac{E}{2L(N_0 + P_{II} T_S / (2\rho M))} \right] + \frac{1}{2} (1 - \rho) \exp \left[-\frac{E}{2LN_0} \right] \right\}^j \times \left\{ 1 - \frac{1}{2} \rho \exp \left[-\frac{E}{2L(N_0 + P_{II} T_S / (2\rho M))} \right] - \frac{1}{2} (1 - \rho) \exp \left[-\frac{E}{2LN_0} \right] \right\}^{L-j}.$$

В этом выражении через $[L/2]$ обозначена целая часть величины $L/2$, а для записи спектральной плотности мощности преднамеренной помехи использовано выражение (1).

На рисунке 2 в логарифмическом масштабе изображены графики зависимости средней вероятности ошибки (СВО) P_e от отношения сигнал-помеха при различных процентах забиваемых частот $\rho = 0.05, 0.1, 0.25, 0.9$ и $E/N_0 = 13.35$ дБ (что соответствует $P_e = 10^{-5}$ при отсутствии преднамеренных помех).

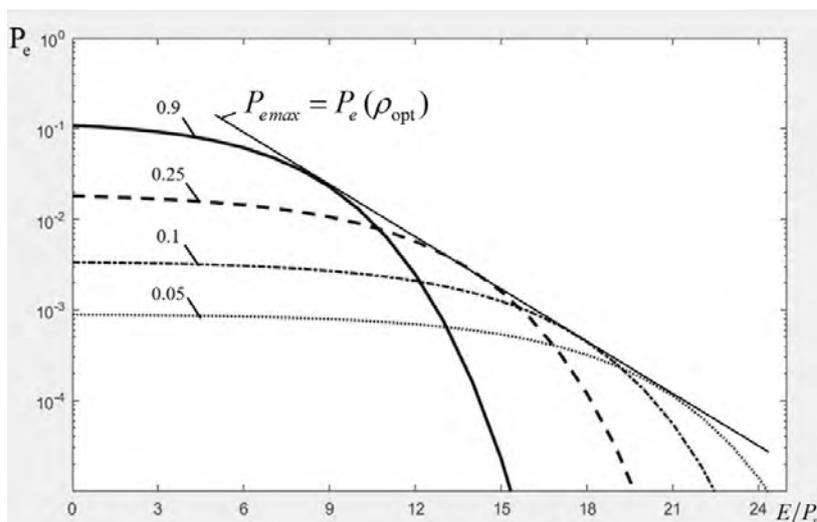


Рис. 2. Зависимость СВО на бит P_e от отношения сигнал-помеха E/P_n

Из графиков на рисунке 2 видно, что для каждого отношения сигнал-помеха существует оптимальное значение $\rho = \rho_{opt}$ и при увеличении отношения E/P_n СВО на бит уменьшается, тем самым улучшается помехоустойчивость. Огибающая графиков показывает максимальную СВО на бит $P_{e\max} = P_e(\rho_{opt})$, при этом зависимость СВО на бит имеет линейный характер, что подтверждает достаточную эффективность подавления систем с ППРЧ путем постановки заградительной помехи на части используемых в системе частот. Это обстоятельство подтверждает также необходимость разработки новых более эффективных методов борьбы с такой помехой.

Дальнейшее исследование авторы связывают с особенностями применения помехоустойчивого кодирования и конкретных видов сигнальных конструкций.

Литература

1. Борисов В.И., Зинчук В.М., Лимарев А.Е. Помехозащищенность систем радиосвязи с расширением спектра сигналов методом псевдослучайной перестройки рабочей частоты / под ред. В.И. Борисова. Изд. 2-е, перераб. и доп. М.: РадиоСофт, 2008. 512 с.
2. Помехозащищенность радиосистем со сложными сигналами / Тузов Г.И. [и др.]. М.: Радио и связь, 1985. 264 с.

Hoang Van Dung, A.A. Paramonov
Russian Technological University

**THE EFFICIENT USE OF FREQUENCY-ENERGY RESOURCE
IN INFORMATION TRANSMISSION SYSTEMS
WITH A FREQUENCY HOPPING SPREAD SPECTRUM
AT A LOW RATE OF TRANSMISSION**

Keywords: *frequency hopping spread spectrum, intentional interference, noise immunity assessment.*

Abstract. *The issues of efficient use of the frequency-energy resource of a radio line in low-speed information transmission systems with a frequency hopping spread spectrum under the influence of the worst noise interference in a part of the frequency band are considered.*

Н.К. Шалаби

МИРЭА – Российский технологический университет

УДК 621.325:621.391

МНОГОФУНКЦИОНАЛЬНЫЕ ГРУППОВЫЕ БЕСПИЛОТНЫЕ АВИАЦИОННЫЕ СИСТЕМЫ С КАНАЛАМИ РАДИОСВЯЗИ МИЛЛИМЕТРОВОГО ДИАПАЗОНА

Ключевые слова: *беспилотный летательный аппарат, миллиметровые радиоволны, беспилотная авиационная система.*

В работе показана актуальность применения групповых беспилотных авиационных систем с каналами радиосвязи миллиметрового диапазона для решения различных технических задач с возможностью интеграции с другими современными радиосистемами.

Введение

Рост динамичности современного образа жизни в обществе постоянно диктует новые требования к различным отраслям науки и техники. Радиотехника является одной из этих отраслей, а также одной из определяющих основ развития современного общества. Наблюдается постоянный рост потребности приёма и передачи разных видов информации в больших объёмах на сверхвысоких скоростях, по возможности повсеместно и непрерывно. С целью удовлетворения таких потребностей радиотехнические системы (РТС) развиваются по нескольким основным направлениям: повышение пропускной способности систем; повышение мобильности радиосистем; повышение степени адаптивности и автоматизации систем; повышение степени связности и интеграции различных систем; повышение уровня децентрализации систем; повышение степени модульности конструкции систем.

Исходя из вышесказанного, чрезвычайно перспективными представляются исследование, применение и развитие РТС с конструкцией и характеристиками, наибольшим образом отражающими

вектор развития радиотехники и с возможностью интеграции с современными и будущими радиотехническими системами и комплексами.

В данной работе в качестве такой системы предложена многофункциональная групповая беспилотная авиационная система (БАС) с использованием миллиметровых волн (ММВ) в радиоканалах связи и управления.

Использование радиоволн мм-диапазона в современных РТС

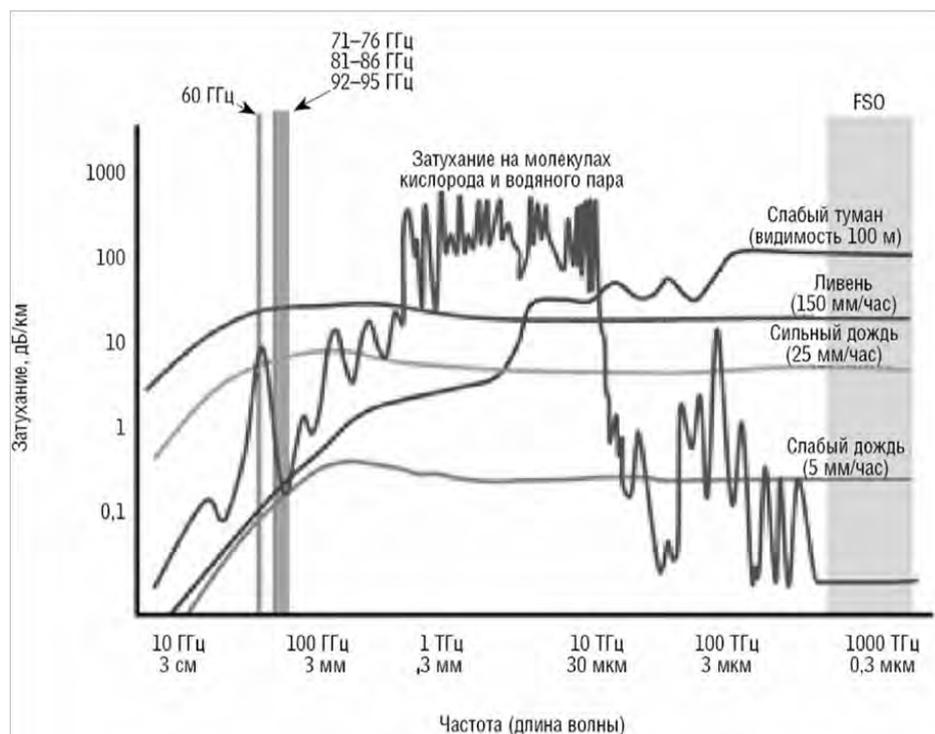


Рис. 1

В настоящее время наблюдается экспоненциальный рост объёма передачи данных как в мобильных радиосетях (к 2020 году достигнет примерно 25 петабайтов / месяц), так и в телекоммуникационных сетях и системах в целом (к 2020 году достигнет примерно 2 зеттабайтов / год). Также постоянно растёт потребность в скорости передачи информации (гигабитные скорости) [1]. Возможности систем передачи ограничены законом Шеннона, причем освоенный спектральный ресурс – уже на исходе. Отсюда возникает необходимость в освоении более высоких свободных частотных диапазонов и применении новых принципов построения радиотехнических систем и сетей [6]. При проектировании РТС удастся резко повысить пространственную плотность радиоканалов и эффективность использования спектрального ресурса в первую очередь за счет пространственного разделения каналов, что хорошо реализуется лишь в диапазонах очень высоких частот.

Радиоволны мм-диапазона имеют специфическую природу, их спектральный диапазон (частота 30 – 300 ГГц, длина волны 10 – 1 мм) находится в верхнем конце радиоволнового диапазона и ниже оптического. Характер их распространения в свободном пространстве имеет много схожего с оптическими волнами, но при этом мм-волны не теряют до конца природные свойства радиоволн. Такая двойная природа распространения ММВ отражается в их радиохарактеристиках, а также в достоинствах и недостатках их применения.

Основными недостатками использования радиоволн с рабочей частотой диапазона крайневисоких частот (КВЧ) являются значительный коэффициент затухания при распространении в атмосфере и чувствительность к метеорологическим условиям.

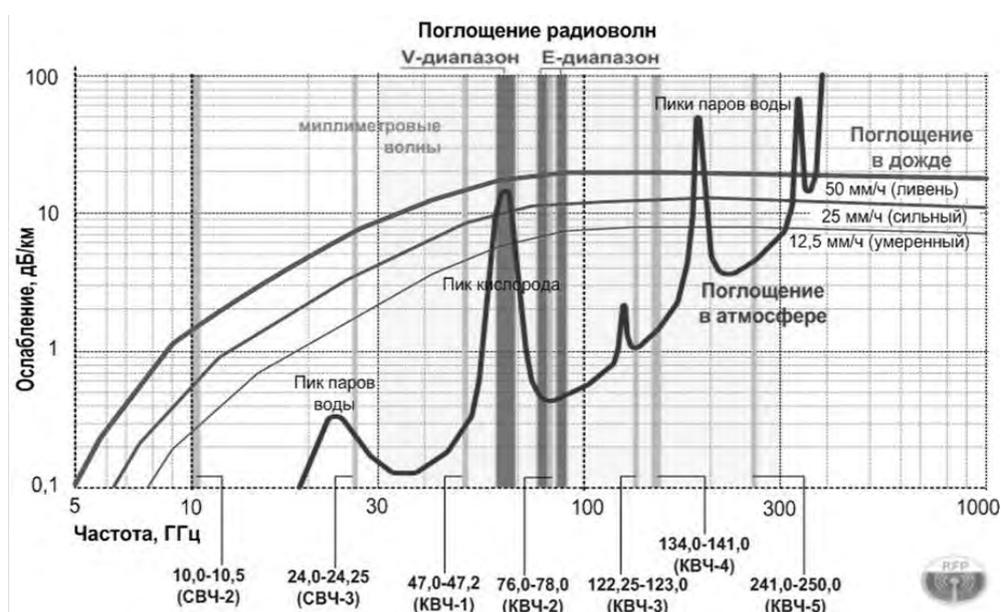


Рис. 2

С другой стороны, потери на атмосферное затухание для радиоволн мм-диапазона сравнимы с потерями в диапазоне сверхвысоких частот (СВЧ), при этом мм-волны имеют более высокую когерентность. Потери от рассеивания и деполяризации при наличии осадков и / или тумана в разы, а иногда и на порядок меньше, чем в оптическом диапазоне; в отличие от оптических волн размеры частиц пыли и дыма значительно меньше длины мм-волны, поэтому такие частицы являются для неё прозрачными [5]. Характеристики ослабления радиоволн в земной атмосфере показаны на рисунках 1 и 2.

ММ-волны как радиоволны, в силу своих высоких частот и малых длин волн, имеют следующие характеристики и достоинства: возможность использования широких полос частот в эфире (порядка 1 ГГц); компактные высоконаправленные антенны; узкая диаграмма направленности (ДН) используемых антенн; низкий коэффициент

отражения от поверхностей; наличие близких друг к другу максимумов затухания и окон прозрачности на частотной шкале, что даёт возможность оператору радиосистемы кардинально менять значения радиотехнических параметров системы небольшим перемещением рабочей частоты; наличие элементной базы, работающей в мм-диапазоне в серийном производстве на основе технологии сверхбольших интегральных схем (СБИС), в том числе в виде монолитных интегральных схем (МИС); экологически более безопасное излучение благодаря высокому затуханию в воде; малая загруженность диапазона КВЧ и упрощённый процесс лицензирования для его использования [6].

Многофункциональная сеть БПЛА

БАС представляет собой радиосистему, которая совмещает в себе процессы извлечения информации, передачи информации и радиоуправления. Гражданские БАС выполняют четыре основные функции: радиообзор площади, радиоанализ структуры, радионаблюдение, обеспечение радиосвязи [8].

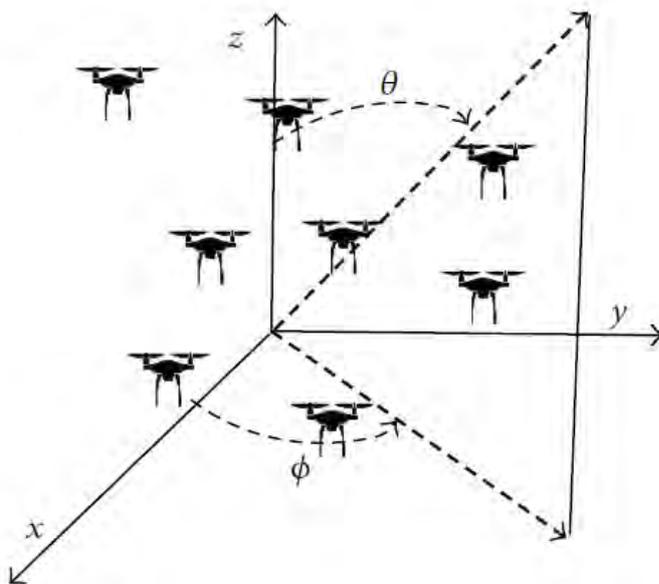


Рис. 3

При организации сети из группы БПЛА (рис. 3) и наземного пункта управления (НПУ) открываются новые горизонты технических возможностей. С одной стороны, при распределении технических ресурсов радиосистемы можно использовать такую сеть для одновременного решения разнообразных технических задач или сосредоточить все ресурсы для решения одной поставленной задачи. С другой стороны, свобода формы пространственного созвездия сети может способствовать повышению дальности действия, пропускной

способности и эффективности радиосистемы в целом. Кроме того, при наличии адаптивных подсистем на бортах и в НПУ такая конфигурация системы поднимает уровень её адаптивности к меняющимся условиям функционирования.

В результате использования гибкости формы радиосети в пространстве, её трёхмерной мобильности и модульности конструкции открывается высокий потенциал интеграции такой радиосети с другими РТС в качестве сегмента многофункционального радиотехнического комплекса, дополнительной или резервной части другой РТС или связующего звена между разными системами [4].

Таким образом, БАС может являться как самостоятельной системой, так и пространственной или функциональной частью более крупной РТС.

Возможность решения поставленных задач группировкой БПЛА определяется, в числе прочего, характеристиками БПЛА, пропускной способностью каналов связи, энергетикой радиолинии и т.д. Ниже даны некоторые оценки этих характеристик.

БПЛА для задач гражданского назначения

Условия решаемой с помощью БАС задачи (радиообзор площади, радиоанализ структуры, радионаблюдение, обеспечение радиосвязи) определяют требования к БАС, что в свою очередь определяет необходимые тактико-технические характеристики используемых ЛА и БАС в целом. Так, задачи гражданского назначения по своим условиям, требованиям и масштабам в большой степени определяют диапазон подходящих для них классов БПЛА. В общем это БПЛА весом в несколько десятков килограммов (5 – 25 кг) с радиусом действия в несколько десятков километров (< 70 км), с продолжительностью полёта в несколько часов (< 6 ч.), с высотой полёта до несколько сотен метров (< 400 м), грузоподъёмностью в несколько килограммов (< 3 кг), со скоростью в несколько десятков километров в час (< 70 км/ч). Используя общепринятую классификацию для российских пользователей, это лёгкие БПЛА малого радиуса действия, низковысотные [10].

Скорости передачи данных в каналах БАС

Для оценки возможностей передачи информации по радиоканалам ММВ внутри и/или с помощью сети БПЛА, расчёты должны полагаться на наиболее жесткие условия и требования.

Из всех радиоканалов, используемых в сети БПЛА (телеметрии, управления, навигации, передачи данных полезной нагрузки), самым ограниченным является канал передачи данных в реальном времени [3]. Канал борт – земля ограничен по мощности, так как передатчик располагается на борту БПЛА; в силу вибраций БПЛА передающая антенна имеет ограничение по направленности и к тому же требования к

скорости передачи в данном случае – относительно других каналов – являются максимальными.

Для многофункциональной БАС следует ориентироваться на задачи визуального наблюдения, так как в силу высоких разрешения и контрастности относительно других частотных диапазонов поток цифровых видеоданных имеет самый большой объём [9].

Объём передаваемых данных в секунду зависит от размера охваченной площади наблюдения и требуемых разрешений в пространстве, времени и цвете. В свою очередь, такие требования определяются самой задачей наблюдения. Существуют множество стандартных форматов видеоизображения, подходящих по своим характеристикам для решения задач видеосъёмки, но с учётом характеристик аппаратуры аэро-видеосъёмки и средств отображения на сегодняшний день и в ближайшем будущем, нас интересует формат с отношением сторон 16 : 9.

К примеру, для насыщенной картинке при передаче спортивного мероприятия желаемым стандартом был бы UHD (ультравысокого разрешения) – 3840 x 2160p (прогрессивная развёртка) x 50 кадров/с x 10 бит/пиксель = 4.147 Гбит/с. При спасательной операции требуемое качество картинки удовлетворил бы стандарт HD (высокого разрешения) – 1280 x 720p x 25 x 8 = 0.184 Гбит/с.

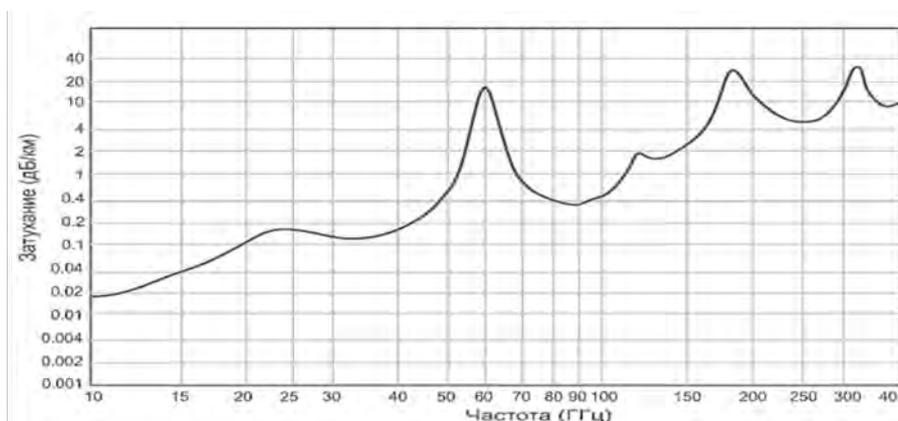


Рис. 4

Анализ бюджета радиолинии борт– земля

При оценке возможностей передачи данных по каналу борт– земля примем наихудшие условия, то есть требования по передаче данных будут завышены, условия функционирования будут приняты самыми плохими, а значения технических параметров системы будут взяты не самыми высокими. Таким образом, результаты анализа будут отражать гарантированную работоспособность системы при самых неблагоприятных реальных условиях.

С увеличением высоты влажность и давление в воздухе убывают по экспоненциальному закону, температура – по линейному. В результате плотность падает и воздух разрежается, степень молекулярного поглощения радиоволн кислородом и водяным паром уменьшается и коэффициенты затухания снижаются. У поверхности земли при тёплой температуре влажность высокая, следовательно, коэффициенты затухания выше [5]. При расчетах использован график, приведенный на рисунке 4, и отражающий зависимость затухания от частоты ММВ при температуре 20°C на уровне моря, а также зависимости погонного затухания ММВ в зависимости от интенсивности дождя (рис. 5).

Для направленных антенн коэффициент усиления растёт с частотой волны. Коэффициент затухания ММВ в окнах прозрачности имеет тенденцию роста с частотой волны. В показательных целях для расчётов выбрана радиочастота первого открытого для использования поддиапазона ММВ.

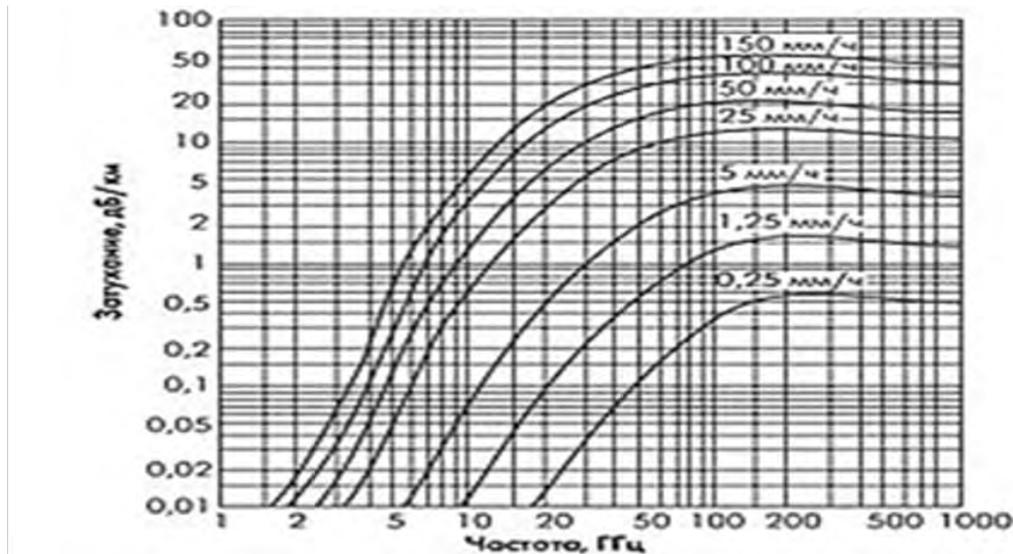


Рис. 5

Ниже приведен примерный расчет дальности передачи информации по каналу БПЛА-НПУ.

Показательный анализ бюджета радиолинии БПЛА-НПУ:

вид модуляции = 2-ФМ (BPSK), спектральная плотность = 1 (бит/с)/Гц;

требуемая вероятность ошибки = 10^{-6} ;

требуемое отношение сигнал шум на один бит $(ОСШ)_{бит} = 10$ дБ;

спектральная плотность шума = $kT = -204$ дБ, где k – постоянная Больцмана, равная

1.38×10^{-23} Дж/К, T – температура в °К;

требуемая скорость передачи = 1 Гбит/с = 90 дБ;

требуемая мощность сигнала на входе приёмника $P_{пр(мин)} = -104$ дБ;

чувствительность приёмника НПУ = -100 дБ;

мощность передатчика БПЛА = 0 дБ.

Для параболических антенн:

$$A_{эфф} = \frac{\pi d^2}{4};$$

диаметр антенны БПЛА $d_{БПЛА} = 0.05$ м;

эффективная площадь антенны БПЛА $(A_{эфф})_{БПЛА} = 20 \times 10^{-4} \text{ м}^2 = -27$ дБ;

диаметр антенны НПУ $d_{НПУ} = 0.5$ м;

эффективная площадь антенны НПУ $(A_{эфф})_{НПУ} = 20 \times 10^{-2} \text{ м}^2 = -7$ дБ;

Коэффициент направленности антенны

$$D = \frac{4\pi}{\lambda^2} A_{эфф} = \frac{4\pi}{\Delta\theta\Delta\varphi} \approx \frac{40000}{(\Delta\theta\Delta\varphi)};$$

Для сигнала с несущей частотой $f = 35$ ГГц, длиной волны $\lambda = 8.57$ мм:

ширина главного лепестка антенны БПЛА $\Delta\varphi = \Delta\theta = 10^\circ = 0.17$ рад;

ширина главного лепестка антенны НПУ $\Delta\varphi = \Delta\theta = 1^\circ = 0.017$ рад;

коэффициент направленности антенны БПЛА $D_{БПЛА} = 26$ дБ;

коэффициент направленности антенны НПУ $D_{НПУ} = 46$ дБ;

погонное затухание в атмосфере $L_{атм} = 0.12$ дБ/км;

погонное затухание в слабом дожде $L_{осадки} (5 \text{ мм/ч}) = 1.2$ дБ/км;

погонное затухание в умеренном дожде $L_{осадки} (12.5 \text{ мм/ч}) = 3$ дБ/км;

погонное затухание в сильном дожде $L_{осадки} (25 \text{ мм/ч}) = 5$ дБ/км.

При распространении радиоволн в свободном пространстве:

$$\frac{P_{пр}}{P_{пер}} = \frac{D_{пер}A_{пр}}{4\pi R^2} = D_{пер}D_{пр} \left(\frac{\lambda}{4\pi R} \right)^2 = \frac{A_{пер}A_{пр}}{\lambda^2 R^2},$$

где $P_{пр}$ – мощность сигнала на входе приёмника, $P_{пер}$ – мощность передатчика, $D_{пер}$ – коэффициент направленности антенны передатчика, $D_{пр}$ – коэффициент направленности антенны приёмника, $A_{пер}$ – эффективная площадь антенны передатчика, $A_{пр}$ – эффективная площадь антенны приёмника, λ – длина волны сигнала, R – дальность распространения радиоволны.

Для ММВ при распространении в атмосфере:

$$20 \lg(R_{км}) + (L_{атм} + L_{осадки}) R_{км} = P_{пер} + A_{пер} + A_{пр} - P_{пр} - 20 \lg(\lambda_{мм}) \quad \text{дБ}$$

$$20 \lg(R_{км}) + (L_{атм} + L_{осадки}) R_{км} = 0 - 7 - 27 + 100 - 20 \lg(8.57) \quad \text{дБ}$$

После преобразования:

$$\left(10^{\frac{-L}{20}} \right)^{R_{км}} \approx \frac{R_{км}}{100}$$

$R_{макс}$ (без осадков) ≈ 50 км

$R_{макс}$ (со слабым дождём) ≈ 13 км

$R_{макс}$ (с умеренным дождём) ≈ 7 км

$R_{макс}$ (с сильным дождём) ≈ 5 км

Выводы

С помощью ММВ группа БПЛА может передавать поток видеоданных без сжатия, в реальном времени и в хорошем качестве на десятки километров через единичный БПЛА. Передаваемый поток может распределяться между несколькими БПЛА в группе. Возможно увеличение дальности передачи путём использования нескольких БПЛА в качестве ретрансляторов в воздухе.

Проведя обзор частот всех окон прозрачности всего диапазона ММВ и выявив оптимальные частоты для различных условий функционирования системы, таких как климат, рельеф местности, требования поставленной задачи, можно увеличить дальность передачи информации. Использование антенных решёток на передаче и на приеме даст возможность регулировать эффективную площадь антенн в зависимости от условий функционирования системы и тем самым также повысить дальность передачи.

Заключение

Радиоканалы диапазона ММВ – со свойственной им пространственной селективностью – отлично вписываются в новую парадигму построения РТС. Радиосеть БПЛА, по природе своей динамики и трёхмерной мобильности и по особенностям функционирования, отлично подходит в качестве объекта для материализации такой парадигмы построения.

В свете вышесказанного применение сетей БПЛА с радиоканалами ММВ является обоснованным, актуальным и эффективным выбором с потенциалом будущего развития и возможности интеграции с радиотехническими системами и сетями будущего.

Литература

1. Радиотехнические и радиооптические системы : учеб. пособие для вузов / Э.А. Засовин [и др.]; под ред. Э.А. Засовина. М.: Круглый год, 2001. 751 с.
2. Austin Reg. Unmanned Aircraft Systems – UAVS design, development and deployment. John Wiley and Sons, 2010. 365 p.
3. Боев Н.М. Анализ командно-телеметрической радиолинии связи с беспилотными летательными аппаратами // Вестник Сибирского государственного аэрокосмического университета им. акад. М.Ф. Решетнева. 2012. Вып. 2 (42). С. 86 - 91.
4. Особенности построения радиосети группы БПЛА / Шлокин В.Н. [и др.] // Системи озброєння і військова техніка. 2013. № 1(33).
5. Гавриленко В.Г., Яшнов В.А. Распространение радиоволн в современных системах мобильной связи. Нижний Новгород: Нижегородский государственный университет им. Н.И. Лобачевского. 2003, 148 с.
6. Шахнович И. Системы беспроводной связи 5G: телекоммуникационная парадигма, которая изменит мир, Краткие тезисы // Связь и телекоммуникация, электроника наука технология бизнес. 2015. №7 (00147).

7. Шахнович И., Вишнеvский В., Фролов С. Радиорелейные линии связи в миллиметровом диапазоне. Новые горизонты скоростей // Связь и телекоммуникация, электроника наука технология бизнес. 2011. №1 (00107).

8. Применение беспилотных летательных аппаратов при разведке труднодоступных и масштабных зон чрезвычайных ситуаций / Янников И.М. [и др.] // Вектор науки ТГУ. № 3 (21) - 2012.

9. Слюсар В. Передача данных с борта БПЛА // Микророботехника, электроника наука технология бизнес. 2010. №3.

10. Корченко А.Г., Ильяш О.С. Обобщённая классификация беспилотных летательных аппаратов // Збірник наукових праць Харківського університету Повітряних Сил. 2012. Вип. 4(33). С. 27-36.

N.K. Shalaby
Russian Technological University

Multifunctional Flying Ad hoc Networks with Millimeter-wave Radio Communication Channels

Keywords: *unmanned air vehicle, millimeter waves, FANET.*

Abstract. *This article reveals the technical relevance of utilizing FANETs with mm-wave radio channels for different technical tasks, with the capability of integration with other advanced wireless systems.*

М.Ю. Ястребов, И.В. Ланева, Д.С. Фролов
*Санкт-Петербургский государственный университет
морского и речного флота им. адмирала С.О. Макарова*

УДК 004.627, 519.21

ПОКАЗАТЕЛИ КАЧЕСТВА ХЕШИРОВАНИЯ И АНАЛИЗ НЕКОТОРЫХ ЭВРИСТИЧЕСКИХ ХЕШ-АЛГОРИТМОВ

Ключевые слова: *анализ алгоритмов, хеширование, показатели эффективности, случайные числа, поиск Фибоначчи, ряды Фаря, бильярдное отражение.*

В статье обсуждаются показатели качества хеширования массива в аспектах равномерности сжатия (распределения частоты коллизий и длин коллизионных цепочек), а также подход к оценке иррегулярности хеширования применительно к эвристическим методам хеширования, основанным на поиске Фибоначчи, рядах Фаря и бильярдном блуждании на окружности с неупругим отражением.

Хеширование в широком толковании этого термина означает преобразование битовой строки в более короткую. Неизбежное нарушение взаимной однозначности исходной строки и её хеш-кода (возникновение коллизий) желательно минимизировать при заданном соотношении максимально возможного и реального объёмов совокупности строк-преобразов. Частота коллизий, как один из показателей эффективности, в случае эвристических процедур (в отличие, например, от хеширования методом деления) не имеет априорной оценки и должна являться предметом исследования в ходе имитационного машинного эксперимента.

Другим важным — криптографическим — аспектом хеширования является (квази)случайность, иррегулярность перемешивания исходной совокупности битовых строк, когда при хешировании нарушается попарная близость в какой-либо метрике преобладающей части пар исходных строк и пар их хеш-кодов. С криптографическим аспектом хеширования связано требование случайного перемешивания ключей $h(K)$ — случайного в том смысле, что хотя функция h задаётся детерминированным алгоритмом, её значения внешне выглядят непредсказуемыми.

Целью хеширования может являться преобразование большого, но потенциально редко заполненного массива в хеш-массив существенно меньшего размера. Объектами хеширования (битовыми строками) выступают адреса элементов исходного одномерного массива (ключи), а хеш-кодами являются адреса элементов хеш-массива. При совпадении хеш-кодов нескольких ключей (возникновении коллизии) из последних формируется цепочка (список с цепной адресацией). Таким образом, адресация реализуется с помощью промежуточного хеш-кода $h(K)$, которому уже однозначно соответствует адрес $adr(K)$:

$$K \xrightarrow{h} h(K) \xrightarrow{f} adr(h(K)),$$

так что

$$adr(K) = f(h(K)).$$

В соответствии со сказанным для трех эвристических алгоритмов хеширования — на основе поиска Фибоначчи (ПФ-хеширования) [1; 2], на основе рядов Фарей (РФ-хеширования) [3] и бильярдного (Б-хеширования), сформулированного ниже, — обсуждаются два аспекта: качество сжатия и мера иррегулярности. Кроме них по введённой системе показателей сжатия анализируются по данным машинных экспериментов также «модельные» алгоритмы хеширования методами деления и умножения [5]. Алгоритмы реализуются с использованием генерации случайного входного потока различных значений ключа.

1. Алгоритмы хеширования. ПФ-хеширование предполагает построение кортежа в трёхсимвольном алфавите, задающем блуждание

при поиске. Затем по кортежу вычисляется хеш-адрес. Размер исходного массива 3^k , размер хеш-массива $F_{k+1}-1$, где $F_{k+1}-1$ – число Фибоначчи.

Пусть ряд Фарея F_n порядка n , содержащий $t(n) = \sum_{j=2}^n \varphi(j) + 2$ членов – несократимых дробей со знаменателями $j \in \{1, \dots, n\}$ (φ – функция Эйлера): $F_n = \{f_1, \dots, f_{t(n)}\}$. При РФ-хеширования массива из $N = t(n)$ элементов с ключом поиска $K \in \{1, 2, \dots, N\}$ вычисляется $r(K) = \{K/n\}$ – дробная часть отношения. Если $f_i = u_i/v_i \leq r(K) < f_{i+1} = u_{i+1}/v_{i+1}$, то хэш-функция

$$h(K) = \begin{cases} \min(v_i, v_{i+1}) \text{ при } u_i \leq \alpha u_{i+1}, \\ h(K) = \max(v_i, v_{i+1}) \text{ в противном случае.} \end{cases}$$

Здесь учитывается, что бóльшие знаменатели в среднем встречаются чаще. Параметр $\alpha > 1$ настраивается в ходе машинного эксперимента.

2. Биллиардное хеширование. Квазистохастическое поведение биллиардной траектории при многократном отражении от гладкой или ломаной замкнутой линии может служить основой для формулирования различных алгоритмов перемешивания элементов массива. В данной статье мы анализируем хеширование на основе серии неупругих отражений от (единичной) окружности (угол падения не равен углу отражения). Заслуживает внимания и биллиардное хеширование на неправильном выпуклом многоугольнике, асимметрия которого позволяет обойтись без случайных добавок к углу отражения.

Биллиардное хеширование на окружности. Если заданы объёмы K и k исходного массива и хеш-массива, окружность разбивается на k дуг $(P_{i-1}P_i)$ точками P_0, P_1, \dots, P_k . Далее, для адреса — натурального числа $a \in \{1, 2, \dots, K\}$ выбираются начальный угол $\alpha(a) \in (0, \pi)$ и количество отражений $n(a)$. Хеш-адрес $h(a)$ определяется как номер j дуги, в которую попадает точка при последнем отражении (рис. 1). Исходное положение точки — середина дуги (P_0P_1) . Варьируя способы задания $\alpha(a)$ и $n(a)$, можно получать различные биллиардные процедуры хеширования.

Программирование алгоритма предполагает: 1) задание функции начального угла $\alpha(a)$ (рис. 2); 2) задание функции числа отражений $n(a)$ – в простейшем случае она может быть постоянной величиной; 3) на очередной i -й итерации:

- вычисление точки M_i пересечения с окружностью луча, исходящего из точки M_{i-1} ;
- вычисление угла β_i падения на касательную в точке M_i ;
- вычисление угла отражения γ_i от касательной (рис. 1); для обеспечения асимметрии к углу отражения прибавляется знакопеременная случайная поправка;
- нахождение номера $j = h(a)$ дуги, содержащей финальную точку.

Решение последних двух задач опирается на соответствующие методы аналитической геометрии. Алгоритм реализуется в полярной системе координат.

Для разброса начальных углов у близких адресов функцию $\alpha(a)$ мы задаём здесь следующим образом: $\alpha(a) = t \bmod \pi$ радиан, где $t = [\{\{\frac{a}{k} + \frac{k}{a}\}10^2\}10^4]$ (по мотивам метода середины квадратов для генерации псевдослучайных чисел). Далее, полагаем $n(a) = n_0 = const$.

3. Показатели сжатия. Эффективность описанных алгоритмов хеширования с точки зрения сжатия должна оцениваться на разреженном исходном массиве, когда $k \leq 2^{-1}K$, поскольку именно разреженность потенциально большого исходного массива является одним из стимулов к хешированию. Показателями эффективности предлагается выбрать: F — частота коллизий по отношению к числу актов хеширования; L — средняя длина цепочек, возникающих при коллизиях; D — дисперсия средней длины (показатель равномерности появления коллизий); E — показатель эффективности использования пространства хеш-массива, определяемый как отношение объёма хеш-массива k к суммарной длине цепочек.

В ходе машинного эксперимента при различных K воспроизводился поток из S различных случайных адресов исходного массива для значений S , равных $0.1K$, $0.2K$, $0.3K$, $0.4K$, $0.5K$, и для каждого из них вычислялись средние значения F , L , D и E . Результаты приведены в таблицах 1–3. Для сравнения эти же показатели рассчитаны для «классических» алгоритмов хеширования методом деления и методом умножения [5] (таблицы 4–5).

Таблица 1

Показатели для ПФ-хеширования

	1300	2600	3900	5200	6500	$k=8$
	0.031	0.016	0.011	0.008	0.006	
	32.5	63.4	95.1	126.8	158.5	
	6.4	25.2	56.6	100.1	156.9	
	457	1905	4296	7441	11501	

Таблица 2

**Показатели для РФ-хеширования
(ресурсоёмкий алгоритм)**

<i>L</i>	14	29	44	59	<i>n = 13</i>
<i>F</i>	0.58	0.70	0.78	0.82	
<i>L</i>	2.55	3.47	4.52	5.59	
<i>E</i>	0.19	0.33	0.45	0.56	
<i>D</i>	4.82	11.75	18.81	25.36	

Таблица 3

Показатели для Б-хеширования

<i>S</i>	2500	5000	7500	10000	<i>k = 10000</i>
<i>F</i>	0.11	0.21	0.30	0.37	
<i>L</i>	1.13	1.27	1.42	1.58	
<i>E</i>	4.00	2.00	1.33	1.00	
<i>D</i>	2.70	3.69	4.46	5.05	

Таблица 4

Показатели для хеширования методом деления

<i>S</i>	2500	5000	7500	10000	<i>k = 10000</i>
<i>F</i>	0.63	0.80	0.87	0.90	
<i>L</i>	2.72	5.03	7.50	10.00	
<i>E</i>	4.00	2.00	0.54	0.05	
<i>D</i>	6.30	8.63	10.38	11.74	

Таблица 5

Показатели для хеширования методом умножения

<i>S</i>	2500	5000	7500	10000	<i>k = 10000</i>
<i>F</i>	0.60	0.77	0.84	0.88	
<i>L</i>	2.48	4.33	6.24	8.16	
<i>E</i>	4.00	2.00	1.33	1.00	
<i>D</i>	6.30	9.12	11.23	13.17	

4. Иррегулярность хеширования. Криптографический аспект хеширования, имея подчинённое значение по сравнению с главной целью этой процедуры — сжатием, тем не менее заслуживает обсуждения. Хеширование реализуется детерминированным алгоритмом, воспроизводящим конечную последовательность (натуральных чисел в случае адресации массива). Поэтому целесообразно наряду со статистическими критериями случайности последовательности, которые при заданном пороговом значении дают дискретный ответ «да/нет», использовать количественные показатели.

Поскольку продуцирование хеш-адресов является отображением в множество меньшего объёма представляется естественным взять в качестве эталона для сравнения алгоритм хеширования делением по заданному модулю. Последнему приписывается минимальная иррегулярность, так как на диапазоне номеров $1 \text{ — } k$ (где k — длина хеш-массива, по модулю которой преобразуются исходные адреса) число инверсий перестановки $(1, 2, \dots, k-1, k)$ равно нулю. Максимальное

количество инверсий $(k-1)k/2$ в зеркальной перестановке $(k, k-1, \dots, 2, 1)$ также свидетельствует о полной регулярности. Таким образом, при формировании показателя I иррегулярности за основу может быть принята статистически оцениваемая удалённость числа инверсий перестановки хеш-адресов, полученной на случайно формируемой монотонной последовательности исходных адресов, от среднего числа $(k-1)k/4$.

Здесь нами лишь обозначен принцип возможного подхода к оценке иррегулярности алгоритмов хеширования, который подлежит дальнейшему уточнению и экспериментальной проверке.

Литература

1. Ястребов М.Ю. Поиск Фибоначчи — обоснование, система счисления и хеширование // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. 2013, № 3 (174). С. 53-58.
2. Ястребов М.Ю. Хеширование на основе рядов Фарея // Системы компьютерной математики и их приложения: материалы XVII Международной научной конференции. Вып. 17. Смоленск: Изд-во СмолГУ, 2016. С. 238-239.
3. Ястребов М.Ю., Бебнев Р.А., Федосов А.Д. Анализ хеширования на основе поиска Фибоначчи // Системы компьютерной математики и их приложения: материалы XVIII Международной научной конференции. Вып. 18. Смоленск: Изд-во СмолГУ, 2017. С. 54-56.
4. Communications of the Association for Computing Machinery. 1971. Vol. 14. P. 228-239.
5. Кнут В. Искусство программирования. Т. 3. Сортировка и поиск. 2-е изд. М.: Изд. дом «Вильямс». 2003. 820 с.

M.Y. Yastrebov, D.S. Frolov

Admiral Makarov State University of Maritime and Inland Shipping

ANALYSIS OF SOME HEURISTIC ALGORITHMS OF HASHING

Keywords: *analysis of algorithms, hashing, measures of efficiency, random numbers, Fibonacci search, Farey series, billiard mirroring, irregularity of hashing.*

Abstract. *The article describes hashing algorithms based on search procedures, Farey series and billiard reflections. A number of performance indicators of such algorithms are introduced. Their values obtained during the simulation computer experiment are given. An approach to estimating hash irregularity is proposed.*

СЕКЦИЯ 2

Параллельные вычисления и многоядерные процессоры

В.В. Борисов, А.С. Федулов

Филиал ФГБОУ ВО «Национальный исследовательский университет» в г. Смоленске

УДК 004.8

МОДЕЛЬ И МЕТОД ИНТЕЛЛЕКТУАЛЬНОГО УПРАВЛЕНИЯ РИСКАМИ НАРУШЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ПРОЦЕССОВ В СОЦИОТЕХНИЧЕСКИХ СИСТЕМАХ¹

Ключевые слова: *интеллектуальное управление рисками, информационная безопасность, социотехническая система, нечеткая модель*

Выполнена формализованная постановка задачи управления рисками нарушения информационной безопасности процессов в социотехнических системах. Предложена модель интеллектуального управления рисками нарушения информационной безопасности процессов в социотехнических системах, основанная на нечетком и нейро-нечетком подходах, а также на методах гибридизации и комплексирования нечетких и нейро-нечетких моделей. Представлен метод интеллектуального управления рисками нарушения информационной безопасности процессов в социотехнических системах, базирующийся на разработанной модели и заключающийся в определении сочетаний управляющих параметров, обеспечивающих снижение социальных и технических затрат на процессы в социотехнических системах при не превышении допустимого уровня риска нарушения информационной безопасности.

Введение

Насущной прагматической проблемой на протяжении последних десятилетий является обеспечение устойчивого функционирования и

¹ Работа выполнена при финансовой поддержке РФФИ, проект № 18-29-03088_мк.

развития различных социотехнических систем (СТС), характеризующихся следующими особенностями:

- ценностно ориентированной направленностью СТС, наиболее ярко проявляющейся именно в их устойчивом функционировании и развитии и, как следствие, в отсутствии явно выраженных целевых показателей и критериев эффективности;
- сложным и нелинейным характером взаимозависимостей между социальными, техническими и иными факторами СТС, а также факторами внешней среды;
- разнородностью, неточностью и неполнотой исходных данных, неопределенностью системных и внешних факторов СТС, а также непосредственных и опосредованных характеристик взаимозависимостей между ними;
- смешанным характером системной динамики изменения социальных и технических факторов при функционировании и развитии СТС;
- сложностью формализации постановок выполняемых в СТС аналитических задач и необходимостью реализации комплексного подхода для их решения [1–3].

В качестве типичных примеров СТС можно привести системы энерго- и теплообеспечения городов и городских поселений, учреждения по предоставлению жилищно-коммунальных и образовательных услуг.

Устойчивость функционирования и развития таких СТС во многом обусловлена качественной и оперативной реализацией протекающих в них информационных процессов.

Под *информационными процессами в СТС* понимаются процессы сбора, передачи, обработки, обобщения и анализа *информационных ресурсов*, к которым относятся данные и информация, необходимые для реализации устойчивого функционирования и развития этих систем.

При этом наблюдается ярко выраженная тенденция, заключающаяся в том, что *нарушение информационной безопасности (ИБ) процессов в СТС*, то есть деструктивное воздействие на информационные ресурсы и информационные процессы, формируемые и реализуемые в рамках различных социальных, технологических, производственных и иных процессов в этих системах, приводит к их частичной или полной остановке, что в итоге влечет за собой существенные финансовые и иные потери.

Под *информационной безопасностью СТС* будем понимать состояние защищенности информационных ресурсов и информационных процессов, реализуемых в рамках социальных, технологических, производственных и иных процессов в этих системах, от деструктивных внутренних и внешних воздействий.

Очевидно, что обеспечение ИБ в СТС должно носить комплексный характер и быть направлено:

- во-первых, на управление рисками нарушения ИБ процессов в СТС;
- во-вторых, на ликвидацию последствий нарушения ИБ процессов в СТС.

При этом приоритетным для обеспечения ИБ процессов в СТС является управление рисками нарушения ИБ процессов в этих системах, которое должно носить превентивный характер.

1. Постановка задачи управления рисками нарушения информационной безопасности процессов в социотехнических системах

В соответствии с перечисленными выше особенностями постановка задачи управления рисками нарушения ИБ процессов в СТС заключается в обеспечении устойчивого функционирования и развития этих систем при неперевышении допустимого уровня риска:

$$S \xrightarrow{P=\{p_l^{jk}\}, l=1\dots L_j, j=1\dots J_k, k=1\dots K} \min,$$

$$S = s_E E + s_H H,$$

при $R \leq R_{(per)}$,

где S – суммарные социальные и технические затраты на процессы в СТС; $P = \{p_l^{jk} \mid l=1\dots L_j, j=1\dots J_k, k=1\dots K\}$ – управляющие параметры процессов в СТС; l – порядковый индекс управляющего параметра j -й стадии k -го процесса в СТС; E и H – общие социальные и технические затраты на процессы в СТС, соответственно; s_E и s_H – удельные социальные и технические затраты на процессы в СТС соответственно; R – обобщенный риск нарушения ИБ процессов в СТС в целом; $R_{(per)}$ – допустимый уровень обобщенного риска нарушения ИБ процессов в СТС.

Допустимый уровень риска ИБ может быть также установлен не только для нарушения процессов в СТС в целом, но и для каждого процесса, а также для отдельных стадий этих процессов, то есть:

$$R_k \leq R_{k(per)}, \quad k=1\dots K,$$

$$R_j^k \leq R_{j(per)}^k, \quad j=1\dots J_k, k=1\dots K,$$

где R_k – риск нарушения ИБ k -го процесса; $R_{k(per)}$ – допустимый уровень риска нарушения ИБ k -го процесса; R_j^k – риск нарушения ИБ j -й стадии k -го процесса; $R_{j(per)}^k$ – допустимый уровень риска нарушения ИБ j -й стадии k -го процесса.

Общие социальные E и технические H затраты СТС зависят от социальных E_k и технических H_k затрат на соответствующие процессы этой системы:

$$E = FE(E_1, \dots, E_k, \dots, E_K),$$

$$H = FH(H_1, \dots, H_k, \dots, H_K),$$

где FE – зависимость между социальными затратами для отдельных процессов и общими социальными затратами в СТС; FH – зависимость между техническими затратами для отдельных процессов и общими техническими затратами в СТС.

Социальные E_k и технические H_k затраты на k -й процесс зависят от социальных E_j^k и технических H_j^k затрат на стадии этого процесса:

$$E_k = FE_k(E_1^k, \dots, E_j^k, \dots, E_{j_k}^k), \quad k = 1 \dots K,$$

$$H_k = FH_k(H_1^k, \dots, H_j^k, \dots, H_{j_k}^k), \quad k = 1 \dots K,$$

где FE_k – зависимость между социальными затратами для j -й стадии k -го процесса и социальными затратами для всего этого k -го процесса; FH_k – зависимость между техническими затратами для j -й стадии k -го процесса и техническими затратами на для всего этого k -го процесса.

Социальные E_j^k и технические H_j^k затраты для j -й стадии k -го процесса зависят от соответствующих им управляющих параметров $\{p_l^{jk} \mid l = 1 \dots L_j\}$:

$$E_j^k = FE_j^k(\{p_l^{jk} \mid l = 1 \dots L_j\}), \quad j = 1 \dots J_k, \quad k = 1 \dots K,$$

$$H_j^k = FH_j^k(\{p_l^{jk} \mid l = 1 \dots L_j\}), \quad j = 1 \dots J_k, \quad k = 1 \dots K,$$

где FE_j^k – зависимость между управляющими параметрами $\{p_l^{jk} \mid l = 1 \dots L_j\}$ и социальными затратами для j -й стадии k -го процесса; FH_j^k – зависимость между управляющими параметрами $\{p_l^{jk} \mid l = 1 \dots L_j\}$ и техническими затратами для j -й стадии k -го процесса.

Обобщенный риск R нарушения ИБ процессов в СТС в целом зависит от рисков R_k нарушения ИБ соответствующих процессов:

$$R = FR(R_1, \dots, R_k, \dots, R_K),$$

где FR – зависимость между рисками нарушения ИБ отдельных процессов и обобщенным риском нарушения ИБ процессов в СТС в целом.

Риск R_k нарушения ИБ k -го процесса зависит от рисков R_j^k нарушения ИБ j -х стадий этого процесса:

$$R_k = FR_k(R_1^k, \dots, R_j^k, \dots, R_{j_k}^k), \quad k = 1 \dots K,$$

где FR_k – зависимость между рисками нарушения ИБ отдельных стадий k -го процесса и риском нарушения ИБ всего k -го процесса.

Риск R_j^k нарушения ИБ j -й стадии k -го процесса зависит от социальных и технических затрат для j -й стадии k -го процесса:

$$R_j^k = FR_j^k(E_j^k, H_j^k), \quad j=1...J_k, k=1...K,$$

где FR_j^k – зависимость между социальными и техническими затратами для j -й стадии k -го процесса и риском нарушения ИБ этой стадии процесса.

2. Модель интеллектуального управления рисками нарушения ИБ процессов в СТС

Зависимости между управляющими параметрами и социальными / техническими затратами СТС на всех стадиях различных процессов в этих системах, а также их воздействия на риски нарушения ИБ этих процессов носят нелинейный характер. Это обусловлено, с одной стороны, спецификой этих процессов, а с другой – влиянием системных и внешних факторов в условиях неопределенности. Поэтому для анализа процессов в СТС и оценки рисков ИБ процессов в этих системах целесообразно использовать нечеткий и нейро-нечеткий подходы [4], а также методы гибридизации и комплексирования нечетких и нейро-нечетких моделей [5].

С учетом этих подходов и методов предлагается модель интеллектуального управления рисками ИБ процессов в СТС [6].

Предлагаемая модель включает в себя несколько каскадов моделей.

Каскад 1 состоит из совокупности моделей для анализа k -х ($k=1...K$) процессов СТС. Эти модели соответствуют результатам декомпозиции процессов в СТС и предназначены для решения частных задач с учетом неопределенности анализируемых характеристик.

Анализ этих процессов может быть основан как на выявленных аналитических зависимостях между системными и внешними характеристиками процессов, так и на предложенном подходе к построению и решению систем уравнений с нечеткими параметрами на основе нечетких численных методов [7].

Каскад 2 состоит из моделей, реализующих зависимости FE_j^k , FH_j^k и предназначенных для оценки социальных и технических затрат E_j^k , H_j^k , на j -х стадиях ($j=1...J_k$) для k -х ($k=1...K$) процессов СТС. Во 2-й каскад также входят модели, которые реализуют зависимости FR_j^k и предназначены для оценки рисков нарушения ИБ j -х стадий ($j=1...J_k$) k -х ($k=1...K$) процессов СТС.

Модели 2-го каскада строятся на основе нечетко-логического подхода и представляют собой согласованные базы нечетких продукционных правил.

Каскад 3 включает в себя совокупности нечетких продукционных моделей для оценки социальных и технических затрат E_k и H_k , для k -х ($k = 1, \dots, K$) процессов СТС. В состав 3-го каскада также входят модели оценки рисков нарушения ИБ процессов в СТС, реализующие зависимости $R_k = FR_k(R_1^k, \dots, R_j^k, \dots, R_{J_k}^k)$.

Каскад 4 включает в себя нечеткие продукционные модели, которые используются для оценки общих социальных и технических затрат E и H соответственно. Помимо моделей E и H в состав 4-го каскада входит модель оценки обобщенного риска нарушения ИБ процессов в СТС в целом, реализующая зависимость $R = FR(R_1, \dots, R_k, \dots, R_K)$.

3. Метод интеллектуального управления рисками нарушения ИБ процессов в СТС

Предлагаемый метод интеллектуального управления рисками нарушения ИБ процессов в СТС к управлению рисками нарушения ИБ процессов в СТС основывается на разработанной модели и в обобщенном виде заключается в следующем:

- *во-первых*, в задании различных сочетаний управляющих параметров $P = \{p_l^{jk} \mid l = 1 \dots L_j, j = 1 \dots J_k, k = 1 \dots K\}$ на каждой стадии для всех процессов в СТС с учетом накладываемых на эти процессы ограничений, в том числе рисков нарушения ИБ j -х стадий k -х процессов $R_j^k \leq R_{j(per)}^k$, ($j = 1 \dots J_k, k = 1 \dots K$);
- *во-вторых*, в моделировании и в определении таких сочетаний этих управляющих параметров, которые обеспечат снижение социальных и технических затрат на процессы в СТС при непревышении допустимого уровня обобщенного риска нарушения ИБ в СТС.

Заключение

В работе обобщены особенности современных СТС. Обоснована необходимость обеспечения ИБ этих систем, прежде всего, за счет управления рисками нарушения ИБ процессов в СТС, которое должно носить превентивный характер.

Выполнена формализованная постановка задачи управления рисками нарушения ИБ процессов в СТС.

Предложена модель интеллектуального управления рисками нарушения ИБ процессов в СТС, основанная на нечетком и нейро-нечетком подходах, а также на методах гибридизации и комплексирования нечетких и нейро-нечетких моделей.

Представлен метод интеллектуального управления рисками нарушения ИБ процессов в СТС, базирующийся на разработанной модели

и заключающийся: во-первых, в задании различных сочетаний управляющих параметров процессов в СТС с учетом ограничений на риски нарушения ИБ процессов СТС; во-вторых, в определении сочетаний управляющих параметров, обеспечивающих снижение социальных и технических затрат на процессы в СТС при непревышении допустимого уровня обобщенного риска нарушения ИБ в СТС.

Литература

1. Дафт Р.Л. Теория организации. М.: ЮНИТИ-ДАНА, 2006. 736 с.
2. Бирман Л.А. Управленческие решения. М.: «Дело» АНХ, 2008. 208 с.
3. Sawyer S., Jarrahi M.H. Sociotechnical approaches to the study of Information Systems / A. Tucker, & H. Topi (Eds.) // Computing Handbook: Information systems and information technology, 3rd ed. Boca Raton, FL: Taylor & Francis, 2014. P. 1–39.
4. Petrosino A., Fanelli A.M., Pedrycz W. Fuzzy Logic and Applications. Springer, 2011. 290 p.
5. Borisov V.V. Hybridization of Intellectual Technologies for Analytical Tasks of Decision-Making Support // Journal of Computer Engineering and Informatics. 2014. Vol. 2. Is. 1. P. 148–156.
6. Борисов В.В., Федулов А.С. Модель интеллектуального управления рисками при обеспечении информационной безопасности процессов в сложных социотехнических системах // Нейрокомпьютеры и их применение: тез. докладов XVII Всероссийской научной конференции. – М.: ФГБОУ ВО МГППУ, 2019. С. 110.
7. Бобков В.И., Борисов В.В., Дли М.И. Подход к исследованию теплопроводности нечеткими численными методами в условиях неопределенности теплофизических характеристик // Системы управления, связи и безопасности. 2017. № 3. С. 73–83.

V.V. Borisov, A.S. Fedulov

Branch of the National Research University
«Moscow Power Engineering Institute» in Smolensk

MODEL AND METHOD FOR INTELLIGENT RISK MANAGEMENT OF INFORMATION SECURITY BREACH OF PROCESSES IN SOCIOTECHNICAL SYSTEMS

Keywords: *intellectual risk management, information security, socio-technical system, fuzzy model.*

Abstract. *A formalized formulation of the problem of managing the risks of violation of information security of processes in sociotechnical systems has been performed. A model of intellectual risk management of information security of processes in sociotechnical systems is proposed, based on fuzzy and neuro-fuzzy approaches, as well as on methods of hybridization and integration of fuzzy and neuro-fuzzy models. The method of intellectual risk management of information security of processes in sociotechnical systems is presented. It is based on the developed model and consists in determining combinations of control parameters that ensure reduction of social and*

technical costs for processes in sociotechnical systems without exceeding the permissible level of information security risk.

Е.И. Гончаров, В.И. Мунерман, Т.А. Самойлова
Смоленский государственный университет

УДК 004.056.55

ВЫБОР ПАРАМЕТРОВ МНОГОМЕРНЫХ МАТРИЦ ДЛЯ ОБОБЩЕННОГО АЛГОРИТМА ШИФРОВАНИЯ ХИЛЛА

Ключевые слова: *алгоритм Хилла, алгебра многомерных матриц, кодирование, декодирование.*

В статье рассмотрено обобщение алгоритма шифрования Хилла. При рассмотрении операций с многомерными матрицами показывается, что они включают в себя все операции с двумерными матрицами как частный случай, что позволяет естественным образом обобщить алгоритм Хилла на многомерные матрицы. Операции алгебры многомерных матриц отличаются тем, что единичная матрица определена неоднозначно, не всегда выполняется закон ассоциативности, возможностью определения множества операций умножения одних и тех же матриц. Эта вариативность существенно повышает криптостойкость алгоритма шифрования. В статье предложен метод выбора параметров многомерных матриц для алгоритма шифрования Хилла.

Постановка задачи. В общем виде алгоритм Хилла имеет следующий вид. Пусть имеются алфавит θ мощности M и кодируемый текст T . Символам алфавита ставятся в соответствие числа $0, 1, \dots, M-1$, а тексту – квадратная матрица A – кодируемая матрица. B – кодирующая матрица, совпадающая по размерам с A и имеющая обратную B^{-1} по mod M . Результатом кодирования текста будет зашифрованная матрица $D = A \times B \pmod{M}$. Декодирование заключается в вычислении $D \times B^{-1} \pmod{M} = ((A \times B) \times B^{-1}) \pmod{M} = A$.

В статье предлагается использовать для кодирования больших текстов не обычные плоские, а многомерные матрицы, рассмотрены многомерно матричные аналоги операций, используемых в алгоритме Хилла, и определены параметры многомерных матриц, необходимые для обобщения алгоритма Хилла. В дальнейшем считается, что операция умножения и обратная к ней операция деления элементов матриц реализована по модулю M .

Элементы алгебры многомерных матриц. Многомерная матрица порядка N и размерности p определяется как система элементов $A_{i_1 i_2 \dots i_p}$, расположенных в точках p -мерного пространства и определяемых координатами (i_1, i_2, \dots, i_p) , где $i_a = 1, 2, \dots, N$; $a = 1, 2, \dots, p$.

Пусть матрицы $A = \parallel a_{i_1 \dots i_p} \parallel$ и $B = \parallel b_{i_1 \dots i_q} \parallel$, p - и q -мерные соответственно. Совокупности индексов этих матриц i_1, \dots, i_p и i_1, \dots, i_q разбиваются на четыре группы, содержащие соответственно κ , λ , μ и ν индексов ($\kappa, \lambda, \mu, \nu \geq 0$). Причем $\kappa + \lambda + \mu = p$, а $\lambda + \mu + \nu = q$. Для полученных групп индексов используются обозначения: $l = (l_1, \dots, l_\kappa)$, $s = (s_1, \dots, s_\lambda)$, $c = (c_1, \dots, c_\mu)$ и $m = (m_1, \dots, m_\nu)$. Тогда матрицы A и B можно представить в виде $A = \parallel a_{lsc} \parallel$ и $B = \parallel b_{scm} \parallel$. Индексы групп s и c в матрицах A и B полностью совпадают. Так же как в операции свертки, индексы разбиения c называются кэлиевыми. Индексы разбиения s называются скоттовыми, а индексы разбиения m , так же как и индексы разбиения l , – свободными.

Матрица $C = \parallel c_{lsm} \parallel$, элементы которой вычисляются по формуле

$$c_{lsm} = \sum_{(c)} a_{lsc} \times b_{scm},$$

называется (λ, μ) -свернутым произведением матриц

A и B . Число всех возможных (λ, μ) -свернутых произведений p -мерной матрицы A на q -мерную матрицу B вычисляется по формуле

$$N_{p,q} = \sum_{\lambda+\mu}^{\min(p,q)} \frac{p!}{\lambda! \mu! (p-\lambda-\mu)!} \cdot \frac{q!}{\lambda! \mu! (q-\lambda-\mu)!}.$$

На рисунке 1 показано, как быстро это количество возрастает с ростом размерностей матриц.

q	p									
	1	2	3	4	5	6	7	8	9	10
1	3	5	7	9	11	13	15	17	19	21
2		15	31	53	81	115	155	201	253	311
3			93	213	411	707	1121	1673	2383	3271
4				639	1151	3239	6063	10453	16909	26001
5					4653	11723	25873	51573	94903	163805
6						35169	90945	209061	437377	847735
7							272835	719253	1705819	3308015
8								2157759	5773279	14034335
9									17319837	46889355
10										140668065

Рис. 1. Зависимость числа (λ, μ) -свернутых произведений от p и q

Замечание. (0,1)-свернутое произведение двух двумерных матриц порядка N совпадает с умножением в традиционной алгебре матриц. Поэтому алгоритм шифрования Хилла есть частный случай своего естественного обобщения на многомерные матрицы.

(λ, μ) -единичная матрица определяется как решение уравнения ${}^{\lambda, \mu}(A \times X) = A$, где $0 \leq \lambda + \mu \leq p$, и обозначается $E(\lambda, \mu)$. Из определения следует, что $E(\lambda, \mu)$ имеет размерность $\lambda + 2\mu$, а каждый её элемент находится как $E_{csm} = \delta_{cm}$ (символ Кронекера). Очевидно, $E(0, 1)$ размерности 2 есть обычная двумерная единичная матрица.

Для каждой пары (λ_0, μ_0) в уравнении ${}^{\lambda, \mu}(A \times X) = A$, $E(\lambda_0, \mu_0)$ определяется однозначно. А в общем случае, для любой многомерной матрицы A размерности p существует $\frac{(p+1)(p+2)}{2}$ различных единичных матриц. Например, при изменении p от 12 до 24 число единичных матриц изменяется от 91 до 325.

Многомерная матрица $A^{-1}(\lambda, \mu)$, которая определяется как решение уравнения ${}^{\lambda, \mu}(A \times X) = E(\lambda', \mu')$, где $0 \leq \lambda' + \mu' \leq p$ и $0 \leq \lambda + \mu \leq p$ называется правой (λ, μ) -обратной к многомерной матрице A относительно единичной матрицы $E(\lambda', \mu')$.

Нахождение обратной матрицы сводится к решению системы линейных уравнений: $\sum_{(c)} A_{lsc} X_{csm} = E_{lsm}$. Она состоит из $N^{\lambda'+2\mu'}$ линейных уравнений, содержащих $N^{\lambda'+2\mu'+\lambda+2\mu-p}$ неизвестных. Существование решений этой системы и обуславливает существование $A^{-1}(\lambda, \mu)$ относительно $E(\lambda', \mu')$.

Выбор параметров многомерных матриц для алгоритма шифрования Хилла. Далее, без нарушения общности, рассматриваются только многомерные матрицы порядка N и «умножение справа». Пусть размерность кодируемой матрицы A равна p . $A = \|A_{lsc}\|$, где $l = (l_1, \dots, l_\kappa)$, $s = (s_1, \dots, s_\lambda)$, $c = (c_1, \dots, c_\mu)$. Размерность кодирующей матрицы B равна q , причем $0 \leq \lambda + \mu \leq q$. $B = \|B_{csm}\|$, где $m = (m_1, \dots, m_\nu)$. Пусть $E(\lambda, \mu)$ некоторая (λ, μ) -единичная матрица. Для того, чтобы определить B^{-1} – правую (λ', μ') -обратную матрицу к матрице B относительно $E(\lambda, \mu)$, необходимо новое разбиение индексов матрицы B : $B = \|B_{csm}\| = \|B_{l's'c'}\|$, где $l' = (l'_1, \dots, l'_\kappa)$, $s' = (s'_1, \dots, s'_\lambda)$, $c' = (c'_1, \dots, c'_\mu)$. Задача построения системы кодирования состоит в выборе пары (λ', μ') таким образом, чтобы выполнялось условие ассоциативности в произведении

$${}^{\lambda', \mu'}(\lambda, \mu(A \times B) \times B^{-1}) = {}^{\lambda, \mu}(A \times {}^{\lambda', \mu'}(B \times B^{-1})).$$

Известно [1], что ассоциативный закон умножения матриц одного и того же порядка $({}^{\lambda',\mu'}({}^{\lambda,\mu}(A \times B) \times B^{-1}) = {}^{\lambda,\mu}(A \times {}^{\lambda',\mu'}(B \times B^{-1})))$ имеет место при условии, что свертываемые совокупности кэлиевых индексов в разных парах перемножаемых матриц не пересекаются между собой и с совокупностями скоттовых индексов. Возможны следующие варианты выбора разбиений.

I. Количество уравнений равно количеству неизвестных в системе уравнений при нахождении B^{-1} , то $\kappa' = \mu'$. Тогда:

а) $\lambda' = \lambda$, $\mu = \mu'$, то есть совокупности кэлиевых индексов в разных парах перемножаемых матриц не пересекаются между собой и со скоттовыми индексами. На рисунке 2 приведен этот случай (кэлиевые индексы отмечены вертикальной штриховкой, а скоттовые – клетчатым узором). Такой выбор параметров делает все матрицы одной размерности $\lambda + 2\mu$.

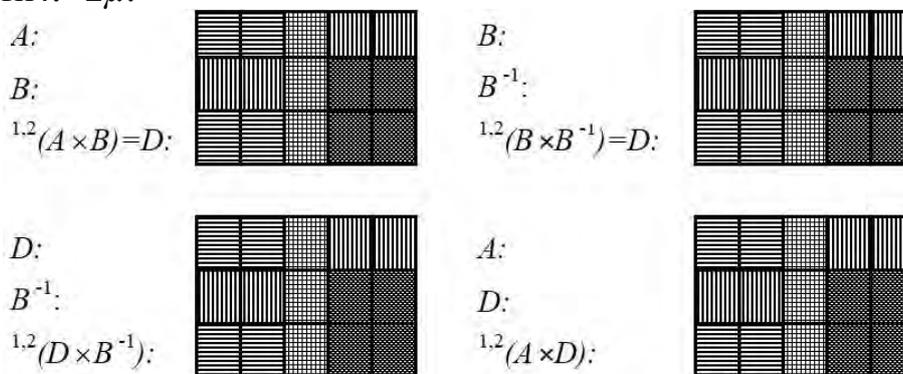


Рис. 2. Конфигурация параметров для случая I. а)

б) $B = \| \| B_{csm} \| \| = \| \| B_{csm's'c'} \| \|$ – разбиение индексов матрицы B , где $m' = (m'_1, m'_2, \dots, m'_{\mu'-\mu-\lambda})$, $m = m's'c'$, $csm' = l'$. Из доказанного в [1], что в этом случае для ассоциативности необходимо, чтобы s и s' не пересекались между собой, с s и s' .

Тогда $\mu' = 2\mu + \lambda$, размерность B^{-1} равна μ' и $\lambda' = 0$. То есть разбиение индексов матрицы B^{-1} состоит только из кэлиевых индексов: $B^{-1} = \| \| B^{-1}_{c'} \| \|$ (рис. 3). Следовательно, размерность матрицы A $p = \mu' = 2\mu + \lambda$, размерность B равна $2p$ и размерность B^{-1} – p .

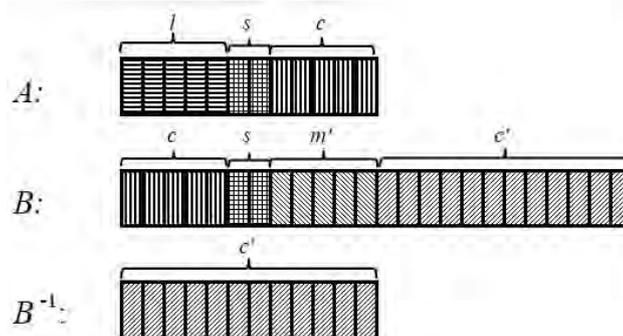


Рис. 3. Конфигурация параметров для случая I. б)

II. Если количество уравнений больше числа неизвестных, то есть $\kappa' > \mu'$, разбиение матрицы $B^{-1} = \|B^{-1}_{c's'm''}\|$ будет состоять из c' , s' , $m'' = (m_1'', m_2'', \dots, m_{\lambda+2\mu-\lambda'-\mu'}'')$. Тогда $\lambda' + \mu' \leq \lambda + 2\mu$, размерность A равна размерности B^{-1} и равна $\lambda + 2\mu$, а размерность $B = \lambda' + \mu' + \lambda + 2\mu$. На рисунке 4 показана возможная конфигурация, но на практике B^{-1} или не будет существовать, или после удаления повторяющихся уравнений задача сведется к одному из предыдущих случаев.

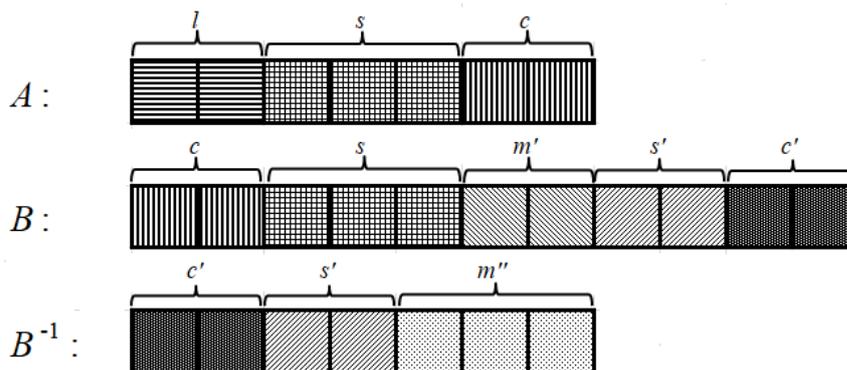


Рис. 4. Конфигурация параметров для случая II

III. Если количество уравнений меньше числа неизвестных, то это приведет к появлению большого количества обратных матриц, что снизит криптоустойчивость системы.

Задача выбора разбиений, сверток произведений, как и подбор обратной матрицы, решается один раз при создании системы кодирования. Именно от этих выборов зависит криптоустойчивость системы. Процесс кодирования заключается в выполнении (λ, μ) -свернутого произведения матриц A и B . $D = {}^{\lambda, \mu}(A \times B)$, где D – зашифрованная матрица. Процесс декодирования заключается в выполнении (λ', μ') -свернутого произведения матриц A и B^{-1} – ${}^{\lambda', \mu'}(D \times B^{-1}) = {}^{\lambda', \mu'}({}^{\lambda, \mu}(A \times B) \times B^{-1}) = {}^{\lambda, \mu}(A \times {}^{\lambda', \mu'}(B \times B^{-1})) = {}^{\lambda, \mu}(A \times E(\lambda, \mu)) = A$. Аналогично строится и алгоритм для «умножения слева».

Вывод. Выбор алгебры многомерных матриц позволяет:

1) качественно увеличить криптостойкость алгоритма Хилла в силу специфики умножения многомерных матриц, вариативности выбора единичной, выбора размерностей.

2) аналогично плоским матрицам возможно распараллеливание операций умножения многомерных матриц и обратных матриц [2].

Литература

1. Соколов Н.П. Введение в теорию многомерных матриц. Киев: Изд-во «Наукова Думка», 1972. 177 с.

2. Мунерман В.И. Архитектура программно-аппаратного комплекса для массовой обработки данных на базе многомерно-матричной модели // Системы высокой доступности. 2015. Т. 11, №. 2. С. 13-18.

E.I. Goncharov, V.I. Munerman, T.A. Samoylova
Smolensk State University

THE METHOD OF SELECTING PARAMETERS OF MULTIDIMENSIONAL MATRIX FOR HILL ENCRYPTION ALGORITHM

Keywords: *Hill's algorithm, algebra of multidimensional matrices, coding, decoding.*

Abstract. *The article discusses the generalization of Hill's encryption algorithm. When considering operations with multidimensional matrices, it is shown that they include all operations with two-dimensional matrices as a special case, which makes it possible to naturally generalize Hill's algorithm to multidimensional matrices. The operations of the algebra of multidimensional matrices differ in that the identity matrix is not uniquely defined, the law of associativity is not always fulfilled, and the possibility of determining the set of multiplication operations of the same matrices is not always fulfilled. This variability significantly increases the cryptographic strength of the encryption algorithm. The article proposes a method for selecting parameters of multidimensional matrices for Hill's encryption algorithm.*

Ю.А. Грибер, Т.А. Самойлова
Смоленский государственный университет

УДК 004.9

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В СОЦИОЛОГИЧЕСКОМ ИССЛЕДОВАНИИ ЦВЕТОНАИМЕНОВАНИЙ И ЦВЕТОВЫХ ПРЕДПОЧТЕНИЙ¹

Ключевые слова: *социологические исследования, цветонаименования, цветовые предпочтения, анализ данных, кластеризация, метрика цветового отличия, SQL, Python.*

¹ Публикация подготовлена в рамках поддержанного РФФИ научного проекта № 17-29-09145 «Картография цвета: диагностика развития цветонаименований русского языка с использованием естественно-научных, историографических, социологических и психологических методов».

В статье рассматриваются основные возможности, которые дают информационные технологии на разных этапах социологических исследований цветоименований и цветовых предпочтений. В качестве исходных данных берутся результаты психолингвистических и социально-психологических экспериментов, содержащие нечисловые значения ответов респондентов в виде перечня цветоименований. Предложена методика обработки данных, основанная на SQL-запросах и возможностях Python-библиотек. Для кластеризации цветовых предпочтений предложен способ расчета метрики их сходства по расстоянию в цветовом пространстве CIELAB.

Прогресс в области информационных технологий и программных продуктов на их основе неразрывно связан с успехами в гуманитарных исследованиях. Особенно это справедливо для совершенствования методик применения информационных технологий в обработке результатов психолингвистических (см., например, [1; 2]) и социально-психологических экспериментов (см., например, [2; 3]), содержащих нечисловые значения цветоименований и поэтому плохо сочетающихся с традиционными средствами анализа числовой информации. Цель данной работы – выбор информационных технологий, обеспечивающих эффективность основных этапов опросных исследований цветовых предпочтений: сбора данных, их очистки, анализа и визуализации. С учетом номинального характера данных наиболее сложным представляется этап анализа, в котором затруднено применение традиционной статистики. Для решения проблемы предлагается нечисловые значения цветоименований обрабатывать эффективными средствами SQL-запросов реляционных СУБД и алгоритмами интеллектуального анализа.

Сбор данных. Наиболее результативным методом сбора информации для изучения цветоименований и цветовых предпочтений является онлайн-эксперимент с помощью инструментов, размещаемых на интернет-ресурсах (Computer Assisted Web Interviewing – CAWI). Такое исследование, хотя и проводится без личного участия интервьюера, имеет большие преимущества в виде демонстрации различных цветных видеоматериалов и изображений. Программные средства его реализации достаточно разнообразны: это языки серверных сценариев PHP, ASP, Django, Ruby, имеющие хорошую репутацию в подобных разработках. При этом данные собираются в таблицы реляционной СУБД, средствами которой они могут легко преобразовываться в популярные форматы, предназначенные для вторичного анализа социологом-исследователем, например в формат Excel. Недостаток изучения цветоименований и цветовых предпочтений в режиме онлайн – невозможность контроля процесса

заполнения анкеты. Поскольку респонденты отвечают на вопросы самостоятельно, из-за невнимательности, неграмотности или по другим причинам они часто допускают опечатки при записи имен цвета. В результате полученная база данных до анализа нуждается в предварительной очистке.

Очистка данных. Наиболее проблемным для последующего компьютерного анализа является перечень цветоименований, полученный с помощью открытых вопросов. В отличие от закрытых вопросов открытые не содержат подсказок, не «навязывают» тот или иной вариант ответа и рассчитаны на получение неформальных мнений респондентов, задаваемых средствами компьютерного ввода. Конечно, возникающие при этом ошибки можно править и вручную, но при больших объемах ввода это становится проблематичным и требует компьютерной обработки программными макросами.

Пример «загрязненных» данных:

<i>корочневый</i>
<i>корраловый</i>
<i>корричневый</i>
<i>коррчневый</i>
<i>кортчневый</i>
<i>корчнево-зеленый.</i>

Исправление хорошо выполняется с помощью машинного словаря и включает два шага: обнаружение ошибок и непосредственную очистку. На первом шаге программный макрос составляет упорядоченное множество уникальных цветоименований – словарь. После этого исследователь вручную создает из словаря таблицу замены, сопоставив с каждым ошибочным цветоименованием соответствующее верное.

Пример таблицы замены:

<i>фиолетовый светло</i>	<i>светло-фиолетовый</i>
<i>фиолето-малиновый</i>	<i>фиолетово-малиновый</i>
<i>фиолетовый</i>	<i>фиолетовый</i>
<i>фиотетовый</i>	<i>фиолетовый</i>
<i>фурия</i>	<i>фуксия</i>
<i>фусия</i>	<i>фуксия</i>

Второй (заключительный) шаг очистки выполняется программным макросом, исправляющим ошибочные цветоименования на правильные, взятые из таблицы замены.

Анализ данных средствами SQL. С учетом хорошей структурированности информации о цветоименованиях и цветовых предпочтениях наиболее подходящим инструментом ее обработки представляется система управления базами данных, включающая язык структурированных запросов SQL (Structured Query Language). Возможности SQL позволяют найти ответы на интересующие

исследователя вопросы без применения сложных алгоритмов анализа, извлекая необходимую информацию и обрабатывая ее различными способами, включая приемы агрегации.

Для примера рассмотрим, как проходит поиск пар несовпадающих цветоименований в ответах респондентов, выполненный средствами SQL. Исходные данные, соответствующие ответам мужчин и женщин, размещаются в двух таблицах – Consistency_Male и Consistency_Female. Обе таблицы имеют одинаковую структуру, включающую два выбранных цветоименования и признак одинаковости выбора: {PRESENT_1ST, PRESENT_2ND, Same}. Обработка данных каждой из таблиц состоит из пяти этапов (имена таблиц в описаниях этапов соответствуют ответам мужчин, обработка ответов женщин проходит аналогично).

Этап 1. Отбор из таблицы Consistency_Male ответов, у которых значение признака одинаковости выбора Same имеет значение False и их размещение в новой таблице Consistency_Male_false выполняет запрос:

```
INSERT INTO Consistency_Male_false (PRESENT_1ST, PRESENT_2ND)
SELECT Consistency_Male.PRESENT_1ST, Consistency_Male.PRESENT_2ND
FROM Consistency_Male
WHERE (((Consistency_Male.Same) = False));
```

Этап 2. Составление двух списков – запросов LIST1, LIST2, содержащих количества уникальных цветоименований (LIST1 – цветоименования PRESENT_1ST, LIST2 – цветоименования PRESENT_2ND):

```
LIST1: SELECT Consistency_Male_false.PRESENT_1ST,
Count(Consistency_Male_false.PRESENT_1ST) AS [Count_PR_1ST]
FROM Consistency_Male_false
GROUP BY Consistency_Male_false.PRESENT_1ST;
LIST2: SELECT Consistency_Male_false.PRESENT_2ND,
Count(Consistency_Male_false.PRESENT_2ND) AS [Count_PR_2ND]
FROM Consistency_Male_false
GROUP BY Consistency_Male_false.PRESENT_2ND;
```

Этап 3. Поиск для каждого цветоименования PRESENT_1ST из LIST1 всех его пар из колонки PRESENT_2ND таблицы Consistency_Male_false и размещение полученного нового списка в таблицу результата Result = {Color_name1, Color_name2}, где Color_name1, Color_name2 – найденная пара несовпадающих цветоименований, поиск которой выполняется объединением списка LIST1 и таблицы оператором:

```
INSERT INTO RESULT (Color_name1, Color_name2)
SELECT List1.PRESENT_1ST, Consistency_Male_false.PRESENT_2ND FROM
List1 INNER JOIN Consistency_Male_false
ON List1.PRESENT_1ST = Consistency_Male_false.PRESENT_1ST
ORDER BY List1.PRESENT_1ST;
```

Этап 4. Поиск для каждого цветоименования Present_2nd из LIST2 всех его пар из колонки PRESENT_1ST таблицы Consistency_Male_false и добавление полученного нового списка к данным таблицы Result (выполняется также оператором INNER JOIN):

```
INSERT INTO RESULT (Color_name1, Color_name2)
SELECT List2.PRESENT_2ND, Consistency_Male_false.PRESENT_1ST FROM
List2 INNER JOIN Consistency_Male_false
ON List2.PRESENT_2ND = Consistency_Male_false.PRESENT_2ND
ORDER BY List2.PRESENT_2ND;
```

Этап 5. Слияние списков, содержащихся в таблице результатов RESULT, в единую упорядоченную структуру средствами оператора ORDER BY с помощью запроса:

```
SELECT * FROM RESULT ORDER BY Color_name1, Color_name2
```

Таким образом, заключительный этап обработки формирует необходимый исследователю список пар несовпадающих в ответах цветоименований.

Интеллектуальный анализ данных (ИАД). Задача интеллектуального анализа – это выявление в данных скрытых связей и закономерностей. Ее решение – всегда сложный процесс, прежде всего для социолога-исследователя. Несмотря на появление в последнее время необходимых многочисленных программных средств с удобным интерфейсом, исследователю оказывается непросто как поставить подобную задачу, так и оценить полезность полученных результатов. Обычно использование элементов интеллектуального анализа реализуется в неполной постановке задачи. Классическим примером является кластерный анализ, проводя который исследователь только задает классифицирующие признаки (их выбор и есть неполная гипотеза), но заранее не может предположить ни состав, ни объем кластеров. Более того, при проведении кластерного анализа приходится перебирать «вручную» гипотезы о количестве кластеров.

Опыт выявления скрытых связей между цветовыми предпочтениями участников эксперимента показывает наибольшую эффективность алгоритмов Apriori и FP-Growth для поиска ассоциативных правил в ответах анкетированных [4] и метода иерархической агломеративной кластеризации для группировки ответов [3]. Реализация алгоритмов в обоих случаях выполняется средствами Python-пакетов pandas, scipy, numpy, scikit-learn. В качестве исходных данных задачи кластеризации берутся ответы респондентов в виде строковых значений цветоименований.

Иерархический алгоритм объединения ответов в кластеры выглядит следующим образом.

Шаг 1. Создается первичный набор кластеров, каждый из которых содержит один ответ-цветоименование.

Шаг 2. Рассчитывается матрица расстояний между кластерами по метрике цветового отличия.

Шаг 3. Определяются два наиболее близких по матрице расстояний кластера.

Шаг 4. Наиболее близкие кластеры объединяются по правилу Уорда [5].

Шаг 5. Шаги 2–4 повторяются до тех пор, пока все объекты не будут объединены в один кластер.

В случае кластеризации цветоименований и цветовых предпочтений традиционное евклидово пространство заменяется цветовой моделью LAB, в которую переводятся полученные вербальные данные. Модель LAB создана международной комиссией по освещению (Communication Internationale de l'Eclairage) с целью стать аппаратно независимой моделью. В отличие от других цветковых моделей LAB впервые позволила описывать оттенки не в терминах элементов, воспроизводимых устройствами, а с использованием трех составляющих цветового зрения человека. Цветовое пространство LAB представляет цвет в трех каналах: один канал выделен для значений яркости (L – Lightness) и два других – для цветовой информации (a и b). Канал a представляет непрерывный спектр от зеленого к красному, канал b – от синего к желтому. Средние значения для a и b соответствуют реальным оттенкам серого (рис. 1).

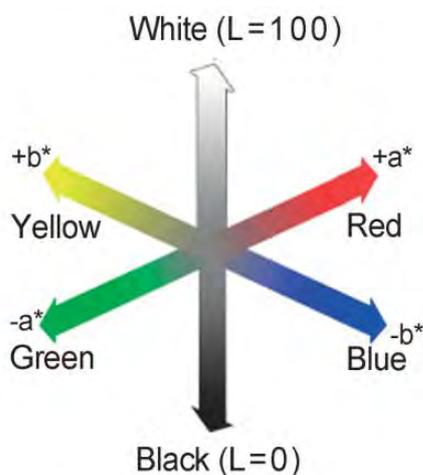


Рис. 1. Каналы цветового пространства LAB

В модели LAB расчет функции расстояния для кластеризации цветоименований и цветовых предпочтений проводится по метрике цветового отличия Delta E2000. Согласно данной метрике, разница между выбранными респондентами цветами i и j – ΔE_{ij} , которой соответствуют значения L_i , a_i , b_i и L_j , a_j , b_j , вычисляется по разработанному LAB набору формул [6]. Суффикс E – первая буква немецкого слова *Empfindung*, которое в широком смысле означает «ощущение». Полученное значение соответствует мере изменения

зрительного восприятия двух данных цветов для понимания того, как человеческий глаз воспринимает разницу между ними. В типичном масштабе значение ΔE будет варьироваться в диапазоне от 0 до 100. Если значение $\Delta E \leq 1$ для двух несоприкасающихся цветов, то обычный человек с трудом воспримет различие.

В результате работы Python-программы формируется дендрограмма иерархической агломеративной кластеризации цветовых предпочтений респондентов (рис. 2).

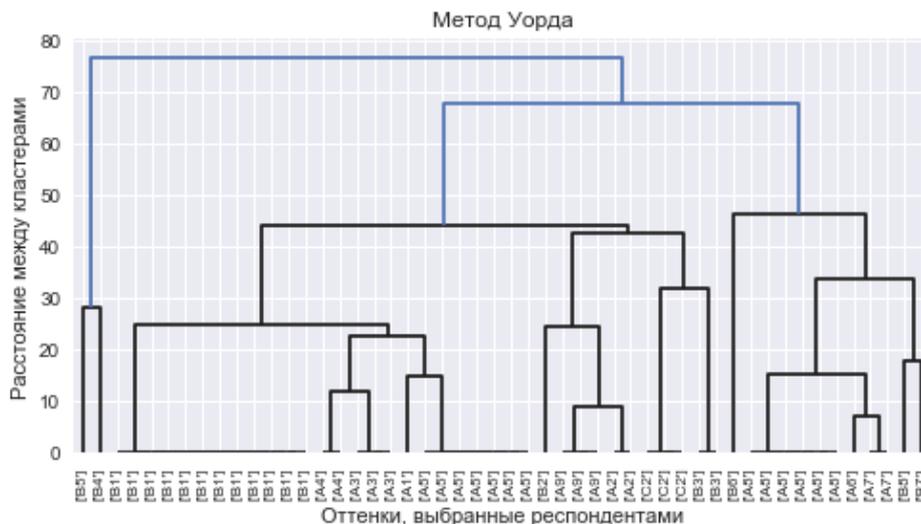


Рис. 2. Дендрограмма иерархической агломеративной кластеризации цветовых предпочтений респондентов

Матрица расстояний `distance_matrix`, непосредственно используемая алгоритмом, вычисляется с учетом метрики цветового отличия.

Фрагмент листинга программы:

```
# Листинг 1
import scipy.cluster.hierarchy as sch
import matplotlib.pyplot as plt
L = sch.linkage(distance_matrix, method= 'ward')
plt.figure(figsize=(6, 10))
sch.set_link_color_palette(['c', 'g', 'r', 'k'])
dn = sch.dendrogram(L, labels=rows, orientation='right')
plt.show()
```

Визуализация данных используется для выявления закономерностей, которые не очевидны в анализе обычных числовых результатов. В настоящее время существует бесплатное программное обеспечение, которое помогает сделать необходимое для исследователя представление данных легко и без лишних усилий. Так, для визуализации результатов обработки цветовых предпочтений хорошо показали себя Python-библиотеки Matplotlib и Tkinter, которые позволяют представить результаты кластеризации, соответствующие

дендрограмме (рис. 2), отображая сгруппированные ответы в соответствующей цветовой палитре (рис. 3).

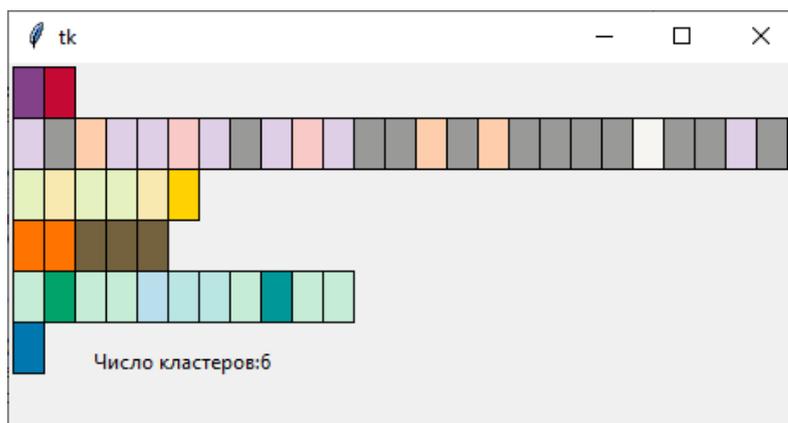


Рис. 3. Визуализация результатов кластерного анализа

В целом использование современных информационных технологий на сегодняшний день является неотъемлемым условием эффективности социологического изучения цветонаименований и цветовых предпочтений. Учет специфики их применения на различных этапах обработки данных положительно влияет на качество полученных результатов и доказывает необходимость дальнейшей активизации работы в этом направлении. При этом применение «простых» для современной программной реализации методов интеллектуального анализа сопровождается весьма трудоемкой интерпретацией результатов, по-прежнему проводимой исключительно социологом-исследователем.

Литература

1. Paramei G.V., Griber Y.A., Mylonas D. An online color naming experiment in Russian using Munsell color samples // *Color Research and Application*. 2018. Vol. 43, Is. 3. P. 358–374.
2. Грибер Ю.А., Самойлова Т.А. Геронтологическая специфика цветовых предпочтений, связанных с выбором цвета фасада жилого дома // *Урбанистика*. 2018. № 3. С. 99–113.
3. Грибер Ю.А., Самойлова Т.А., Двойнев В.В. Цветовые предпочтения пожилых людей в различных типах жилого интерьера // *Урбанистика*. 2018. № 4. С. 3649.
4. Грибер Ю.А., Самойлова Т.А. Поиск ассоциативных правил в результатах экспериментального исследования цветовых предпочтений // *Системы компьютерной математики и их приложения*. Смоленск: Изд-во СмолГУ, 2018. Вып. 19. С. 118–124.
5. Воронцов К.В. Алгоритмы кластеризации и многомерного шкалирования. М.: МГУ, 2007.
6. Sharma G., Wu W., Dalal E.N. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations // *Color Research and Application*. 2005. Vol. 30, No. 1. P. 21–30.

Yu.A. Griber, T.A. Samoylova
Smolensk State University

INFORMATION TECHNOLOGIES IN THE SOCIOLOGICAL RESEARCH OF COLOR NAMES AND COLOR PREFERENCES

Keywords: *sociological research, color naming, color preferences, data analysis, clustering, color difference formula, SQL, Python.*

Abstract. *The article discusses the key opportunities that information technologies provide at different stages of sociological research on color naming and color preferences. The initial data is taken from the results of psycholinguistic and psychological experiments and contains non-numeric meanings of the respondents' answers in the form of a list of color names. For processing data, we offer a technique based on SQL queries and capabilities of Python libraries. For clustering color preferences, we propose a method for calculating their similarities by distance in the CIELAB color space.*

**Г.М. Григорьева, А.И. Миронов,
В.И. Мунерман, В.Ю. Ходченков**

Смоленский государственный университет

УДК 004.022:004.027

АЛГОРИТМ СЛОЖЕНИЯ ВЕКТОРОВ НА КВАНТОВОМ КОМПЬЮТЕРЕ

Ключевые слова: *квантовые вычисления, квантовый параллелизм, векторный процессор.*

В статье рассматривается возможность создания векторного процессора на основе квантовых вычислений. Необходимость решения этой проблемы вытекает из того факта, что существует широкий класс задач, формализованных в векторной и матричной алгебрах. Квантовый параллелизм позволяет одновременно складывать несколько пар чисел. Однако основные подходы в литературе сводятся к созданию обычных сумматоров, таких как сумматоры современных компьютеров. В статье предлагается метод построения квантового векторного процессора для сложения k -мерных (k – степень 2) векторов. Сумматор такого процессора состоит из k элементов,

которые содержат пару координат в интервале $[0, 1]$. Приведен вычислительный эксперимент.

Введение. В статье обсуждается возможность построения эффективных средств обработки больших объемов данных (Big data). Реляционные СУБД в основном используются для обработки структурированных данных. Разработчики этих СУБД реализуют различные методы оптимизации запросов, в том числе с помощью методов параллельной обработки данных. Повышение эффективности обработки реляционных баз данных также возможно за счет использования параллельных вычислений на основе алгебры многомерных матриц, которая изоморфна реляционной алгебре [1]. Алгоритмы, реализующие операции алгебры многомерных матриц, легко распараллеливаются с использованием компьютерных систем с SIMD-архитектурой. Одна из разновидностей SIMD систем – это векторные процессоры. Разрабатываемые в настоящее время квантовые компьютеры наиболее подходят для реализации векторных процессоров. Это объясняется тем фактом, что они могут одновременно выполнять простые операции с произвольным фиксированным количеством данных. Далее мы рассмотрим возможность построения квантового векторного сумматора, который будет выполнять сложение двух единичных векторов, размерность которых равна $2t$ (t определяется количеством доступных кубитов). Такой квантовый сумматор может служить основой для построения векторного процессора, поскольку любой вектор 2-мерного векторного пространства может быть представлен в виде линейной комбинации единичных векторов.

Принцип построения квантового векторного сумматора. Предложенный подход основан на сохранении в первой координате кубита первой координаты нормализованного двумерного вектора, в первой координате второго кубита первой координаты второго вектора и последующем дублировании для специфического измерения сложившейся последовательности кубит, в результате чего будут получены данные для нахождения суммы векторов (вторая координата суммы вычисляется аналогично в одно время с первой). Это открывает возможность атомарной обработки единичных векторов, которые могут составлять основу произвольного векторного пространства четной размерности. Требование четности объясняется тем, что из двумерных векторов можно построить пространство только такой размерности.

Описание алгоритма. Предположим, есть два вектора a и b , такие что: $a \in Q^2$, $b \in Q^2$

И норма каждого из этих векторов: $\|a\| = \|b\| = r$.

Положим $a' = \frac{a}{r}$, $b' = \frac{b}{r}$.

Тем самым мы приводим оба вектора к единичной норме.

Затем значения a' , b' поступают на вход квантового сумматора. Первая координата первого вектора записывается в первую координату первого кубита как $\sqrt{a1'}$. Первая координата второго вектора записывается в первую координату второго кубита как $\sqrt{b1'}$ (см. рис. 1). После этого последующие: третий, пятый, ..., $2n + 1$ кубиты принимают значение, эквивалентное первому кубиту, аналогично четвертый, шестой, ..., $2n + 2$ кубиты принимают значение, эквивалентное второму кубиту. С увеличением количества дублирующих векторов возрастает точность вычисления. Далее, следует увеличение первых координат $2n+1$ -го и $2n$ -ого кубита в k^n раз. Дополнительные кубиты нужны для того, чтобы в результате процесса измерения кубит получить не одно из базисных значений (0 или 1), а совокупность базисных состояний, которая определит насколько значение координаты изначальных кубит близко к 0 или 1 (аналогично как и их произведение, если значимым считать не одно измерение а два последовательных) при помощи изменения вероятности с заданным шагом и последующем измерении кубит. Процесс измерения кубит – это процесс перехода состояния кубита в одно из своих базисных состояний, зависящий от координат кубита – они выступают в роли вероятностей перехода кубита в соответствующие базисные состояния. Измерим каждый кубит. В результате будет получена последовательность нулей и единиц – измеренных кубитов в базисных состояниях. Затем находится номер m , с которого каждые две позиции не содержат пары нулей, что указывает на то, что, более вероятно, кубит переместится во второе базисное состояние, нежели в первое. Следовательно, возможно определить произведение первых координат пар кубитов по формуле:

$$\frac{1}{a1' \times b1'} = k^{2m} \times \left(1 + \frac{1}{k^2}\right) \quad (1)$$

Поскольку число m означает число, после которого знак неравенства квадратов координат изменился, учитывая выбор шага k и количество дополнительных кубитов, получены пределы определения $a1' \times b1'$, само значение можно взять как среднее между пределами.

Пусть правая часть (1) – X .

Точно так же, зная число p , с которого последовательность будет содержать единицы в каждой позиции, следующее выражение определяется как:

$$\frac{1}{1 - (a1' + b1') + a1' \times b1'} = k^{2p} \times (1 + k^2) \quad (2)$$

Пусть правая часть (2) – Y .

Учитывая вышеизложенное:

$$a1' + b1' = \frac{1}{X} - \frac{1}{Y} + 1$$

Аналогично для вторых координат векторов.

Таким образом, получена сумма первой и второй координат векторов, что означает, что может быть получена сумма векторов при замене:

$$a + b = (a' + b') \times r$$

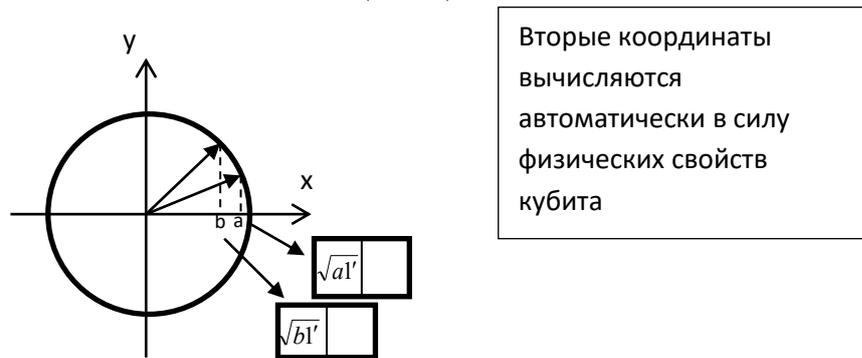


Рис. 1. Графическое описание координат

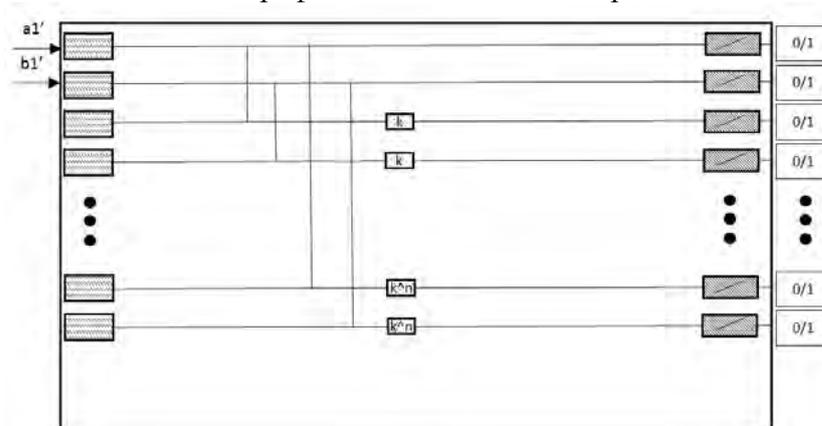


Рис. 2. Схема, описывающая предложенный алгоритм

Графическая интерпретация координат векторов приведена на рисунке 1. На рисунке 2 приведена графическая интерпретация алгоритма сложения векторов:

- координаты векторов **a** и **b** поступают на вход соответствующих кубитов;
- вертикальные линии показывают состояние передачи кубита;
- далее «точечный» прямоугольник – это процесс увеличения первой координаты соответствующего кубита кратно значению, указанному внутри прямоугольника;
- после этого штрихованный прямоугольник – это измерение состояния кубита, которое превращает кубит в одно из базовых состояний ноль или единицу;

- далее получается последовательность нулей и единиц (белые прямоугольники с вариациями значений);

- Затем следует обработка последовательности, как описано выше.

Заключение. Таким образом, операция сложения 2-мерных векторов одной нормы реализуется на основе способности квантового компьютера обрабатывать произвольное фиксированное количество данных. Эту реализацию можно рассматривать как основу для создания процессора на основе алгебры многомерных матриц, поскольку, как упоминалось ранее, любой 2-мерный вектор, выступающий в качестве элемента этой алгебры, можно рассматривать как линейную комбинацию базовых нормализованных векторов.

Создание такого процессора повлечет за собой возможность использования SIMD-архитектуры для распараллеливания алгоритмов, реализующих алгебру многомерных матриц. Это решение, в свою очередь, повысит эффективность обработки реляционных баз данных.

Литература

1. Emelchenkov .E.P. Levin N.A., Munerman V.I. Algebraic Approach to Optimizing the Development and Operation of Database Management Systems // Systems and Means of Information. 2009, Add. Is. P. 114–137.

**G.M. Grigorieva, M.A. Mironov,
V.I. Munerman, V.Yu. Khodchenkov**
Smolensk State University

VECTOR ADDITION ALGORITHM IN A QUANTUM COMPUTER

Keywords: *quantum computing, quantum parallelism, vector processor.*

Abstract. *The article discusses the possibility of creating a vector processor based on quantum computing. The need to solve this problem follows from the fact that there is a wide class of problems that are formalized in vector and matrix algebras. Quantum parallelism allows one to simultaneously add a number of pairs of numbers. However, the main approaches in the literature are reduced to the creation of conventional adders like modern computers adders. The article proposes a method for constructing a quantum vector processor for adding k -dimensional ($k - \text{degree } 2$) vectors. The adder of such a processor consists of k elements, which contains a pair of coordinates in the interval $[0, 1]$. The computational experiment is given.*

РЕАЛИЗАЦИЯ МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ В МЕДИЦИНСКОЙ ПРАКТИКЕ

Ключевые слова: *интеллектуальный анализ данных, большие данные, нейронная сеть.*

В статье рассматривается подход построения системы компьютерной поддержки для принятия решений в медицинской практике на основе диагностических данных пациента. Предложено решение, которое предоставляет врачам информацию, необходимую для выбора наиболее подходящего метода лечения. Для построения и обучения нейронной сети использовались библиотеки Python на языке. В статье дается описание архитектуры системы интерфейса и методики ее взаимодействия с пользователями. Достижимость решения описанной задачи была подтверждена вычислительным экспериментом.

Введение. Современная медицинская наука и практика характеризуются активным внедрением систем интеллектуального анализа данных (ИАД) [1]. Целью этих систем является организация компьютерной поддержки специалистов по постановке диагноза и выбору лучшего метода лечения. Решение этих проблем производится на основе информации, поступающей из системы электронных историй болезни. Эта информация характеризуется неоднородностью: лабораторные данные, показатели здоровья и ряд других. Поскольку в настоящее время накоплено большое количество информации о лечении пациентов с различными заболеваниями, технологии обработки больших данных могут быть использованы для реализации систем комплексного интеллектуального анализа медицинских данных (СКИАМД) [3].

Разработка и внедрение медицинской диагностики заболеваний и их осложнений не только не заменяют традиционное обследование пациента, но и не решают основной проблемы – выбора правильного диагностического решения. Проявление одного и того же заболевания и развитие одних и тех же осложнений различается у разных пациентов и не всегда соответствует «классической» картине заболевания [2].

Основной проблемой инфекционных заболеваний является наличие осложнений, которые могут привести к летальному исходу. Следовательно, для благоприятного исхода лечения существует необходимость в дополнительном контроле пациентов, которым грозит

развитие осложнений. Это требует поиска способов распознавания проблем со здоровьем на ранних стадиях лечения и в стационаре.

Таким образом, выбор методов и алгоритмов интеллектуального анализа данных и разработка на их основе специализированных систем поддержки принятия решений является актуальной задачей, а развитие медицинской кибернетики и современных вычислительных технологий открывает новые возможности для ее решения.

В статье рассмотрена проблема выбора методов и алгоритмов для ИАД, способных решать конкретные медицинские задачи, и разработка на их основе специализированной СКИАМД.

Выбор данных. Данные представляют собой три связанные таблицы, которые хранятся в общей базе данных. Распределение данных по таблицам соответствует распределению данных по разделам классификации. Связь между таблицами осуществляется через идентификатор пациента (ID), который позволяет отдельно обрабатывать данные, а затем выбирать результат для каждого пациента в одной таблице. Распределение данных по таблицам происходит в соответствии с логическими разделами.

Раздел первый «Temp list» содержит информацию о колебаниях температуры пациента, находящегося в больнице.

Второй раздел «Лабораторные данные» содержит данные клинической лабораторной диагностики.

Третий раздел «Основной» включает в себя несколько разделов электронной медицинской карты сразу: Anamnesis Vitae, Anamnesis Morbi – описание течения заболевания со слов пациента, описание хронических и перенесённых заболеваний и многое другое.

Все перечисленные разделы и их факторы участвуют в интеллектуальном анализе данных.

Этот набор основан на медицинской карте пациента в соответствии с регистрационной формой № 003 / у и медицинской карте пациента, получающего медицинскую помощь в амбулаторных условиях, в соответствии с регистрационной формой № 025 / у.

Выбор метода принятия решений. Предлагаемая сеть построена на основе структур сетей Хэмминга и Ворда. Сети Ворда предполагают обработку данных блоками с использованием различных дополнительных придаточных функций, именно поблочную обработку данных мы и заимствуем из сетей Ворда. Сети Хэмминга в классическом случае обрабатывают кортеж данных, а количество входных нейронов соответствует размерности этого кортежа, именно этот основной принцип и был перенесён в нашу гибридную нейронную сеть. В сети используется функция активации логистический сигмоид.

Мелкие круги на рисунке 1 – это входные нейроны, в которые попадают все входные данные. Круги в бликах и соединённые с ними –

это нейроны скрытого слоя, который обрабатывает данные. Светлый кружок – это нейрон для вывода результата, который дает логический ответ и указывает, есть ли у пациента риск развития осложнений.

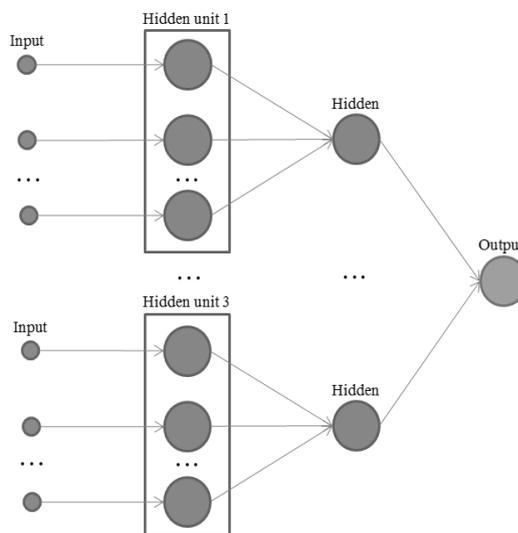


Рис. 1. Архитектура построенной нейронной сети

Нейроны входного слоя получают данные из соответствующих таблиц и отправляют их на первый обрабатывающий слой по таблицам, то есть каждый раздел использует свой собственный блок нейронов первого скрытого слоя. Во втором скрытом слое каждый блок соответствует одному нейрону обработки. Выходной слой состоит из одного нейрона, который представляет данные в требуемой форме.

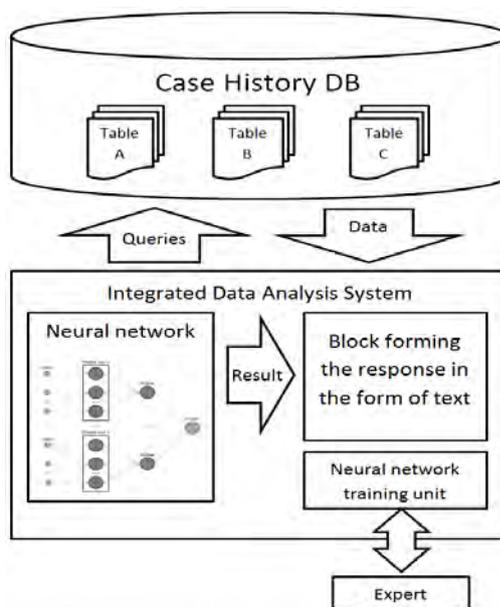


Рис. 2. Процесс работы и обучения нейронной сети в составе SKIAMD

На рисунке 2 приведена схема работы предполагаемой SKIAMD. В верхней части изображена база данных, связанная с системой ИАД стрелками, отображающими потоки данных. Нейронная сеть

обрабатывает данные из базы и передаёт своё решение в блок представления, который, в свою очередь, направляет решение специалисту.

Описание эксперимента. Чтобы проверить предложенный подход и создать модель СКИАМД, был проведен вычислительный эксперимент.

Оборудование:

- Процессор: Intel® Core™ (i5-6200U @ 2,30 ГГц);
- ОЗУ: 12 ГБ.

Программное обеспечение:

- Операционная система: Windows 10;
- СУБД Postgre;
- STATISTICA Automatic Neural Networks [3];
- Язык программирования: Python.

Целью эксперимента было сравнение нейронных сетей разных типов и решение задачи улучшения подхода. Это необходимо для снижения риска системных ошибок при работе в реальных условиях. Были исследованы нейронные сети Хеминга, Ворда, комбинированная сеть на их основе и несколько видов персептронов.

Для статистического анализа были отобраны медицинские записи двухсот пациентов (находились в стационаре в 2014–2016 годах), содержащие факторы, описанные выше, и представлены в удобной для обработки форме, то есть в наборах нулей и единиц.

Оценка качества нейронной сети основана на этих данных с использованием STATISTICA Automatic Neural Networks.

Результаты эксперимента приведены на рисунке 3

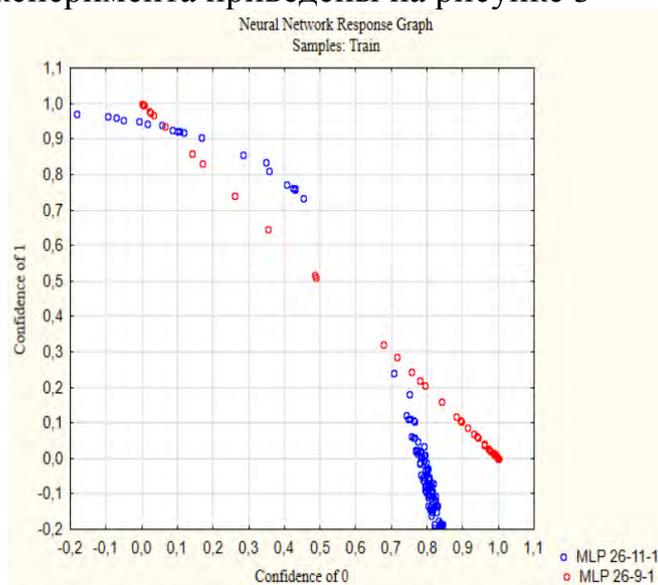


Рис. 3. График точности нейронной сети

Здесь показан график надежности решений двух нейронных сетей, наиболее четко отражающий принцип выбора подходящего типа и структуры.

Для ясности мы определим два фактора, которые являются критерием для подходящей нейронной сети. Линейность результатов, расположение точек графика на одной линии. А также сходимость на концах графика, в идеальном случае в точках – (0,1) и (1,0).

На рисунке 3 для ясности результатов указаны только два графика нейронных сетей. Красные точки обозначают нейронную сеть, которая показала лучший результат по обоим критериям. Синие точки отмечают нейронную сеть с наибольшим отклонением от линейности, эта нейронная сеть имеет отличную конвергенцию в конце дуги, но достоверность этих результатов недостаточно высока.

Из результатов эксперимента следует, что гибридная нейронная сеть сконструированного нами типа оказалась наиболее эффективной.

Эксперимент позволил не только проверить нейронные сети, но и рассчитать начальные значения весов нейронов, с которых может быть продолжено обучение.

Заключение. Для решения проблемы была выбрана гибридная нейронная классификационная сеть. Выбор этого типа сети обусловлен серией тестов в программе «STATISTICA», которые показали тип нейронной сети, наиболее подходящий для решения задачи классификации тестовой выборки историй болезни.

Литература

1. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. М.: Финансы и статистика, 2016. 424 с.
2. Анохин П.К. Узловые вопросы теории функциональной системы. М.: Наука, 2015. 197с.
3. Кузнецова А. В., Сенько О. В. Возможности использования методов Data Mining при медико-лабораторных исследованиях для выявления закономерностей в массивах данных // Врач и информационные технологии. 2015, №2.

G.M. Grigoryeva, V.Yu. Khodchenkov
Smolensk State University

REALIZATION OF DATA MINING METHODS IN MEDICAL PRACTICE

Keywords: *data mining, big data, neural network.*

Abstract. *The article discusses the approach of building computer support for decision making in medical practice based on patient diagnostic data. A solution has been proposed that provides doctors with information*

that is necessary for choosing the most suitable method of treatment. The description of an information system and a neural network was provided. Python libraries were used to build and train the neural network. The article provides a description of the system architecture and interfaces of its interaction with experts and users. The achievability of the described problem solution was proved by a computational experiment.

О.Е. Диев, В.И. Мунерман
Смоленский государственный университет

УДК 004.657

КОНТЕЙНЕРНЫЙ МЕТОД ГЕНЕРАЦИИ ЭЛЕМЕНТОВ БАЗ ДАННЫХ В РАСПРЕДЕЛЁННЫХ СИСТЕМАХ

Ключевые слова: массовая обработка данных, параллельное программирование, программно-аппаратные комплексы, PostgreSQL, грид-кластер, контейнеры, Docker.

В работе рассматривается задача распараллеливания JOIN запроса в реляционной СУБД при помощи локальной грид-системы с использованием контейнерных технологий, в частности с применением ПО Docker. В результате была получена новая концепция локальной грид-системы. Рабочие станции в грид-системе были разделены на три вида: полнофункциональная (на машине предустановлена ОС, СУБД, БД), частично функциональная (соответственно ОС, СУБД) и потенциальная (только ОС). Далее для каждого вида рабочей станции генерируется контейнер для дальнейшего исполнения на машине исполнителя.

Архитектура контейнерной технологии. Приведем описания контейнеризации на примере ПО Docker. Docker – это программное обеспечение, используемое для автоматизации развертывания и управления приложениями в изолированной среде под управлением различных операционных систем. Пользовательское приложение и все, что ему требуется (языковые и библиотечные средства), «упаковываются» в специализированный объект – контейнер (контейнерный образ), который может быть перенесён на любую вычислительную систему, операционная система которой поддерживает Docker-платформу. Контейнерный образ представляет собой автономный, исполняемый пакет программного обеспечения, который включает в себя все необходимое для его запуска: код, среда выполнения, системные инструменты, системные библиотеки,

настройки. Контейнеры изолируют программное обеспечение от окружения, что позволяет игнорировать различия между средами разработки и промежуточными средами. Естественно, контейнеры одного и того же приложения для различных операционных систем и вычислительных средств могут быть различными. Кроме того, Docker предоставляет среду по управлению контейнерами. Эта среда представляет собой стандартную RunTime-систему для всех видов контейнеров. Docker реализует управление полным жизненным циклом контейнера и выполняет основные функции жизненного цикла, такие как передача и хранение данных, как системных, так и пользовательских, управление контейнером, низкоуровневые хранилища и сетевые приложения. Программа, реализующая среду управления контейнерами (Docker использует Containerd), предназначена для встраивания в системную часть конкретного программно-аппаратного комплекса для параллельной обработки симметрично горизонтально распределенных данных. Среда управления недоступна для непосредственного использования прикладным программистом или конечным пользователем.

Основными достоинствами данной системы являются:

- ускоренный процесс разработки. Нет необходимости устанавливать вспомогательные инструменты в частности СУБД PostgreSQL, его можно запускать в контейнерах;
- удобная инкапсуляция приложений;
- простое масштабирование.

Задача по реализации симметричного горизонтального распределения в условиях локальной грид-системы. Для реализации симметричного горизонтального распределения данных [4] выбрана SQL-реляционная модель СУБД PostgreSQL. Выбор этой модели обусловлен тем, что:

- эта модель изоморфна теоретико-множественной (файловой) модели данных, которая обеспечивает возможность симметричного горизонтального распределения;

- она обеспечивает выполнение важнейшего требования грид-технологии – это позволяет симметрично горизонтально распределить таблицы-операнды операции Join и затем параллельно выполнить эту операцию над фрагментами таблиц. Кроме того, операции симметричного горизонтального распределения таблиц: создание индексов для индексно-последовательного доступа к таблицам, распределение таблиц алгоритмом бустрофедона могут быть реализованы в процедурно-ориентированном диалекте языка SQL – PL/PGSQL. Таким образом, выбор SQL-реляционной модели данных позволяет использовать вычислительные средства, реализованные на

различных платформах: различные операционные системы, различные, но поддерживающие SQL-реляционную модель, СУБД. Это полностью решает проблему интероперабельности. Применение грид-технологии в сочетании с СУБД PosrgreSQL позволяет создавать для конкретных приложений локальные параллельные программно-аппаратные комплексы или локальные грид-системы.

Рабочие станции и их виды. В данной работе выделяется три возможных вида рабочих станций: *полнофункциональная*, *частичнофункциональная* и *потенциальная*.

- Полнофункциональная – рабочая станция, имеющая свою ОС, СУБД и БД (с набором хранимых процедур).
- Частично функциональная – рабочая станция, имеющая свою ОС и СУБД.
- Потенциальная – рабочая станция, которая имеет у себя только предустановленную ОС.

Процесс взаимодействия инициатора параллельной реализации JOIN показан на рисунке 1.

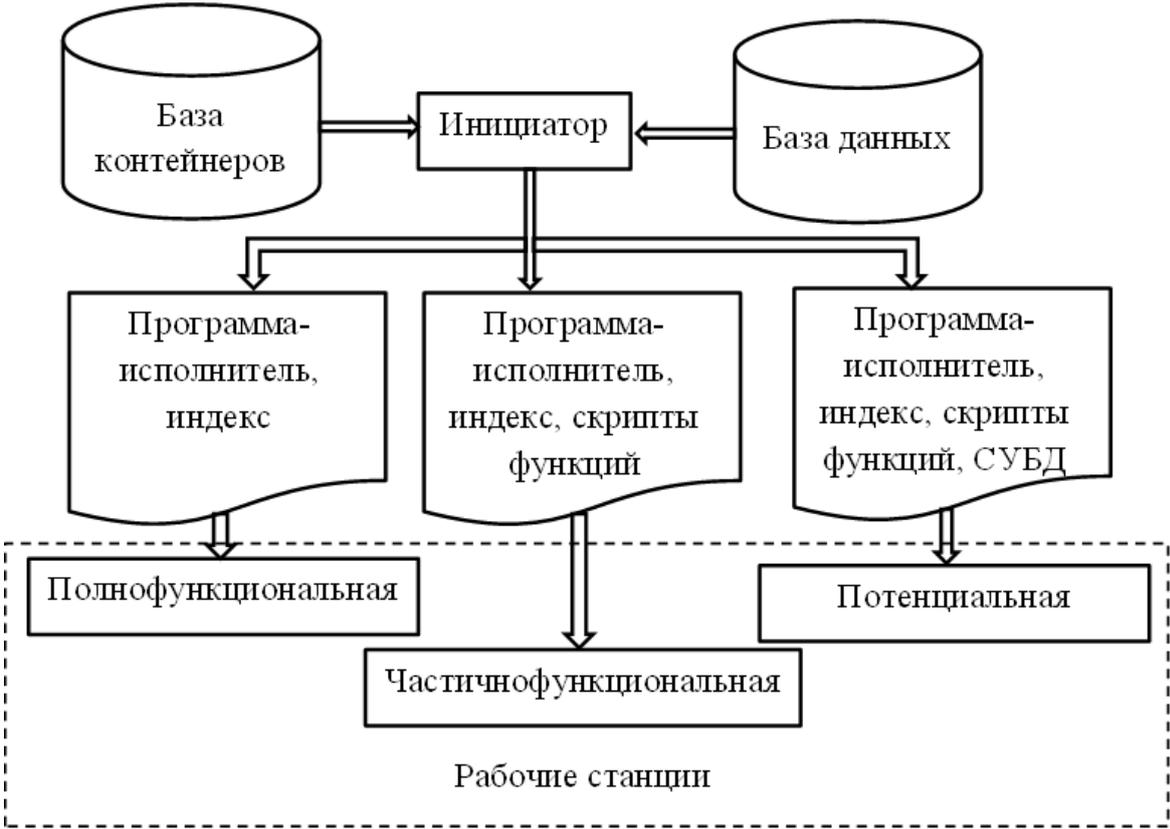


Рис. 1. Процесс взаимодействия рабочих станций в локальных грид

Инициатор с помощью системы Docker определяет количество и виды доступных рабочих станций. Каждой рабочей станции в зависимости от её вида из баз данных, доступных Инициатору, передаются контейнеры, содержащие необходимое для

функционирования этой рабочей станции программное обеспечение и данные.

Соответственно представленным рабочим станциям формируются три вида контейнеров с различным набором инструментов. Контейнер для полнофункциональной рабочей станции будет состоять из данных, над которыми необходимо выполнить JOIN и программы, вызывающей хранимые процедуры СУБД, а именно: первое – хранимая процедура на загрузку предоставленных данных, второе – хранимая процедура на выполнение операции JOIN над этими данными и последующей отправкой результата на машину отправителя данного контейнера. Контейнер для частичнофункциональной рабочей станции будет включать в себя набор программ и данных по аналогии с контейнером для полнофункциональной рабочей станции, а также файл с набором запросов к СУБД на создание хранимых процедур и написанной программы, которая будет отправлять данные запросы в СУБД. Контейнер для потенциальной рабочей станции будет включать в себя саму СУБД, БД (с хранимыми процедурами и данными, над которыми требуется выполнение операции JOIN) и написанную программу для исполнения хранимых процедур в БД.

Описание эксперимента. Эксперимент проводился в следующих условиях:

- рабочие станции – IBM PC с процессорами типа Intel Core и оперативными памятьями от 4 Гб;
- локальная вычислительная сеть;
- однородная СУБД Postgre;
- Иницилирующая программа разработана на языке программирования C#, скрипты хранимых процедур на языке PL/SQL.

Эксперимент состоит из трёх этапов.

• Первый этап – поиск рабочей станции со свободными ресурсами из локальной грид-системы. Как только такая машина была найдена определяется к какому виду относится эта машина (полнофункциональной, частичнофункциональной или потенциальной).

• Второй этап – генерация контейнера под тот вид машины, который был найден на первом этапе.

• Третий этап – отправка контейнера на машину исполнителя с последующим исполнением содержимого пересылаемого контейнера на машине исполнителя.

Заключение. Таким образом, разработана система параллельной реализации операции JOIN с использованием симметричного горизонтального распределения данных и контейнерной технологии. Эксперимент показал возможность подобной реализации такого вида параллельной обработки данных.

Литература

1. <https://www.postgresql.org/docs/10/static/index.html>.
2. Левин Н.А., Мунерман В.И. Модели обработки больших объемов данных в системах массового параллелизма // Системы высокой доступности. 2013, Т. 9, №1. С. 35-43.
3. Мунерман В.И. Построение архитектур программно-аппаратных комплексов для повышения эффективности массовой обработки данных // Системы высокой доступности. 2014, Т.10, №4. С. 3-16.
4. Мунерман В.И., Мунерман Д.В. Параллельная реализация симметричного горизонтального распределения данных на основе сетевых технологий // Международный научный журнал «Современные информационные технологии и ИТ-образование». 2017. Vol. 13, No. 3. S. 1. Доступно на: <<http://sitito.cs.msu.ru/index.php/SITITO/article/view/288>>. дата доступа: 17.03. 2019 doi: <https://doi.org/10.25559/SITITO.2017.3.521>.

О.Е. Diev, V.I. Munerman
Smolensk State University

CONTAINER METHOD OF GENERATION OF ELEMENTS OF DATABASES IN DISTRIBUTED SYSTEMS

Keywords: *massively data processing, parallel programming, software and hardware complexes, PostgreSQL, grid cluster, containers, Docker.*

Abstract. *The paper considers the task of paralleling a JOIN query in a relational DBMS using a local grid system using container technologies, in particular using Docker software. A new concept of a local grid system was obtained. The workstations in the grid system were divided into three types: full-featured (OS is preinstalled, DBMS, DB is pre-installed), partially functional (OS, DBMS respectively) and potential (OS only). Then for each type of workstation a container is generated for further execution on the executor's machine.*

А.А. Жарков
Смоленский государственный университет

УДК 004.056.53, 004.056.55

ОДИН ПОДХОД К ПОВЫШЕНИЮ КРИПТОСТОЙКОСТИ СИСТЕМ КОДИРОВАНИЯ ДЛЯ ИНТЕРНЕТА ВЕЩЕЙ

Ключевые слова: *защита информации, несанкционированный доступ, методы шифрования, метод Диффи-Хелмана, блочные алгоритмы шифрования, криптоустойчивость, умные среды, интеллектуально-ориентированные среды.*

В статье рассматривается проблема безопасности и целостности информационной системы, содержащей в себе устройства концепции Интернета вещей. Анализируется возможность предотвращения несанкционированного доступа к интеллектуально-ориентированному комплексу. Рассмотрен некоторый подход к повышению сохранности конфиденциальной информации, где за основу взяты известные методы. Подразумевая случай «диверсии на производстве» и разглашении основополагающих для системы алгоритмов, напрямую или косвенно связанных с шифрованием данных внутри интеллектуально-ориентированной среды, построена идея защиты информационных систем «умных» сред, состоящих из устройств, вычислительная мощность которых недостаточно велика для выполнения сложных вычислений. В данной работе установлена основная причина сложности задачи защиты информации от несанкционированного доступа в интеллектуально-ориентированных средах, связанная с доступностью данных системы и вмешательством в неё со стороны. Приведены разработанные для решения поставленной задачи некоторые возможные модификации нескольких известных алгоритмов. А также приведены некоторые примеры защиты системы от рассмотренных в статье видов угроз информационной безопасности. Представлены алгебраические модели, используемые в предложенном подходе.

В настоящем существует проблема защиты информации в Интернете вещей в связи с низкой вычислительной способностью микроконтроллеров из-за невысоких тактовых частот. И в связи с потенциальной возможностью нанесения ущерба владельцу интеллектуально-ориентированной среды за счёт кражи личных данных и возможности изменения её функционирования необходимо разработать некоторую методику, которая позволит защищать данные в системах концепции Интернета вещей. [3] Также необходимо учесть, что объём данных, передаваемый в подобных сетях, крайне невелик, вследствие чего задача вычисления криптоаналитиком исходного сообщения изрядно упрощается из-за малого перебора ключа (зачастую от размера сообщения зависит требуемый размер ключа).

Рассмотрим основные криптографические алгоритмы.

Среди криптосистем выделяют два вида алгоритмов шифрования данных: алгоритмы асимметричного и симметричного шифрования. Самым эффективным методом шифрования признаётся асимметричное шифрование [1].

Концепция криптосистем с асимметричным шифрованием данных в общем случае такова:

1. Вычисляемая пара ключей получателем В должна быть простой.
2. Отправитель А, зная открытый ключ В и сообщение М, может легко вычислить криптограмму С.
3. Получатель В, используя секретный ключ и криптограмму С, может легко восстановить исходное сообщение М.
4. Противник, зная открытый ключ, при попытке вычислить секретный ключ наталкивается на тяжёлую вычислительную проблему.
5. Противник, зная открытый ключ и криптограмму, при попытке вычислить исходное сообщение М наталкивается на тяжёлую вычислительную проблему.

В качестве недостатков асимметричных криптосистем выделяют:

- 1) отсутствие математического доказательства необратимости используемых функций;
- 2) такой вид шифрования требует больших вычислительных мощностей, поскольку при шифровании и расшифровке используются весьма ресурсоемкие операции;
- 3) существует необходимость защиты открытых ключей от подмены.

Выделенный набор минусов подчёркивает невозможность использования маломощными системами данного типа алгоритмов. Модифицировать такие алгоритмы практически невозможно, так как они строятся на сложной математической задаче.

Перейдём к следующему рассматриваемому виду алгоритмов.

Алгоритмы симметричного шифрования – алгоритмы шифрования, использующие один и тот же ключ для зашифровывания и дешифрования информации или ключ расшифровывания легко вычислить из ключа зашифрования.

Основной недостаток симметричного шифрования заключается в необходимости согласования ключей. Для решения данной проблемы можно использовать алгоритм Диффи-Хелмана [2], позволяющий устройствам с низкими вычислительными способностями, используя «несложные» операции, не затрачивать много времени на вычисление. Также в условиях внутренней диверсии или распознавания алгоритма защищённая им интеллектуально-ориентированная среда становится уязвима для потенциальных злоумышленников, воспользовавшихся подбором ключа «атакой на основе шифротекста» [4]. На этот недостаток нельзя не обратить внимание, так как при такой системе становится практически невозможным полагаться на выбор алгоритма шифрования, ориентируясь только на его сложность. В остальном же алгоритмы симметричного шифрования можно считать достаточно проработанной и эффективной мерой с минимальным количеством недостатков, особенно на фоне асимметричного шифрования.

Чтобы устранить данный недостаток, предлагается использовать композиции неодинаковых алгоритмов. Причиной неодинаковости алгоритмов является шанс существования такого ключа, который даёт тот же шифротекст, что и двойное шифрование одним и тем же алгоритмом. Но и шифрование неодинаковыми алгоритмами может не привести к желаемому результату в случае, если криптоаналитик в курсе, какие алгоритмы использовались изначально, что откроет ему возможность, используя атаку на основе подобранный шифротекста, попасть в систему.

Таким образом предлагается расширить вариантивность композиций алгоритмов не только за счёт неодинаковых криптографических симметричных алгоритмов, но и других алгоритмов, меняющих структуру исходного сообщения. К примеру, к таким алгоритмам можно отнести алгоритмы сжатия, хеширования и прочие [1].

Так или иначе, даже уменьшив шансы криптоаналитика на успешное использование атак на основе частотного анализа, остаётся шанс, что в случае настойчивого криптоанализа в течение некоторого времени из-за статичности системы шифрования с помощью атаки подбором шифротекста злоумышленник сможет получить доступ в систему. Данную проблему предлагается решать методом динамического создания композиций алгоритмов предложенного множества. Очевидно, что при таком решении должен быть некоторый вычислительный сервер для создания таких композиций. Но в частном случае Интернета вещей наличие центрального вычислительного сервера не является редкостью. Поэтому будем считать, что имеется в наличии центральный шифрующий сервер.

Поддерживать динамику предлагается некоторой модификацией генетического алгоритма. В связи с небольшими потребностями к памяти у данного семейства алгоритмов (в отличие от «жадных» алгоритмов), простоте распараллеливания и достаточным временем на формирование различных решений приведенный алгоритм хорошо подходит к решению данной проблемы. Также выбранный метод хорош в условиях данной задачи тем, что в случае временной уязвимости памяти центрального сервера шифрования, злоумышленник не сможет быть уверенным в стабильности множества заимствованных им решений [5].

Определим A как конечное множество алгоритмов мощности b , пригодных для решения задачи создания шифрующих композиций. Оно будет содержать в себе алгоритмы симметричного шифрования, алгоритмы сжатия данных, а также алгоритмы хеширования, не требующие ключа. Для каждого алгоритма сжатия обозначим условие однозначного разжатия данных получателем. Для остальных алгоритмов

определим их вариативные части, которые могут видоизменяться (и будем использовать их для процесса «мутирования») и поместим их в множество возвратных функций P (мутагенов). Например, для алгоритмов хеширования – это различные вариации возвратных функций; для некоторых алгоритмов кодирования – методы преобразования ключа и пр.

Для каждой i -той итерации генетического алгоритма уточним, что, пройдя этапы селекции, мутирования и отбора, выжившие особи становятся членами множества M , расширяя множество возможных решений для криптоаналитика таким образом, что его мощность становится равной количеству $b_i^2 + b_i$.

Время, которое можно потратить на генерацию алгоритмов и передачу нового алгоритма устройствам в сети, будет равно времени, необходимому на вскрытие самого слабого алгоритма в данном множестве.

Определим функцию мутации алгоритма мутагеном (однозначно-возвратным преобразованием)

$$m : A \times P \rightarrow A (m(a, p) = k, a \in A, p \in P, k \in A).$$

Также задана функция генерации новых алгоритмов:

$$g : M \times M \rightarrow M (g(m_1, m_2) = m_3, m_1 \in M, m_2 \in M, m_3 \in M)$$

Таким образом, задаётся четвёрка $D = (A, P, m, g)$ – двухосновная универсальная алгебра, состоящая из операций мутирования и роста популяции.

Определим X, K, Y некоторыми конечными множествами открытых текстов, ключей и зашифрованных сообщений соответственно. На прямом произведении множеств X и K задана функция:

$$f : X \times Y \rightarrow Y (f(x, \chi) = y, x \in X, \chi \in X, y \in Y).$$

И поставим в соответствие f семейство отображений $f_\chi : X \rightarrow Y, \chi \in K$ где f_χ – ограничение f на множестве $X \times \{\chi\}$.

Введённая четвёрка $A = (K, X, Y, f)$ определяет трёхосновную универсальную алгебру, сигнатура которой состоит из функциональной единственной операции f – функции шифрования.

Литература

1. Schneier Bruce. Applied Cryptography. Protocols, Algorithms and Source Code in C. Wiley, 1993. 618 p.
2. Simon Singh. The Code Book: the evolution of secrecy from Mary Queen of Scots to quantum cryptography. New York: Doubleday, 1999. 416 p.
3. Hakima Chaouchi. The Internet of Things. London: Wiley-ISTE, 2010.
4. Churchhouse R. Codes and Ciphers: Julius Caesar, the Enigma and the Internet/ Cambridge: Cambridge University Press, 2002.
5. Davis L. Handbook of genetic algorithms. NY: Van Nostrand Reinhold, 1991.

A.A. Zharkov
Smolensk State University

A WAY TO INCREASE CRYPTOGRAPHIC SECURITY WITHIN INTERNET OF THINGS

Keywords: *internet of things, unauthorized access, encryption methods, the Diffie-Hellman method, block encryption algorithms, crypto-stability, intelligent environments, intellectually-oriented environments.*

Abstract. *This article discusses a security problem and the integrity of the information system that contains "smart" devices. The possibility of preventing unauthorized access to the intellectually-oriented complex of objects of the Internet concept of things is analyzed. Based on some known methods and algorithms, directly or indirectly related to the protection of information from unauthorized access, the idea of protecting information systems of "smart" environments consisting of devices whose computational capability is not good enough for performing complex calculations is built. Implying the case of "sabotage" and disclosure of the underlying algorithms for the system, directly or indirectly related to data encryption within an intelligently oriented environment, an idea of protecting information systems of "smart" environments consisting of devices whose computational power is not large enough to perform complex calculations is built. The analysis of basic methods of obtaining the informative part of packets transmitted in information systems containing "smart" devices is carried out. Some possible modifications of several known algorithms developed for the solution of this problem are given. And also, some examples of protection of the system from the types of hacking considered in the article are given. Algebraic models used in the proposed approach are presented.*

В.Н. Захаров
*Федеральный исследовательский центр «Информатика и
управление» Российской академии наук*

УДК 004.021

О ПРОБЛЕМЕ «P VERSUS NP»

Ключевые слова: *проблема «P versus NP», теория сложности вычислений, детерминированные машины Тьюринга, недетерминированные машины Тьюринга, NP-полнота.*

В 2000 году Математический институт Клэя (США) опубликовал список так называемых проблем тысячелетия, за решение каждой из

которых был обещан приз в один миллион долларов США. Этот список содержал семь нерешенных математических проблем, которые были определены как «важные классические задачи, решение которых не найдено вот уже в течение многих лет» [1]. К настоящему времени только одна из этих задач (гипотеза Римана) решена в 2010 году российским математиком Григорием Перельманом. Среди нерешенных проблем из списка наибольшее внимание привлекает проблема «P versus NP», относящаяся к теории алгоритмов и которую часто также называют «проблемой перебора». Возможно, это связано с простотой её формулировки. В самом общем виде суть проблемы можно сформулировать следующим образом: Верно ли, что если достаточно просто проверить, что представленное решение некоторой проблемы является правильным, то и само это решение можно найти достаточно просто? В статье представлена точная формулировка проблемы в терминах машин Тьюринга, определены классы сложности, показаны некоторые соотношения между этими классами, а также приведены сведения об опубликованных попытках решения проблемы.

Проблему «P versus NP» (называемую также «проблемой перебора») сформулировали в 1971 году Стефен Кук (S.A. Cook) [2] и независимо от него Леонид Левин (L.A. Levin) [3; 4]. Официальное описание этой проблемы, написанное С. Куком, представлено на сайте Института Клэя в ряду проблем тысячелетия (MillenniumPrizeProblems) [5]. Он использовал для объяснения проблемы термины машин Тьюринга, работающих полиномиальное время. Там же описаны некоторые возможные подходы к решению проблемы и некоторые промежуточные результаты.

В данной статье для полноты картины представим определение машины Тьюринга, данное в [5]. Из той же работы позаимствуем определение машин Тьюринга, работающих полиномиальное время. Далее опишем введенные ранее определения некоторых классов сложности машин Тьюринга и приведем полученные соотношения между ними [6]. В заключительной части статьи проведем обзор, суммирующий известные (объявленные) сообщения о попытках решения проблемы.

1. Машина Тьюринга

Машина Тьюринга M состоит из устройства управления с конечным множеством состояний (то есть конечной программой), к которому присоединена головка чтения-записи, способная перемещаться по бесконечной ленте. Лента разделена на клетки, в каждой из которых может сохраняться одна буква из конечного алфавита Γ . Этот алфавит включает пустую букву λ . Каждая машина имеет некоторый входной алфавит Σ , являющийся подмножеством Γ ; Σ не включает пустую букву

λ . На каждом шаге вычисления M находится в некотором состоянии q из конечного множества Q состояний машины M . Перед началом работы входное слово в алфавите Σ записывается в последовательные клетки ленты, а все остальные клетки делаются пустыми (в них помещается буква λ). Головка машины размещается так, чтобы она воспринимала самую левую букву входного слова, а машина приводится в начальное состояние q_0 . На каждом шаге машина M находится в некотором состоянии q , а ее головка размещена на некоторой клетке ленты, содержащей букву s . Машина выполняет действие, которое зависит от двойки $\langle q, s \rangle$. Это действие определяется функцией переходов машины (или ее программой) δ . Действие состоит из следующих поддействий:

- запись буквы в клетку, на которой находится головка;
- перемещение головки влево или вправо на одну клетку;
- переход в новое состояние.

Формально машина Тьюринга M есть четверка $\langle \Sigma, \Gamma, Q, \delta \rangle$, где Σ, Γ, Q – конечные непустые множества, причем $\Sigma \subseteq \Gamma, \lambda \in (\Gamma \setminus \Sigma)$. В множестве состояний Q имеются три специальных состояния: $q_0, q_{\text{accept}}, q_{\text{reject}}$. Функция переходов δ есть отображение типа

$$\delta: ((Q \setminus \{q_{\text{accept}}, q_{\text{reject}}\}) \times \Gamma) \rightarrow (Q \times \Gamma \times \{-1, 1\}).$$

Если $\delta(q, s) = \langle q', s', h \rangle$, то это интерпретируется следующим образом. Если машина M находится в состоянии q и воспринимает букву s , то q' – ее новое состояние, s' – буква, которую она записывает на ленту; головка перемещается влево или вправо в зависимости от значения h : при -1 сдвиг влево, а при 1 – вправо. Предполагается, что множества Q и Γ различны.

Конфигурация машины M – это слово XqY , где $X, Y \in \Gamma^*, Y$ – непустое слово, а $q \in Q$. Конфигурация XqY означает, что машина M находится в состоянии q , на ее ленте записано слово XY , а головка находится на клетке, содержащей самую левую букву слова Y .

Пусть C и C' – две конфигурации. $C \vdash_M C'$, если $C = XqsY, \delta(q, s) = \langle q', s', h \rangle$ и имеет место одно из следующих соотношений:

- $C' = Xs'q'Y, h = 1$ и Y – непустое слово;
- $C' = Xs'q'A, h = 1$ и Y – пустое слово;
- $C' = X'q'as'Y, h = -1$ и $X = X'a$ для некоторого $a \in \Gamma$;
- $C' = q'\lambda s'Y, h = -1$ и X – пустое слово.

Конфигурация XqY называется *останавливающей*, если $q \in \{q_{\text{accept}}, q_{\text{reject}}\}$. Заметим, что для каждой неостанавливающей конфигурации C однозначно определяется конфигурация C' такая, что $C \vdash_M C'$.

Вычисление машины M на входном слове X – это однозначно определяемая последовательность C_0, C_1, \dots конфигураций такая, что:

- $C_0 = q_0X$ (или q_0A , если A – пустое слово, а X – входное слово);
 - $C_i \vdash_M C_{i+1}$ для каждого i , для которого в вычислении имеется C_{i+1} .
- В каждую конфигурацию входит *промежуточное* слово, которое написано на ленте;
- последовательность либо бесконечна, либо она заканчивается останавливающей конфигурацией. Если вычисление конечно, то число шагов в ней по крайней мере на единицу меньше числа возможных различных конфигураций, в противном случае число шагов бесконечно. Говорят, что M допускает слово X , если вычисление на этом слове конечно и заключительная конфигурация содержит состояние $q_{\text{акцепт}}$;
 - если вычисление конечно, то *длиной вычисления* или *временем вычисления* называется количество конфигураций в вычислении (число шагов вычисления).

2. Недетерминированные машины Тьюринга

Функция переходов δ для недетерминированных машин Тьюринга допускает для одной и той же пары, принадлежащей $(Q \setminus \{q_{\text{акцепт}}, q_{\text{реjekt}}\} \times \Gamma)$, несколько троек, принадлежащих $(Q \times \Gamma \times \{-1, 1\})$.

Назовем *правильной последовательностью конфигураций* (или правильным путем) для недетерминированной машины Тьюринга N такую последовательность конфигураций $C_0, C_1, \dots, C_i, C_{i+1}, \dots, C_t$, что $C_0 \vdash_N C_1 \vdash_N \dots C_i \vdash_N C_{i+1} \vdash_N \dots C_t$, где переходы делаются в соответствии с функцией переходов недетерминированной машины Тьюринга N , а завершается эта последовательность конфигурацией C_t , которая содержит состояние $q_{\text{акцепт}}$ или $q_{\text{реjekt}}$. *Числом шагов* вычисления называется число членов правильной последовательности конфигураций. Конечно, число шагов зависит от конкретного правильного пути для одного и того же входного слова.

3. Машины Тьюринга, работающие полиномиальное время

Обозначим через $t_M(X)$ число шагов в вычислении машины Тьюринга M с входным алфавитом Σ на входном слове $X \in \Sigma^*$. Если машина не останавливается на этом слове, положим $t_M(X) = \infty$. Для натурального числа n определим $T_M(n)$ как наибольшее время работы на словах длины n , то есть

$$T_M(n) \stackrel{\text{def}}{=} \max \{t_M(X) \mid (X \in \Sigma^*) \& (|X| = n)\}.$$

Говорят, что машина M *работает полиномиальное время*, если существует такое натуральное число m , что для любого n $T_M(n) \leq \text{poly}_m(n)$. В частности, это означает, что такая машина имеет конечное вычисление на любом слове $X \in \Sigma^*$. Класс всюду определенных машин Тьюринга, которые работают полиномиальное время, ограниченное полиномом $\text{poly}_m(|X|)$, называется $M(P_m)$, $M(P)$ $\stackrel{\text{def}}{=} \bigcup_{m=1}^{\infty} M(P_m)$.

Будем обозначать класс языков, распознаваемых такими машинами Тьюринга, через $L(P_m)$ или иногда просто P_m . Тогда $L(P) \stackrel{\text{def}}{=} \bigcup_{m=1}^{\infty} L(P_m)$ или $P \stackrel{\text{def}}{=} \bigcup_{m=1}^{\infty} P_m$.

4. Недетерминированные машины Тьюринга, работающие полиномиальное время

Будем говорить, что предикат (язык) A распознается недетерминированной машиной Тьюринга за полиномиальное время, ограниченное полиномом $\text{poly}_n(|X|)$, если найдется недетерминированная машина Тьюринга N , такая, что для любого слова $X \in A$ существует вычисление машины Тьюринга N , заканчивающееся состоянием с q_{accept} , и длина которого (число шагов) ограничена полиномом $\text{poly}_n(|X|)$. Множество таких слов для машины N будем обозначать $\text{Lang}(N)$. Для слов, не входящих в A , вычисления машины N , заканчивающегося состоянием с q_{accept} , с длиной (числом шагов), ограниченной полиномом $\text{poly}_n(|X|)$, не существует, то есть $\text{Lang}(N) = A$. Множество таких машин Тьюринга будем обозначать $M(\mathbf{NP}_n)$. Будем обозначать множество предикатов, распознаваемых машинами из $M(\mathbf{NP}_n)$, через $L(\mathbf{NP}_n)$ (или просто \mathbf{NP}_n). То есть \mathbf{NP}_n – это множество предикатов (языков), вычисляемых на недетерминированных машинах Тьюринга за время, ограниченное полиномом $\text{poly}_n(|X|)$. Определим $L(\mathbf{NP}) \stackrel{\text{def}}{=} \bigcup_{n=1}^{\infty} L(\mathbf{NP}_n)$.

Существует и другое определение класса \mathbf{NP} . Оно выглядит следующим образом. Класс $L(\mathbf{NMP}_n^m)$ – это класс предикатов (языков), которые представимы в виде $Q(X) \stackrel{\text{def}}{=} \exists Y (|Y| \leq \text{poly}_n(|X|)) P_m(X\#Y)$, где P_m – машина Тьюринга (детерминированная), работающая полиномиальное время, которое ограничено некоторым полиномом степени m , то есть $P_m \in M(P_m)$. Иными словами, язык $(A \in L(\mathbf{NMP}_n^m)) \stackrel{\text{def}}{=} (A = \{X : \exists Y (|Y| \leq \text{poly}_n(|X|)) P_m(X\#Y)\})$. Слово Y при этом можно интерпретировать в качестве предъявленного доказательства.

Определим $L(\mathbf{NMP}_n) \stackrel{\text{def}}{=} \bigcup_{m=0}^{\infty} L(\mathbf{NMP}_n^m)$, $L(\mathbf{NMP}^m) \stackrel{\text{def}}{=} \bigcup_{n=0}^{\infty} L(\mathbf{NMP}_n^m)$, а $L(\mathbf{NMP}) \stackrel{\text{def}}{=} \bigcup_{n=1}^{\infty} L(\mathbf{NMP}_n)$. Очевидно, что $L(\mathbf{NMP}) = \bigcup_{m=0}^{\infty} L(\mathbf{NMP}^m) = \bigcup_{n=1}^{\infty} L(\mathbf{NMP}_n)$, а также, что $L(\mathbf{NMP}_0) = L(P)$.

Общеизвестно, что два приведенных выше определения класса предикатов, разрешимых на недетерминированных машинах Тьюринга за полиномиальное время, эквивалентны, например [7]. В [6] приведено доказательство следующих теорем о соответствии введенных классов сложности:

Теорема 1 ($(L(\mathbf{NP}_n) \subseteq L(\mathbf{NMP}_{2n}^3))$).

Теорема 2 ($L(\mathbf{NMP}_n^m) \subseteq L(\mathbf{NP}_{(m+1)n})$), следствием которых является

следующее утверждение о гипотезе принадлежности слова к предикату из класса **NP**: Число шагов алгоритма проверки гипотезы о принадлежности слова к предикату из класса **NP** не превосходит некоторого полинома третьей степени от суммарной длины этого слова и длины предъявленного доказательства.

5. О соотношении классов сложности P и NP

Проблема соотношения классов сложности P и NP является одной из нерешенных «проблем тысячелетия». Она заключается в вопросе, равны ли классы предикатов $L(P)$ и $L(NP)$, или же класс $L(P)$ строго принадлежит классу $L(NP)$.

6. NP -полнота и сводимость по Карпу

NP -предикат P_1 сводится (по Карпу) к NP -предикату P_2 , если существует вычислимая за полиномиальное время функция f , такая что $\forall X(P_1(X) \sim P_2(f(X)))$ [8].

NP -предикат называется NP -полным, если к нему можно свести любую другую машину из класса NP за полиномиальное время. Другими словами, NP -предикат Q NP -полон, если для любого NP -предиката P существует такая вычислимая за полиномиальное время функция f_P , что $\forall X(Q(X) \sim P(f_P(X)))$.

В литературе описаны несколько сотен NP -полных проблем (формально сводимых к NP -полным предикатам), например, [9]. Одной из них является 3-SAT проблема, описанная в [2] и [3]. Там же показано, что 3-SAT проблема является NP -полной. Заключается эта проблема в том, чтобы по формуле 3-КНФ, представляющей собой конъюнкцию троек, где каждая тройка – это дизъюнкция трех элементов, каждый из которых представляет собой булевскую переменную или ее отрицание, определить, существуют ли значения переменных, при которых эта формула принимает значение истина (T).

В [6] приводится доказательство теоремы о том, что язык 3-КНФ формул, которые принимают значение T при некоторых значениях входящих в них переменных, принадлежит классу $L(NP_3)$.

7. Подходы к решению проблемы « P versus NP »

Ввиду достаточной простоты формулировки Проблема « P versus NP » привлекает достаточно много людей, пытающихся решить данную Проблему. В Интернете создана страница, в которой отслеживаются публикации, говорящие о решении Проблемы [10]. К сожалению, последнее обновление этой страницы датируется 26 сентября 2016 года. На эту дату на странице зафиксировано 116 ссылок, большинство из которых на публикации, размещенные в открытом архиве публикаций <http://arxiv.org/abs/>. Следует заметить, что во всех опубликованных «решениях» были обнаружены ошибки.

Интересно распределение «решений» по «полученному результату». Так, в 61 материале из списка доказывається утверждение « P равно NP », в 51 – что « P не равно NP », а в четырех статьях из списка доказывається, что проблема определения, равны ли классы P и NP , неразрешима.

Подавляющее большинство «доказательств P равно NP » заключались в попытке представить алгоритм, работающий полиномиальное время и решающий одну из NP -полных проблем. Пока во всех представленных доказательствах обнаруживались ошибки. Большинство «доказательств P не равно NP » проводились одним из двух методов: диагонализацией или с использованием булевских схем. В этой части были достигнуты значительные продвижения и получены интересные промежуточные результаты, однако Проблема остается и по сей день нерешенной.

Любопытно, что в Плате фундаментальных исследований Российской академии наук на период 2011–2025 годов [11] в числе важнейших научных проблем, которые намечается решить в прогнозируемый период, указано «Решение "проблемы перебора" (называемой также P - NP проблемой) в теории алгоритмов».

Надо признаться, что автор совместно с коллегой – Владимиром Александровичем Козмидиади, ушедшим от нас в 2017 году, – также затратили значительное количество времени и усилий в попытке решить Проблему. Был момент в 2012 году, когда казалось, что Проблему удалось решить. Была даже напечатана небольшим тиражом брошюра [12]. Автор 20 июня 2012 года написал письмо профессору Бостонского университета (США) Леониду Левину, одному из авторов формулировки Проблемы, с сообщением о том, что подготовлен препринт с доказательством Проблемы и с просьбой о совете относительно дальнейших шагов. Надо сказать, что автор был знаком с Л. Левиным по учебе в школе-интернате и в Московском университете. Интересно привести фрагмент ответного письма Леонида Левина, отправленного буквально в тот же день.

Dear Victor: Nice to hear from you, and please say hi to Dr. Kozmidiadi (if he remembers me).

You are asking a difficult question. As you can imagine, there are many people who claim extraordinary results, like P _vs_ NP , Riemann Hypothesis, etc., as there were many Fermat T. provers in the past. It was very hard to find somebody willing to read a proof of Fermat Theorem. When it was finally proven, it followed great many smaller insights, so skepticism was much reduced by these previous achievements.

Of course, you would need to put it at some permanent public archive, like arxiv.org, in order for anybody to look at it. At least people will see that you are sufficiently serious to risk your entire lifetime reputation on this proof. You should realize that an erroneous proof would ruin your reputation forever. But exactly for this reason, I would suggest, to precede such posting with maximal preparation.

You need to make the proof very clean, so nobody thinks that you are hiding errors in some obscure places. It would be great if formal steps are supplemented with intuitive comments making clear the overall structure and the purpose of each part of the proof. Notations should also be set carefully, to avoid awkwardness.

You should check everything VERY carefully yourself, then with friends. Of course, it better be in English (Math texts are easy to translate).

Перевод «Дорогой Виктор: рад слышать тебя, и, пожалуйста, передай привет доктору Козмидиади (если он помнит меня).

Ты задаешь сложный вопрос. Как ты можешь представить, есть много людей, которые объявляют о получении экстраординарных результатов, вроде P_vs_NP , Гипотезы Римана и т.п., так же как было много доказывающих теорему Ферма в прошлом. Было очень трудно найти желающего прочесть доказательство теоремы Ферма. Когда она наконец была доказана, последовало множество более мелких результатов, так как скептицизм был в значительной степени уменьшен этими предыдущими достижениями.

Конечно, вам потребуется выложить (результат) в какой-нибудь постоянный публичный архив, вроде arXiv.org, для того, чтобы любой мог посмотреть его. По крайней мере люди увидят, что вы достаточно серьезны, рискуя всей своей жизненной репутацией, выставляя это доказательство. Вы должны понимать, что ошибочное доказательство навсегда разрушит вашу репутацию. И точно по этой же причине я бы предложил предварить выкладывание максимальной подготовкой.

Вам следует сделать доказательство очень ясным, так чтобы никто не подумал, что вы скрываете ошибки в каких-то "мутных" местах. Будет очень хорошо, если формальные шаги будут дополнены интуитивными комментариями, делающими ясной всю структуру и цель каждой части доказательства. Обозначения должны быть также введены очень тщательно, чтобы избежать непонимания.

Вам следует проверить все ОЧЕНЬ тщательно самим, затем с товарищами. Конечно, лучше иметь текст на английском (математические тексты легко переводить)».

Я очень благодарен Леониду Левину за четкое и откровенное письмо – образец настоящего подхода ученого. Следуя его совету и тщательно проверяя написанное, 12 июля автору самому удалось обнаружить ошибку в доказательстве одной из теорем (при подсчете числа шагов). К счастью, удалось это сделать до публичного обнародования доказательства.

На данное время Проблема остается нерешенной и по-прежнему привлекает внимание многих исследователей.

Литература

1. <http://www.claymath.org/millennium-problems>.
2. Cook S.A. The complexity of theorem Proving Procedures // Proceedings of the 3rd ACM Symposium on Theory of Computing. 1971. P. 151–158.
3. Левин Л.А. Универсальные задачи перебора // Проблемы передачи информации. 1973. Т. 9, № 3. С. 115–116.
4. Levin L. Universal sequential search problems (in Russian) // ProblemyPeredachiInformatsii. 1973. Vol. 9(3). P. 115–116. English translation in B.A. Trakhtenbrot, A survey of Russian approaches to Perebor (brute-force search) algorithms. Annals of the History of Computing. 1984. No. 6. P. 384-400.

5. Cook S.A. The P versus NP Problem. URL: <http://www.claymath.org/sites/default/files/pvsnp.pdf> (дата обращения: 15.02.2019).
6. Zakharov V.N., Kozmidiadi V.A. On Relationships between Complexity Classes of Turing Machines // Computational Mathematics and Mathematical Physics. 2017. Vol. 57, No. 4, P. 726-738.
7. Интернет-Университет Информационных Технологий. Лекция: Класс NP: сводимость и полнота. <http://www.INTUIT.ru>.
8. Karp R.M. Reducibility Among Combinatorial Problems / Raymond E. Miller and James W. Thatcher (ed.) // Complexity of Computer Computations (Symposium Proceedings). New York: Plenum Press, 1972. P. 85–103.
9. Garey M.R. and Johnson D.S. Computers and Intractability: a Guide to the Theory of NP-Completeness. San Francisco: W.H. Freeman and Co., 1979.
10. <https://www.win.tue.nl/~gwoegi/P-versus-NP.htm>.
11. План фундаментальных исследований Российской академии наук на период до 2015 года. <http://www.ras.ru/FStorage/Download.aspx?id=0bd21e6b-b3d7-45d1-ad74-d84ed16bf7e6>.
12. Козмидиади В.А., Захаров В.Н. Многоголовые автоматы и проблема «P versus NP» // ИПИ РАН. 2012. 136 с.

V.N. Zakharov

Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences (FRC CSC RAS)

ABOUT THE PROBLEM "P VERSUS NP"

Keywords: *problem «P versus NP», computational complexity theory, deterministic Turing machines, nondeterministic Turing machines, NP-completeness.*

Abstract. *In 2000, the Clay Mathematical Institute (USA) published a list of the so-called millennium problems, for solving each of which a prize of \$ 1 million was promised. This list contained seven unsolved mathematical problems that were identified as “important classical problems whose solution has not been found for many years” [1]. At present, only one of these problems (the Riemann hypothesis) has been solved in 2010 by the Russian mathematician Gregory Perelman. Among the unresolved problems from the list, the most attention is drawn to the problem “P versus NP”, which relates to the theory of algorithms and which is often also called the “bust problem”. Perhaps this is due to the simplicity of its formulation. In the most general form, the essence of the problem can be formulated as follows: Is it true that if it is rather simply to verify that the presented solution of a certain problem is correct, then this solution itself can be found quite simply? The article presents the exact formulation of the problem in terms of Turing machines, defines the complexity classes, shows some correlations between these classes, and also provides information about published attempts to solve the problem.*

ОБ ОПРЕДЕЛЕНИИ ЭКВИВАЛЕНТНОСТИ ГРАФИЧЕСКИХ ОБРАЗОВ СРЕДСТВАМИ МОБИЛЬНЫХ ОС

Ключевые слова: *распознавание образов, мобильные средства, мобильные ОС, теория распознавания объектов, чтение штрих кода, распознавание лиц, распознавание текста.*

В статье рассматривается проблема представления информации с помощью визуального объекта, необходимости его определения. На базе такого объекта, представленного в дальнейшем графическим образом, рассматриваются возможности определения эквивалентности графических образов, сферы применения выбранной технологии, а также возможность ее реализации средствами мобильных операционных систем. Анализируются имеющиеся подходы и решения в данной сфере, их результаты и возможные улучшения ввиду того, что область использования данной технологии обширна и требует глубокого изучения.

Постановка задачи. В настоящее время имеется потребность в представлении информации с помощью визуального объекта, обладающего неким набором свойств и методов его преобразования. Это позволяет систематизировать данные, находить новые методы анализа и определять связи между ними, а также получать новую, неизвестную до этого информацию. Обусловлено все это быстрым ростом объема данных в силу новых технологий передачи и обработки информации, появлением сети Интернет и обеспечением каждого человека вычислительными средствами.

Ввиду широкого распространения мобильных устройств, представителями которых являются не только планшеты и телефоны, но и специальное оборудование, обеспеченное поддержкой мобильных операционных систем, можно заключить, что существует необходимость анализа применения способов и методов обработки графических образов технологическими средствами на базе мобильных операционных систем.

Анализ использования технологии

Ввиду того, что многие разработанные решения базируются на скрытых алгоритмах, некоторые методы и способы описаны кратко.

Одной из базовых задач технологии определения графических образов является считывание штрих или QR кода, представленного графической информацией на какой-либо поверхности и состоящего из

полос черного и белого цветов. Чтение штрих-кода осуществляется посредством проецирования и отражения от имеющейся поверхности света и преобразованием полученной информации в бинарный цифровой сигнал. Полученный сигнал декодируется.

Обработка изображения в таком случае зависит от принципа чтения штрих-кода. По их классификации существует несколько методов считывания: ПЗС (прибор с зарядной связью) метод, лазерный метод, реп метод для линейных штрих-кодов, а также множество методов чтения двумерных штрих-кодов.

Камеры, установленные в современных мобильных устройствах, позволяют осуществлять чтение штрих-кода или QR кода по принципу, схожему с заложенным в сканер. Пример применения данной технологии средствами мобильных устройств – замена наличных платежных средств электронными. QR код используется в качестве метки считывания данных о платежеспособном клиенте для оплаты счетов. Таким образом, можно сделать вывод о том, что современные вычислительные устройства со считывающими камерами и соответствующим программным обеспечением способны решить проблему преобразования графической информации (штрих и QR кода) в цифровую.

В сфере теории распознавания образов базовой и основополагающей технологией является ее практическое приложение – технология распознавания лиц. Имеется несколько направлений исследований проблемы распознавания лиц, к которым относятся:

- нейropsихологические модели;
- нейрофизиологические модели;
- информационно-процессуальные модели;
- компьютерные модели распознавания.

Хотя большинство систем распознавания объектов не раскрывают алгоритмы, можно предположить, что широко используется метод перцептивного хеширования [1].

В настоящее время технологию распознавания лиц в мобильных операционных системах широко используют для обеспечения безопасности пользовательских данных, начиная с разблокирования мобильного устройства при помощи сканирования лица и заканчивая блокировкой персональных данных конкретного типа.

Примером является технология Face ID, разработанная компанией «Apple» и используемая в операционных системах «iOS». Для получения данных лица проецируются и анализируются более 30 000 точек сканирования для составления подробной структурной карты лица, а также создается изображение лица в инфракрасном спектре. Карта глубины и инфракрасное изображение преобразуются в математическое представление, которое позже проходит этап сравнения с

зарегистрированными данными лица пользователя. Этап первой регистрации проходит при первом запуске мобильного устройства. Алгоритм преобразования карты глубин и инфракрасного изображения не раскрывается. Также используются методы машинного обучения, позволяющие определить пользователя и разблокировать устройство в тех случаях, когда пользователь изменился внешне. К таким изменениям относятся как наличие или отсутствие макияжа, так и небритость, наличие аксессуаров одежды, контактных линз и др.

Другой основной технологией теории распознавания образов является технология распознавания объектов. Распознавание объектов может происходить как по фотографии, тексту, обложке книги, изображению с витрин магазинов и других видов информации, представленной в виде графического образа и нанесенного на какую-либо поверхность, так и представленной в виде реального объекта.

В таком случае для решения задачи определения такого объекта предлагается модель вида:

$$P(O_c, O, Par) = \begin{cases} 0 \\ 1 \end{cases},$$

где P – предикат, O_c – текущий объект, полученный в ходе сканирования с использованием камеры мобильного устройства, $O = \{O_1, O_2, \dots, O_n\}$ – совокупность объектов, с которыми идет сравнение текущего, $Par = \{Par_1, Par_2, \dots, Par_k\}$ – совокупность параметров, передаваемых предикату, где n – число объектов для сравнения, k – число параметров, которые задаются для той или иной задачи практическим способом. Правая часть равенства представлена 0 и 1 и характеризует совпадение текущего объекта с каким-либо из имеющейся базы (1 означает совпадение). Такое представление обусловлено необходимостью применения нечеткой логики. Перцептивное хеширование может быть использовано как технология представления и сравнения изображений, поскольку совпадение определяется по расстоянию Хэмминга в двоичном векторном пространстве.

Примером технологии распознавания таких образов является «Google Lens», распространяемая компанией «Google» в качестве отдельного одноименного приложения, а также в качестве встроенных средств в приложения «Google Ассистент» и «Google Фото». Приложение и встроенные средства базируются на средствах машинного обучения открытой программной библиотеки «TensorFlow», разработанной также компанией «Google».

Ввиду своей открытости и поддержки со стороны разработчиков на программно-аппаратном уровне, библиотека «TensorFlow» является широко распространенным набором инструментов для программистов. Работа библиотеки строится на базе построения графа вычислений

алгебраических выражений в префиксной форме (рис. 1). Библиотека конструирования процедур выполнения вычислений достаточно проста в использовании.

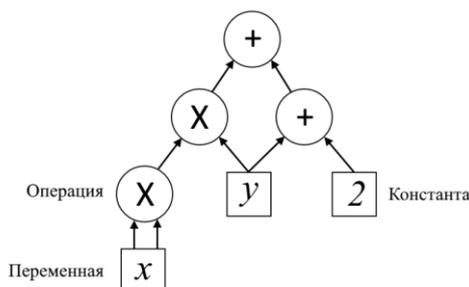


Рис. 1. Пример графа вычисления функции $f(x, y) = x^2y + y + 2$

Анализ «Google Lens» показал, что способы и методы распознавания объектов на текущий момент входят в базовый пакет инструментов «ML Kit», представленный компанией «Google» в мае 2018 года. Он позволяет распознавать объекты по фото, текст, лица, сканировать штрих-коды и QR коды, получать информацию об объектах достопримечательности.

Из менее распространенных ввиду их частного применения, закрытости или сервис-ориентированности разработчикам предоставляются API следующих планов:

- Kairos – распознавание лиц;
- Mobile Vision – ранние наработки будущего пакета «ML Kit», используемые отдельно от основного пакета;
- OpenCV – низкоуровневая открытая библиотека алгоритмов компьютерного зрения, изначально базируемая для стационарных аппаратов;
- Tesseract – изначально базирующийся как программное средство для UNIX-подобных и Windows систем, но затем перестроенный под библиотеку набор инструментов оптического распознавания символов.

Согласно статистике, за предыдущий год большую популярность среди всех аппаратных средств занимает «TensorFlow» (рис. 2).

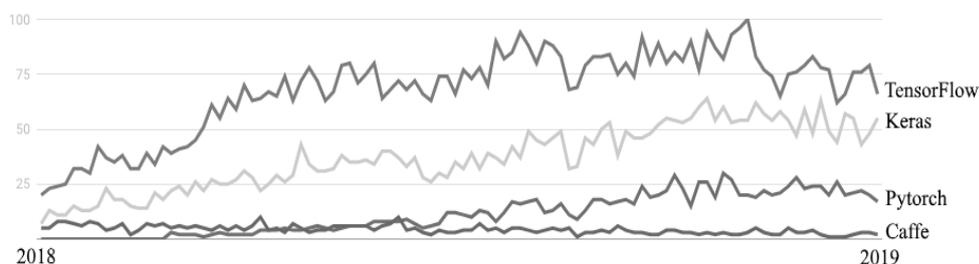


Рис. 2. Статистика использования машинного обучения на базе различных библиотек

Вывод. Проведенный анализ показал, что изучение технологии определения эквивалентности графических образов в настоящее время является одним из главных направлений развития научно-технического прогресса. С практической точки зрения такая технология может быть применена во многих сферах деятельности человека.

Говоря об использовании данной технологии в мобильных операционных системах следует отметить факт наличия достаточного количества предложенных методов и средств для решения частных задач распознавания объектов, стоит упомянуть и тот факт, что часть таких методов и средств базируются на скрытых алгоритмах, а некоторые из них являются ресурсоемкими. Это означает, что необходимо дальнейшее изучение данной технологии и ее улучшение.

Литература

1. Monga V., Evans B.L. Perceptual Image Hashing Via Feature Points: Performance Evaluation and Tradeoffs // IEEE Signal Processing Society. 2006. P. 3452-3465.
2. Бринк Х., Ричардс Дж., Феверолф М. Машинное обучение. М: Изд-во «Питер», 2017. 266 с.
3. Потапов А. Распознавание образов и машинное восприятие. М: Изд-во «Политехника», 2007. 552 с.
4. Зуев А.С. Применение методов определения эквивалентности графических образов // Студенческая наука – 2018: сборник статей. Смоленск: СмолГУ, 2018.
5. Abrahams S., Hafner D., Erwitte E., Scarpinelli A. TensorFlow for machine intelligence: A hands-on introduction to learning algorithms–Bleeding Edge Press, 2016. 298 p.
6. Google ML Kit. Machine learning for mobile developers. 2018. URL: <https://developers.google.com/ml-kit>.

E.P. Emelchenkov, A.S. Zuev
Smolensk State University

ABOUT THE DETERMINATION OF EQUIVALENCE FOR GRAPHIC IMAGES BY RESOURCES OF MOBILE OS

Keywords: *images recognition, mobile resources, mobiles os, object recognition theory, barcode reading, face recognition, text recognition.*

Abstract. *This article reviews the problem of presenting information using a visual object and the need of its define. On the basic of such object, presented hereafter graphically, the article reviews the possibilities of determining the equivalence of graphic images, the scope of selected technology's application, and the possibility of its implementation by mobile operational systems' resources. Existing approaches and solutions in this area, their results and possible improvements are analyzed in view of the fact that the area of technology's use is extensive and requires in-depth study.*

ГЕНЕРАЦИЯ ХРАНИМЫХ ПРОЦЕДУР ДЛЯ РЕАЛИЗАЦИИ АЛГОРИТМА БУСТРОФЕДОНА

Ключевые слова: *генерация хранимых процедур, алгоритм бустрофедона, индексно-последовательный метод доступа, метаданные.*

В работе описано создание хранимой процедуры для реализации алгоритма бустрофедона, рассмотрен индексно-последовательный метод как наиболее популярный метод индексирования. Показана работа с метаданными, рассмотрено решение проблемы вычислительной сложности операции JOIN посредством работы с метаданными.

В статье рассматривается предложенный авторами метод для автоматизации разработки хранимых процедур, реализующих симметричное горизонтальное распределение таблиц, участвующих в операции Join для последующего параллельного выполнения этой операции. Параллельная реализация этой операции над таким образом распределенными данными подробно рассмотрена в работах [1; 2].

Особенность предложенного метода заключается в том, что вся подготовка к параллельному выполнению операции Join производится на основе использования метаданных, характеризующих свойства таблиц-операндов. Это позволяет существенно сократить время распределения таблиц по запоминающим устройствам вычислителей, выполняющих операцию над фрагментами таблиц-операндов. Для создания метаданных, используемых при симметричном горизонтальном распределении таблиц, используется индексно-последовательный метод доступа (ISAM). Эти метаданные представляют собой специальные индексные таблицы, которые содержат сведения о количествах строк, соответствующих каждому значению ключа операции Join. Симметричное горизонтальное распределение таблиц реализуется алгоритмом бустрофедона [1; 3].

Далее рассматривается реализация предложенного метода, представляющая собой CASE-систему, генерирующую на основе описания таблиц (их системных метаданных) и свойств операции Join хранимую процедуру, реализующую создание индексных таблиц и распределение таблиц-операндов алгоритмом бустрофедона.

Пусть в базе данных содержатся таблицы $T_1(K, F_1)$ и $T_2(K, F_2)$, над которыми необходимо выполнить операцию Join. F_1, F_2 – наборы

атрибутов, которые участвуют в вычислениях, но не участвуют в идентификации, а $K=\{K_1, \dots, K_m\}$ – набор идентифицирующих атрибутов, одинаковый в обеих таблицах, и используемый в операции Join в выражении

ON $T_1.K_1=T_2.K_1$ AND ... AND $T_1.K_m=T_2.K_m$.

Требуется распределить эти таблицы по заданному числу вычислителей таким образом, чтобы вычислительные сложности операций Join над соответствующими друг другу фрагментами незначительно различались. Для этого нужно выполнить запросы вида ($i=1, 2$):

INSERT INTO $indT_i$ SELECT K, SUM (1) AS M_i FROM T_i GROUP BY K

Результаты этих запросов – индексные таблицы $indT_1(K, M_1)$ и $indT_2(K, M_2)$, где M_1 и M_2 – количества строк, соответствующих фиксированному значению ключа K. Процесс построения индексных таблиц показан на рисунке 1.

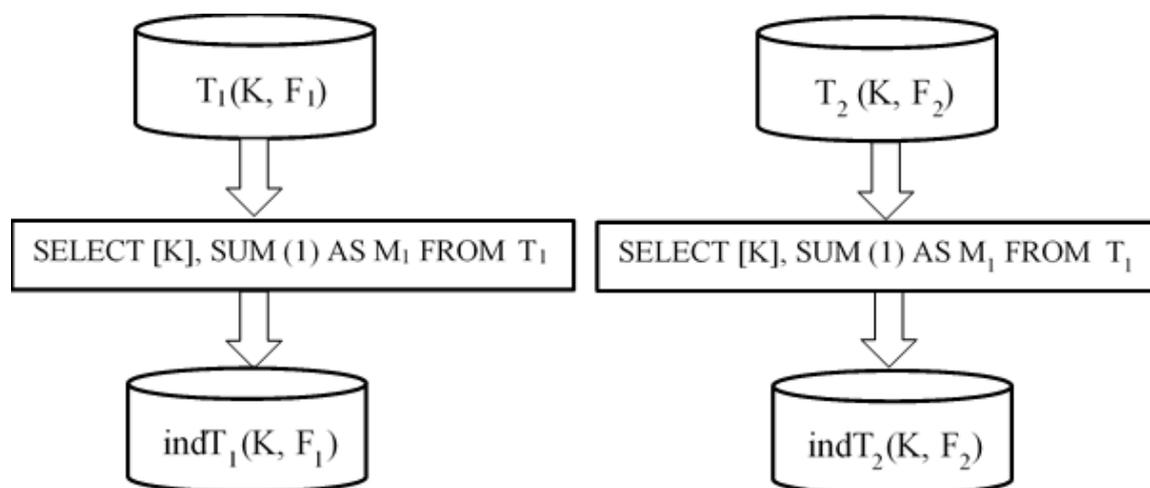


Рис. 1. Формирование индексных таблиц

Затем, запросом

SELECT $indT_1.K, M_1 * M_2$ AS M FROM $indT_1$ INNER JOIN $indT_2$

ON $indT_1.K = indT_2.K$ строится сводная таблица, которая используется алгоритмом бустрофедона.

Произведение $M_1 * M_2$ определяет вычислительную сложность операции Join над фрагментами таблиц T_1 и T_2 , которая вычисляется по формуле $\sum_{(K')} M_1 \cdot M_2$ (K' – набор значений ключа в соответствующих фрагментах).

Интерфейс программиста, использующего CASE-систему, позволяет ему по заданным именам таблиц $T_1(K_1, \dots, K_m, F_1)$ и $T_2(K_1, \dots, K_m, F_2)$, получить список их полей (рис. 2а) и установить связи между ключами (рис. 2б).

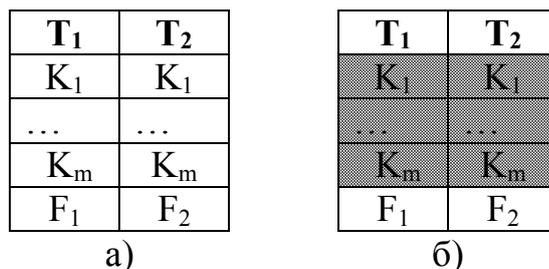


Рис. 2. Поля таблиц и связи между ключами Join

Система реализует эту возможность с помощью запросов вида
SELECT * FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME= T_i, (i=1, 2). Поскольку для генерации
индексных таблиц потребуются типы полей K₁, ... , K_m, то для их
получения выполняются последовательности запросов вида:
SELECT DATA_TYPE, CHARACTER_MAXIMUM_LENGTH
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME= T_i AND COLUMN_NAME=K_j,
(i=1, 2; j=1, ..., m).

На основе полученной от программиста информации о таблицах
(системных метаданных) и связях их ключей в операции Join
(оперативных метаданных) генерируется хранимая процедура с
заголовком

```
CREATE PROCEDURE [dbo].[Create_Index_T1_T2]
AS
```

В теле процедуры генерируются операторы, выполняющие
следующие действия:

1. Если в базе данных уже существуют индексные таблицы для
таблиц T₁ и T₂ (indT₁ и indT₂), то эти индексные таблицы удаляются и
создаются заново: таблицы

```
IF OBJECT_ID('[dbo].[indT1]') IS NOT NULL
DROP TABLE [indT1]
IF OBJECT_ID('[dbo].[indT2]') IS NOT NULL
DROP TABLE [indT2]
CREATE TABLE[indT1] (
K1 <тип поля>,
...
Km <тип поля>,
M int
);
CREATE TABLE[indT2] (
K1 <тип поля>,
...
Km <тип поля>,
```

M int);

2. Затем генерируются два запроса на заполнение созданных индексных таблиц:

```
INSERT INTO[indT1]([K1], ... [Km], M)
SELECT [T1].[K1], ... [T1].[Km], SUM(1) AS M
FROM[T1]
GROUP BY [K1], ... [Km] ORDER BY [K1], ... [Km]
INSERT INTO[indT2]([K1], ... [Km], M)
SELECT [T1].[K1], ... , [T2].[Km], SUM(1) AS M
FROM[T1]
GROUP BY [K1], ... [Km] ORDER BY [K1], ... [Km]
```

Эти запросы переносят в индексные таблицы наборы фиксированных значений ключей K_1, \dots, K_m и групповой операцией SUM(1) вычисляют и для каждого такого набора число соответствующих ему строк. Сортировка ORDER BY $[K_1], \dots, [K_m]$ включена в запрос исключительно с технической целью: для облегчения отладки сгенерированных процедур симметричного горизонтального распределения таблиц.

3. На последнем этапе генерируется оператор создания сводной индексной таблицы и оператор выхода из процедуры.

```
INSERT INTO[CommInd]([K1], ... [Km], M)
SELECT [indT1].[K1], ... [indT1].[Km], indT1.M * indT2.M AS M
FROM indT1 INNER JOIN[indT2]
ON [indT1].[K1] =[indT2].[K1] AND ... AND [indT1].[Km]
=[indT2].[Km]
RETURN 0
```

Этот вариант построения индексной таблицы целесообразен в том случае, когда суммарная размерность типов ключей K_1, \dots, K_m существенно меньше суммарной размерности типов остальных полей в обеих таблицах. В этом случае операция Join над данными фактически подменяется аналогичной операцией над метаданными существенно меньшего объема и, соответственно выполняется за гораздо меньшее время. В противном случае может быть сгенерирована другая (более сложная) последовательность операторов, которая генерирует таблицу CommInd за один проход таблиц indT₁ и indT₂.

Предложенная CASE-система разработана в системе программирования Microsoft Visual Studio 2017 для системы управления базами данных Microsoft Sql Server 2016 на языке программирования C#. Процедура подготовки метаданных для симметричного горизонтального

распределения таблиц на основе полученных из базы данных системных метаданных генерируется в виде одной командной строки, которая помещается в свойство Text объекта SqlCommand. Эта строка содержит последовательность операторов языка Transact SQL. Затем, метод ExecuteNonQuery этого объекта вызывает запрос на создания скрипта, который размещает текст процедуры в разделе Programming базы данных и транслирует ее.

Таким образом, можно утверждать, что предложенная в работе CASE-система обеспечивает генерацию достаточно сложного программного обеспечения с помощью простого и доступного интерфейса. Полученные в результате работы сгенерированной хранимой процедуры метаданные обеспечивают дальнейшее использование основного алгоритма бустрофедона.

Литература

1. Мунерман В.И. Опыт массовой обработки данных в облачных системах (на примере Windows Azure) // Системы высокой доступности. 2014. Т. 10, №. 2. С. 3-8.
2. Мунерман В.И., Мунерман Д.В. Параллельная реализация операций обработки файлов // Системы компьютерной математики и их приложения. Смоленск: Изд-во СмолГУ, 2016. Вып. 17.
3. Кравченко В.В., Мунерман В.И., Симакова А.А Анализ алгоритма бустрофедона на больших массивах данных. // Системы компьютерной математики и их приложения. Смоленск: Изд-во СмолГУ, 2014. Вып. 15.

M.S. Ivanov, V.I. Munerman
Smolensk State University

GENERATION OF PROCEDURES FOR THE IMPLEMENTATION OF THE BUDROPHEDONE ALGORITHM

Keywords: *generation of stored procedures, algorithm of a bustrofedon, index-sequential access method, metadata.*

Abstract. *The work describes the creation of a stored procedure for the implementation of the algorithm of boustrophedon, considered an index-sequential method, as the most popular indexing method. The work with metadata is shown, the solution of the problem of computational complexity of the JOIN operation by means of working with metadata is considered.*

РЕКУРСИВНОЕ ВЫЧИСЛЕНИЕ ДЕТЕРМИНАНТА МНОГОМЕРНОЙ МАТРИЦЫ

Ключевые слова: рекурсивный алгоритм; многомерная матрица; детерминант; системы счисления.

В статье предложен рекурсивный алгоритм вычисления детерминанта многомерной матрицы. Рассмотрено представление индексов соответствующих элементов матрицы в виде n -ичных чисел и создание матрицы таких индексов. Представлен рекурсивный алгоритм, позволяющий построить все трансверсали обходом матрицы индексов. На базе построенных трансверсали вычисляется детерминант многомерной матрицы.

Основные понятия алгебры многомерных матриц. Основные определения, используемые в статье, даны в книге [1].

Любая система из n_1, \dots, n_p элементов $A_{i_1 \dots i_p}$, где $i_\alpha = 1, \dots, n_\alpha$, поля P , расположенных в точка p -мерного пространства, определяемых координатами i_1, \dots, i_p образует p -мерную $n_1 \times \dots \times n_p$ – матрицу над P . При $n_1 = \dots = n_p$, это p -мерная матрица порядка n над P .

$$A = \left\| A_{i_1 i_2 \dots i_p} \right\| \left(i_1, i_2, \dots, i_p = 1, 2, \dots, n \right) \quad (1)$$

Совокупность элементов матрицы (1) с фиксированными значениями индекса $\overline{i_\alpha}$ ($1 \leq \alpha \leq p$) образует простое сечение ориентации (i_α) , являющееся $(p-1)$ -мерной матрицей порядка n .

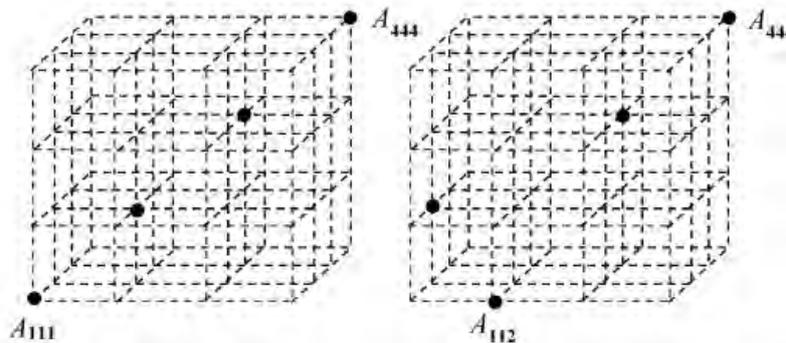


Рис. 1. Диагональ $A_{111} A_{222} A_{333} A_{444}$ и транверсаль $A_{112} A_{211} A_{333} A_{444}$ трехмерной матрицы

Элементы матрицы, взятые в количестве, не превосходящем n , называются трансверсальными, если ни одна их пара не принадлежит

одному и тому же простому сечению какой-либо ориентации. Совокупность n трансверсальных элементов образует трансверсаль матрицы. Число всех трансверсалий матрицы равно $(n!)^{p-1}$.

На рисунке 1 приведен пример трансверсали для трехмерной матрицы порядка 4.

Возьмём в матрице (1) какую-нибудь трансверсаль:

$$A_{i_1^{(1)}i_2^{(1)} \dots i_p^{(1)}}, A_{i_1^{(2)}i_2^{(2)} \dots i_p^{(2)}}, \dots, A_{i_1^{(n)}i_2^{(n)} \dots i_p^{(n)}}.$$

Значения $i_v^{(1)}, i_v^{(2)}, \dots, i_v^{(n)}$ индексов i_v ($v=1, 2, \dots, p$) в элементах трансверсали образуют перестановку из чисел $1, 2, \dots, n$. Составим произведение этих элементов:

$$A_{i_1^{(1)}i_2^{(1)} \dots i_p^{(1)}} A_{i_1^{(2)}i_2^{(2)} \dots i_p^{(2)}} \dots A_{i_1^{(n)}i_2^{(n)} \dots i_p^{(n)}}. \quad (2)$$

Пусть m – четное неотрицательное число, $m \leq p$. Выберем из индексов i_1, i_2, \dots, i_p какие-нибудь m индексов и определим число инверсий I_μ в перестановке $i_\mu^{(1)}, i_\mu^{(2)}, \dots, i_\mu^{(n)}$, образуемой значениями индекса i_μ ($1 \leq \mu \leq m$). Умножая произведение (2) на $(-1)^{\sum_{\mu=1}^m I_\mu}$, получим:

$$(-1)^{\sum_{\mu=1}^m I_\mu} A_{i_1^{(1)}i_2^{(1)} \dots i_p^{(1)}} A_{i_1^{(2)}i_2^{(2)} \dots i_p^{(2)}} \dots A_{i_1^{(n)}i_2^{(n)} \dots i_p^{(n)}}. \quad (3)$$

Значения индекса i_1 естественно предполагать идущими в натуральном порядке.

Индексы i_μ ($\mu=1, 2, \dots, m$), от которых зависит знак выражения (3), называются альтернативными, остальные $p-m$ индексов – неальтернативными.

Алгебраическая сумма выражений вида (3) есть p -мерный детерминант порядка n $\left| A_{\substack{\pm \pm \pm \dots \pm + + \\ i_1 i_2 \dots i_m i_{m+1} \dots i_p}} \right|_n$ с сигнатурой $\begin{pmatrix} \pm \pm & \pm + & + \\ i_1 i_2 & \dots & i_m i_{m+1} & \dots & i_p \end{pmatrix}$, где альтернативные индексы обозначены знаком \pm , а неальтернативные – знаком $+$.

Таким образом, имеем:

$$\left| A_{\substack{\pm \pm \pm \dots \pm + + \\ i_1 i_2 \dots i_m i_{m+1} \dots i_p}} \right|_n = \sum (-1)^{\sum_{\mu=1}^m I_\mu} A_{i_1^{(1)}i_2^{(1)} \dots i_p^{(1)}} A_{i_1^{(2)}i_2^{(2)} \dots i_p^{(2)}} \dots A_{i_1^{(n)}i_2^{(n)} \dots i_p^{(n)}}$$

Задача. Требуется вычислить детерминант p -мерной матрицы порядка n , используя рекурсивный алгоритм нахождения трансверсалий.

Важно отметить, что для ускорения работы алгоритма будут рассмотрены лишь случаи, когда порядок матрицы $n = 2^k, k = 1, 2, 3$.

Нетрудно обобщить используемый алгоритм на произвольный порядок матрицы.

Представление индексов многомерной матрицы. Пусть дана матрица (1).

Индексы её элемента $A_{i_1 i_2 \dots i_p}$ могут быть представлены как соответствующие разряды p -значного n -ичного числа, $\overline{i_p \dots i_2 i_1}$ ($i_\alpha = 0, 1, \dots, n-1; \alpha = 0, 2, \dots, p-1$) [2].

Утверждение. Пусть n – произвольное натуральное число, $n \geq 1$; тогда существует и единственное представление любого натурального числа x в виде:

$$x = i_{p-1}n^{p-1} + \dots + i_1n + i_0.$$

Тогда, если определить индексы элементов пространственной матрицы как n -ичные числа, саму матрицу естественно рассмотреть как матрицу-столбец $n^p \times 1$. Индекс каждого элемента матрицы представляется в виде n -ичного числа, и каждое такое число представляется единственным образом в виде натурального десятичного числа.

Таким образом, построена биекция между индексами элементов пространственной матрицы и номерами строк матрицы-столбца. В этом виде удобно представлять пространственную матрицу произвольного порядка и размерности в программе, так как легко получить номер элемента матрицы в столбце и обращаться к нему по этому номеру, используя перевод числа из n -ичной системы счисления в десятичную.

Задание матрицы индексов. Рассмотрим сечения матрицы (1) ориентации (i_1). Пусть $i_1 = 0$. Тогда выбранное сечение составлено элементами с индексами:

$$\overline{0_{p-1}0_{p-2} \dots 0_1 0_0}, \overline{0_{p-1}0_{p-2} \dots 0_1 1_0}, \dots, \overline{0_{p-1}n-1_{p-2} \dots n-1_1 n-1_0}.$$

Таких элементов n^{p-1} . Представим индексы $\overline{0_{p-1}0_{p-2} \dots 0_1 0_0}$ элемента матрицы как число в n -ичной системе счисления, получим $\overline{0_{p-1}0_{p-2} \dots 0_1 0_0}$. Прибавляя к этому числу единицу n^{p-1} раз, мы пересчитаем все числа до $\overline{0_{p-1}n-1_{p-2} \dots n-1_1 n-1_0}$ включительно. Занесем такие числа последовательно в матрицу-столбец $n^{p-1} \times 1$.

Пусть $i_1 = 1$. Тогда выбранное сечение составлено элементами с индексами:

$$\overline{1_{p-1}0_{p-2} \dots 0_1 0_0}, \overline{1_{p-1}0_{p-2} \dots 0_1 1_0}, \dots, \overline{1_{p-1}n-1_{p-2} \dots n-1_1 n-1_0}.$$

Представим индексы $\overline{1_{p-1}0_{p-2} \dots 0_1 0_0}$ элемента матрицы как число в n -ичной системе счисления, получим $\overline{1_{p-1}0_{p-2} \dots 0_1 0_0}$. Прибавляя к этому

числу единиц n^{p-1} раз, мы пересчитаем все числа до $1_{p-1}n-1_{p-2}\dots n-1_1n-1_0$ включительно. Последовательно занесем такие числа в матрицу-столбец $n^{p-1} \times 1$ и присоединим её справа к матрице, полученной на предыдущем шаге.

Продолжим аналогично до $i_1 = n-1$; получим матрицу $n^{p-1} \times n$, содержащую индексы всех элементов исходной пространственной матрицы.

$$\left(\begin{array}{ccc} 0_{p-1}0_{p-2}\dots 0_0 & \dots & n-1_{p-1}0_{p-2}\dots 0_0 \\ 0_{p-1}0_{p-2}\dots 1_0 & \dots & n-1_{p-1}0_{p-2}\dots 1_0 \\ \dots & \dots & \dots \\ 0_{p-1}n-1_{p-2}\dots n-1_0 & \dots & n-1_{p-1}n-1_{p-2}\dots n-1_0 \end{array} \right) \quad (4)$$

Алгоритм построения трансверсалей. Необходимо построить все трансверсали исходной матрицы. Для этого следует воспользоваться рекурсивным алгоритмом обхода матрицы (4).

Суть алгоритма: перебор индексов элементов для выделения трансверсальных элементов.

Естественно представить все возможные варианты перебора в виде графа. Таким образом, исходная задача сводится к полному обходу графа.

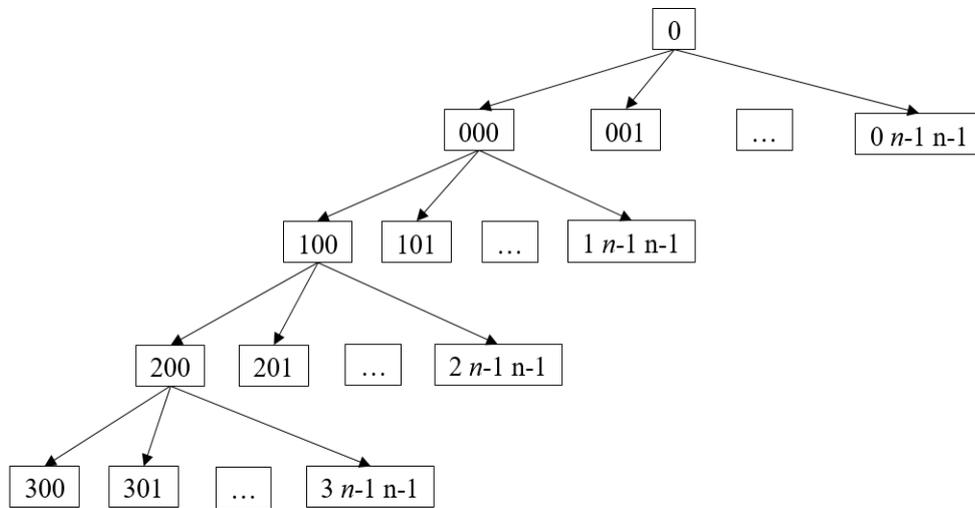


Рис. 2. Представление матрицы (4) для трехмерной матрицы порядка 4 в виде графа

Граф, представленный на рисунке 2, ациклический и односвязный, соответствует определению дерева. Корень дерева не участвует в построении трансверсалей, поэтому можно разбить исходный граф на подграфы исключением корня. Полученные графы также будут деревьями. Тогда узлы первого слоя исходного дерева станут их корнями. Такое разбиение удобно тем, что понижает возможную глубину рекурсии.

Для каждого полученного дерева выполняется следующий алгоритм.

1. Фиксация корня.

2. Для вершин, начиная с корня, выполняется процедура O :

если не построена трансверсаль и есть нерассмотренные вершины, выполняется последовательный перебор прилежащих к вершине ребёр;

если найдено ребро, ведущее к вершине с трансверсальным элементом, фиксируется значение в вершине и сама вершина, после чего для неё выполняется процедура O ;

если трансверсаль построена, зафиксированные элементы сохраняются, после чего совершается возврат на один шаг назад и алгоритм продолжается;

если нерассмотренных вершин нет, совершается возврат на один шаг назад.

Таким образом, будут построены все трансверсали без повторов.

Далее для каждой трансверсали нужно составить произведение (3), считая перестановки по указанным индексам, и просуммировать полученные произведения. В итоге будет найден детерминант p -мерной матрицы порядка n с указанной сигнатурой.

Поскольку предложенный рекурсивный алгоритм сводится к полному обходу вершин дерева, к нему могут быть применены методы распараллеливания вычислений, подобные предложенным в работах [3; 4].

Таким образом, в работе приведено решение проблемы вычисления детерминантов многомерной матрицы.

Литература

1. Соколов Н.П. Введение в теорию многомерных матриц. Киев: Наукова думка, 1972. С.5-15.

2. Захаров В.Н., Мунерман В.И. Параллельная реализация обработки интенсивно используемых данных на основе алгебры многомерных матриц // Аналитика и управление данными в областях с интенсивным использованием данных: XVII Международная конференция DAMDID/RCDL. 2015. С. 217-223.

3. Лупин С.А., Посыпкин М.А. Технологии параллельного программирования. Форум, 2008.

4. Посыпкин М.А., Сигал И.Х. Применение параллельных эвристических алгоритмов для ускорения параллельного метода ветвей и границ // Журнал вычислительной математики и математической физики. 2007. Т. 47, №. 9. С. 1524-1537.

P.L. Pjin, V.I. Munerman
Smolensk State University

RECURSIVE COMPUTATION OF THE MULTIDIMENSIONAL MATRIX DETERMINANT

Keywords: *recursive algorithm, multidimensional matrix, determinant, number system.*

Abstract. *The recursive algorithm for computing the determinant of a multidimensional matrix is proposed in the article. The representation of the indices of the corresponding elements of the matrix in the form of n-ary numbers and the creation of a matrix of such indices are considered. The recursive algorithm is presented that allows to construct all transversals by traversing the matrix of indices. the determinant of the multidimensional matrix is calculated on the basis of the constructed transversals.*

А.В. Кирикова, В.И. Мунерман, Т.А. Самойлова
Смоленский государственный университет

УДК 004.22:004.27

РЕАЛИЗАЦИЯ ПОИСКА ИЗОБРАЖЕНИЙ В БАЗАХ ДАННЫХ

Ключевые слова: *параллельное программирование, big data, поиск изображений, перцептивное хеширование, базы данных.*

В статье рассматривается метод повышения эффективности решения задачи поиска изображений в базе данных. Этот метод основан на использовании перцептивного хеширования изображений, трех уровней распараллеливания данных и процедур поиска изображений. Для реализации параллельной обработки данных используются принцип симметричного горизонтального распределения данных и возможности современных процессоров (SIMD регистры и соответствующие им инструкции). Приведены результаты вычислительного эксперимента, подтверждающие эффективность предложенного метода.

Введение. В статье рассматривается метод повышения эффективности решения задачи поиска изображений в базе данных. В настоящее время эта задача актуальна для обработки больших данных во многих предметных областях, например в геоинформационных системах [1].

Данный метод основан на использовании перцептивного хеширования изображений, трех уровней распараллеливания данных и процедур поиска изображений.

Подразумевается, что данные, необходимые для поиска изображений, делятся на два типа: эталонные, долговременно хранящиеся в базе данных, среди которых осуществляется поиск, и

оперативные данные – набор поисковых образов. Два изображения считаются близкими (сравнимыми, схожими), если расстояние Хэмминга между их ключами не превышает заданного значения.

Для уменьшения времени поиска предлагаются следующие подходы:

- сокращение перебора (access scan) за счет введения понятия **вес ключа**;

- использование параллельной обработки данных программно-аппаратным комплексом с **SIMD архитектурой** на всех уровнях вычислений: от выборки данных из базы до сравнения идентификаторов.

Очевидно, что эталонных данных значительно больше, чем оперативных, поэтому предлагается способ распределения данных в программно-аппаратном комплексе, показанный на рисунке 1.

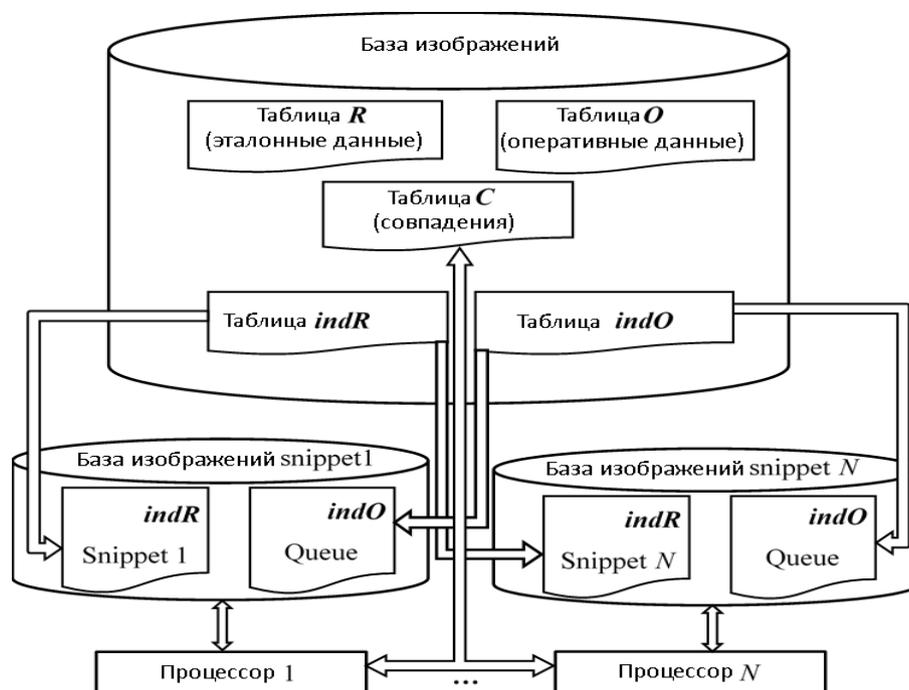


Рис. 1. Организация и распределение данных для поиска изображений

Реализация и анализ метода поиска изображений в базе данных.

Для реализации и анализа предложенного метода поиска изображений в базе данных с использованием ключей изображений – значений перцептивной хэш-функции был проведен вычислительный эксперимент. Он осуществлялся в следующих условиях:

- вычислительная система – рабочая станция с процессором Intel Core i7, оперативной памятью – 16 Гбайт, дисковой памятью – 1 Тбайт;

- программное обеспечение – операционная система Windows 10, система программирования Microsoft Visual Studio 2017, СУБД Microsoft Sql Server 2016;

- языки программирования – Assembler, C++, C#, Transact SQL.

На первом этапе эксперимента проводилась оценка метода сравнения ключей с использованием регистров ХММ. В оперативной памяти было случайным образом сгенерировано 14 миллионов 64-битовых ключей. Для сравнения с ними одного ключа с использованием регистров ХММ потребовалось 7 секунд, без их использования – 41 секунда. Из этого можно сделать вывод о том, что ускорение практически пропорционально числу используемых регистров ХММ.

На втором этапе эксперимента исследовалась производительность предложенного метода распределенного поиска изображений в базе данных. Для этого была сделана оценка количества строк (Q_i) в классах эквивалентности $indR_{w_i}$ таблицы $indR$. Очевидно, что верхняя граница (супремум) Q_i определяется по формуле $C_n^{w_i} = \frac{n!}{w_i!(n-w_i)!}$ (n – размерность ключа). На рисунке 2 распределение значений Q_i для принятого в эксперименте 64-битового ключа показано штриховой линией. В реальных условиях Q_i принимает случайные значения, которые не больше супремума. На рисунке 2 случайное распределение значений Q_i изображается сплошной линией. Количество классов эквивалентности в таблице $indR$ определяется размерностью ключа. В эксперименте их шестьдесят три, так как каждый ключ содержит от одной до шестидесяти трех единиц.

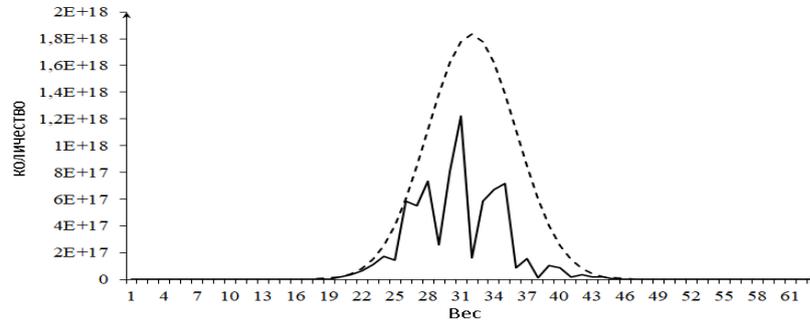


Рис. 2. Распределение значений Q_i

Распределение классов эквивалентности, которое использовалось в эксперименте, показано на рисунке 3. Для этого была создана таблица $indR$, содержащая 3249491 случайным образом сгенерированных ключей.

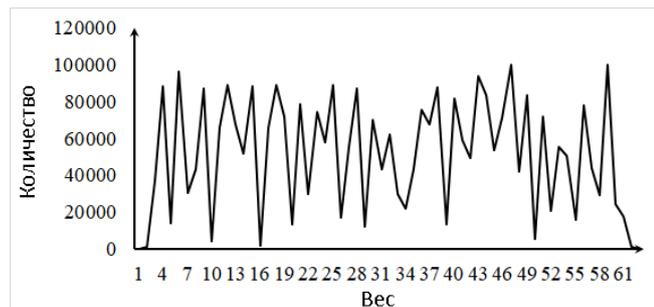


Рис. 3. Распределение значений Q_i для эксперимента

Симметричное горизонтальное распределение этой таблицы *indR* производилось по четырем фрагментам. Была использована хранимая процедура. Общее время выполнения процедуры, включая удаление данных из таблиц-фрагментов, построение таблицы *indRw* и заполнение всех четырех фрагментов таблицы *indR* данными, составило 39 секунд. На рисунке 4 видно, что количество строк в таблицах-фрагментах примерно одинаково: *Snp1R* – 816830, *Snp2R* – 826194, *Snp3R* – 829139, *Snp4R* – 777328.

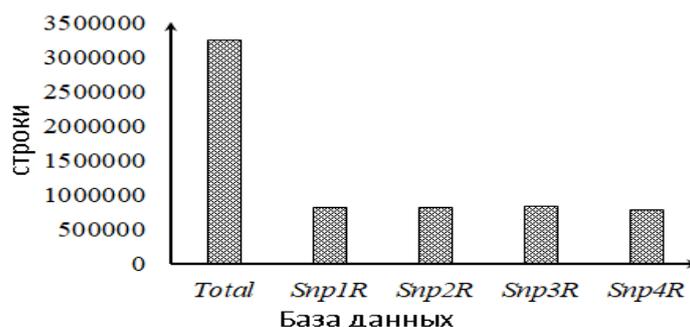


Рис. 4. Результат симметричного горизонтального распределения таблицы *indR*

На третьем, последнем, этапе эксперимента проводилось исследование процедуры поиска изображений в базе данных. Таблица *indO* была сгенерирована пять раз с различным (возрастающим) числом строк: 12880, 17212, 25876, 51904, 129868. Эту таблицу распределена по четырем фрагментам. Затем, четырьмя потоками на четырех ядрах, была параллельно выполнена процедура слияния фрагментов таблиц *indR* и *indO*. В ходе слияния для всех пар ключей из таблиц *indR* и *indO*, для которых выполняется условие $|w - w_i| \leq d$, вычислялось расстояние Хэмминга. Если вычисленное значение не превышало d , формировалась строка таблицы *C*, содержащая оба ключа. Эти строки отправлялись в базу данных образов для дальнейшей обработки.

На рисунке 5 показана зависимость времени слияния таблиц *indR* и *indO* от числа строк в таблице *indO*. На графике видно, что увеличение времени выполнения слияния пропорционально увеличению числа строк в ней. Из этого следует, что увеличение числа процессоров (горизонтальное масштабирование) существенно сокращает время поиска изображений.

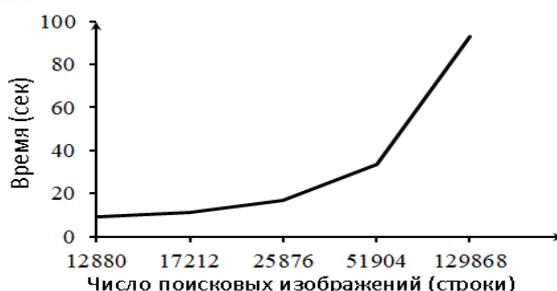


Рис. 5. Зависимость времени поиска от числа строк в таблице *indO*

Рассмотренный в статье метод поиска изображений в базе данных и приведенное в ней описание вычислительного эксперимента позволяют сделать следующие выводы:

- использование перцептивного хеширования изображений позволяет выполнять операции сравнения не самих данных (изображений), а соответствующих им метаданных;

- введение понятия «вес ключа» позволяет существенно сократить перебор ключей в процессе их сравнения;

- использование принципа симметричного горизонтального распределения данных позволяет существенно сократить время сравнения ключей за счет уменьшения количества строк в фрагментах обрабатываемых данных и параллельной обработки этих фрагментов;

- сокращение времени сравнения ключей также достигается за счет использования SIMD регистров процессоров и соответствующих им команд;

- применение технологии «in-database» позволяет использовать возможности параллельной обработки данных, реализованные в современных СУБД.

Сокращение времени поиска изображений пропорционально количеству процессоров, что свидетельствует о высокой масштабируемости программно-аппаратных комплексов, разрабатываемых на основе предложенной архитектуры.

Литература

1. High performance analysis of big spatial data / D. Haynes [et al.]. 2015. IEEE International Conference on Big Data (Big Data). IEEE, 2015.

2. Monga V., Evans. Perceptual image hashing via feature points: performance evaluation and tradeoffs. IEEE Transactions on Image Processing 15.11 (2006). P. 3452-3465.

3. Pablo, et al. "Social computing for image matching / P. Chamoso [et al.]. PloS one 13.5 (2018): e0197576.

4. Munerman V.I. The experience of massive data processing in the cloud using windows azure (as an example) // High availability systems. 2014. Vol. 10, №. 2. P. 8-13.

A.V. Kirikova, V.I. Munerman, T.A. Samoilova

Smolensk State University

IMPLEMENTATION OF IMAGE SEARCH IN DATABASES

Keywords: *parallel programming, big data, image search, perceptual hashing, databases.*

Abstract. *The article discusses a method to increase the efficiency of solving the problem of finding images in a database. This method is based on the use of perceptual hashing of an image, three levels of data parallelization and image search procedures. To implement parallel data processing, the principle of symmetric horizontal data distribution and the capabilities of*

modern processors (SIMD registers and corresponding instructions) are used. The results of a computational experiment, confirming the effectiveness of the proposed method, are presented.

Е.В. Кириллов, К.В. Мельник, В.И. Мунерман
Смоленский государственный университет

УДК 004.89:004.75

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ ВЫВОДА АССОЦИАТИВНЫХ ПРАВИЛ НА ОСНОВЕ NUMA- АРХИТЕКТУРЫ

Ключевые слова: *искусственный интеллект, ассоциативные правила, базы данных, многомерные матрицы, SQL, NUMA.*

В статье рассматривается задача параллельной реализации алгоритма вывода ассоциативных правил на базе NUMA-архитектуры. Предложенные алгоритмы реализуют решение задачи в технологиях in-memory и in-database. Приводится описание архитектуры программно-аппаратного комплекса на базе NUMA-архитектуры, а также результаты экспериментального анализа быстродействия предложенных алгоритмов на разных объемах данных в зависимости от выбранной модели данных.

Одной из важных проблем анализа данных является задача поиска ассоциативных правил. Пусть имеется множество объектов (транзакций) $I = \{i_1, \dots, i_k\}$ и множество свойств (товаров) $P = \{p_1, \dots, p_n\}$. Ассоциативным правилом называется выражение вида $X \rightarrow Y$, где $Y \subset I$, $X \cap Y = \emptyset$.

Ассоциативные правила – это закономерности между событиями. Пример такой закономерности – правило, указывающее, что из события А следует событие В с некоторой вероятностью. Правило $X \rightarrow Y$ имеет поддержку s (support), если $s\%$ транзакций содержат $X \cup Y$. Достоверность c (confidence) правила показывает вероятность, что из X следует Y . Правило $X \rightarrow Y$ имеет достоверность c , если $c\%$ транзакций, содержащих X , содержат также Y [1].

Для решения задачи параллельного вывода ассоциативных правил нами были выбраны две алгебраические модели: реляционная (в технологии in-database) и многомерно-матричная (в технологии in-memory) [2], а в качестве архитектуры программно-аппаратного комплекса – NUMA-архитектура.

NUMA (non-uniform memory access) – архитектура многопроцессорных компьютеров, в которой память является логически

общей, но разделенной физически между вычислительными узлами. Каждый узел содержит несколько равноправных процессоров, имеющих доступ к локальной памяти [3].

Для решения поставленной задачи была выбрана следующая структура таблицы транзакций T_0 : поле tid , содержащее номер транзакции n , и поле $item$, содержащее товар из n -й транзакции.

Реализация алгоритма вывода ассоциативных правил на основе реляционной модели данных заключается в построении на каждой итерации промежуточной таблицы с помощью соответствующего SQL-запроса.

База данных содержит следующий набор таблиц: $T_0(I, P_1)$ – исходные данные, каждая строка содержит идентификатор объекта и одно из его свойств; $CopyT_0(I, P_1)$ – копия таблицы T_0 , используемая для ускорения процесса обработки данных; $T_1(I, P_1, P_2), \dots, T_{k-1}(I, P_1, \dots, P_k)$ – таблицы, получаемые на итерациях и содержащие данные для вывода ассоциативных правил на очередной итерации, используемые на следующей итерации как исходные данные.

Запрос, исполняемый на l -й итерации, имеет следующий общий вид:

```

 $Q_l = \text{INSERT INTO } T_l(I, P_1, \dots, P_{l+1})$ 
 $\text{SELECT } T_{l-1}.I, T_{l-1}.P_1, \dots, T_{l-1}.P_l, T_0.P_1 \text{ AS } P_{l+1}$ 
 $\text{FROM } T_{l-1} \text{ INNER JOIN } T_0 \text{ [на итерации } 1 - CopyT_0]$ 
 $\text{ON } T_{l-1}.I = T_0.I \text{ AND } \pi(T_{l-1}.I, T_{l-1}.P_1, \dots, T_{l-1}.P_l, T_0.P_1),$ 
где  $\pi(T_{l-1}.I, T_{l-1}.P_1, \dots, T_{l-1}.P_l, T_0.P_1)$  – предикат, запрещающий дублирование комбинаций свойств.

```

Итак, если для построения ассоциативных правил необходимо знать количества всех объектов, содержащих комбинации свойств от 1 до $k \leq s$, то выполняются запросы Q_1, \dots, Q_{k-1} . Ассоциативные правила выводятся на втором этапе из таблиц T_0, \dots, T_{k-1} .

Алгоритм, основанный на умножении многомерных матриц, также состоит из последовательных итераций. Исходными данными в многомерно-матричной модели является двумерная матрица, содержащая данные об объектах и соответствующих свойствах [4].

На первом шаге вычисляются трехмерная матрица $M_1 = {}^{1,0}(M_0 \times M_0) = (m_{ij}^1), i = 1, \dots, n, j = 1, \dots, s, m_{ij}^1 = m_{ij}^0 \times m_{ij}^0$ и двумерная матрица $T_1 = {}^{0,1}(M_0 \times M_0) = (t_{jj}^1), j = 1, \dots, s$, где $t_{jj}^1 = \sum_{i=1}^n m_{ij}^0 \times m_{ij}^0$.

На l -й итерации вычисляются $l+1$ -мерная матрица $M_l = {}^{1,0}(M_{l-1} \times M_0) = \left(m_{ij\dots j}^l \right), m_{ij\dots j}^l = m_{ij\dots j}^{l-1} \times m_{ij}^0$, которая будет использована на следующей итерации, и матрица

$$T_l = {}^{0,1} (M_{l-1} \times M_0) = \left(t_{ij\dots j}^l \right), t_{ij\dots j}^l = \sum_{i=1}^n m_{ij\dots j}^{l-1} \times m_{ij}^0. \quad \text{Матрицы} \quad T_1, \dots, T_l$$

используются для вывода ассоциативных правил.

Операции (1,0) и (0,1) называются (λ, μ) -свернутым произведением многомерных матриц [5].



Рис. 1. Архитектура программно-аппаратного комплекса

В предлагаемой нами архитектуре программно-аппаратного комплекса (рис. 1) используется n вычислителей для хранения и обработки фрагментов базы данных. Физические диски вычислителей объединены в виртуальное дисковое пространство. Центральный вычислитель при необходимости разделяет некоторую исходную базу данных на БД-фрагменты, распределяет их между вычислителями и передает команду (по протоколу TCP/IP) для начала вычислений ассоциативных правил. После окончания процесса каждый вычислитель передает результат своих вычислений на центральный вычислитель. По прибытии всех результатов центральный вычислитель производит их обработку и выдает искомые правила пользователю.

Данная архитектура может быть реализована как в локальной сети, так и в глобальных сетях Интернет и Интранет.

Чтобы рассчитать поддержку кандидатов и правил, а также достоверность правил, нужно узнать мощность n -предметных наборов, то есть сколько раз они встречаются в разных транзакциях. Для вычисления необходимых характеристик в случае распределенной базы данных нужно посчитать мощности n -предметных наборов в БД-фрагментах на вычислителях, переслать полученные промежуточные данные на центральный вычислитель, выполнить на нём операцию Join или операцию сложения многомерных матриц над всеми фрагментами данных, сложив при этом соответствующие мощности. Так мы получим

мощность n-предметных наборов, распределенных среди всех БД-фрагментов.

Также для расчета поддержки и достоверности правил необходимо узнать количество уникальных tid во всех БД-фрагментах. На каждом вычислителе нужно подсчитать количество уникальных tid в БД-фрагменте, передать эту информацию на центральный вычислитель, который просуммирует все полученные числа. Данная сумма будет использоваться при вычислении поддержки и достоверности.

Деление БД нужно осуществлять таким образом, чтобы БД-фрагменты содержали попарно непересекающиеся множества транзакций, так как вычислители не знают, какие tid хранятся на других вычислителях, а значит, подсчет уникальных tid будет невозможен (без дополнительного обмена данными, что нерационально). Желательно также, чтобы БД-фрагменты содержали примерно одинаковое количество строк.

Следует отметить, что в рамках предложенной архитектуры целесообразно записывать данные по мере их поступления сразу в БД-фрагменты на вычислителях, чтобы не затрачивать время на разделение базы данных и пересылку фрагментов.

Для оценки эффективности данной архитектуры был проведен вычислительный эксперимент. Программно-аппаратный комплекс на базе NUMA-архитектуры состоит из центрального вычислителя и четырех вычислителей, на каждом из которых установлен четырехъядерный процессор. На каждом вычислителе ассоциативные правила выводятся одновременно в четырех потоках. В таблице 1 представлено сравнение времени вывода ассоциативных правил при разных объемах данных на NUMA-архитектуре и на SMP-архитектуре (состоящей из одного компьютера с четырехъядерным процессором).

Таблица 1

Время вывода ассоциативных правил

Исходная таблица T0 (млн. строк)	Алгоритм на основе реляционной модели		Многомерно-матричный алгоритм	
	SMP	NUMA (4 вычислителя)	SMP	NUMA (4 вычислителя)
1	25 сек	7 сек	14 сек	5 сек
2	40 сек	14 сек	30 сек	7 сек
3	54 сек	19 сек	49 сек	11 сек

Время передачи данных между центральным вычислителем и вычислителями не включалось в общее время вычислений. Это обусловлено:

1) тем, что это время существенно зависит от пропускной способности сети;

2) тем, что передача больших объемов данных производится только в двух случаях: при первом распределении и при перераспределении после накопления большого числа изменений; в остальных случаях передаются незначительные объемы корректирующих данных.

Рассмотренная реализация параллельного вывода ассоциативных правил на основе NUMA-архитектуры позволила существенно ускорить процесс поиска закономерностей.

Литература

1. Зайко Т.А., Олейник А.А., Субботин С.А.. Ассоциативные правила в интеллектуальном анализе // Вестник НТУ «ХПИ». 2013. №39. С. 82–96.

2. Мунерман В.И. Модели обработки больших объемов данных в системах массового параллелизма // Системы высокой доступности. 2013. Т. 9, №1. С. 35–43.

3. Максименко В.Н., Филиппов А.А. Анализ и сравнение микроархитектур SMP и NUMA для создания вычислительных систем и платформы IMS // Т-Comm: Телекоммуникации и транспорт. 2010. №10. С. 55–57.

4. Захаров В.Н., Мунерман В.И., Самойлова Т.А. Параллельные методы вывода ассоциативных правил в технологиях in-database и in-memory // Конвергентные когнитивно-информационные технологии: сб. тр. науч. конф. М., 2017. С. 219–225.

5. Соколов Н.П. Введение в теорию многомерных матриц. Киев: Наукова думка, 1972. 176 с.

E.V. Kirillov, K.V. Melnik, V.I. Munerman
Smolensk State University

PARALLEL ASSOCIATION RULES MINING BASED ON NUMA-ARCHITECTURE

Keywords: *artificial intelligence, association rules, databases, multidimensional matrices, SQL, NUMA.*

Abstract. *The article deals with the problem of parallel implementation of the algorithm for deriving associative rules based on the NUMA architecture. The proposed algorithms implement the solution of the problem in in-memory and in-database technologies. A description of the architecture of the software and hardware complex based on the NUMA architecture is given. The results of experimental analysis of the performance of the proposed algorithms on different data volumes depending on the selected data model are presented.*

МОДЕЛЬ МУЛЬТИАГЕНТНОЙ СИСТЕМЫ ЛОГИСТИЧЕСКОГО ПРЕДПРИЯТИЯ

Ключевые слова: *мультиагентная система, логистика, оптимизация, планирование, искусственный интеллект.*

В статье рассматривается одно из возможных применений мультиагентных систем в логистической отрасли. Описывается решение задачи составления расписания с применением мультиагентных систем. В данном варианте решения модель системы состоит из двух типов агентов: агенты складов и агенты транспортных средств. Агенты складов рассчитывают текущее состояние доставки каждой посылки: анализируют время, проведенное посылкой в пути, определяя приоритетность ее отправки на фоне остальных посылок, и выбирают кратчайший путь ее доставки. Агенты транспортных средств определяют расчетное время прибытия в пункт выдачи и в последующие пункты на своем пути, а также денежные затраты на маршрут. Агенты склада анализируют эти данные и назначают маршруты транспортным средствам.

Задача формирования оптимальных расписаний является сложной и трудоемкой задачей для логистических компаний, так как при ее решении необходимо учесть множество факторов: маршруты движения, количество, грузоподъемность и скорость доступных транспортных средств, стоимости перевозок – таким образом, чтобы минимизировать собственные затраты. Несмотря на то, что подобные задачи достаточно глубоко рассматривались учеными, на практике расписание часто составляют вручную. Наилучший выход в данной ситуации – автоматизированное составление расписания с учетом текущих экономических потребностей. Предлагается решение данной задачи с применением технологии мультиагентных систем.

В основе мультиагентного подхода лежит понятие программного агента, который реализован и функционирует как самостоятельный элемент программного обеспечения или элемент искусственного интеллекта. Суть мультиагентных технологий заключается в принципиально новом методе решения задач. В отличие от классического способа, когда проводится поиск некоторого четко определенного алгоритма, позволяющего найти наилучшее решение

проблемы, в мультиагентных технологиях решение получается автоматически в результате взаимодействия множества самостоятельных целенаправленных программных модулей – агентов[2].

Программный агент – это некий автономный компонент, который действует в интересах владельца или пользователя и направлен на достижение поставленных целей [1]. Главной особенностью агента является то, что он способен работать в различных, но однородных окружениях, самостоятельно функционировать и общаться с другими агентами. Мультиагентная система является децентрализованной, что позволяет ей быть более стабильной и устойчивой к резким изменениям нагрузки.

Рассмотрим модель мультиагентной системы на примере логистической компании, занимающейся перевозом частных посылок.

Пусть имеется логистическое предприятие, занимающееся перевозкой товаров между складами и пунктами выдачи с помощью некоторого набора транспортных средств. Грузоподъемность каждого транспортного средства ограничена и может отличаться от грузоподъемности других транспортных средств.

Каждая посылка в этой системе является уникальной, что отражено в идентификаторе посылки (номере отслеживания). Зная информацию о номере отслеживания, можно получить основные необходимые характеристики посылки: вес, пункт отправки и конечный пункт выдачи посылки. Каждый склад и пункт выдачи имеет свою локальную базу данных, содержащую информацию о посылках в наличии в данный момент времени в этом пункте. База данных также содержит информацию о конкретном секторе склада, где размещается посылка, сотруднике, принявшем посылку, весе посылки, сроке доставки и т.д.

В глобальную базу передаются не все данные о посылке, а исключительно основные характеристики: вес, текущее местоположение, пункт назначения, максимальный срок доставки, что позволяет отправителям и получателям отслеживать текущее местоположение посылки.

В предложенной модели используются два типа агентов: агент склада (пункта выдачи) и агент транспортного средства. Агент каждого типа в ходе работы взаимодействует с агентами как этого же, так и другого типа. Для экономии времени и ресурсов на определенном этапе алгоритма агенты одного типа выбирают между собой наилучшее из найденных решений и только потом передают это решение агентам другого типа, избавляя их от потребности в дополнительных вычислениях. Краткая схема взаимодействия агентов представлена на рисунке 1.



Рис. 1. Схема взаимодействия агентов

На складах и пунктах выдачи устанавливается программный агент мультиагентной системы. Главная цель агента – передать все отправления на следующий склад на пути к пункту выдачи. Он работает с локальной базой данных склада, а именно: с идентификаторами посылок, историей пунктов доставки, конечным пунктом выдачи, максимальным сроком доставки и схемой расположения складов, рассчитывая кратчайший маршрут из текущей точки до места назначения любым из известных методов (например, с помощью алгоритма Дейкстры).

Далее агент присваивает каждой посылке приоритет, базируясь на данных об истории пройденных пунктов и сроках доставки. В распоряжении агента находится схема расположения пунктов выдачи. С ее помощью он определяет, какой процент пути посылка уже прошла, а также процент времени, который был затрачен на доставку посылки из пункта отправления в данный пункт. Чем больший процент времени был потрачен на доставку в данный пункт выдачи и чем меньший процент пути был пройден посылкой, тем более высокий приоритет ей присваивается. Приоритет пересчитывается для всех посылок с некоторой периодичностью для поддержания информации о нем в актуальном состоянии. Посылки, которые уже находятся в целевом пункте выдачи, получают минимальный (нулевой) приоритет. Также отметим, что сотрудник логистической компании может вручную менять приоритет заданного отправления, если условия его доставки являются специфическими (например, срок доставки меньше стандартного или компания понесет большие убытки в случае опоздания доставки) и это не вызовет сбоев в дальнейшей работе системы.

Таким образом, агент формирует список посылок с соответствующими им приоритетами отправки и пунктами маршрута, которые необходимо пройти.

Следующим этапом работы агента является его взаимодействие с агентами второго типа. Агент первого типа отправляет запрос агентам

второго типа, передавая информацию о наиболее приоритетных посылках: вес, текущее местоположение и пункт назначения.

Агент второго типа устанавливается на транспортные средства предприятия. Он работает со схемой пунктов доставки и грузоподъемностью данного транспортного средства. Когда к нему приходит заявка на отправку посылки от агента первого типа, он рассчитывает денежные затраты на ее доставку, время прибытия в пункт для принятия посылки и время окончания маршрута. Если транспортный агент уже имеет некий маршрут, не проходящий через текущий склад, то он передает время, соответствующее сумме времени на завершение текущего маршрута и времени на достижение пункта выдачи из конечной точки текущего маршрута. Параметры денежных затрат для данного средства регулируются пользователем. В них может включаться средняя стоимость топлива, обслуживания транспортного средства и т.д.

Подсчеты затрат для агента в таком случае состоят из двух частей: затраты на дорогу до пункта, откуда будет забрана посылка, и затраты на дорогу до конечного пункта назначения. Если доставка проходит через несколько пунктов либо может иметься несколько путей доставки, то рассчитываются затраты на передвижения между разными пунктами для всех путей. Дополнительно выполняется проверка на превышение грузоподъемности транспортного средства на каждом отрезке пути. Если проверка прошла успешно на протяжении всего пути или хотя бы отдельных его отрезков, то полученные данные передаются остальным агентам второго типа. Для каждого отрезка пути выбираются список агентов-лидеров с наименьшими затратами на его преодоление и список наиболее близко расположенных к текущей точке агентов. Списки отправляются агенту склада.

Агент склада перебирает самые дешевые варианты агентов в поисках транспортных средств для каждого отрезка пути, которые могут доставить посылку до наступления максимального срока доставки. Если таких вариантов не найдено, то выбираются наиболее близко расположенные агенты (агенты, для которых время прибытия в текущий пункт минимально) для каждого отрезка пути и им назначается маршрут.

Если транспортное средство с назначенным маршрутом заполнено не полностью на некотором отрезке пути, то агент второго типа направляет запрос агентам первого типа тех пунктов, через которые он будет проезжать, для поиска посылок с меньшим приоритетом и весом, не превышающим свободную грузоподъемность средства на данном отрезке пути.

Результатом работы данной системы является автоматически составленное расписание доставки посылок. Такое расписание – экономически выгодно для компании, что достигается за счет

рационального использования каждого транспортного средства, наименьших денежных затрат на доставку и соблюдения сроков доставки.

Преимущество данного подхода по сравнению со стандартными методами решения задачи составления расписания состоит в том, что система, построенная по такой модели, полностью автоматизирована, стабильна, гибка и масштабируема. Она способна адекватно реагировать на неожиданное изменение условий, добавление или потерю агента, не зависит от работы одного конкретного компонента системы.

Литература

1. Иванов К.К., Лужин В.М., Кожевников Д.В. Программные агенты и мультиагентные системы // Молодой ученый. 2017. № 7. С. 11-13
2. Методологии проектирования мультиагентных систем / М.Г. Зубарева [и др.] // Технические науки в России и за рубежом: материалы VI Междунар. науч. конф. (г. Москва, ноябрь 2016 г.). М.: Буки-Веди, 2016. С. 3-8.

N.Yu. Kovaleva
Smolensk State University

MODEL OF MULTI-AGENT PROGRAM SYSTEM OF LOGISTICS ENTERPRISE

Keywords: *multiagent system, logistics, optimization, planning, artificial intelligence.*

Abstract. *The article discusses one of the possible applications of multi-agent systems in the logistics industry. The solution of the problem of scheduling using multi-agent systems is described. In this solution, the system model consists of two types of agents: warehouse agents and vehicle agents. Warehouse agents calculate the current delivery status of each parcel: analyze the time spent by the parcel on the way, determining the priority of its shipment against the background of the remaining parcels, and determine the shortest route for its delivery. Vehicle agents determine the estimated time of arrival and subsequent points on their way, as well as the cash costs of the route. Warehouse agents analyze this data and assign routes to vehicles.*

ОСОБЕННОСТИ РАЗРАБОТКИ ИНТЕРНЕТ-МАГАЗИНА ПРОДАЖ СРЕДСТВАМИ MS VISUAL STUDIO

Ключевые слова: *информатика, информационные системы, базы данных, программирование, IT- технологии, интернет-магазин.*

В статье обсуждаются общие подходы к созданию интернет-магазинов продаж. В качестве примера раскрываются отдельные аспекты разработки интернет-магазина «BDCosmetics». Характеризуются роли пользователей среды информационной системы. Рассматриваются возможности администратора, поставщика и покупателя интернет-магазина продаж «BDCosmetics». Описываются технические особенности реализации программного кода информационного ресурса в визуальной среде объектно-ориентированного программирования MS Visual Studio. Особое внимание уделено вопросам хранения данных в системе и организации взаимодействия пользователей при одновременном доступе к системным данным. Актуальность статьи связана с возрастанием интереса к проблеме разработки и эксплуатации программной среды интернет-магазинов продаж.

Современные тренды развития рынка товаров и услуг диктуют необходимость внедрения в сферу экономической деятельности последних достижений IT-технологий [1; 2; 3]. Эффективное функционирование любого бизнеса уже немыслимо без использования возможностей Интернета. Все большее число компаний создает интернет-площадки для параллельного ведения экономической деятельности либо вовсе переносит свою работу в область мировой паутины.

Одним из направлений применения IT-разработок в сети Интернет является создание различного рода web-ресурсов [4; 5]. При этом особой популярностью в последнее время пользуется разработка интернет-магазинов [6]. Любая преуспевающая компания в XXI веке обязана иметь собственный web-ресурс. Компания в сфере товаров и услуг должна не только владеть информационным web-ресурсом, но и вести свой бизнес посредством интернет-магазина. В связи с этим становится актуальным вопрос об использовании современных возможностей IT-технологий при разработке такого рода ресурсов [7].

Для проектирования и разработки интернет-ресурсов на сегодняшний день сфера информационно-коммуникационных технологий предлагает достаточно богатый выбор программных средств [8]. Набор программных сред разработки варьируется от средств специального назначения для отдельных web-компонентов до инструментальных средств общего назначения. При этом наиболее востребованными оказываются мощные программные оболочки, которые позволяют подключать как общие, так и специализированные пользовательские библиотеки. К последним можно отнести Embarcadero RAD Studio и Microsoft Visual Studio. Так, MS Visual Studio включает множество языков программирования, которые могут программно взаимодействовать друг с другом на платформе NET.

Остановимся более подробно на отдельных особенностях разработки интернет-магазина продаж средствами MS Visual Studio. Рассмотрение будем вести на примере web-проекта «BDCosmetics».

Добавим в класс формы строковое поле *String* – строку подключения к базе данных. В строке подключения в значении параметра *DataSource* указывается путь к файлу с базой данных. Затем добавим строку создания объекта типа *OleDbDataAdapter*. Объект этого типа организует пересылку наборов данных с вызываемым процессом. Адаптеры данных содержат набор из четырех внутренних объектов команд. Это команды чтения, вставки, изменения и удаления информации.

После создания адаптера данных сформируем объект *DS* типа *DataSet*, который позволит оперировать набором данных. Набор данных представляет аналог промежуточного буфера для данных, которые могут отображаться на программных формах. Он служит удобным механизмом для чтения и обновления данных, а также инкапсулирует множество таблиц и связей между ними.

Для вывода содержимого базы данных на web-форму необходим объект *DataGridView*. Он служит для представления данных в табличной форме. Переменная *Command* будет содержать строку SQL-запроса для вывода всех записей таблицы *Products*. После этого данные из таблицы *Products* отобразятся на форме.

Собственно данные, которыми оперирует информационная система, можно как хранить непосредственно в ней, так и работать с ними внешним образом. Последний способ более удобен, так как позволяет подключать базы данных разных форматов, подходящих под структуру таблиц проекта интернет-магазина. В проекте интернет-магазина продаж «BDCosmetics» база данных создана в среде MS Access. Она находится в папке проекта и имеет следующий адрес – *MagazinBD/Workwiththe/BD.mdb*. Подключаемая база данных состоит из четырех таблиц. Таблица *Basket_Products* – это содержимое корзины

покупателя, таблица *Employers* – данные о сотрудниках администрации, *Products* – каталог магазина, *Users* – данные пользователей.

После запуска web-приложения перед пользователем появляется начальная страница интернет-магазина. На ней расположены две кнопки: «Зарегистрироваться» и «Уже зарегистрирован». Кнопка «Зарегистрироваться» предназначена для пользователя, данных которого ещё нет в системе. Кнопка «Уже зарегистрирован» – для пользователя, данные которого в системе есть. Она позволяет войти в систему. Данная кнопка переносит пользователя на форму авторизации. Пользователем может быть администратор, поставщик или обычный покупатель. Для каждого из них в программе предназначен отдельный интерфейс, предоставляющий пользователю web-ресурса определенный набор функциональных возможностей в соответствии с его статусом.

Форма авторизации имеет для заполнения два поля – «Логин» и «Пароль». Данные берутся из базы данных Access – *BD.mdb*. Если введенных параметров нет в базе данных, программа не допустит пользователя в магазин. После успешной аутентификации появится окно следующего вида (рис. 1).



Рис. 1. Окно интернет-магазина продаж после входа в систему

Рассмотрим вид интерфейса и возможности покупателя. Для начала покупок следует нажать кнопку «Открыть базу данных». После нажатия этой кнопки в компоненте *DataGridView* на форме проекта отобразится содержимое интернет-магазина. Покупатель имеет возможность выполнить сортировку товаров. Для этого следует выбрать кнопку «Перейти к сортировке», после чего откроется новое окно – окно сортировки. Для выполнения сортировки данных таблицы нажимаем на кнопку «Перейти к сортировке». Сортировка базы данных возможна как по одному полю, так и по двум полям. Поля, по которым она будет выполняться, пользователь выбирает самостоятельно. Сортировку можно выполнить как по возрастанию, так и по убыванию. Для каждого поля можно выставить своё значение.

Покупатель также имеет возможность найти нужный ему товар. Для этого следует нажать на кнопку «Перейти к поиску». После этого откроется окно поиска. Поиск можно выполнить по следующим параметрам: «Поиск по одному полю», «Контекстный поиск» и «Поиск по диапазону».

В случае, когда искомый товар в базе данных найден, пользователь может добавить его к списку выбранных им элементов. Для этого в правом верхнем углу начальной страницы есть кнопка «Перейти в корзину». Покупатель может добавить товар в корзину двумя способами. Первый способ – вписать код нужного товара в ячейку и нажать кнопку «Добавить в корзину». Второй способ – щелкнуть по строке таблицы с нужным товаром. Программа самостоятельно запишет код товара в ячейку. После этого покупатель также должен нажать кнопку «Добавить в корзину». Корзина содержит товары, которые туда добавил покупатель. Для каждого покупателя формируется собственная корзина товаров, информация о которой при их выборе отражается в общей сводной таблице базы данных интернет-магазина. Для заказа товаров нужно нажать кнопку «Заказать». Программа выдаст сообщение об успешном добавлении товаров.

Рассмотрим интерфейс и возможности администратора. Администратор обладает наиболее широкими правами при работе с данными интернет-магазина продаж. При входе в аккаунт администратора откроется окно добавления строки. Для начала работы с программой следует открыть базу данных. Для этого переходим в раздел «Открытие» и нажимаем кнопку «Открыть базу данных».

Администратор имеет возможность добавления, удаления и редактирования записей. Для удаления товара из базы данных следует ввести его id и нажать кнопку «Удалить строку». Для редактирования – ввести новые данные и нажать кнопку «Редактировать строку».

Интерфейс для поставщика схож с интерфейсом администратора. Поставщик также может добавлять и редактировать строки товаров. Отличие его функций от возможностей администратора заключается в том, что он не может удалять записи о товарах.

Остановимся далее на особенностях программной реализации отдельных элементов интерфейса и основных программных функций. В качестве примера рассмотрим реализацию сортировки по одному полю.

Форма сортировки представляет собой окно, состоящее из следующих элементов (рис. 2):

- кнопки «Сортировать» типа *Button*, которой в программе присвоено имя *mBtnSortOne*;
- таблицы типа *DataGridView* для вывода на экран каталога товаров с программным именем *dgvSortOne*;

- выпадающего списка типа *ComboBox*, позволяющего выбрать поле, по которому будет проводиться сортировка, с именем `mcbSortOne`;
- двух переключателей «По возрастанию» и «По убыванию» типа *RadioButton* с именами `mrbsortOneEsc` и `mrbsortOneDesc` соответственно.



Рис. 2. Сортировка товаров интернет-магазина продаж по одному полю

С помощью данных элементов, расположенных на форме, пользователь выбирает, по каким параметрам следует проводить сортировку. Отметим, что выпадающий список имеет семь вариантов строк сортировки, по которым можно проводить упорядочивание табличных данных. Сортировка данных в таблице товаров интернет-магазина осуществляется с помощью следующего программного кода:

```
private void mBtnSortOne_Click(object sender, EventArgs e)
{
    ADBSC.SortOne(mcbSortOne, mrbsortOneEsc, mrbsortOneDesc,
        dgvSortOne);
}
```

Собственно реализация сортировки данных содержится в классе `AppDBSortC` в методе `SortOne`. В этом методе описаны различные варианты сортировки, которые зависят от того, какая строка выбрана из выпадающего списка. Различным вариантам сортировки в программе присвоены значения от 0 до 6 элементов *ComboBox*.

При выборе сортировки по ID программа обращается к следующей части программного кода:

```
public void SortOne (MetroComboBox _C, MetroRadioButton
    _RE, MetroRadioButton _RD, DataGridView _dgv)
{
    if (_C.SelectedIndex == 0)
        {if (_RE.Checked == true)
            {...}
            Connection.ConnectionString =
                "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=BD.mdb";
            OleDbCommandcomSort = new OleDbCommand("SELECT * FROM
                Products ORDER BY ID", Connection);
            ...}
        if (_RD.Checked == true)
            {...}
        }
```

```

        Connection.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0; Data Source=BD.mdb";
        OleDbCommandcomSort = new OleDbCommand("SELECT * FROM
Products ORDER BY ID DESC", Connection);
        ...}
    }
    ...}

```

Подключение к базе данных в этой сортировке осуществляется с помощью свойства *ConnectionString*, в котором указаны драйвер для подключения и расположение файла с табличной информацией о продажах. Сортировка осуществляется посредством SQL-запроса с помощью инструкции *SELECT* и фразы *ORDER BY*. При этом в первом случае сортировка проводится по возрастанию, а во втором – по убыванию, на что указывает параметр *DESC*.

Упорядочивание данных по возрастанию или убывания осуществляется на основе выбора пользователем одного из двух соответствующих положений селектора переключателей на форме. Остальные варианты упорядочивания товаров интернет-магазина продаж реализованы аналогичным образом. Для этого также используется инструкция *SELECT* с указанием соответствующего поля таблицы базы данных для сортировки во фразе *ORDER BY*.

Для поиска товаров с заданными характеристиками в интернет-магазине продаж «BDCosmetics» разработана собственная событийная процедура. Она реализована в программном коде с помощью инструкции SQL-запроса *SELECT* с указанием поля поиска. Пользователь может осуществлять поиск товаров по всем полям табличных данных интернет-магазина «BDCosmetics». Добавление, удаление и редактирование записей о товарах интернет-магазина продаж также выполняется на основе структурированного языка запросов SQL.

Приведенный программный код наглядно демонстрирует возможности использования стандартных паттернов проектирования баз данных при разработке интернет-магазина продаж [9]. Такой подход, реализованный средствами MS Visual Studio, унифицирует процессы программной реализации. Язык программирования C# раскрывает возможности генерирования структуры и основных программных модулей web-приложения по форме. Язык структурированных запросов SQL предлагает общие методы поиска и сортировки данных. В свою очередь, вместе они позволяют выработать общие подходы к программной разработке подобного рода web-продуктов.

Литература

1. Баженов Р.И., Лопатин Д.К. О применении современных технологий в разработке интеллектуальных систем // Журнал научных публикаций аспирантов и докторантов. 2014. № 3 (93). С. 263-264.

2. Козлов С.В. Использование математического аппарата импликативных матриц при создании и сопровождении информационных систем // International Journal of Open Information Technologies. 2017. Т. 5, № 12. С. 16-23.
3. Козлов С.В., Суин И.А. О некоторых подходах математического описания и анализа многомерной структуры информационных систем // Системы компьютерной математики и их приложения. 2018. № 19. С. 177-182.
4. Мунерман В.И., Самойлова Т.А. Обучение методам разработки информационно-аналитических систем на основе облачных технологий (на примере MICROSOFT AZURE) // Системы высокой доступности. 2016. Т. 12, № 4. С. 3-11.
5. Максимова Н. А. Описание работы web-сервиса // Постулат. 2017. № 8 (22). С. 17.
6. Винокуров А.С., Баженов Р.И. Разработка мобильного приложения для музыкального магазина в среде Android Studio // Постулат. 2016. № 9 (11). С. 8.
7. Козлов С.В. Использование соответствия Галуа как инварианта отбора контента при проектировании информационных систем // Современные информационные технологии и ИТ-образование. 2015. Т. 2, № 11. С. 220-225.
8. Суин И.А., Козлов С.В. Особенности разработки интерактивных приложений // Постулат. 2017. № 11. С. 29.
9. Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы / пер. с англ. А.А. Минько. М.: Вильямс, 2010. 400 с.

S.V. Kozlov, A.V. Demkovich
Smolensk State University

FEATURES OF DEVELOPMENT OF ONLINE STORE OF SALES BY MEANS OF MS VISUAL STUDIO

Keywords: *informatics, information systems, databases, programming, IT technologies, online store.*

Abstract. *In article the general approaches of creation of online stores of sales are discussed. As an example separate aspects of development of «BDCosmetics» online store reveal. Roles of users of the environment of an information system are characterized. The possibilities of the administrator, the supplier and the buyer of online store of sales «BDCosmetics» are considered. Authors describe technical features of implementation of a program code of an information resource in the visual environment of object-oriented programming MS the Visual Studio. Special attention is paid to questions of data storage in a system and the organization of interaction of users at simultaneous access to system data. The relevance of article is connected with increase of interest in a problem of development and operation of the program environment of online stores of sales.*

ОБЩИЕ ПРИНЦИПЫ РАЗРАБОТКИ ИНТЕРНЕТ-МАГАЗИНА ТОВАРОВ С ИСПОЛЬЗОВАНИЕМ JAVA-ТЕХНОЛОГИЙ

Ключевые слова: *информатика, информационные системы, базы данных, программирование, IT-технологии, интернет-магазин.*

В статье обсуждаются актуальные вопросы создания интернет-магазинов товаров с использованием новейших IT-технологий. В качестве примера раскрываются особенности разработки интернет-магазина «Chronicles». Особое внимание уделено использованию возможностей Java-технологий при управлении ресурсами web-сервисов магазина. Охарактеризованы отдельные аспекты взаимодействия с web-сервисами, web-программирования и работы с базами данных. Рассмотрена специфика обработки HTTP-запросов в программной среде web-ресурса. Актуальность статьи связана с поиском универсальных методов проектирования и разработки интернет-магазинов.

Современная web-индустрия охватывает все большие сферы человеческой деятельности [1]. С каждым днем создается все больше новых интернет-ресурсов, при этом растет не только их число, но и многообразие. Среди них достаточно широкую нишу занимают интернет-магазины товаров и услуг. Проектирование и непосредственная программная реализация такого рода web-проектов требует от разработчика использования новейших достижений в области IT-технологий [2; 3; 4]. При этом общие принципы, методы и подходы при разработке web-ресурсов конкретного вида, в частности интернет-магазинов, выработаны еще в недостаточной мере.

В последнее время при создании разного рода web-ресурсов, в том числе интернет-магазинов товаров, особую популярность приобрели Java-технологии. Это обусловлено их относительной универсальностью и аппаратной независимостью программной реализации от предметной области. Они позволяют эффективно организовать взаимодействие с различного рода web-ресурсами [5], организовать хранение и использование в запросах информации web-сервисов к базе данных интернет-приложения [6], а также предлагают достаточно широкий список современных пользовательских библиотек для web-программирования.

Проект «Chronicles», реализованный с применением разнообразных методов и приемов Java-технологий, содержит в себе множество под-проектов, которые затрагивают важные части создания программного софта, такие как web-сервисы, web-программирование, работа с базами данных. Благодаря этому можно создать шаблон для разработки интернет-магазинов товаров. Spring Framework (или, коротко, Spring) – это универсальный фреймворк с открытым исходным кодом для Java-платформы.

В шапке сайта пользователь видит несколько ссылок: каталог, акции, подборки, новинки, новости магазина, личный кабинет. На странице акций и новостей он находит как актуальную, так и архивную информацию, которая хранится в базе данных. Нажав на название акции или новости, можно перейти на её индивидуальную страницу, где представлена более полная информация. Для реализации проекта понадобятся такие классы как контроллеры и модели. Каталог товаров, списки акции и новостей, данные пользователей хранятся в базе данных. Важную роль играет файл `pom.xml`. В нем прописываются зависимости, которые используются в данном проекте. То есть библиотеки самостоятельно подтягиваются в проект. Иначе говоря, не надо файлы скачивать и устанавливать – их необходимо просто прописать в проекте, и они сами скачаются. После этого в нужных классах мы просто их импортируем. Зависимости прописываются в тэге `<dependencies>...</dependencies>`. В файле `pom.xml` также указывается, какого типа файл будет создаваться как результат сборки. Например, `<packaging>war</packaging>`.

`DispatcherServlet` – это сервлет, который определяет, в какой контроллер отправлять поступивший запрос. Он обеспечивает механизм для обработки запросов. `DispatcherServlet` действует как контроллер внешнего вида (`front controller`) для web-приложений, основанных на Spring Framework. `DispatcherServlet` наследуется от `javax.servlet.http.HttpServlet` обычно конфигурируется в файле `web.xml`. Файл находится в директории `WEB-INF`. Файл `web.xml` это дескриптор развертывания, который определяет, какие URL будут передаваться определенному сервлету. Он хранит информацию о конфигурации приложения. `DispatcherServlet` объявляется в тэге `<servlet>...</servlet>`. В нем указывается имя (тэг `<servlet-name>`), класс `<servlet-class>` (`<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>`) и порядок загрузки сервлета (тэг `<load-on-startup>`). Далее прописывается тэг `<servlet-mapping>`. В нем указывается, какой сервлет будет обрабатывать какие web-адреса. Например,

```
<servlet-mapping>
  <servlet-name>dispatcher</servlet-name>
  <url-pattern>*.htm</url-pattern>
</servlet-mapping>
```

В данном случае все поступающие HTTP-запросы, оканчивающиеся на «htm», будут обрабатываться диспетчером «dispatcher». Наиболее распространенные типы HTTP запросов – это GET и POST. GET-запрос используется, чтобы запросить содержимое конкретного ресурса, POST-запрос – для передачи данных на сервер. В классе HttpServlet есть методы doGet и doPost, которые определены для реакции на запросы пользователей GET и POST. В качестве параметров методы doGet и doPost принимают объекты HttpServletRequest и HttpServletResponse. Например, HttpServletRequest arg0 – это запрос, который приходит к сервлету, а HttpServletResponse arg1 – это ответ сервлета. В большинстве случаев приложения имеют один DispatcherServlet. DispatcherServlet использует классы конфигурации Spring для обнаружения компонентов делегата, в которых он нуждается для отображения запроса, разрешения представления, обработки исключений и т.д. Также в файле web.xml можно указать начальную страницу. Она указывается в тэге <welcome-file-list>.

Spring MVC предоставляет подход, основанный на аннотациях. То есть нет необходимости расширять базовый класс для обработки исключений, входных параметров запроса и прочего. Аннотация @Controller определяет класс как контроллер Spring MVC. То есть класс обрабатывает запрос пользователя, после чего создает требуемую модель, а потом передает ее для отображения. Аннотация @GetMapping указывает, на какие запросы будет реагировать контроллер. Например, @GetMapping("/Catalog") будет реагировать на вызов страницы каталога товаров. То есть указывается URL, с которым связывается данный метод. Объект ModelAndView (модель – сами данные; view (представление) – как эти данные отображать) используется для передачи значений из Spring MVC-контроллера в HTML.

Код контроллера, связывающегося с таблицей, содержит в себе объявление драйверов, установку соединения с СУБД, SQL-запрос, который забирает указанные данные из таблицы и передает их в список.

После этого соединение закрывается. Отправляется объект ModelAndView. DispatcherServlet на основании полученного ModelAndView ищет, какое представление ему использовать (View Resolver), и получает в ответе имя представления View. В представление передаются данные (model), и обратно посылается ответ от представления.

После чего пользователь видит страницу, где представлен список товаров, которые продает магазин. Наиболее удобное представление товара – сетка, в каждой ячейке которой представлена одна единица товара. В ней содержится основная информация. Самое удобное – это указание названия и цены. Если пользователь заинтересован данным товаром, он нажимает на название. После этого на сервер отправляется

GET-запрос, в котором как параметр передается «id» товара. Когда запрос приходит на сервер, тот отправляет его в контроллер. Контроллер принимает параметр «x», равный «id», который является номером товара. Так как «id» является первичным ключом, всегда будет выбран запрашиваемый товар. После чего пользователю открывается индивидуальная страница товара. На данной странице информация по товару выводится полностью. Соответственно, запрос к базе данных составляется иначе. Выбираются все столбцы конкретной строчки с помощью присваивания параметру «id» значения «x».

Для поиска конкретного товара используется SQL-запрос. Пользователь вводит в строку поиска название товара. Запрос передается в виде параметра, после чего происходит поиск в базе. При установлении соответствия выбирается нужная строка, значения столбцов передаются как список, закрывается подключение, и пользователю показывается страница с нужным товаром.

Чтобы пользователям было удобно запоминать понравившийся товар, можно сделать регистрацию. В контроллере создается метод, в котором в базу данных передаются сведения, вводимые пользователем. Для этого метода используется аннотация `@PostMapping`. Метод будет обрабатывать POST-запрос, соответствующий данному URL. Пользователь вводит данные в форме, расположенной на HTML странице. Информация передается в базу данных посредством SQL-запроса, например, "INSERT INTO public.\"user\" (name, email, age) VALUES (?, ?, ?)". Фраза «VALUES (?, ?, ?)» содержит значения, которые передаются в таблицу. Данные должны быть указаны для каждого столбца в таблице. В таблице «user» создается новая строка, в которую заносятся данные нового пользователя.

Чтобы запустить приложение, необходимо развернуть на компьютере локальный сервер. В настройках указываются URL и порт. Они задаются автоматически, но их можно изменить. Также нужно собрать проект. Для этого создается исполняемый WAR-файл программы, содержащий модули со всеми зависимостями. Он будет «пересобираться» перед каждым запуском.

Таким образом, проект интернет-магазина товаров демонстрирует применение общих методов и частных приемов при разработке web-ресурсов подобного рода. Использование при его создании и последующей эксплуатации Java-технологий позволяет эффективно обрабатывать большие объемы информации при взаимодействии с интернет-приложением. Это обуславливает востребованность выявления общих принципов проектирования и их последующего использования при разработке интернет-магазинов как одного из основных видов web-ресурсов.

Литература

1. Ересь А.В., Баженов Р.И. Разработка web-ориентированной экспертной системы профориентационного тестирования // Постулат. 2018. № 6 (32). С. 80.
2. Козлов С.В. Использование соответствия Галуа как инварианта отбора контента при проектировании информационных систем // Современные информационные технологии и ИТ-образование. 2015. Т. 2, № 11. С. 220-225.
3. Козлов С.В. Интерпретация инвариантов теории графов в контексте применения соответствия Галуа при создании и сопровождении информационных систем // International Journal of Open Information Technologies. 2016. Т. 4., № 7. С. 38-44.
4. Козлов С.В. Использование математического аппарата импликативных матриц при создании и сопровождении информационных систем // International Journal of Open Information Technologies. 2017. Т. 5, № 12. С. 16-23.
5. Радионов С.В., Шайдулов А.А., Баженов Р.И. Управление проектом с помощью web-средств PLAN.IO // Постулат. 2018. № 8 (34). С. 6.
6. Максимова Н.А. Описание работы web-сервиса // Постулат. 2017. № 8 (22). С. 17.

S.V. Kozlov, P.I. Kondratchikova
Smolensk State University

THE GENERAL PRINCIPLES OF DEVELOPMENT OF ONLINE SHOP OF GOODS WITH USE OF JAVA TECHNOLOGIES

Keywords: *informatics, information systems, databases, programming, IT technologies, online shop.*

Abstract. *In the article discusses current issues of creating online shops of goods using the latest IT technologies. As an example features of development of Chronicles online shop are revealed. Special attention is paid to the use of Java-technology capabilities in managing the resources of the web services of the online shop. Authors characterized separate aspects of interaction with web services, web programming and work with databases. The specificity of processing HTTP requests in the web-resource software environment is considered. The relevance of the article is related to the search for universal methods of designing and developing online shops.*

С.В. Козлов, И.О. Меженцев
Смоленский государственный университет

УДК 004

ВЗАИМОДЕЙСТВИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ C# С МЕЖПЛАТФОРМЕННОЙ СРЕДОЙ РАЗРАБОТКИ UNITY

Ключевые слова: *информатика, информационные системы, программирование, C#, среда разработки Unity, технология разработки видеоигр.*

В статье обсуждаются проблемы создания видеоигр, которые, в свою очередь, подталкивают к созданию новых решений для развития компьютерной науки. В качестве примера рассматривается разработка видеоигры с помощью межплатформенной среды Unity. Для взаимодействия со средой разработки используется язык программирования C#, который имеет свои особенности в данной среде. На основе проведенного исследования, делается вывод о сложности и особенностях создания видеоигры и о существовании решения для упрощения разработки с помощью адаптированных друг для друга сред.

В настоящее время информационно-коммуникационные технологии проникают во множество сфер человеческой деятельности. Это обусловлено двусторонними процессами. С одной стороны, развитие ряда сфер уже невозможно без новейших достижений науки и техники в области IT-технологий. С другой – собственно развитие сферы IT-индустрии обуславливает использование инструментальных средств разработки, инвариантных относительно предметной области при решении все большего круга практико-ориентированных задач [1; 2; 3].

Так, сегодня существует серьезная проблема современной науки, возникшая из-за видеоигр. Видеоигры представляют собой необычный продукт, с помощью которого развиваются современные личности. Не менее важным является то, что видеоигры подталкивают современную науку в области технологий к новым открытиям и разработкам, так как именно они выступают двигателем данной отрасли, именно для создания игр требуются большие ресурсы. Благодаря видеоиграм становится возможным вообразить жизненные ситуации или ситуации, которые в принципе не могут существовать.

Однако при создании видеоигр возникает множество проблем [4; 5]. Так очень часто создание игр, особенно крупных проектов, ассоциируют с тем, что вы едете на поезде, перед которым нет железной дороги. Чтобы достичь нужного пункта, то есть завершить проект, вам нужно бежать перед поездом и строить железную дорогу. Ассоциация возникает в силу того, что во время разработки игр меняются технологии, меняется конечное видение игры, приходится перерабатывать проект, вносить порой даже очень серьезные изменения, затрагивающие большую часть проекта. К тому же каждая игра имеет особый характер, не существует двух похожих игр, в то время как в целом их бесчисленное множество. Так как все игры различны, требуется различная реализация этих игр. На помощь приходят среды разработки, которые имеют в себе базовые технологии для создания игр, и вам лишь требуется правильно воспользоваться ими. Конечно, такие

системы не идеальны и в рамках больших проектов для конкретного жанра игр создают собственную среду разработки, но готовые среды можно модифицировать, и, дополнив их своими возможностями, вы можете сделать то, что вам нужно.

Разработка современных компьютерных игр ведется с использованием разнообразных методов и средств ИТ-технологий. В частности, это выражается в использовании определенных библиотек визуальных сред программирования и их языков. Так, например, одним из наиболее часто применяемых объектно-ориентированных языков программирования является язык *C#* программной оболочки *MS Visual Studio* [6]. Рассмотрим особенности организации взаимодействия с межплатформенной средой разработки компьютерных игр *Unity* и ее интерфейсом при помощи языка программирования *C#*.

Для изучения среды *Unity* придумаем игру, которую мы бы хотели реализовать. Например, пусть это будет герой, цель которого – забраться как можно выше по отвесной горе. Он будет хвататься за камни, а вы, в свою очередь, должны вовремя коснуться экрана, для того чтобы персонаж сменил направление своего движения. Цель игры – набрать как можно больше очков, то есть забраться как можно выше по горе.

Разработчиком данной игры является один человек, и возможности для написания полноценной среды ее разработки нет, так как подобные проекты создаются большими командами и зачастую не один год. На данный момент существует несколько бесплатных программных сред разработки компьютерных игр, среди них *GameMaker*, *Construct 2*, *Unity*, *Unreal Engine 4*. Мы воспользуемся средой разработки *Unity* версии 2018.2.18f1, так как у нее есть ряд преимуществ перед другими подобными системами. Например, использование языка программирования *C#*, кроссплатформенность, хорошая документация, форум, *Asset Store* с уже готовыми ресурсами.

При создании игры мы опустим такие моменты, как звуковое сопровождение, анимация, настройка графики и др. Порой эти этапы занимают даже большую часть времени, особенно этап программной реализации графики [7; 8], чем разработка игровой механики, которой мы и займемся при создании игры.

Итак, в проекте у нас одна сцена, на которой и будет реализована игра. Для начала мы реализуем генерацию камней на нашей сцене, для этого создадим скрипт, который назовем *GenerateMap.cs*, с одноименным классом. Классы автоматически создаются с тем же именем, поэтому дальше мы не будем это оговаривать. Затем «повесим» данный скрипт на пустой объект на сцене, для того чтобы взаимодействовать с ним. Такого прикрепления уже достаточно, чтобы скрипт запустился при создании сцены. Все классы автоматически наследуются от базового класса *MonoBehaviour*, который расположен в

пространстве имен *UnityEngine*; это пространство имен содержит основные методы для взаимодействия кода и среды *Unity*.

В начале класса мы объявим несколько публичных полей, среди которых будут поля с типами *GameObject*, *Transform*. Первый тип является базовым запечатанным классом. С помощью данного типа осуществляется большинство связей объектов на сцене со скриптом. Иначе говоря, *GameObject* хранит в себе ссылку на объект, который находится на сцене. Тип *Transform* является производным от типа *Component*, который, в свою очередь, входит в свойства *GameObject*. Поля следует объявить с модификатором доступа «public». Это необходимо для того, чтобы взаимодействовать с полями напрямую из инспектора, например, использовать окно для отображения свойств объектов сцены, ресурсов и т.д. Теперь мы можем просто перетащить префабы – заготовленные слепки объекта, которые могут хранить в себе множество компонент и характеристик, – в поля скрипта. Бывают ситуации, когда нам необходимо, чтобы поле было помечено модификатором доступа «private», – в таких случаях доступ из инспектора будет закрыт. Но есть возможность сделать его открытым для редактирования из инспектора: для этого поле помечается атрибутом [*SerializeField*]. Вообще данный атрибут в первую очередь используется при сериализации объекта, но так как у него есть данное специфическое свойство, то можно использовать в наших целях и его.

Также в скрипте мы объявим коллекцию *List* типа *Vector3*. Тип *Vector3* – это внутренний тип *Unity* [9]. Он находится в пространстве имен *UnityEngine* и является структурой. В себе он хранит, как несложно догадаться, три значения типа *float*. Вообще тип *float* является самым используемым в коде, так как через него удобно взаимодействовать с координатами; этот тип также использует внутриигровое пространство *Unity* в качестве *unit'ов* как размерность пространства.

Далее в методе *Start()* мы создадим первые камни на нашей сцене. Метод *Start()* является базовым и вызывается единожды при инициализации сцены. В данном методе мы используем такую функцию, как *Instantiate*, в которую передаем три параметра: объект-префаб, позицию, поворот. Этот метод отвечает за создание объекта на сцене из заготовленного нами префаба в нужных нам координатах, там, где находится правая рука персонажа. Метод *Instantiate()* мы вызываем непосредственно в еще одном методе, имя которого – *Destroy()*. В нашем случае метод принимает два параметра. Первый – это объект, который следует удалить со сцены, а второй – время, через которое следует удалить этот самый объект. Удаление объекта мы делаем для того, чтобы не занимать лишнюю память, ведь если мы уже прошли данные камни, то они нам не понадобятся. В указанном методе мы с помощью

цикла вызываем метод *CreateNewPos()*, в котором создадим последующие наши камни.

В методе *CreateNewPos()* мы используем статический метод *Range()* из запечатанного класса *Random*. Данный метод позволяет нам генерировать случайное число из заданного диапазона. В Unity в классе *Random* создана, а не заимствована из языка *C#*, своя реализация для генерации случайных чисел для более удобного использования. Также в этом методе будут генерироваться монеты на камнях, которые будут подбираться в случае успешного нажатия.

Далее мы создадим сразу три скрипта: *PlayerController.cs*, *LeftHandDetector.cs* и *RightHandDetector.cs*. Последние два будут иметь схожий код, различия лишь в том, что один будет навешен на левую кисть нашего персонажа, второй на его правую кисть.

Начнем реализацию скрипта *LeftHandDetector.cs*. Объявим публичное поле с типом *Text* и именем *coinText*, также подключим директиву *UnityEngine.UI*, отвечающую за связь с игровым интерфейсом, при помощи которого мы можем создать поле с типом *Text*. Это поле будет отвечать за вывод собранных нами монет. Наш скрипт содержит два базовых метода для взаимодействия с коллайдерами на сцене, имена которых *OnTriggerEnter2D* и *OnTriggerExit2D*. Первый выполняется, если коллайдер объекта, на котором висит скрипт, пересекает другой коллайдер. А второй, в свою очередь, выполняется, когда прекращается пересечение, иначе говоря, выходит из зоны коллайдера. В данных методах мы будем проверять, коснулись ли мы коллайдера помеченного тегом «*Rock*». На сцене объектам можно присваивать собственные имена с помощью свойства *Tags*. Если коснулись, то мы будем считать, что рука находится в зоне камня, и присваивать полю *isRockedLeftHand* значение *true*, в противном случае – наоборот. Также мы проверяем, есть ли на этом камне монетка, тогда мы вызываем так называемую корутину *ScoreCoint()*. Корутина – простой и удобный способ запускать функции, которые должны работать параллельно в течение некоторого времени. Корутины представляют собой простые *C#* итераторы, возвращающие *IEnumerator* и использующие ключевое слово *yield*. В данной корутине мы проверяем, сменилось ли вращение персонажа, и если это так, то прибавляем значение монет. Скрипт для правой кисти имеет такой же вид с некоторыми изменениями, например направление движения, поэтому описывать мы его не будем.

Далее в скрипте *PlayerController.cs* сделаем обработчик нажатия на клавишу мыши (касания экрана) с помощью статического метода *GetMouseButtonDown()*, который находится в классе *Input*. И по клику у нас будет изменяться значение переменной «*direction*», отвечающей за направление вращения персонажа. Вся вышеописанная логика будет

находиться в методе *Update()*, который обрабатывает каждый кадр. Также в методе *Update()* реализуем логику смены направления вращения нашего персонажа. В зависимости от значения переменной «*direction*» мы будем вращать его с помощью метода *RotateAround()*, находящегося в классе *Transform*. Для этого у свойства «*transform*» мы вызовем метод и передадим ему три параметра: первый – вокруг чего мы будем совершать вращение, второй – градус поворота, третий – скорость.

Нажатие будет приходиться не точно в центр области нашего камня: для красоты движения, если мы нажали на экран, когда кисть находилась не в центре камня, мы сдвинем кисть в нужное место. Для этого воспользуемся статическим методом *SmoothDamp()*, который применим к свойству *position* указанного *transform'a*. В нашем случае данный метод принимает три основных параметра: первый – текущая позиция, второй – позиция, куда нужно сместить, третий – скорость смещения. Данный метод мы используем также для движения камеры за персонажем, чтобы оно было плавным.

Итак, мы рассмотрели на примере создания игры «*Rock Climber*» особенности организации взаимодействия языка программирования *C#* с межплатформенной средой разработки *Unity*. Специфика реализации методов игрового проекта «*Rock Climber*» доказывает эффективность встраивания программного кода языка *C#* в модули платформы среды разработки компьютерных игр *Unity*. Таким образом, создание игровых проектов раскрывает широкие горизонты для совместного использования различных средств и методов ИТ-технологий.

Литература

1. Козлов С.В. Математические особенности использования возможностей программного комплекса «Advanced Tester» как инструмента функционального анализа системных данных // International Journal of Open Information Technologies. 2019. Т. 7, № 2. С. 21-30.
2. Козлов С.В. Применение методов функционального анализа при формировании оптимальных стратегий обучения школьников // Международный журнал экспериментального образования. 2016. № 3-2. С. 182-185.
3. Размахнина А.Н., Баженов Р.И. О применении экспертных систем в различных областях // Постулат. 2017. № 1 (15). С. 38.
4. Козлов С.В. Содержание и особенности разработки игровых мультимедийных проектов обучения математике в среде программирования LAZARUS // Психология, социология и педагогика. 2015. № 6(45). С. 88-96.
5. Шрейер Дж. Кровь, пот и пиксели. Обратная сторона индустрии видеоигр. М.: Бомбора, 2019. 368 с.
6. Хокинг Дж. Unity в действии. Мультиплатформенная разработка на *C#* / пер. с англ. И. Рузмайкиной. СПб.: Питер, 2016. 336 с.
7. Козлов С.В., Суин И.А. Особенности реализации графических решений при разработке пользовательских приложений в объектно-ориентированных средах программирования // Международный студенческий научный вестник. 2017. № 5. С. 53.

8. Суин И.А., Козлов С.В. Особенности разработки интерактивных приложений // Постулат. 2017. № 11. С. 29.

9. Документация Unity. URL: <https://docs.unity3d.com/Manual/index.html> (дата обращения: 25.02.2019).

S.V. Kozlov, I.O. Mezhentcev
Smolensk State University

INTERACTION OF THE C# PROGRAMMING LANGUAGE WITH THE CROSS-PLATFORM UNITY DEVELOPMENT ENVIRONMENT

Keywords: *informatic, informatic system, programming, C#, Unity development environment, video game development technology.*

Abstract. *The article discusses the problem of creating video games, which in turn push for the creation of new solutions for the development of computer science. As an example, we consider the development of a video game using the cross-platform UNITY environment. To interact with the development environment using the programming language C#, which has its own characteristics in this environment. On the basis of the conducted development, it is concluded that the complexity and features of the video game development and the existence of a solution to simplify the development with help of adapted for each other media.*

С.В. Козлов, И.А. Суин
Смоленский государственный университет

УДК 004.62

О НЕКОТОРЫХ АСПЕКТАХ ПРИМЕНЕНИЯ ИНВАРИАНТНЫХ МЕТОДОВ ФУНКЦИОНАЛЬНОГО АНАЛИЗА ДАННЫХ В РАЗЛИЧНЫХ ПРЕДМЕТНЫХ ОБЛАСТЯХ

Ключевые слова: *информатика, информационные системы, базы данных, программирование, информационно-коммуникационные технологии, функциональный анализ, соответствие Галуа, инвариантные методы, графовая модель.*

В статье обсуждаются вопросы применения методов функционального анализа данных в различных предметных областях. В качестве примеров рассматриваются информационные системы компьютерных сетей и тестирования знаний учащихся. Для изучения

особенностей указанных систем предлагается использовать методологию соответствия Галуа как одного из методов функционального анализа данных. В информационной системе компьютерной сети соответствие Галуа применяется для анализа надежности ее компонентов. В информационной системе тестирования знаний учащихся оно используется для выявления элементов усвоения учебного материала. Это позволяет выявить структурные элементы информационных систем, на основе дальнейшего изучения которых можно генерировать и исследовать различные сценарии их поведения. На основе проведенного анализа делается вывод об инвариантности соответствия Галуа относительно системных данных рассматриваемых предметных областей. Это позволяет говорить об эффективности применения соответствия Галуа как метода функционального анализа данных в информационных системах.

В настоящее время вопросы проектирования и сопровождения информационных систем носят особо важный характер в силу цифровизации практически всех без исключения сфер человеческой деятельности. Одним из приоритетных направлений исследований в этой области выступает проблема поиска методов анализа многомерных системных данных, позволяющих следить за внутренними процессами и предпринимать действия по управлению системой эффективным образом [1]. При этом востребованными оказываются методологии, математический аппарат которых является инвариантным относительно некоторого множества рассматриваемых предметных областей [2]. В качестве таковых можно выделить целую группу математических методов, основанных на функциональном анализе данных структуры и содержания информационной системы.

При описании моделей информационных систем различных предметных областей мы часто встречаем перечни однотипных элементов, объединенных в схемы. Данные схемы могут быть организованы разными способами в силу специфики рассматриваемой предметной области или ряда других параметров. Часто это препятствует составлению общих методов анализа подобных систем, которые можно было бы использовать при решении широкого круга практических задач. В то же время значительную часть схем в информационных системах можно представить в виде графовых моделей. С одной стороны, это позволяет анализировать данные информационных систем различных предметных областей, исходя из общих принципов их представления, с другой – открывает перспективы разработки методов анализа системных данных, отвечающих принципу инвариантности.

В числе таких инвариантных методов следует упомянуть ряд функциональных соотношений области математического анализа, в частности соответствие Галуа [3]. Они позволяют как исследовать собственно элементы информационных систем, так и изучать связи между ними. При этом, как правило, именно анализ связей между группами системных элементов дает возможность изучить латентные параметры, оказывающие существенное влияние на функционирование системы в целом.

Охарактеризуем инвариантность методов функционального анализа на примерах информационных систем компьютерной сети и тестирования знаний учащихся. Первая из них реализована посредством программного комплекса «Net Reliability Checker» [4], а вторая – в виде многомодульной программы «Advanced Tester» [5].

В программной среде «Net Reliability Checker» пользователю предоставляется возможность спроектировать модель существующей компьютерной сети. Целью такого проектирования является анализ ее свойств. Полученная информация используется для повышения уровня качества компьютерной сети, а также выявления характеристик её элементов для последующего их изучения. Одним из таких характеристических свойств может выступать параметр надежности компонентов сети. Используя инструменты программы, пользователь создает в ее основной рабочей области графовую модель, где вершины – это узлы сети, а ребра – связи между ее узлами [6; 7].

Каждый узел сети графовой модели характеризуется в программе рядом параметров, таких как скорость обработки информации, пропускная способность канала связи, объем памяти, тип операционной системы и др. Некоторые из данных параметров имеют ключевое значение для анализа надежности сетевых элементов, в то время как значениями других параметров при данной проверке можно пренебречь.

Когда модель полностью спроектирована, пользователь переходит к закладке «Тест». Здесь ему предоставляется возможность создать тест, который будет использован для проверки уровня качества смоделированной сети. Тестирование сети будет состоять из нескольких независимых тестовых проверок, каждая из которых позволяет получить информацию о ее состоянии. Для проведения тестирования пользователю необходимо выбрать набор проверок. В свою очередь, для каждой из них следует указать задействованные в проверке узлы – откуда и куда был отправлен сигнал, какие узлы спроектированной сети были задействованы и так далее.

После проведения тестирования программа выводит на экран графовую модель сети, где каждому узлу присвоен коэффициент надежности. Данный коэффициент выводится на основе количественных результатов тестирования, которое проверяет соответствующие

параметры смоделированной сети. Этой информации уже достаточно для исходных суждений о некоторых характеристиках имеющейся сети. Так как необходимый минимальный порог параметра надежности задается пользователем, он уже может делать некоторые выводы относительно имеющейся графовой модели. Одним из наиболее полезных результатов является выделение проблемных узлов, коэффициент надежности которых ниже заданного параметра. Однако данная программа позволяет этим не ограничиваться.

Используя методы функционального анализа, в частности метод соответствия Галуа, программа может обнаружить ряд дополнительных латентных характеристик [8]. Прежде всего она позволит не только выявить список узлов, коэффициент надежности которых меньше указанного значения, но и вскрыть вероятные причины попадания узла в этот список. Это может произойти в силу наличия некоторых отрицательных факторов на узлах, связанных с анализируемым узлом. Также программа помогает делать выводы о значительном количестве узлов сети не только после анализа всех имеющихся узлов, но и после анализа их фиксированной группы меньшего размера [9].

Подобная методология уже применялась ранее, но в иной области. Была разработана программная оболочка «Advanced Tester», предназначенная для моделирования информационного пространства «Средняя школа – вуз». Данная программа позволяла оптимально решать следующие задачи: формирование систем учебных заданий и построение индивидуальных траекторий обучения, отвечающих заданным параметрам, генерирование системы индивидуальных заданий в соответствии с личностными запросами учащегося.

Программа «Advanced Tester» содержит ряд баз данных с различным функционалом, прежде всего базу данных, включающую учебный материал по изучаемой дисциплине. Например, фундаментальное ядро составляет материал по математике, информатике и физике. При этом методы, заложенные в работу программной среды, инвариантны относительно предметной области [10]. Материал разбит на отдельные элементы, как правило, это относительно завершенные предметные блоки, например, группа параграфов или отдельная тема для изучения. На основе данного разбиения строится графовая модель, где узлы – это элементы знаний, а ребра указывают на связь этих элементов знаний между собой. Точнее, направленное ребро от одного элемента знаний к другому элементу знаний указывает, что второй элемент знаний опирается на первый [11].

Также программа содержит базу данных тематических задач, каждая из которых связана с конкретным элементом знаний. Эти задачи обладают рядом параметров, таких как сложность, полнота и эффективность. Помимо вышеперечисленных параметров задача

обладает рядом неочевидных характеристик. Так как элементы знаний в основном имеют связи или базируются на других элементах знаний, многие задачи связаны не только с элементами знаний, по которым проводится тестирование, но и с предыдущими элементами знаний, усвоение которых влияет на успех решения задачи.

После изучения имеющегося материала ученику предоставляется тест, состоящий из фиксированного перечня задач. После прохождения теста на основе полученной информации делаются выводы относительно того, как обучающийся усвоил тот или иной материал. Особенность данной программы заключается в том, что, обращая внимание на одну конкретную задачу и исследуя результаты только её успешного или неуспешного выполнения, мы с определенной долей вероятности можем делать выводы об усвоении учеником совершенно других элементов знаний, которые не были связаны непосредственно с изначальным.

На основе набора задач также выстраивается граф, где в роли узлов выступают задачи, а в качестве ребер – связи между теоретическим материалом и навыками практических действий, проверяемыми в них. По результатам решения такого фиксированного перечня задач учеником программа строит таблицу надежности на основе математического аппарата импликативных матриц [12; 13]. В данной таблице каждому ребру графовой модели присваивается числовое значение успешности изучения. Далее полученные результаты заносятся в соответствующую сводную таблицу, имеющую динамические характеристики. Данные параметры будут естественным образом изменяться с изучением учебного материала. Целью же применения соответствия Галуа как инструмента функционального анализа данных, является исследование характера происходящих изменений и выявление причин данных процессов.

Таким образом, применяя метод соответствий Галуа, анализируя фиксированный набор из общего множества результатов, мы можем делать выводы относительно всего множества в целом. В данном примере это не только позволяет изучать степень усвоения учеником учебного материала, но и дает возможность более тщательно подбирать для него индивидуальную программу обучения, основываясь на полученных данных.

Как можно заметить, предложенные варианты использования методов соответствий Галуа весьма схожи в исполнении. Хотя предметная область и определяет специфику полученных результатов, а также дальнейшие семантические выводы, методология использования математического аппарата остается неизменной. Таким образом, это только лишний раз подчеркивает основную идею – универсальность рассмотренного метода функционального анализа данных.

Литература

1. Муха В.С. Математические модели многомерных данных // Доклады Белорусского государственного университета информатики и радиоэлектроники. 2014. № 2 (80). С. 143-158.
2. Козлов С.В., Суин И.А. О некоторых подходах математического описания и анализа многомерной структуры информационных систем // Системы компьютерной математики и их приложения. Смоленск, 2018. С. 177-182.
3. Кон П.М. Универсальная алгебра / пер. с англ. Т.М. Баранович; под ред. А.Г. Куроша. М.: Мир, 1968. 351 с.
4. Суин И.А. О некоторых вопросах разработки моделей компьютерных сетей. Постулат. 2018. №6(32). С. 98.
5. Козлов С.В. Математические особенности использования возможностей программного комплекса «Advanced Tester» как инструмента функционального анализа системных данных // International Journal of Open Information Technologies. 2019. Т. 7., № 2. С. 21-30.
6. Кристофидес Н. Теория графов. Алгоритмический подход. М.: Мир, 1978. – 432 с.
7. Зыков А. А. Основы теории графов. – М: Вузовская книга, 2004. 664 с.
8. Козлов С.В. Использование соответствия Галуа как инварианта отбора контента при проектировании информационных систем // Современные информационные технологии и ИТ-образование. 2015. Т. 2, № 11. С. 220-225.
9. Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы / пер. с англ. А.А. Минько. М.: Вильямс, 2010. 400 с.
10. Козлов С.В. Применение методов функционального анализа при формировании оптимальных стратегий обучения школьников // Международный журнал экспериментального образования. 2016. № 3-2. С. 182-185.
11. Козлов С.В. Интерпретация инвариантов теории графов в контексте применения соответствия Галуа при создании и сопровождении информационных систем // International Journal of Open Information Technologies. 2016. Т. 4, № 7. С. 38-44.
12. Соколов Н.П. Введение в теорию многомерных матриц. Киев: Наукова думка, 1972. 176 с.
13. Козлов С.В. Использование математического аппарата импликативных матриц при создании и сопровождении информационных систем // International Journal of Open Information Technologies. 2017. Т. 5, № 12. С. 16-23.

S.V. Kozlov, I.A. Suin
Smolensk State University

ABOUT SOME ASPECTS OF APPLICATION OF INVARIANT METHODS OF THE FUNCTIONAL ANALYSIS OF DATA IN VARIOUS SUBJECT DOMAINS

Keywords: *informatics, information systems, databases, programming, information and communication technologies, functional analysis, compliance of Galois, invariant methods, graph model.*

Abstract. *In article discuss of application the functional analysis data methods in various subject domains are discussed. As examples information systems of computer networks and testing of knowledge of pupils are considered. For studying features of the considered systems it is offered to use methodology of compliance Galois as one of data functional analysis methods. In information system of computer network applied Galois's compliance to analysis of reliability its components. In information system of testing of knowledge of pupils it used for identification of assimilation elements a training material. It allows to reveal structural elements of information systems on the basis of which further studying it is possible to generate and investigate various scenarios of their behavior. On the basis of carried-out analysis the conclusion about invariancy of compliance Galois rather system these considered subject domains is drawn. It allows speaks about efficiency of use'd compliance Galois as method of the functional analysis data in information systems.*

А.И. Макаров

Смоленский государственный университет

УДК 519.686.4

ПРОБЛЕМА КОЭФФИЦИЕНТА ИНФОРМАЦИОННОЙ ЭНТРОПИИ ДЛЯ ПОКАЗАТЕЛЬНОГО ЗАКОНА РАСПРЕДЕЛЕНИЯ

Ключевые слова: *информационная энтропия, надежность, контроль, проектирование систем, показательный закон распределения, выборка.*

В статье рассматривается проблема коэффициента информационной энтропии на примере показательного закона распределения. Базируясь на основах математической статистики, понятии энтропии и вычислительном эксперименте, автор предлагает способ вычисления коэффициента информационной энтропии. Приведены расчеты и рассуждения, доказывающие его достоверность и исключающие возможность использования известного метода расчета коэффициента информационной энтропии для показательного закона распределения.

Любая программа создает информацию и может обмениваться ею с другими программами, этот обмен является дискретным сигналом. Вычислительная система передает и принимает информацию – это тоже создание дискретного сигнала. Любое передаваемое сообщение можно рассматривать как источник дискретного сигнала. Дискретный сигнал

отличается тем, что в любой момент времени он может находиться в одном из нескольких состояний. Рассматривая именно такой источник сигнала в своей работе [1], Клод Шеннон ввел понятие информационной энтропии: «Назовем величину $H = - \sum p_i * \log(p_i)$ энтропией множества вероятностей $p_1 \dots p_n$ ». Эту величину принято считать количественной мерой возможности выбора или мерой количества информации.

Однако, помимо количества информации, данную величину можно рассматривать как меру хаотичности прохождения какого-либо процесса (к примеру, длительность сеанса пользователя, промежутки времени между приходящими на сервер запросами, время работы устройства до поломки [2]). Но именно этому применению энтропии в теории информации уделяется предельно мало внимания, так как мера случайности в процессе практически не информативна.

Большее значение имеет коэффициент информационной энтропии. Это величина, позволяющая сравнить информационную энтропию процесса с информационной энтропией модели данного процесса. Такое сравнение позволяет определить, как сильно реальный процесс отличается от собственной модели.

Особую ценность это имеет при проектировании информационных систем на этапе первоначального тестирования, так как именно на этом этапе важно определить отклонения от проекта и внести правки. Но и на более поздних этапах коэффициент информационной энтропии может сыграть важную роль. Так, например, в случае с запросами к функционирующему серверу при серьезном отклонении расчетного значения информационной энтропии от модельного можно обнаруживать аномалии, которые, среди прочего, могут быть вызваны кибератаками либо неисправностями оборудования или программного обеспечения.

Для расчета коэффициента информационной энтропии предлагается значение информационной энтропии реального процесса разделить на значение информационной энтропии математической модели данного процесса. Причем информационная энтропия модели рассчитывается как энтропия непрерывного процесса: $H = - \int f(t) * \log(f(t)) dt$.

Но при попытке внедрения данных расчетов на производстве выявился серьезный недочет данной формулы. Так как числитель дроби рассчитывается по дискретной формуле и выборке, он оказался зависимым от количества интервалов разбиения, в то время как знаменатель, рассчитываемый по непрерывной формуле, от числа интервалов не зависит.

В случае использования дискретной модели функция будет зависеть от двух параметров: собственно параметра показательного

распределения и числа интервалов разбиения, тогда как при непрерывной модели функция зависит только от параметра распределения. Для проверки данного утверждения были проведены расчеты в системе компьютерной математики Wolfram Mathematica 11.2. Далее приводится листинг программы с комментариями.

Для расчета информационной энтропии показательного распределения с параметром λ и заданным количеством интервалов разбиения *count* следует рассчитать вероятность попадания значения аргумента в каждый из интервалов разбиения.

$$p_{\text{показательное}}[i, \lambda] = \\ = \text{Simplify}[(1 - e^{-(\lambda * i * 1/\text{count})}) - (1 - e^{-(\lambda * (i - 1) * 1/\text{count})})];$$

Далее приводится исследование значения информационной энтропии.

$$K_{\text{показательное-дискретное}}[\text{count}, \lambda] = \\ = \text{Simplify}[\sum_{i=1}^{\text{count}} (p_{\text{показательное}}[i, \lambda] \\ * \text{Log}[1/p_{\text{показательное}}[i, \lambda]])];$$

Полученный результат:
$$\sum_{i=1}^{\text{count}} e^{-\frac{i\lambda}{\text{count}}} (-1 + e^{\frac{\lambda}{\text{count}}}) \text{Log}\left[\frac{e^{\frac{i\lambda}{\text{count}}}}{-1 + e^{\frac{\lambda}{\text{count}}}}\right]$$

Для дальнейших расчетов логарифм представлен в виде разности.

$$K1_{\text{показательное-дискретное}}[\text{count}, \lambda] = \\ = \sum_{i=1}^{\text{count}} e^{-\frac{i\lambda}{\text{count}}} (-1 + e^{\frac{\lambda}{\text{count}}}) \left(\frac{i\lambda}{\text{count}} - \text{Log}[-1 + e^{\frac{\lambda}{\text{count}}}] \right);$$

Рассмотрена функция энтропии от числа интервалов разбиения на монотонность.

$$g[z, \lambda] = \text{Simplify}[K1_{\text{показательное-дискретное}}[z, \lambda] \\ - K1_{\text{показательное-дискретное}}[z - 1, \lambda]];$$

$$dg[z, \lambda] = \text{Simplify}[\partial_z g[z, \lambda]];$$

$$\text{Limit}[g[z, \lambda], z \rightarrow \infty];$$

$$\text{Limit}[dg[z, \lambda], z \rightarrow \infty];$$

$$\text{Plot3D}[dg[z, \lambda], \{z, 5, 100000\}, \{\lambda, 0, 1\}, \text{AxesLabel} \rightarrow \text{Automatic}]$$

Пределы обеих функций получились равными нулю.

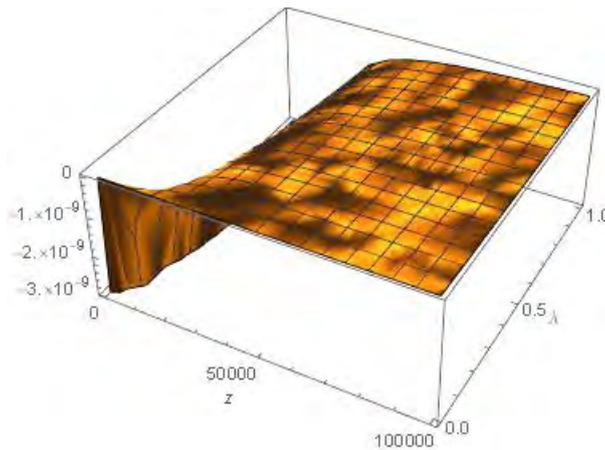


Рис. 1. Графическое представление функции

Производная функции оказалась отрицательной при количестве отрезков, не превышающем 100 000. В практическом применении даже такое количество отрезков является чрезвычайно большим, так как обычно рекомендуется брать величину \sqrt{n} , а значит, для такого числа измерений должно быть не менее 10^{10} , либо величину $5 \lg n$ [4], тогда измерений нужно не менее 10^{20000} . Ограничения на λ наложены для упрощения представления графика с учетом того, что математическими преобразованиями любое значение λ_0 можно свести в интервал $(0,1]$.

Из того, что производная функции оказалась отрицательной, можно сделать вывод, что сама функция является монотонно убывающей. При этом функция остается положительной. Значит, при увеличении числа интервалов разбиения значение информационной энтропии возрастает.

Рассмотрим теперь формулу для вычисления информационной энтропии при непрерывном законе распределения.

$$\begin{aligned}
 f(x) &= \lambda e^{-\lambda x} \\
 \int_0^{+\infty} \lambda e^{-\lambda x} \ln \left(\frac{e^{\lambda x}}{\lambda} \right) dx &= \lambda \int_0^{+\infty} e^{-\lambda x} (\lambda x - \ln(\lambda)) dx = \\
 &= \lambda^2 \int_0^{+\infty} x e^{-\lambda x} dx - \lambda \ln(\lambda) \int_0^{+\infty} e^{-\lambda x} dx \\
 &= \int_0^{+\infty} -\lambda x e^{-\lambda x} d(-\lambda x) + \ln(\lambda) \int_0^{+\infty} e^{-\lambda x} d(-\lambda x) \\
 &= 1 - \ln(\lambda)
 \end{aligned}$$

Полученная функция зависит только от параметра λ .

Из данных рассуждений можно заключить, что высказанное выше предположение верно и в случае использования дискретной модели функция будет зависеть от двух параметров: собственно параметра показательного распределения и числа интервалов разбиения, в то время как при непрерывной модели функция зависит только от параметра распределения.

Литература

1. Шеннон К. Работы по теории информации и кибернетике. М., 1963.
2. Письменный Д.Т. Конспект лекций по теории вероятностей и математической статистике. Айрис-пресс, 2005.
3. Heinhold I., Gaede K.W. Ingenieur statistic. München; Wien, Springer Verlag, 1964. 352 p.
4. Шторм Р., Федоров Н.Н., Федоров М.Г. Теория вероятностей; Математическая статистика; Статистический контроль качества: пер. с нем. М.: Мир, 1970.

A.I. Makarov
Smolensk State University

THE PROBLEM OF INFORMATION ENTROPY COEFFICIENT FOR AN EXAMINATIVE DISTRIBUTION

Keywords: *informational entropy, reliability, control, system design, exponential distribution, sampling.*

Abstract. *The article deals with the problem of the coefficient of information entropy on the example of the exponential distribution. A method for calculating the coefficient of information entropy is proposed based on the statistic's basics, the concept of entropy and computational experiment. Calculations and arguments are presented that prove its reliability and exclude the possibility of using the well-known method of calculating the information entropy coefficient for the exponential distribution.*

В.И. Мунерман, Д.В. Мунерман
Смоленский государственный университет

УДК 004.652

СООТВЕТСТВИЕ ОПЕРАЦИЙ В МНОГОМЕРНО- МАТРИЧНОЙ И РЕЛЯЦИОННОЙ МОДЕЛЯХ ДАННЫХ

Ключевые слова: *алгебра многомерных матриц; реляционная модель данных; гомоморфизм алгебр.*

В статье рассмотрены две операции различных алгебр: операция (λ, μ) -свернутого произведения многомерных матриц и операция Join реляционной алгебры. Показано, что можно установить однозначное соответствие между множеством логических многомерных матриц и

множеством отношений. Доказано, что при этом результату операции Join двух отношений соответствует результат операции (λ, μ) -свернутого произведения их образов. Приведен пример, иллюстрирующий полученный результат.

Цель работы состоит в доказательстве соответствия двух важных операций алгебры многомерных матриц [1] и реляционной алгебры.

Пусть i_1, \dots, i_p – совокупность индексов, принимающих значения от 1 до n_α ($\alpha = 1, \dots, p$) соответственно. Тогда p -мерная матрица – это совокупность $T = \{a_{i_1 \dots i_p}\}$ элементов некоторого типа, над которым определены аддитивная и мультипликативная операции. Многомерная матрица $A = \parallel A_{i_1 \dots i_p} \parallel$ называется логической многомерной матрицей (ЛММ), если ее элементы принадлежат множеству $\{0, 1\}$ и над ними определены аддитивная операция дизъюнкции и мультипликативная операция конъюнкции [2].

Пусть ЛММ $A = \parallel A_{i_1 \dots i_p} \parallel$ и $B = \parallel B_{i_1 \dots i_q} \parallel$, p и q -мерные соответственно. Совокупности индексов этих матриц i_1, \dots, i_p и i_1, \dots, i_q разбиваются на четыре группы, содержащие соответственно κ , λ , μ и ν индексов ($\kappa, \lambda, \mu, \nu \geq 0$). Причем $\kappa + \lambda + \mu = p$, а $\lambda + \mu + \nu = q$. Для полученных групп индексов используются обозначения: $l = (l_1, \dots, l_\kappa)$, $s = (s_1, \dots, s_\lambda)$, $c = (c_1, \dots, c_\mu)$ и $m = (m_1, \dots, m_\nu)$. Тогда матрицы A и B можно представить в виде $A = \parallel A_{lsc} \parallel$ и $B = \parallel B_{scm} \parallel$. Индексы групп s и c в матрицах A и B полностью совпадают. Так же как в операции свертки, индексы разбиения c называются кэлиевыми. Индексы разбиения s называются скоттовыми, а индексы разбиения m , так же как и индексы разбиения l , свободными.

Матрица $C = \parallel C_{lsm} \parallel$, элементы которой вычисляются по формуле $C_{lsm} = \bigvee_{(c)} A_{lsc} \wedge B_{scm}$, называется произведением матриц A и B .

Алгоритм реализации этой операции состоит в следующем:

- 1) перемножаются все пары элементов с фиксированными значениями индексов групп l, s, m , у которых полностью совпадают значения индексов группы c ;
- 2) суммируются все произведения с одинаковыми значениями индексов группы c .

Произведение многомерных матриц называется (λ, μ) -свернутым произведением и обозначается ${}^{\lambda, \mu}(A \times B)$. Из определения (λ, μ) -

свернутого произведения следует, что для любой пары многомерных матриц можно построить много различных произведений, подбирая различные значения λ и μ .

Далее рассматривается наиболее употребляемая разновидность реляционной операции Join. Пусть $R_1(F_1)$ и $R_2(F_2)$ – два отношения, схемы которых содержат совокупности атрибутов $F_1 = \{F_{11}, \dots, F_{1m}\}$ и $F_2 = \{F_{21}, \dots, F_{2n}\}$. Предполагается, что оба отношения имеют одинаковые наборы атрибутов S_1, \dots, S_λ и C_1, \dots, C_μ . Тогда их схемы можно представить как $R_1(S_1, \dots, S_\lambda, C_1, \dots, C_\mu, L_1, \dots, L_\kappa, F'_1)$ и $R_2(S_1, \dots, S_\lambda, C_1, \dots, C_\mu, M_1, \dots, M_\nu, F'_2)$, где наборы атрибутов L_1, \dots, L_κ и M_1, \dots, M_ν , как правило, различны. Обычно наборы атрибутов $S_1, \dots, S_\lambda, C_1, \dots, C_\mu, L_1, \dots, L_\kappa$ и M_1, \dots, M_ν рассматриваются как ключи базы данных, а F'_1 и F'_2 – как атрибуты, которые участвуют в вычислениях.

SQL-запрос, реализующий рассматриваемую операцию, имеет вид:

```
SELECT S1, ..., Sλ, L1, ..., Lκ, M1, ..., Mν, Σ(F'1, F'2)
FROM R1 INNER JOIN R2
ON R1.S1 = R2.S1 AND ... AND R1.Sλ = R2.Sλ
GROUP BY S1, ..., Sλ, L1, ..., Lκ, M1, ..., Mν
```

Предикат ON задает отношение эквивалентности, которое разбивает отношения R_1 и R_2 на классы эквивалентности, превращая их в фактор-множества. Классы эквивалентности обозначаются R_1^* и R_2^* .

$\sum_{(R_1^* \times R_2^*)} (F'_1, F'_2)$ – функция, реализующая одну из групповых операций, стандартных для используемой СУБД или определенных пользователем. Такая функция определена на множестве всех кортежей каждого декартова произведения $R_1^* \times R_2^*$ соответствующих классов эквивалентности таблиц R_1 и R_2 . Результат вычисления этой функции – кортеж, построенный на основе атрибутов, содержащихся в наборах F'_1 и F'_2 . Тогда операцию Join можно формально определить как $\bigcup_{(S_1, \dots, S_\lambda)} \sum_{R_1^* \times R_2^*} (F'_1, F'_2)$.

Приведенные определения позволяют установить соответствие между ЛММ и отношениями в реляционной модели данных, а также между операциями (λ, μ) -свернутого произведения и Join.

Пусть $R_1(F_1)$ и $R_2(F_2)$ – два отношения, схемы которых имеют вид $R_1(S_1, \dots, S_\lambda, C_1, \dots, C_\mu, L_1, \dots, L_\kappa, F'_1)$ и $R_2(S_1, \dots, S_\lambda, C_1, \dots, C_\mu, M_1, \dots, M_\nu, F'_2)$. Очевидно, что в реальных отношениях, хранящихся в базе данных, могут содержаться не все возможные классы эквивалентности. Это может привести к коллизиям при установлении соответствия между отношениями и ЛММ. Для того, чтобы избежать коллизий, вводится понятие универсального неопределенного значения Θ для кортежей F'_1 и F'_2 . Тогда, если кортежи $(s_1, \dots, s_\lambda, c_1, \dots, c_\mu, l_1, \dots, l_\kappa, F'_1)$ или $(s_1, \dots,$

$s_\lambda, c_1, \dots, c_\mu, m_1, \dots, m_\nu, F'_2)$ физически отсутствуют в отношении, предполагается, что отношение логически содержит кортежи $(s_1, \dots, s_\lambda, c_1, \dots, c_\mu, l_1, \dots, l_\kappa, \Theta)$ или $(s_1, \dots, s_\lambda, c_1, \dots, c_\mu, m_1, \dots, m_\nu, \Theta)$, в дальнейшем – Θ -кортежи. Таким образом, можно утверждать, что оба отношения содержат все возможные кортежи, то есть все возможные классы эквивалентности.

Поскольку домены атрибутов $S_1, \dots, S_\lambda, C_1, \dots, C_\mu, L_1, \dots, L_\kappa$ и M_1, \dots, M_ν не более чем счетные, а на практике – конечные множества, то их элементы можно пронумеровать, и тогда каждому атрибуту соответствует индекс.

Тогда отношениям $R_1(F_1)$ и $R_2(F_2)$ можно поставить в соответствие ЛММ $A = \|A_{i_1 \dots i_p}\| = \|A_{isc}\|$ и $B = \|B_{i_1 \dots i_q}\| = \|B_{scm}\|$ следующим образом:

1) доменам атрибутов $S_1, \dots, S_\lambda, C_1, \dots, C_\mu, L_1, \dots, L_\kappa$ и M_1, \dots, M_ν ставятся в соответствие множества значений индексов разбиений $l = (l_1, \dots, l_\kappa)$, $s = (s_1, \dots, s_\lambda)$, $c = (c_1, \dots, c_\mu)$ и $m = (m_1, \dots, m_\nu)$, то есть размерности матриц однозначно определяются количеством атрибутов, идентифицирующих классы эквивалентности отношений;

2)

$$A_{isc} = \begin{cases} 0, & \text{если соответствующий класс эквивалентности } R_1 \\ & \text{состоит из единственного } \Theta\text{-кортежа,} \\ 1, & \text{в противном случае;} \end{cases}$$

3)

$$B_{scm} = \begin{cases} 0, & \text{если соответствующий класс эквивалентности } R_2 \\ & \text{состоит из единственного } \Theta\text{-кортежа,} \\ 1, & \text{в противном случае.} \end{cases}$$

При таком построении ЛММ A и B , соответствующих отношениям $R_1(F_1)$ и $R_2(F_2)$, наборы атрибутов $S_1, \dots, S_\lambda, C_1, \dots, C_\mu$ соответствуют наборам скоттовых и кэлиевых индексов, и результат конъюнкции элементов ЛММ, соответствующих классам эквивалентности, будет следующим:

$$A_{isc} \wedge B_{scm} = \begin{cases} 0, & \text{если по крайней мере один из } A_{isc} \text{ и } B_{scm} \text{ соответствует классу} \\ & \text{эквивалентности, состоящему из единственного } \Theta\text{-кортежа,} \\ 1, & \text{в противном случае.} \end{cases}$$

По определению отношение-результат рассматриваемого варианта операции Join строится из пар физически присутствующих классов эквивалентности отношений-операндов с одинаковыми значениями экземпляров атрибутов S_1, \dots, S_λ . Но этим классам эквивалентности в обеих ЛММ-операндах ${}^{\lambda, \mu}(A \times B)$ соответствуют элементы, равные 1. Следовательно, соответствующий элемент ЛММ-результата также будет равен 1, и она будет образом полученного в операции Join отношения-результата.

Соответствие между множеством всех отношений и множеством всех ЛММ однозначное, так как двум отношениям с различно именованными атрибутами, содержащими данные из различных предметных областей, может соответствовать одна и та же ЛММ. Следовательно, рассмотренное соответствие алгебры ЛММ и реляционной модели данных по операции Join и операции (λ, μ) -свернутого произведения – гомоморфизм.

Пример. Пусть в реляционной базе данных заданы отношения и запрос, показанные на рисунке 1.

R_1			R_2			Результат запроса		SELECT S, SUM($F_1 \times F_2$) FROM R_1 INNER JOIN R_2 ON $R_1.S = R_2.S$ GROUP BY S
C	S	F_1	S	C	F_2	S	$\sum_{(R_1 \times R_2)} (F_1 \times F_2)$	
x	a	3	a	p	7	a	240	
y	a	6	a	q	3	d	361	
w	a	7	a	r	6	e	198	
x	b	2	a	s	3	f	176	
z	b	7	a	t	3		###	
v	b	3	c	r	9			
x	d	6	c	t	7			
y	d	1	d	r	5			
z	d	7	d	s	5			
v	d	2	d	t	9			
w	d	3	e	p	2			
y	e	7	e	q	8			
v	e	4	e	s	6			
x	f	0	e	t	2			
z	f	7	f	p	1			
w	f	9	f	s	8			
		###	f	t	2			
					###			

Рис. 1. Отношения $R_1(S, C, F_1)$ и $R_2(S, C, F_2)$, запрос и результат запроса

На рисунке 2 показаны ЛММ A и B , соответствующие этим отношениям, и одномерная матрица – результат их произведения ${}^{0,1}(A \times B)$. Как видно на рисунке, эта матрица содержит элементы, равные 1, при тех значениях индекса s , которые соответствуют значениям атрибута S , идентифицирующим классы эквивалентности отношения-результата операции Join.

		ЛММ A							ЛММ B							$^{0,1}(A \times B)$					
$c \rightarrow$	$s \downarrow$	x	y	z	v	w	$s \rightarrow$	$c \downarrow$	a	b	c	d	e	f	$s \rightarrow$	a	b	c	d	e	f
a		1	1	0	0	1	p		1	0	0	0	1	1		1	0	0	1	1	1
b		1	0	1	1	0	q		1	0	0	0	1	0							
c		0	0	0	0	0	r		1	0	1	1	0	0							
d		1	1	1	1	1	s		1	0	0	1	1	1							
e		0	1	0	1	0	t		1	0	1	1	1	1							
f		1	0	1	0	1															

Рис. 2. Таблицы, запрос и результат запроса

Проведенные в статье рассуждения позволяют сделать вывод о том, что алгебра многомерных матриц и реляционная модель данных гомоморфны относительно операции Join. Это позволяет применить методы параллельной реализации операции умножения многомерных матриц для распараллеливания соответствующей реляционной операции.

Литература

1. Соколов Н.П. Введение в теорию многомерных матриц. Киев: Наукова Думка, 1972.
2. Мунерман В.И. Построение архитектур программно-аппаратных комплексов для повышения эффективности массовой обработки данных // Системы высокой доступности. 2014. Т. 10, №. 4. С. 3-16.

V.I. Munerman, D.V. Munerman
Smolensk State University

THE COMPLIANCE OF OPERATIONS OF MULTI-DIMENSIONAL MATRIX ALGEBRA WITH OPERATIONS OF THE RELATIONAL DATA MODEL

Keywords: *algebra of multidimensional matrices; relational data model; algebra homomorphism.*

Abstract. *The article considers two operations of different algebras: an operation (λ, μ) -contracted product of multidimensional matrices and the operation Join of the relational algebra. It is shown that a single-valued transformation between the set of logical multi-dimensional matrices and the set of relations can be established. It is proved that the result of the operation Join two relations corresponds to the result of the operation (λ, μ) -contracted product of their images. An example illustrating the result obtained is given.*

ОЦЕНКА КАЧЕСТВА РЕЗУЛЬТАТОВ СИСТЕМ МАШИННОГО ПЕРЕВОДА

Ключевые слова: *оценка качества перевода, нейронные сети, статистический автоматизированный перевод, анализ.*

В статье рассматривается структура систем машинного перевода и освещается проблема оценки качества результатов машинного перевода на примере двух автоматизированных систем перевода – Яндекс.Переводчик и Google neural machine translate, в основе которых находятся различные подходы к осуществлению машинного перевода: гибридный и нейронный соответственно. Критериями оценки качества перевода выступили гладкость и адекватность перевода. По результатам сравнительного анализа переводов текстов статей научного и публицистического стиля на тему общей лингвистики, параллельного программирования и компьютерных технологий делаются выводы относительно качества машинного перевода и определяется минимальный набор средств, необходимых для достижения эквивалентного перевода. Исследование находится на начальном этапе, далее планируется расширить тематическую область текстового материала, осуществить глубокий анализ ошибок, полученные сведения учесть при разработке программы перевода текстов, применяющей разные API-ключи.

Целью данной работы является анализ различных стратегий машинного перевода посредством описательного и сравнительного методов исследования, в результате которого возможно будет определить минимальный набор средств для осуществления качественного перевода. Несмотря на то, что автоматизированный перевод – явление не новое, однозначного и полного решения проблемы перевода с одного языка на другой нет, чем обуславливается актуальность исследования.

В статье представлены результаты анализа работы систем машинного перевода, таких как Яндекс.Переводчик и Google Neural Machine Translation (GNMT), где перевод осуществляется посредством использования рекуррентной нейронной сети. Следует отметить, что система Яндекс.Переводчик использует гибридный подход: перевод осуществляется с использованием статистической модели и нейронной сети, затем классификатор «Catboost» оценивает результаты и выбирает

наиболее релевантный перевод. Система машинного перевода от компании «Google» базируется на использовании нейронной сети.

Использование параллельных корпусов лежит в основе систем статистического и нейронного машинного перевода. Параллельный корпус – это собрание построчно выровненных текстов. Сначала выравниваются отдельные слова, затем предложения.

Пусть длина предложения на языке E равна I , а длина предложения на языке F равна J , тогда вероятность длины предложения в F равна $P(J|E)$. Если все выравнивания $A = a_1, a_2, \dots, a_j$ равновероятны (их $(1+I)^j$), тогда вероятность (выравниваний): $P(A|E) = P(A|E, J)P(J|E) = \frac{P((J|E))}{(1+I)^j}$.

Пусть $t(f_x, e_y)$ – вероятность перевода слова e_y в слово f_x , тогда $P(F|E)$ можно определить как

$$P(F|E) = \sum_A P(F|E, A)P(A|E) = \sum_A \frac{P((J|E))}{(1+I)^j} \prod_{j=1}^J t(f_j, e_{(a_j)})$$

Процесс декодирования заключается в нахождении наиболее вероятного выравнивания [5]:

$$P(S) \times P(T|S) = P(S, T)$$

$$\hat{S} = \operatorname{argmax}_S P(S|T) \operatorname{argmax}_S P(S, T)$$

$$\hat{S} = \operatorname{argmax}_{0 \leq i \leq I} t(f_i, e_i) \quad 1 \leq i \leq J.$$

В статистической модели предложение при переводе дробится на слова / фразы (n-граммы). Предполагается, что у любого слова / фразы в одном языке есть соответствующее по смыслу слово / фраза в другом языке. Такая пара обозначается как (S, T) , а $P(T|S)$ – вероятность, с которой переводчик выберет пару (S, T) . Чем больше данная вероятность, тем «лучше» перевод слова / фразы, что можно подсчитать с помощью формулы Байеса: $P(S|T) = \frac{P(S)P(T|S)}{P(T)}$.

Из языковой модели и модели перевода выстраивается вероятность распределения пар S и T [2]:

Языковая модель – S – модель перевода – T – декодер – \hat{S}

$$P(S) \times P(T|S) = P(S, T) \quad \hat{S} = \operatorname{argmax}_S P(S|T) \operatorname{argmax}_S P(S, T)$$

Несовершенство статистических методов перевода состоит в отсутствии системного подхода к тексту как объекту со своими внутренними связями и характеристиками. Так, предложение «Besides being able to detect falls, the shoes can also reportedly track the wearer's movements and weight» будет переведено статистическим переводчиком как «Кроме того, будучи в состоянии обнаружить падает, то обувь можно также сообщается, отслеживать движения пользователя и веса». Нетрудно увидеть, что части предложения переведены точно, но смысл теряется из-за отсутствия каких-либо связей между ними.

Попыткой найти решение этой проблемы стало привлечение двунаправленных рекуррентных нейронных сетей в разработку алгоритмов машинного перевода. Перевод осуществляется посредством матричных вычислений за счёт представления слова в виде вектора и его прохождения через различные слои нейронной сети.

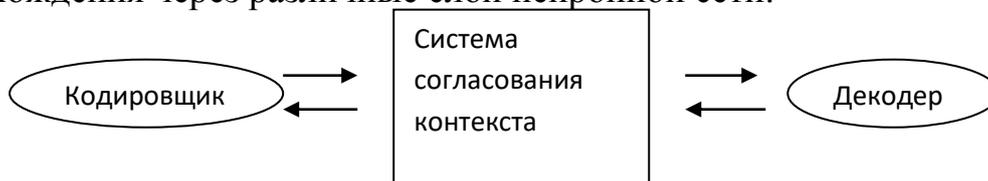


Рис. 1. Схема перевода слов в нейронной сети

В нейронной сети от «Google» имеется дополнительный слой softmax для вычисления распределения вероятностей перевода слова [6].

Для оценки качества результатов работы систем машинного перевода Яндекс.Переводчик и GNMT были использованы критерии полноты передачи смысла (adequacy) и гладкости перевода (fluency) [4].

Адекватный перевод подразумевает полное соответствие высказываний языка-источника и языка, на который осуществляется перевод. Кроме того, в выходном высказывании не должно быть пропущенных, непереуведенных или искаженных слов / фраз.

Под *гладкостью перевода* понимается грамматическая корректность выходного высказывания, а также соответствие идиоматических выражений языков, участвующих в процессе перевода, если таковые имеются.

Так, предложение «Besides being able to detect falls, the shoes can also reportedly track the wearer's movements and weight» будет переведено как «Помимо возможности обнаружения падений, обувь также может отслеживать движения и вес владельца» в соответствии с вышеприведенными критериями. Получившееся предложение логически и грамматически правильно построено и передает полностью смысл исходного высказывания.

Степень соответствия перевода данным критериям была оценена по пятибальной шкале.

Таблица 1

Баллы	Критерии оценивания	
	Критерий и его значение	
	Передача смысла (adequacy)	Гладкость перевода (fluency)
5	Полная	Безупречная
4	По большей части	Хорошая
3	Достаточная	Достаточная
2	Частичная	Частичная
1	Недостаточная	Недостаточная

Материалом исследования послужили статьи электронного издания «PCMag», а также аннотации статей из научных изданий «Information technology journal», «Australian journal of computer science», «Research journal of information technology», «Technologies», что в итоге составило 80 текстов различного объема. Перевод осуществлялся в двух направлениях: английский – русский и английский – французский.

В результате анализа полученных данных можно сделать вывод о том, что перевод, выполненный GNMT, в целом лучше и качественнее, чем у Яндекс.Переводчик, что наглядно представлено в таблицах ниже.

Таблица 2

Результаты оценки текстов перевода языковой пары «английский-русский»

Анг-ру		«PCMag»	Аннотации научных статей	Параллельное программирование	Лингвистика
Яндекс.Переводчик	Передача смысла	83	90	62	69
Яндекс.Переводчик	Гладкость перевода	65	69	57	65
GNMT	Передача смысла	85	93	68	70
GNMT	Гладкость перевода	72	81	66	68

Таблица 3

Результаты оценки текстов перевода языковой пары «английский-французский»

Англ-фр		«PCMag»	Аннотации научных статей	Параллельное программирование	Лингвистика
Яндекс.Переводчик	Передача смысла	82	75	69	66
Яндекс.Переводчик	Гладкость перевода	64	60	65	69
GNMT	Передача смысла	87	76	70	70
GNMT	Гладкость перевода	74	58	68	70

Тем не менее результаты перевода аннотаций научных статей с английского на французский язык гибридной системой Яндекс.Переводчик были оценены на 0,5% выше, чем перевод GNMT, что видно из таблицы 4.

Таблица 4

Итоговый процент по двум критериям (передача смысла и гладкость перевода)

Материалы исследования	Языки	Яндекс.Переводчик	GNMT
Аннотации	Англ-рус	79,5	87
	Англ-фр	67,5	67
Статьи «PCMag»	Англ-рус	74	78,5
	Англ-фр	73	80,5
Параллельное программирование	Англ-рус	59,5	67
	Англ-фр	67	69
Лингвистика	Англ-рус	67	69
	Англ-фр	67,5	70
Среднее значение качества перевода в двух направлениях: англ-рус и англ-фр		69,375	73,5

Следует отметить, что перевод статей электронного издания «PCMag», выполненный в языковой паре английский – французский, лучше представлен GNMT, в то время как аннотации научных статей и их вводная часть переведены примерно одинаково.

Обе системы машинного перевода достаточно хорошо справились с передачей смысла при переводе аннотаций научных статей с английского на русский язык (Яндекс.Переводчик – 90, GNMT – 93), в то время как этот показатель оказался значительно ниже при переводе статей электронного издания «PCMag» и вводной части научных статей по лингвистике и параллельному программированию.

Обратная ситуация складывается при переводе статей электронного издания «PCMag»: смысл передан лучше при переводе с английского на французский (Яндекс.Переводчик – 82, GNMT – 87), тогда как оценка данного критерия колеблется в районе 70 для научных статей.

Гладкость перевода с английского на русский язык выше при переводе аннотаций научных статей (Яндекс.Переводчик – 69, GNMT – 81), что можно объяснить специфической структурой текста аннотации и наличием таких пар текстов при публикации статей в научных журналах. С другой стороны, та же структура присуща аннотациям и на французском языке, однако различия в структуре предложений на английском и французском – синтаксисе – не позволили избежать большого числа грамматических ошибок, отсутствия согласования, а также неправильного выбора части речи системами машинного перевода. В то же время русский язык, характеризующийся свободным порядком слов, предоставляет больше возможностей для выражения смысла.

Этот показатель выше среди текстов публицистического стиля при переводе с английского на французский (GNMT – 74), так как общеупотребительная лексика преобладает над специфическими терминами семантического поля «компьютерные технологии».

По итогам исследования можно сделать вывод о том, что использование нейронных сетей в системах машинного перевода позволило повысить качество решения коммуникативной задачи на уровне целого текста, однако проблема передачи смысла остается открытой, так как в тексте перевода:

- встречаются искаженные и непереведенные слова;
- отсутствует связь между аббревиатурами и их расшифровкой либо эта связь не сохраняется на протяжении всего текста;
- нет единообразия перевода общих и авторских терминов.

Из чего следует, что существует необходимость расширения и непрерывного пополнения базы данных параллельных корпусов.

Гладкость перевода остается насущной проблемой из-за большого количества грамматических, синтаксических и лексических ошибок,

особенно при переводе сложных и объемных предложений, что было выявлено при работе с научными статьями, где научный стиль изложения (сложные конструкции и пунктуация, специфические термины, сокращения и обозначения) доставляет определенные трудности системам перевода.

В дальнейшем исследовании предполагается классифицировать выявленные недочеты перевода текстов, расширить их тематику и продолжить анализ работы систем автоматизированного перевода с целью поиска оптимального набора средств машинного перевода. Однако уже сейчас очевидно, что результаты работы нейронной сети выше, но несовершенны, что может быть компенсировано внедрением в программу перевода мультязычного словаря, где переводчик смог бы уточнить значение интересующих его слов и выражений для получения адекватного перевода.

Литература

1. Карасев И.В., Артюшина Е.А. Системы машинного перевода // Успехи современного естествознания. 2011. № 7. С. 117-118.
2. The mathematics of statistical machine translation: Parameter estimation / P.F. Brown [et al.] // Computational Linguistics. 1993. Vol. 19, No2. P. 263-311.
3. Jehl L. Machine translation for Twitter. Master's thesis. The University of Edinburgh, 2010.
4. Koehn P. Statistical Machine Translation. Cambridge, 2010.
5. Введение в обработку текстов. Лекция 9. ВКМ МГУ. URL: <http://tpc.at.ispras.ru> (дата обращения: 11.01.2019).
6. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. URL: <https://ai.googleblog.com> (дата обращения: 27.01.2019).

A.A. Nankevich

Smolensk State University

QUALITY ESTIMATION OF MACHINE TRANSLATION RESULTS

Keywords: *quality estimation of translated texts, neural networks, statistical machine translation, analysis.*

Abstract. *The article discusses the structure of machine translation systems, and highlights the problem of assessing the quality of machine translation results on the example of two automated translation systems – Yandex.Translate and Google neural machine translate, which are based on different approaches to the implementation of machine translation: hybrid and neural, respectively. The criteria for assessing the quality of translation were fluency and adequacy of translation. Based on the results of comparative analysis of translations of scientific and journalistic texts on the*

topic of general linguistics, parallel programming and computer technologies, conclusions are given regarding the quality of machine translation, and the minimum set of tools required to achieve equivalent translation is determined. The research is at its initial stage, then it is planned to expand the thematic area of the text material, to carry out a deep analysis of errors, to take into account the information obtained while developing a text translation program that uses different API keys.

Е.А. Недорезова

Смоленский государственный университет

УДК 004.912

ОБОСНОВАНИЕ ВЫБОРА МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОЙ ОБРАБОТКИ ТЕКСТОВ С ЧАСТИЧНО ОПРЕДЕЛЕННОЙ СЕМАНТИКОЙ

Ключевые слова: классификация текстов; тексты с частично определенной семантикой; машинное обучение; характеристика текстов; методы классификации; метод опорных векторов; оценка качества классификации.

Предоставлены результаты анализа методов интеллектуальной обработки текстов, проведено сравнение данных методов, указаны преимущества каждого из них по сравнению с другими. Дается определение текстов с частично определенной семантикой, описаны примеры данных текстов, встречаемых в повседневной жизни, приводится их характеристика. Осуществлена оценка качества классификации методом опорных векторов и наивным байесовским классификатором.

Классификация текстов является одной из основных задач информационно-аналитических систем и особенно такого направления, как компьютерная лингвистика. Задачи обработки текстов и методы их решения могут быть различны в зависимости от объемов текстовой информации и степени определенности семантики. В данной статье остановимся на рассмотрении текстов с частично определенной семантикой.

Описание класса задач обработки текстов с частично определенной семантикой. В лингвистике под *семантикой* текста понимается его смысловое значение. *Текстами с частично определенной семантикой* будем называть группу текстов, тематика которых определена (текстам сопоставлен один класс – тема из группы классов),

но не проведена классификация внутри определенной темы. Данные тексты уже заведомо отнесены к некоему классу, и перед нами ставится задача определения подкласса в рамках одной группы текстов. Иными словами, тексты содержат одну и ту же информацию по теме и содержанию, но не проведена классификация внутри одной темы, например, по настроению, стилю высказывания и т.д.

Данные тексты содержат два уровня семантики:

1-й уровень. Предметная область.

2-й уровень. Качественная характеристика текста.

К текстам такого типа можно отнести следующие примеры.

1. Редакционная почта (газета, журнал, любое издательство и т.д.). Тема / класс текстов уже определены заранее. Ставится задача: провести классификацию внутри заданной темы, а именно – разнести письма по рубрикам (будет ли это обращение в рубрику «Спорт», «Кулинария», «Здоровье» и т.д.).

2. Отзывы покупателей о товаре / отзывы пользователей какого-либо интернет-портала (магазин, справочник и т.д.). Необходимо определить настроение сообщения / отзыва от пользователя: пользователь доволен, возникли вопросы или сложности, крайне недоволен (классификация может быть различной в зависимости от потребностей и функционала интернет-портала).

3. Форумы, социальные сети. Ставится задача выявления троллинга в комментариях на форумах и в социальных сетях.

Характеристика сложности выбранного класса задач. Классификацию любого набора текстов можно осуществить по следующим критериям (лингвистическим качествам текста):

1) информационно-структурные качества текста, а именно логичность, точность;

2) тональные, или стилистические характеристики: правильность, чистота речи.

Данные критерии включают большее число качеств, но выше были перечислены только те, которые будут более полезны при классификации текстов методами машинного обучения.

Логичность текста определяет такие его качества, как последовательность в изложении, непротиворечивость мысли. В выбранном классе задач данное требование выполняется, так как тексты уже заранее классифицированы по тематике (определен 1-й уровень семантики). Также реализуется качество точности (предполагает соответствие речи / текста смысловому содержанию).

Выполнение критерия правильности речи будет зависеть от типов текстов (официальный документ, публицистика, сообщения посетителей форума и т.д.). Данное качество может быть как отражено в тексте, так и нет.

Важным для нас является критерий чистоты речи. Чаще всего данный критерий не выполняется, так как речь (и письменная, и устная) может содержать большое число заимствований, профессионализмов.

Выбор и описание методов решения задачи интеллектуальной обработки текстов с частично определенной семантикой. Выбор наилучших методов для решения поставленной задачи будет основан на следующих критериях:

- 1) простота использования;
- 2) удовлетворение потребностей заказчика (максимальная точность результата);
- 3) время обучения и время работы.

Рассмотрим следующие методы классификации текстов, покажем их особенности, достоинства и недостатки, области применения и их применимость для решения задачи обработки текстов с частично определенной семантикой.

Метод опорных векторов (SVM, support vector machine) является методом машинного обучения с учителем (а именно методом линейной классификации) и позволяет решать задачи классификации и регрессионного анализа.

Метод заключается в построении в пространстве объектов оптимальной разделяющей гиперплоскости. Гиперплоскость считается оптимальной, если элементы каждого класса удалены от гиперплоскости достаточно далеко. Метод опорных векторов осуществляет поиск объектов, находящихся на границах между классами, и соотнесение этих объектов с классами. Объекты множества, лежащие на границе областей и называются опорными векторами (см. рис. 1). Классификация считается хорошей, если в области между границами классов нет объектов.

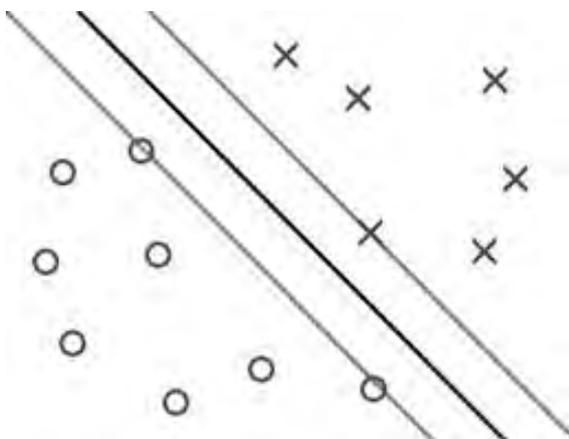


Рис. 1. Опорные векторы

Для случая линейного классификатора имеется одна особенность данного метода, выделяющая его среди других методов классификации,

а именно: если удалить все точки, кроме опорных векторов, то алгоритм метода не изменится. Это свойство делает данный метод уникальным и отличает его от других методов, например, метода k-ближайших соседей, наивного байесовского классификатора и др., где все точки обучаемой выборки используются для оптимизации функции.

Также метод опорных векторов обладает следующим преимуществом: он сводится к задаче выпуклой оптимизации, которая имеет единственное решение, следовательно, удовлетворяет критерию максимальной точности результата.

Недостатком метода является его неустойчивость к шумам в исходных данных (обучающей выборке).

Наивный байесовский классификатор – один из наиболее простых в реализации алгоритмов классификации, и при этом качественно не хуже, чем более сложные методы.

Наивная байесовская модель является вероятностным методом обучения. В многомерной модели текстовый документ – это вектор бинарных атрибутов, показывающих, встретилось ли в документе то или иное слово. Наивный байесовский алгоритм предполагает, что наличие какого-либо признака в классе не связано с наличием какого-либо другого признака.

Исходные данные (обучающую выборку) необходимо для начала преобразовать в частотную таблицу, а затем сформировать таблицу правдоподобия, рассчитав соответствующие вероятности.

Season	Storm
Summer	Yes
Winter	Yes
Autumn	No
Spring	Yes
Winter	No
Summer	No
Autumn	Yes
Summer	No
Spring	Yes

Frequency table		
Season	Yes	No
Winter	1	1
Spring	2	0
Summer	1	2
Autumn	1	1

Likelihood table				
Season	Yes	No		
Winter	1	1	=2/9	0,22
Spring	2	0	=2/9	0,22
Summer	1	2	=3/9	0,33
Autumn	1	1	=2/9	0,22
	=5/9	=4/9		
	0,56	0,44		

(а)
(б)
(в)

Рис. 2. а – исходные данные, б – частотная таблица, в – таблица правдоподобия

Затем по теореме Байеса рассчитывается апостериорная вероятность каждого класса. Класс с наибольшей апостериорной вероятностью будет результатом прогноза или классификации.

Преимущества данного метода:

1) гарантирует легкость реализации и быстроту работы. Наивный байесовский классификатор достаточно быстро обучается, поэтому его можно использовать для обработки текстовой информации в режиме реального времени;

2) превосходит многие другие алгоритмы, так как требует меньшего объема обучающих данных;

3) обеспечивает возможность многоклассовой классификации. Это позволяет прогнозировать вероятности для множества значений переменной.

Недостатки:

1) в случае, если в тестовой выборке присутствует класс, который не встречался в обучающей выборке, алгоритм присвоит нулевую вероятность данному значению («нулевая частота») и не сможет построить прогноз;

2) значения прогноза могут не всегда являться достаточно точными.

Данный метод широко применяется в фильтрации спама, рекомендательных системах, подборе контекстной рекламы, в системах документооборота и многих других.

Сверточная нейронная сеть лучше всего подходит для решения задач компьютерного зрения, распознавания образов и классификации изображений. Алгоритм обладает рядом недостатков, которые будут сильно выражены при использовании данного метода для интеллектуальной обработки текстов с частично определенной семантикой, а именно:

- продолжительное время обучения (в нашем случае это играет важную роль, так как тексты небольшого объема);

- необходимость подбора большого количества примеров для обучения (в сравнении с SVM данный метод проигрывает, так как для SVM не так важно количество примеров, как отсутствие шумов);

- сложность определения количества ядер для каждого слоя.

Генетические и эволюционные алгоритмы имеют широкое применение для решения сложных неформализованных задач, для которых не разработано сериальных методов. Алгоритм может длительное время работать над поиском решения, и не обязательно найденное решение будет оптимальным.

Результаты. Для проверки качества классификации текстов с частично определенной семантикой были выбраны метод опорных векторов и наивный байесовский классификатор. Эксперимент проводился на размеченном корпусе – отзывах покупателей интернет-магазина. Каждому отзыву поставлен в соответствие определенный класс: «хороший», «плохой», «нейтральный».

Оценка качества классификации была проведена по критериям полноты, точности, *F*-меры.

Accuracy (accuracy_score) – доля документов по которым классификатор принял правильное решение, или вероятность того, что класс будет проставлен правильно.

Гораздо более информативными критериями оценки качества классификации являются *точность* (precision) и *полнота* (recall).

Точность класса – это доля документов, действительно принадлежащих данному классу, по отношению к тем документам, которые система отнесла к этому классу. Полнота – это доля найденных документов, принадлежащих классу, относительно всех документов этого класса в тестовой выборке.

Еще одним критерием качества является F -мера (f1-score) – среднее гармоническое точности и полноты.

```

Загружаем обучающие данные из БД...
    считано: 50
Обучение классификатора...
accurasy_score: 0.5

```

	precision	recall	f1-score	support
плохой	0.14	0.50	0.22	2
хороший	0.67	0.25	0.36	8
нейтральный	0.70	0.70	0.70	10
avg / total	0.63	0.50	0.52	20

Рис. 3. Оценка качества классификации наивным байесовским классификатором

```

Загружаем обучающие данные из БД...
    считано: 50
Обучение классификатора...
accurasy_score: 0.48

```

	precision	recall	f1-score	support
плохой	0.33	0.12	0.18	8
хороший	0.60	0.33	0.43	9
нейтральный	0.47	1.00	0.64	8
avg / total	0.47	0.48	0.42	25

Рис. 4. Оценка качества классификации методом опорных векторов

Параметр *Accuracy* у обоих методов почти одинаковый. Параметры *precision* и *recall* показывают лучший результат при использовании метода наивного Байеса. Также и F -мера показывает лучший результат при данном методе.

Приведенные выше результаты показывают, что наивный байесовский классификатор обеспечивает лучшее качество классификации по сравнению с методом опорных векторов. Но оба

метода хорошо применимы для решения поставленной задачи и по многим параметрам обходят более сложные методы.

Рассмотренные выше примеры алгоритмов машинного обучения и области их применения показывают, что для решения поставленной задачи могут использоваться различные методы, которые отличаются друг от друга как временем работы, обучения, так и точностью результатов.

В нашем случае для решения задачи интеллектуальной обработки текстов с частично определенной семантикой не обязательно применение сложных методов, требующих длительного времени обучения, и эффективнее оказываются методы, предоставляющие точный результат, работая более простыми целевыми функциями.

Литература

1. Батура Т. В. Методы автоматической классификации текстов // Программные продукты и системы. 2017. Т. 30, №. 1.
2. Валгина Н. С. Теория текста. М.: Логос, 2003. С. 81.
3. Дхиллон Д. Использование языка Python, пакета SciKit и средств классификации текста для профилирования поведения клиентов. Copyright IBM Corporation, 2014. 10 с.
4. Лифшиц Ю. Автоматическая классификация текстов // Алгоритмы для Интернета. 2005. 8 с.
5. Учителев Н.В. Классификация текстовой информации с помощью SVM // Информационные технологии и системы. 2013, №1. С. 335-340.

Е.А. Nedorezova
Smolensk State University

JUSTIFICATION OF THE CHOICE OF METHODS OF INTELLECTUAL PROCESSING OF TEXTS WITH PARTIALLY CERTAIN SEMANTICS

Keywords: *classification of texts; texts with partially certain semantics; machine learning; text characterization; classification methods; support vector machine; grading quality assessment.*

Abstract. *The results of the analysis of the methods of intellectual text processing are presented, the comparison of these methods is carried out, the advantages of each method compared with other methods are indicated. The definition of texts with partially defined semantics is given, examples of these texts occurring in everyday life are given, the characteristics of texts with partially defined semantics are given. The quality assessment of the classification by the support vector method and the Naive Bayes classifier are carried out.*

РАЗРАБОТКА ВЕБ-СЕРВИСА ДЛЯ РЕШЕНИЯ СТАТИСТИЧЕСКИХ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ ЭЛЕКТРОННЫХ ТАБЛИЦ

Ключевые слова: *корпоративные информационные системы, веб-сервис, MVC-технология, Google Sheets, уравнение линейной регрессии.*

В статье рассматривается способ конструирования инструментальных средств для решения статистических задач с использованием электронных таблиц. В качестве средства разработки таких средств использовались веб-сервис ASP.NET MVC и табличный процессор Google Sheets. Приведено описание эксперимента, в ходе которого была решена задача поиска коэффициентов уравнения линейной регрессии. Обоснованы преимущества предложенного подхода.

Введение. В статье рассматривается способ конструирования инструментальных средств для решения статистических задач с использованием электронных таблиц. Статистический анализ играет важную роль в корпоративных информационных системах (КИС), в которые должны включаться инструментальные средства для накопления информации и принятия решений [1]. В частности, в [1] указано, что КИС возможно создать только в том случае, если имеются основания для обобщения и анализа данных. Стоит отметить, что под КИС [2] подразумеваются системы, в которых функционально объединяются системы для решения задач автоматизации учета и управления производством, финансами, снабжением и сбытом, кадрами и информационными ресурсами. Из определения КИС следует, что они автоматизируют бизнес-процессы принятия стратегических решений, что можно рассматривать как статистическую задачу. В силу автоматизирования КИС такие задачи решаются с помощью современных вычислительных средств.

Описание использованных программных средств. Для создания инструментальной системы был использован табличный процессор Google Sheets. Согласно [3], электронная таблица или табличный процессор – прикладная программа, служащая для обработки данных, представленных в виде двумерной таблицы. Согласно описанию (см. [4]), данный сервис предоставляет стандартные функции электронных таблиц, такие как возможность использования формул, функций,

построение графиков и диаграмм, а также форматирование данных. В отличие от более распространенных электронных таблиц типа Microsoft Excel или OpenOffice Calc, Google – таблицы бесплатны, не требуют обязательной установки и позволяют работать с данными объемом до 5 млн. ячеек [5].

Посредником между пользователями и табличным процессором выступает веб-сервис, имеющий архитектуру Model-View-Controller, – ASP.NET MVC. Применение средств Microsoft было возможно для проведения эксперимента, хотя многие современные технологии (например, Java [6]) поддерживают MVC. Открытость современных СУБД (например, [7] или [8]) позволяет разрабатывать подобные системы с использованием открытых технологий.

Архитектура приложения. Архитектура приложения представлена на рисунке 1. В его основе лежит база данных задач, хранящаяся совместно с веб-сервисом. Работу с базой данных задач ведет системный аналитик, в обязанности которого входит работа с таблицей задач базы данных. Главная функция базы данных задач – хранение формул, решающих ту или иную задачу, в строковом виде. Основная проблема заключается в том, что необходимо перенести эти формулы в табличный процессор с пользовательскими данными, неизвестными заранее. Поэтому основная цель веб-сервиса – преобразование формул в зависимости от пользовательского запроса. Описываемый веб-сервис по запросу пользователя формирует базу данных результатов, хранящуюся в облаке веб-сервиса. Для связи приложения с табличным процессором используются Google Sheets API, позволяющие работать с содержимым и оформлением данных [9]. Пользователь заходит на страницу выбора решаемой задачи, выбирает нужную ему задачу и отправляет приложению запрос на обработку нужного ему объема данных. Специальный метод контроллера обращается к базе данных задач и подставляет в выбранную пользователем формулу объем данных. Далее преобразованная формула отправляется в табличный процессор и помещается в ячейку сразу после ячеек, в которые будут введены пользовательские данные. В результате пользователь переходит по ссылке к таблице с формулой, вводит в нее свои данные и получает результат.

В качестве примера задачи для сервиса взят поиск коэффициентов для уравнения линейной регрессии [10].

Описание эксперимента. Пусть пользователь веб-сервиса имеет по 5 значений независимых и зависимых переменных и хочет найти по этим данным коэффициенты уравнения линейной регрессии. Он выбирает на сервисе нужную задачу и вводит объем необходимых данных для обработки (в данном случае – 5).



Рис. 1. Архитектура приложения

Веб-сервис считывает переменную – необходимый объем данных и переходит к основной процедуре контроллера, содержащей несколько методов. Первый метод отправляет запрос на создание экземпляра электронной таблицы сервису Таблиц Google. Если таблица создана успешно, то результат запроса на создание файла таблицы конвертируется с помощью метода `JsonConvert.SerializeObject`, превращающего объект (в данном случае – результат запроса к табличному процессору) в строку JSON. После этого метод получает идентификатор таблицы и ссылку на нее для последующей передачи пользователю.

Следующий метод вставляет в созданную электронную таблицу формулу, необходимую для решения задачи. Таблица формул `FormTable` состоит из трех полей: код задания `CodeTask`, его расшифровка `TaskName` и формула для задачи `Formula`. Для облегчения работы с базой данных формул для приложения создана модель.

Таблица 1

Общий вид таблицы формул `FormTable`

Id	CodeTask	TaskName	Formula
1	EV	Expected value	=SUM(A1:AN)/N
2	SV	Sample variance	=VAR(A1:AN)*N/(N-1)
3	LR	Linear regression	=LINEST(B1:Bn;A1:An)
4	ST	Students'T-test	=TTEST(A1:AN;B1:BN; tails; type)

Сначала метод определяет код решаемой задачи. Затем он формирует переменную–запрос к таблице формул, являющуюся стандартным запросом SQL вида `SELECT * FROM FormTable WHERE CodeTask = {0}`. Далее приложение обращается к модели с запросом, в котором код задания соответствует выбранной задаче. В ответ метод получает формулу общего вида, имеющую для задачи линейной регрессии вид:

$$=LINEST(B1:Bn;A1:An), \quad (1)$$

где n – объем пользовательских данных, B – столбец для зависимой переменной y , A – столбец для независимой переменной x . Далее значение n заменяется на число данных, необходимое пользователю (в данном случае – 5). После этого метод контроллера создает экземпляр класса $List<T>$, являющийся формулой, которая будет отправлена в таблицу. Далее метод подставляет объект с формулой в последнюю строку диапазона ячеек и, используя идентификатор посылает сервису таблиц Google запрос на обновление, который отправляется на сервис.

Последний метод контроллера служит для передачи ссылки на созданную таблицу в представление. После перехода по сгенерированной ссылке пользователь получает таблицу следующего вида.

Таблица 2

Готовая таблица с формулой

...	...
=ЛИНЕЙН(B1:B5;A1:A5)	

Таблица 3

Готовая таблица с пользовательскими данными

3,54	10,8
4,68	17,6
4,07	12,7
3,84	13
3,79	13,3

Выводы. Предложенный подход обеспечивает параллельную работу нескольких пользователей, поскольку этот способ автоматизирует работу системных аналитиков (разработчиков) и пользователей КИС, потому что:

- во-первых, современные СУБД работают при выполнении запросов параллельно;
- во-вторых, реализующий запрос метод контроллера при независимых обращениях выполняются параллельно.

Также использование MVC-технологии делает предложенный подход более универсальным.

Литература

1. Новикова Г.М. Корпоративные информационные системы: учеб. пособие. – М.: РУДН, 2008. 94 с.
2. Корпоративные информационные системы: учеб. пособие / В.А. Погонин [и др.]. Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2012. 144 с.
3. American Heritage Dictionary of the English Language (5th ed.). Houghton Mifflin Harcourt, 2011. 2084 p.
4. Google Sheets. URL: https://support.google.com/docs/topic/9054603?hl=ru&ref_topic=1382883 (дата обращения: 16.02.2019).

5. Google Drive Help. URL: <https://support.google.com/drive/answer/37603> (дата обращения: 17.02.2019).

6. Spring Framework Documentation. URL: <https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html> (дата обращения: 17.02.2019).

7. PostgreSQL. URL: <https://www.postgresql.org/about/> (дата обращения: 01.03.2019).

8. MySQL 8.0 Reference Manual. URL: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html> (дата обращения: 02.03.2019).

9. Comprehensive access to spreadsheets. URL: <https://developers.google.com/sheets/api/> (дата обращения: 17.02.2019).

10. David A. Freedman. Statistical Models: Theory and Practice. Revised Edition. Cambridge University Press, 2009. 456 p.

E.I. Parfenova, V.I. Usachev
Smolensk State University

DEVELOPMENT OF WEB SERVICE FOR STATISTICS TASKS SOLVING WITH USING OF SPREADSHEETS

Keywords: *enterprise information system; web service; MVC pattern; Google Sheets; equation of linear regression.*

Abstract. *The variant of creation the program for statistics tasks solving with the using of spreadsheets is considered. Web service ASP.NET MVS and spreadsheet program Google Sheets are used as the programming tool for creation of this program. An experiment in which the coefficients of linear regression equation were found was held. The rationale for the benefits of the proposed approach has been made.*

Е.Н. Петров, Л.Г. Гагарина
НИУ «МИЭТ»

УДК 004.852

РАЗРАБОТКА МЕТОДИКИ КЛАССИФИКАЦИИ НЕСТРУКТУРИРОВАННОЙ ИНФОРМАЦИИ НА ПРИМЕРЕ БИБЛИОГРАФИЧЕСКИХ ДАННЫХ С ПОМОЩЬЮ УСЛОВНО- СЛУЧАЙНЫХ ПОЛЕЙ

Ключевые слова: *классификация, условно-случайные поля, неструктурированные данные, библиографические данные, логарифмические линейные функции.*

В статье рассматривается разработка методики классификации неструктурированной информации на примере библиографических данных с помощью условно-случайных полей. Формализуется задача классификации библиографических данных с помощью условно-случайных полей. На основе существующих стандартов библиографических данных выбираются метки и разрабатывается признаковое представление, позволяющее с помощью разработанной методики соотносить данные с соответствующими им метками. Проводится программная реализация разработанной методики, и в ходе эксперимента подтверждается ее эффективность при решении задачи обработки библиографических данных.

Неструктурированные данные, представляющие собой информацию, не имеющую заранее определенной структуры, как правило представляются в форме текста на естественном языке и содержат даты, цифры и факты. Такая организация, а точнее ее отсутствие, затрудняет анализ данных с помощью традиционных программ и требует больших трудо- и времязатрат для ручного структурирования оператором. Рутинный труд по преобразованию данных подвержен влиянию человеческого фактора, сказывающегося на качестве выполняемой работы и ее эффективности.

По разным оценкам, от 70 до 90% всей потенциально полезной деловой информации представлено в неструктурированной форме, что, с учетом взрывного роста объемов информации за последние годы, снижает возможности ее эффективной обработки и анализа.

Одной из областей, использующих неструктурированную информацию, являются библиографические данные, которые необходимо обрабатывать в библиотеках, вузах, наукоемких предприятиях и других организациях, связанных с издательской или научной деятельностью.

Отчеты о научной деятельности организаций, научные статьи, учебные издания, отчеты по научно-исследовательской работе, научно-квалификационные и выпускные квалификационные работы содержат большие объемы библиографической информации, которая может достигать ста и более позиций. При ручном оформлении этих данных согласно принятым стандартам, наиболее распространенным из которых является ГОСТ 7.0.5-2008, зачастую встречаются неточности: пропуск опциональных и обязательных полей библиографического описания, нарушение порядка следования и недостаток или избыток пунктуационных символов. Указанные отступления от стандартов усложняют поиск и определение использованных источников, затрудняя научную и отчетную работу.

Очевидно, что ручная обработка большого объема библиографических данных является неэффективной как по времени, так и по качеству; с другой стороны, выполнение данной работы компьютером гарантирует отсутствие ошибок по невнимательности, универсальность обработки различных форматов данных, структуризацию и быстроту обработки и, как следствие, своевременную отчетность.

Таким образом, является актуальной задача классификации библиографических данных с помощью компьютера, которая заключается в переводе этих данных из различных форматов в единое представление.

Для решения описанной проблемы необходимо формализовать задачу классификации библиографических данных с помощью условно-случайных полей, разработать набор меток и признаковое представление. На основе выполненной формализации необходимо разработать алгоритм и методику классификации и подтвердить ее эффективность с помощью эксперимента. Прделанная работа позволит использовать разработанный алгоритм при создании программных средств для повышения точности обработки неструктурированной информации на примере библиографических записей.

Разработанное формализованное представление задачи классификации библиографических данных с помощью условно-случайных полей формулируется следующим образом.

Пусть Y – множество меток классов, X – множество слов. Вероятность того, что последовательность меток $y \in Y_n$ длины n верно классифицирует последовательность слов $x \in X_n$ в составе библиографической записи, вычисляется по формуле:

$$p(y|x;w) = \frac{\exp \sum_{j=1}^J w_j F_j(x,y)}{Z(x,w)},$$

$$Z(x,w) = \sum_{y \in Y} \exp \sum_{j=1}^J w_j F_j(x,y),$$

где $F_j(x,y)$ – функция признака j , отображение $F_j : X^n \times Y^n \rightarrow \{0, 1\}$, w_j – весовой коэффициент функции j , а $Z(x, w)$ – коэффициент нормировки; тогда задача классификации библиографических записей сводится к нахождению такой последовательности меток, что:

$$\tilde{y} = \operatorname{argmax}_y p(y|x;w) = \operatorname{argmax}_y \sum_{j=1}^J w_j F_j(x,y).$$

В ходе разработки методики сформулированы рекомендации по подбору данных, их подготовке и нормализации; выделены ключевые

классы для задачи классификации библиографических данных, а также разработано их признаковое представление; определен способ оценки классификатора с помощью измерения его точности, полноты и F-меры.

Разработанная методика классификации библиографических данных на основе условно-случайных полей формулируется следующим образом.

Этап 1: подбор данных для обучения модели.

Этап 2: подготовка и нормализация данных.

Этап 3: выделение признаков классов и их функций, обучение модели с помощью метода условно-случайных полей.

Этап 4: проверка адекватности обучения модели с помощью измерения точности, полноты и F-меры классификации.

Этап 5: использование модели.

В работе подбор данных для обучения модели осуществлен на основе публикаций авторов кафедры «Информатика и программное обеспечение вычислительных систем» НИУ «МИЭТ» за последние пять лет.

Подготовка библиографических данных включает в себя их форматирование, очистку и семплирование.

Форматирование является распространенной операцией при подготовке данных к работе и обычно состоит в переводе записей из различных источников (текстовых документов, таблиц, веб-страниц) в единый текстовый вид.

Очистка данных заключается в удалении служебных пометок к библиографическим записям, которые добавлены по правилам оформления первичных источников (веб-страницы библиографических баз, отчеты о научной деятельности), но непосредственно к библиографическим данным не относятся.

Семплирование данных необходимо для избавления от дубликатов и извлечения из общего массива данных обучающей и тестовой выборки.

На основе аналитического обзора основных стандартов представления библиографических данных для работы выбран следующий набор меток: автор статьи, разделитель, название журнала / конференции, город издательства, число страниц, название издательства, название статьи, том, год.

В ходе исследования тестовой выборки в соответствии с выбранными метками сформирован следующий набор функций признаков:

a. признак начала записи,

b. признак конца записи,

c. признак первой заглавной буквы для текущего элемента,

d. признак буквы для текущего элемента,

e. признак цифры для текущего элемента,

f.признак совпадения со строкой “//” для текущего элемента,
g.признак окончания слова на точку для текущего элемента,
h.признак слова в верхнем регистре для текущего элемента,
i.признак присутствия в слове символа “№” для текущего элемента,
j.признак первой заглавной буквы для следующего элемента,
k.признак буквы для следующего элемента,
l.признак цифры для следующего элемента,
m.признак совпадения со строкой “//” для следующего элемента,
n.признак окончания слова на точку для следующего элемента,
o.признак слова в верхнем регистре для следующего элемента,
p.признак присутствия в слове символа “№” для следующего элемента,
q.признак первой заглавной буквы для предыдущего элемента,
r.признак буквы для предыдущего элемента,
s.признак цифры для предыдущего элемента,
t.признак совпадения со строкой “//” для предыдущего элемента,
u.признак окончания слова на точку для предыдущего элемента,
v.признак слова в верхнем регистре для предыдущего элемента,
w.признак присутствия в слове символа “№” для предыдущего элемента.

При этом признаки j-q) вычисляются в том случае, если признак начала записи текущего элемента равен нулю. Признаки p-w) вычисляются, если признак конца записи для текущего элемента равен нулю.

Согласно разработанной методике, адекватность обучения модели проверяется с помощью точности, полноты и F-меры, вычисляемых на основании истинно-положительных (TP), ложно-положительных (FP) и ложно-отрицательных (FN) результатов классификации тестовых данных с помощью обученной модели.

$$\text{Точность: } \frac{TP}{TP + FP}$$

$$\text{Полнота: } \frac{TP}{TP + FN}$$

$$F_1 = 2 \frac{\text{Точность} \cdot \text{Полнота}}{\text{Точность} + \text{Полнота}}$$

Использование этих метрик позволяет оценить долю верно распознанных элементов класса по отношению ко всем элементам, отнесенным к этому классу, долю верно распознанных элементов определенного класса, а также их среднее гармоническое.

На основе данной методики разработан алгоритм классификации библиографических данных, использующий рекомендации и признаковое представление для эффективной обработки.

Алгоритм состоит из следующих основных частей:

- предобработка данных (выбор, очистка и форматирование библиографической записи);
- векторизация библиографической записи (вычисление функций признаков для каждого элемента библиографической записи);
- передача векторизованного представления библиографической записи в обученную модель и классификация;
- группировка классифицированных элементов (формирование последовательностей из элементов одного класса).

Рассмотрена программная реализация разработанной методики и алгоритма в виде программного комплекса классификации библиографических данных. В ходе вычислительного эксперимента, включающего в себя оценку результатов классификации по формальным критериям, отобраным в рамках разработанной методики, получены значения точности, полноты и F-меры для каждого из рассматриваемых классов (табл. 1).

Таблица 1

Точность, полнота и F-мера классификатора

Метка	Точность	Полнота	F-мера
T_AUTHOR	1.000	1.000	1.000
T_DELIMITER	0.938	1.000	0.968
T_JOURNAL	0.940	0.959	0.949
T_LOCATION	0.800	0.800	0.800
T_PAGES	1.000	1.000	1.000
T_PUBLISHER	1.000	0.500	0.667
T_TITLE	1.000	1.000	1.000
T_TJSEP	1.000	1.000	1.000
T_VOLUME	1.000	0.895	0.944
T_YEAR	0.800	0.842	0.821
Среднее	0.969	0.968	0.967

Проведена оценка практического применения результатов разработки и оценка перспектив дальнейших исследований. Результаты исследования позволяют выполнять обработку и структуризацию библиографических данных на русском языке без использования явно

заданного формата, опираясь только на признаки отдельных структурных компонентов библиографической записи, а не на их последовательность или структуру записи в целом.

Такой подход к обработке позволяет осуществлять перевод библиографических данных из одного представления или формата в другой с минимальным участием оператора, в том числе исправлять ошибки форматирования и следования структурных компонентов в рамках одного стандарта оформления.

Литература

1. Lafferty J., McCallum A., Pereira F. Conditional random fields: probabilistic models for segmenting and labeling sequence data // International Conference on Machine Learning, 2001.
2. Sha F., Pereira F. Shallow parsing with conditional random fields // Proceedings of Human Language Technology. NAACL, 2003.
3. Wallach H.M. Efficient training of conditional random fields. Master's thesis, University of Edinburgh, 2002.
4. Sutton C., McCallum A., Rohanimanesh K. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. Journal of Machine Learning Research. 2008. Vol. 8. P. 693–723.

E.N. Petrov, L.G. Gagarina

National Research University of Electronic Technology – MIET

DEVELOPMENT OF CRF-BASED CLASSIFICATION OF UNSTRUCTURED DATA IN THE FIELD OF BIBLIOGRAPHIC INFORMATION

Keywords: *classification, CRF, unstructured data, bibliographic data, log-linear functions.*

Abstract. *In this article we consider the classification problem of unstructured data. One of possible solutions could be a development of a new CRF-based method for automated data classification. The article includes formalization of the problem, label selection based on contemporary bibliographical standards and feature functions development, which is the core element of the method development. Software prototype which implements the method is developed to verify the efficiency.*

АЛГОРИТМЫ АППРОКСИМАЦИИ НЕЯВНО ЗАДАННЫХ ОБЛАСТЕЙ¹

Ключевые слова: *системы уравнений, системы неравенств, метод неравномерных покрытий, аппроксимация множества решений.*

В статье рассматривается проблема построения аппроксимации множеств, задаваемых системой ограничений. Отдельно исследуются случаи систем уравнений, систем неравенств и смешанных систем. Предлагается подход, позволяющий строить внешнюю аппроксимацию множеств решений таких систем. Для случая системы неравенств также строится внутренняя аппроксимация. Приводится описание алгоритмов и утверждений, обосновывающих корректность получаемых аппроксимаций.

Многие задачи на практике сводятся к аппроксимации множества, задаваемого неявно с помощью систем ограничений. В качестве ограничений могут рассматриваться уравнения и неравенства. Одним из самых известных и наиболее удачных применений подобного рода систем является определение рабочей области роботов параллельной структуры [1]. В работах [2–6] представлены различные варианты постановок и способов решения этой задачи.

В нашей работе даны формальные описания алгоритмов аппроксимации рабочей области множества решений систем неравенств и систем уравнений. Также предложен гибридный подход, позволяющий получать множество решений системы, в которую входят как уравнения, так и неравенства.

Аппроксимация множества решений системы уравнений.
Рассмотрим систему уравнений общего вида:

$$\begin{aligned} g_i(x) &= 0, i = 1, \dots, m, \\ x &\in [a, b], \end{aligned} \tag{1}$$

где $g_i(x): R^n \rightarrow R$ – непрерывные отображения, а $[a, b] = \{x \in R^n: a_i \leq x_i \leq b_i, i = 1, \dots, n\}$ – n -мерный параллелепипед, задающий интервалы изменения параметров. Пусть X – множество всех решений системы (1).

¹ Работа выполнена при частичной финансовой поддержке РФФИ, проект 17-07-00510.

Для нахождения аппроксимации множества X применяется приведенный ниже алгоритм CovEq, результатом работы которого будет список параллелепипедов A , образующих аппроксимацию.

Алгоритм CovEq

$L := \{[a, b]\}$

while $L \neq \emptyset$ do

 take $P \in L, L := L \setminus \{P\}$

$m := \max_{i=1, \dots, m} \min_{x \in P} g_i(x), M := \min_{i=1, \dots, m} \max_{x \in P} g_i(x)$

 if $m \leq 0 \leq M$ then

 if $d(P) \leq \delta$ then

$A := A \cup \{P\}$

 else

$\{P_1, P_2\} = \text{divide } P$

$L := L \cup \{P_1, P_2\}$

 endif

 endif

done

Принцип работы алгоритм CovEq состоит в том, что исходный параллелепипед делится на параллелепипеды меньшего размера до тех пор, пока диаметр параллелепипеда не станет меньше заданной величины δ . Обрабатываемые параллелепипеды заносятся в список L . Деление происходит всегда вдоль максимального ребра, что обеспечивает конечность алгоритма. При этом параллелепипеды, заведомо не содержащие точек из множества решений системы (5), отбрасываются, для чего выполняется сравнение величин $m := \max_{i=1, \dots, m} \min_{x \in P} g_i(x), M := \min_{i=1, \dots, m} \max_{x \in P} g_i(x)$ с нулем. Если $m > 0$ или $M < 0$, то параллелепипед P не содержит ни одной точки из X и может быть исключен из дальнейшего рассмотрения.

К сожалению, точное нахождение величин m, M возможно только в относительно простых случаях. В большинстве случаев используются оценки \hat{m}, \hat{M} : $\hat{m} \leq m \leq M \leq \hat{M}$. В качестве примеров можно рассмотреть интервальные [7] или Липшицевы оценки [8]. При этом алгоритм CovEq заменяется алгоритмом CovEqEst, полученным из CovEq заменой величин m, M на \hat{m}, \hat{M} .

По построению отсеву подвергаются только параллелепипеды, не содержащие точек из X , поэтому справедливо следующее утверждение относительно алгоритма CovEqEst.

Утверждение 1. Множество A , построенное в результате работы алгоритма CovEqEst, содержит все множество X : $X \subseteq A$.

Решение системы неравенств. Система неравенств в общем виде имеет следующий вид:

$$\begin{aligned} h_i(x) &\leq 0, i = 1, \dots, k, \\ x &\in [a, b], \end{aligned} \quad (2)$$

где $h_i(x): R^n \rightarrow R$ – непрерывные отображения, а $[a, b] = \{x \in R^n: a_i \leq x_i \leq b_i, i = 1, \dots, n\}$ – n -мерный параллелепипед, задающий интервалы изменения параметров. Пусть X – множество всех решений системы (2).

Систему (2) можно заменить эквивалентной системой:

$$\begin{aligned} \psi(x) &\leq 0, \\ x &\in [a, b], \end{aligned}$$

где $\psi(x) = \max_{i \in \{1, \dots, k\}} h_i(x)$.

Для аппроксимации множества X применяется алгоритм CovIneq, результатом работы которого будут списки параллелепипедов A , B , аппроксимирующих внутренность и границу множества X соответственно.

Алгоритм **CovIneq**

$L := \{[a, b]\}$

while $L \neq \emptyset$ **do**

take $P \in L, L := L \setminus \{P\}$

$z := \min_{x \in P} \psi(x), Z := \max_{x \in P} \psi(x)$

if $Z < 0$ **then**

$A := A \cup \{P\}$

else if $z \leq 0$ **then**

if $d(P) \leq \delta$ **then**

$B := B \cup \{P\}$

else

$\{P_1, P_2\} = \text{divide } P$

$L := L \cup \{P_1, P_2\}$

endif

endif

done

Так же как и в случае равенств, точные значения максимума и минимума можно заменить на оценки \hat{z}, \hat{Z} : $\hat{z} \leq z \leq Z \leq \hat{Z}$. Аналогично алгоритм CovIneq заменяется алгоритмом CovIneqEst, полученным из CovIneq заменой величин z, Z на \hat{z}, \hat{Z} . Справедливо следующее утверждение относительно алгоритма CovIneqEst.

Утверждение 2. Для множеств A и B , построенных в результате работы алгоритма *CovIneqEst*, выполняются следующие соотношения:

$$\begin{aligned} A \subseteq \text{Int } X \subseteq X \subseteq A \cup B, \\ \partial X \subseteq B. \end{aligned}$$

Решение смешанной системы. Смешанная система равенств и неравенств в общем виде имеет следующий вид:

$$\begin{aligned} h_i(x) \leq 0, i = 1, \dots, k, \\ g_j(x) = 0, j = 1, \dots, m, \\ x \in [a, b], \end{aligned} \quad (3)$$

где $g_i(x): R^n \rightarrow R$, $h_i(x): R^n \rightarrow R$ – непрерывные отображения, а $[a, b] = \{x \in R^n: a_i \leq x_i \leq b_i, i = 1, \dots, n\}$ – n -мерный параллелепипед, задающий интервалы изменения параметров. Пусть X – множество всех решений системы (3).

Систему (3) можно заменить эквивалентной системой:

$$\begin{aligned} \psi(x) \leq 0, \\ g_j(x) = 0, j = 1, \dots, m, \\ x \in [a, b], \end{aligned}$$

где $\psi(x) = \max_{i \in \{1, \dots, k\}} h_i(x)$.

Для нахождения аппроксимации множества X применяется алгоритм *CovAll*, результатом работы которого будет список параллелепипедов A , образующих искомое покрытие.

Алгоритм *CovAll*

$L := \{[a, b]\}$

while $L \neq \emptyset$ **do**

take $P \in L, L := L \setminus \{P\}$

$m := \max_{i=1, \dots, m} \min_{x \in P} g_i(x), M := \min_{i=1, \dots, m} \max_{x \in P} g_i(x), z := \min_{x \in P} \psi(x),$

if $m \leq 0 \leq M$ **and** $z \leq 0$ **then**

if $d(P) \leq \delta$ **then**

$A := A \cup \{P\}$

else

$\{P_1, P_2\} = \text{divide } P$

$L := L \cup \{P_1, P_2\}$

endif

endif

done

В результате работы алгоритма формируется множество A , являющееся внешней аппроксимацией множества X . Заменяя точные значения максимумов и минимумов ограничений m, M, z на соответствующие оценки $\hat{m}, \hat{M}, \hat{z}$, получаем алгоритм CovAllEst, в отношении которого справедливо следующее утверждение.

Утверждение 3. Множество A , построенное в результате работы алгоритма CovAllEst, содержит все множество X : $X \subseteq A$.

Заключение

В работе рассмотрены три алгоритма CovEq, CovIneq и CovAll, предназначенные для построения аппроксимации множества, заданного системой уравнений, неравенств и смешанной системой соответственно. Также рассмотрены варианты, когда не удается точно вычислить минимумы и максимумы ограничений и используются оценки. Приведены утверждения, обосновывающие корректность получаемых аппроксимаций.

При применении указанных алгоритмов на практике, как правило, возникает проблема высокой вычислительной сложности. Эту проблему удается частично преодолеть путем применения методов параллельных вычислений [4].

Литература

1. Merlet J. P. Parallel robots. Springer Science & Business Media, 2006.
2. Анализ рабочей области робота DexTAR-dexterous twin-arm robot / Д.И. Малышев [и др.] // International Journal of Open Information Technologies. 2018. Т. 6, №. 7.
3. Отыскание множеств решений систем нелинейных неравенств / Ю.Г. Евтушенко [и др.] // Журнал вычислительной математики и математической физики. 2017. Т. 57, №. 8. С. 1248-1254.
4. Параллельный алгоритм для аппроксимации рабочей области робота / А.Ю. Горчаков [и др.] // International Journal of Open Information Technologies. 2019. Т. 7, №. 1.
5. Kaloorazi M.H.F., Masouleh M.T., Caro S. Determining the maximal singularity-free circle or sphere of parallel mechanisms using interval analysis // Robotica. 2016. Т. 34, №. 1. С. 135-149.
6. Merlet J.P. On the workspace of suspended cable-driven parallel robots // 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016. С. 841-846.
7. Шарый С.П. Конечномерный интервальный анализ. Институт вычислительных технологий СО РАН. 2016.
8. Sergeyev Y.D., Kvasov D.E. Lipschitz global optimization // Deterministic Global Optimization. Springer, New York, NY, 2017. С. 1-17.

M.A. Posypkin

Federal Research Center «Computer Science and Control» of Russian
Academy of Sciences

APPROXIMATION ALGORITHMS FOR IMPLICITLY DEFINED REGIONS

Keywords: *systems of equations, systems of inequalities, method of non-uniform coverage, approximation of the set of solutions.*

Abstract. *The article deals with the problem of constructing an approximation of sets given by a system of constraints. The cases of systems of equations, systems of inequalities and mixed systems are considered separately. An approach is proposed to construct an external approximation of the sets of solutions of such systems. For the case of a system of inequalities, an internal approximation is also constructed. The description of algorithms and statements justifying the correctness of the approximations are given.*

Т.А. Самойлова, А.И. Павлюков

Смоленский государственный университет

УДК 004.85

НЕЙРОУПРАВЛЕНИЕ ДВИЖЕНИЕМ В РЕАЛЬНОМ ВРЕМЕНИ

Ключевые слова: *нейроуправление движением, реальное время, анимация, фазово-функциональная нейронная сеть, многомерный массив, Python, C#, Theano.*

Предложена программная реализация нейросетевого подхода к задаче управления движением в реальном времени. Рассмотрен относительно новый эффективный алгоритм обучения нейронной сети, использующий базу данных, содержащую рельеф местности и описание движений объекта. Программирование алгоритма выполнено средствами языков Python, c# и модулей библиотек Theano и cuDNN, дающих возможность эффективно быстро определять, оптимизировать и оценивать многомерные массивы. Предложенный алгоритм обучения управляет объектом в реальном времени, требует небольшого объема памяти и обеспечивает высокое качество движения.

В настоящее время все больше внимания уделяется решению проблем построения систем управления динамическими объектами в

реальном времени на основе нейронных сетей. Такие системы находят идеальное применение в компьютерной анимации, робототехнике, производственной, авиационно-космической и деловой сферах. Известно, что необходимым условием использования традиционных методов оптимального и адаптивного управления в этих областях является наличие большого объема априорной информации об объекте управления, например результатов математического моделирования. Благодаря способности нейронных сетей к самообучению, им такой объем информации в ходе управления не требуется. Нейросетевые модели позволяют выявить сложную нелинейную зависимость между входом и выходом системы управления на основании предварительного обучения, используя наборы больших данных.

В ходе развития методов нейроуправления стали применяться известные способы построения нейронных сетей [1; 2]. Ключевая проблема их применения в задачах управления – невысокое качество, неточное обучение сети и требование значительных вычислительных ресурсов с продолжительным временем обработки. Для решения этой проблемы предлагается использовать новую архитектуру нейронной сети, созданную исследователями из Эдинбургского университета, которая носит название «Нейронная сеть с фазовым функционированием» (Phase-Functioned Neural Network – PFNN)[3]. Машинное обучение такой нейронной сети выполнялось на данных управления анимацией персонажей в среде Python 3.6 средствами пакетов Numpy / Scipy / Theano/ Pillow/ scipy/ CUDA/ cuDNN. Управление непосредственно движением объектов реализовано в среде Unity средствами языка C# на основе предварительно обученной сети. Работа программной системы состоит из трёх этапов: стадии предварительной обработки, стадии обучения фаз нейронной сети и стадии выполнения. На этапе предварительной обработки осуществляется сбор данных о местности, где будет происходить движение, формируется карта высот рельефа местности, которая сохраняется в базе данных. На втором этапе PFNN обучается с использованием этих данных, создавая регрессионную модель прогнозирования движения объекта. На этапе выполнения формируется необходимая для управления движением информация, которая непосредственно вводится в обученную регрессионную модель.

Предварительная обработка данных. Чтобы создать систему управления, где персонаж автоматически адаптируется к геометрии среды, были подготовлены данные для обучения PFNN, включающие информацию о перемещении персонажа на разных участках местности [4]. При этом использовалась открытая база данных, созданная с использованием технологии 3D-моделирования Motion Capture

(MoCap – захват движения) [5] для записи движений человека. Процесс захвата движений показан на рисунке 1.

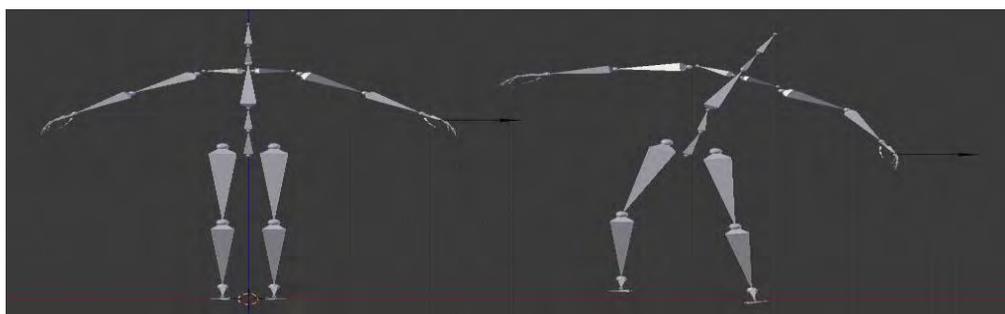


Рис.1. Процесс захвата движений

Средствами среды unity этим данным ставится в соответствие карта высот местности [6] с разрешением одного пикселя на дюйм. Из карты высот произвольно выбираются ориентации и местоположения для приблизительно 20 000 патчей площадью 3х3 метра. Процесс подготовки геометрии происходит в два периода. В первом периоде для каждой анимации цикла в этом движении путем линейного поиска находится 10 наиболее подходящих патчей из базы данных, выполняется подгонка каждого патча и выбор тех, которые минимизируют функцию ошибки. Во втором периоде средствами радиальной базисной функции с линейным ядром результат уточняется, определяя ландшафт так, чтобы ноги персонажа находились точно на земле во время контакта. С помощью процесса подгонки происходит редактирование карты высот таким образом, чтобы ноги персонажа касались земли. Общая обработка данных и время подбора для всей базы данных движения составляет около 4 часов на процессоре Intel i5-4700 с частотой 3,4 ГГц.

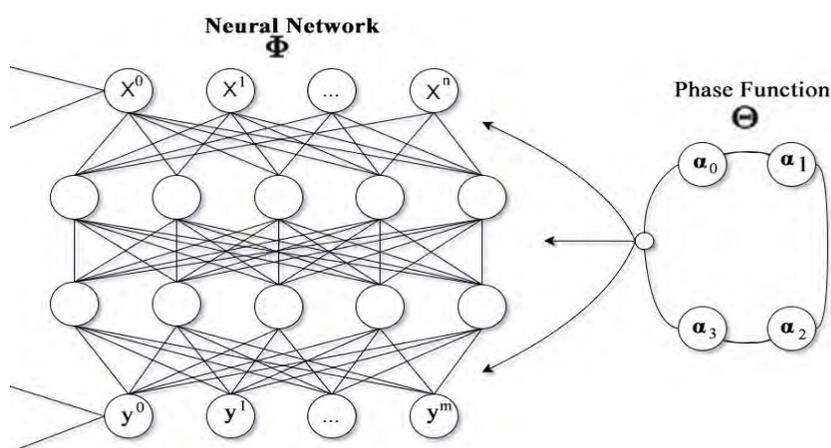


Рис. 2. Архитектура PFNN

Обучение нейронной сети. Нейронная сеть (PFNN) работает по принципу генерации в каждый момент времени многомерного массива – весов синаптических связей, зависящих от фазы – переменной, представляющей синхронизацию цикла движения. Однажды

сгенерированные, веса сети используются для выполнения регрессии от контрольных параметров в этом кадре до соответствующего положения тела. Ключевым моментом было использование авторегрессионной модели, где поза прогнозируется [7] на основе предыдущих поз. Эта модель подходит для работы с реальным временем, таким как дроны к примеру, в котором поза обновляется почти в каждом кадре. На рисунке 2 представлена архитектура PFNN, где веса вычисляются на основании периодической функции фазы p , называемой фазовой функцией.

Входные данные x содержат информацию о текущей позе объекта. Выходы y – это изменённые фазы, соответствующие следующей позе персонажа в новом кадре (кадр – это отдельный фрагмент действия). В PFNN сетевые веса α вычисляются в каждом кадре посредством отдельной функции, называемой фазовой функцией, которая принимает в качестве входной фазы p и параметров β следующие: $\alpha = \Theta(p; \beta)$. Θ обозначается как кубический сплайн, он позволяет начальным и конечным контрольным точкам с одинаковым количеством параметров (пропорционально числу контрольных точек) плавно изменяться по отношению к входному параметру p . Каждая контрольная точка α_k представляет собой определенную конфигурацию весов для нейронной сети α , а функция Θ выполняет плавную интерполяцию между этими нейронными весами конфигурации.

Во время обучения нейронной сети для каждого кадра i переменные x_i, y_i и фазы p_i помещаются в матрицы X, Y и P . Используются средние значения x_{μ}, y_{μ} и стандартные отклонения x_{σ}, y_{σ} с целью нормализации данных, которые затем дополнительно масштабируются весами x_w, y_w , дающие относительные значения каждого измерения [8]. Во время обучения необходимо обеспечить, чтобы многомерный массив включал управляющие параметры X и фазовые параметры P , участвующие в вычислении выходных переменных Y как функции. Поэтому обучение – это проблема оптимизации многомерного массива в отношении фазовой функции $\beta = \{\alpha_0 \alpha_1 \alpha_2 \alpha_3\}$.

Процесс обучения проводится в течение 20 эпох и занимает около 30 часов на графическом процессоре NVIDIA GeForce GTX 670. Результат обучения – значения сетевых синаптических весов, сохраненных в виде простых двоичных файлов.

Этап выполнения. Каждый кадр – фрагмент движения – включает формирование данных, необходимых для управления движением и собственно управление, когда эти данные посылаются с фазой p на вход модели обученной нейронной сети. Фаза p может быть сохранена или увеличена с использованием вычисленного изменения фазы, модулированного для цикла в диапазоне $0 \leq p \leq 2\pi$. Результат из предыдущего кадра берется в качестве входных данных для

следующего, образуя прошлые / будущие траектории $t^p t^d$. Задание будущего элемента траектории вычисляется путём смешивания [9] двух траекторий: указанного пользователем направления движения на площадке и траектории, сгенерированной PFNN в предыдущем кадре. Таким образом, скорость и направление движения смешиваются в каждом будущем кадре со скоростью и направлением движения, предсказанными PFNN в предыдущем кадре (t_{i+1}^p, t_{i+1}^d) . Это смешивание реализуется командой $TrajectoryBlend(a_0, a_1, t, \tau) = (1 - t^\tau)a_0 + t^\tau a_1$. Здесь значение t колеблется от 0 до 1, когда траектория выходит за рамки, а τ представляет собой дополнительное смещение, контролирующее отзывчивость персонажа. Также устанавливается желаемая походка персонажа (представленная в виде двоичного вектора) и другая информация, например, высота потолка и места, где надо прыгать, а не забираться. Эти данные задаются либо в результате взаимодействия с пользователем (так, мы используем правую кнопку на клавиатуре, чтобы указать, что походка должна переключиться на пробежку), либо путем проверки расположения траектории относительно элементов окружающей среды (например, когда траектория проходит через определенную область, указывающую, что прыжковое движение активировано).

После того как переменные, относящиеся к будущей траектории, заданы, реализуется последний шаг – проецирование положения траектории вертикально на геометрию сцены и извлечение высоты для подготовки t^h . Это и создаёт показанное на рисунке 3 движение персонажа в реальном времени, которое выглядит естественным, но остается отзывчивым, так как воспринимаемая отзывчивость возникает, когда персонаж быстро реагирует на изменения в желаемом направлении.

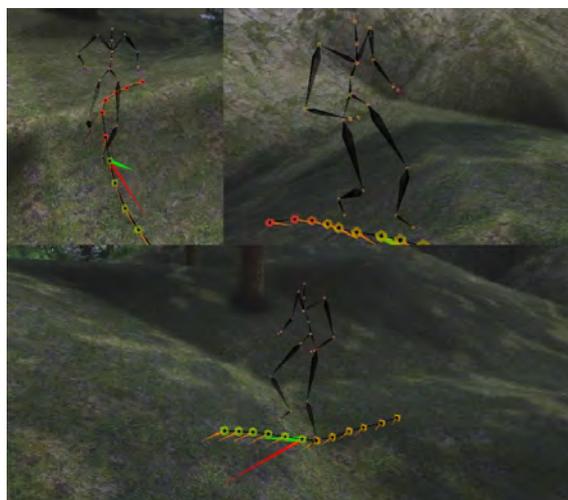


Рис. 3. Движение персонажа в реальном времени

Заключение. Предложена программная реализация нейросетевого алгоритма управления движением в реальном времени, основанного на обучении фазово-функциональной нейронной сети. Преимуществом является динамическое изменение веса сети в зависимости от фазы, вместо того чтобы сохранить их статичными, как в стандартных нейронных сетях; тем самым значительно увеличивается выразительность регрессии, при этом сохраняется компактная структура. Это позволяет сети учиться на большом объеме данных, где геометрия окружающей среды и данные о движении человека связаны. Несмотря на свою компактную структуру, сеть может учиться на большом объеме данных благодаря многомерному массиву, который плавно изменяется с течением времени для создания большого разнообразия сетевых конфигураций.

После обучения система чрезвычайно быстрая и компактная, требует только миллисекунд времени на просчет входных данных и задействует несколько мегабайт памяти, обеспечивая при этом высокую точность движения. Предложенный подход показал свою эффективность и может быть использован для построения систем управления движением в реальном времени. В дальнейшем на основании этого подхода планируется решать задачи управления робототехническими объектами, повысив скорость обучения PFNN параллельной обработкой многомерного массива синаптических весов.

Литература

1. Суханов Н.В. Схема управления летательным аппаратом на основе нейронных сетей. URL: <https://mai.ru/upload/iblock/6c1/6c19168006748d378a6f2a491eac3e88.pdf>
2. Бахшиев А.В., Гунделах Ф.В. Разработка и исследование бионической модели нейронной сети для управления движением робототехнических систем. URL: <http://journal-niss.ru/journal/archive/13/paper6.pdf>.
3. Holden D.A Deep Learning Framework for Character Motion Synthesis and Editing. URL: https://github.com/AliJalalifar/Character_Animation.
4. A Fully Convolutional Network for Semantic Labeling of 3D Point Clouds. URL: <https://arxiv.org/pdf/1710.01408.pdf>.
5. Moeslund T.B. A survey of advances in human motion capture and analysis // Computer Vision and Image Understanding. 2006. Vol. 104/ P. 90–126
6. Mazen A.B. Trajectory Optimization for Full-Body Movements with Complex Contacts/2013.
7. Modeling Human Motion with Quaternion-based Neural. URL: <https://arxiv.org/pdf/1901.07677.pdf>.
8. Holden D.A Deep Learning Framework for Character Motion Synthesis and Editing. URL: https://github.com/AliJalalifar/Character_Animation.
9. The Expanding Universe of Ubiquitin and Ubiquitin-Like Modifiers. URL: <http://www.plantphysiol.org/content/160/1/2>.

A.I. Pavlykov, T.A. Samoylova,
Smolensk State University

NEURAL MOVEMENT IN REAL TIME

Keywords: *motion neurocontrol, real time, animation, phase-functional neural network, multidimensional array, Python, C#, Theano, prediction.*

Abstract. *A software implementation of the neural network approach to the problem of real-time motion control is proposed. A relatively new efficient neural network learning algorithm using a database containing a terrain relief and a description of object motions is considered. The algorithm was programmed using Python, C#, and Theano and cuDNN library modules, which make it possible to efficiently quickly determine, optimize, and evaluate multidimensional arrays. The proposed learning algorithm controls the object in real time, requires a small amount of memory and provides high quality motion.*

Т.А. Самойлова, И.М. Струнин
Смоленский государственный университет

УДК 004.932.2

СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЙ МЕТОДОМ МАРКЕРНОГО ВОДОРАЗДЕЛА

Ключевые слова: *сегментация изображений, метод маркерного водораздела, язык Python, библиотека OpenCV.*

Обсуждаются современные подходы к задаче сегментации изображений. Рассмотрен относительно новый эффективный алгоритм, основанный на модификации функции сегментации с использованием значений маркеров фона и маркеров переднего плана. Для решения задачи использовались язык программирования Python и модули библиотеки компьютерного зрения OpenCV. Проведены эксперименты, в ходе которых показано, что выбранный метод позволяет проводить эффективную сегментацию изображений высокого разрешения.

Сегментация изображений – фундаментальный процесс в системах компьютерного зрения, выполняющих анализ графической информации при решении многих практических задач [1; 2]. В этом процессе происходит группирование разрозненных участков или фрагментов изображения в область, принадлежащую одному объекту, или разделение какого-либо участка изображения на области, принадлежащие разным объектам. При этом группирование осуществляется по различным признакам, таким как яркость, цвет, движение в одном направлении и т.п. На сегодняшний день известно большое количество методов сегментации изображений, использующих разные признаки [3]. Поэтому перед разработчиками систем компьютерного зрения встает непростая проблема выбора алгоритмов, наиболее адекватных их задачам.

Суть предлагаемого метода сегментации. Для решения задачи сегментации изображений высокого разрешения выполнена реализация алгоритма маркерного водораздела [5] средствами библиотеки OpenCV языка Python. Преобразование водораздела – метод сегментации, основанный на топологической теории математической морфологии. Название соответствует геологическому водоразделу или водосборному бассейну, который разделяет смежные водосборные бассейны. Метод обрабатывает изображение как топографическую карту с яркостью каждой точки, представляющей ее высоту, и находит линии, проходящие вдоль вершин гребней. Преобразование включает два этапа. Один – это процесс сортировки, а другой – процесс погружения. На первом этапе значение каждого пикселя изображения сортируется от низкого до высокого уровня серого. Затем в процессе погружения каждое локальное минимальное значение оценивается и маркируется с использованием структуры «первым вошел – первым вышел» (FIFO). Граничный пункт между водосборными бассейнами – это водораздел. Очевидно, водораздел обозначает максимальную точку входного изображения. Поэтому, чтобы получить границу информации об изображении, в качестве входного изображения используется его градиент:

$$g(x, y) = \text{grad}(f(x, y)), \quad (1)$$

где x, y – координаты пикселя, $f(x, y)$ обозначает исходное изображение.

Однако алгоритм простого водораздела считает "бассейнами" самые незначительные объекты, поэтому в нем возникает проблема чрезмерной сегментации. Для уменьшения этой проблемы была реализована сегментация водораздела на основе маркеров. Маркер – связанный компонент, принадлежащий изображению. Он имеет два

типа: внешний, связанный с фоном, и внутренний, связанный с объектами интереса.

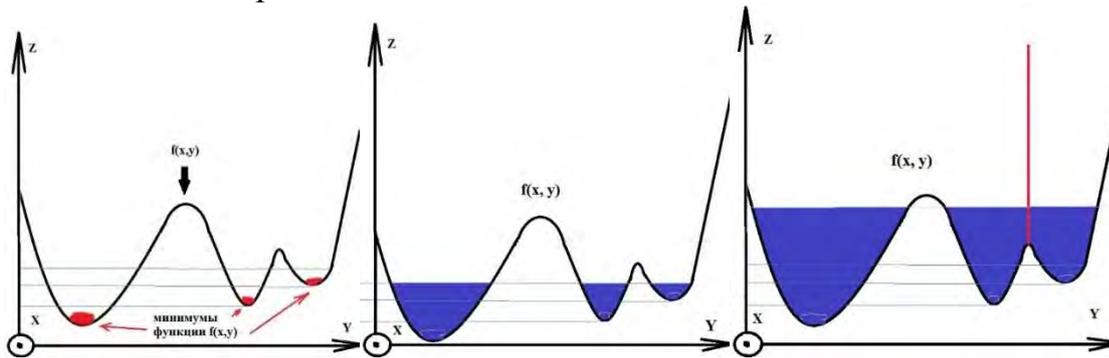


Рис. 1. Иллюстрация метода маркерного водораздела

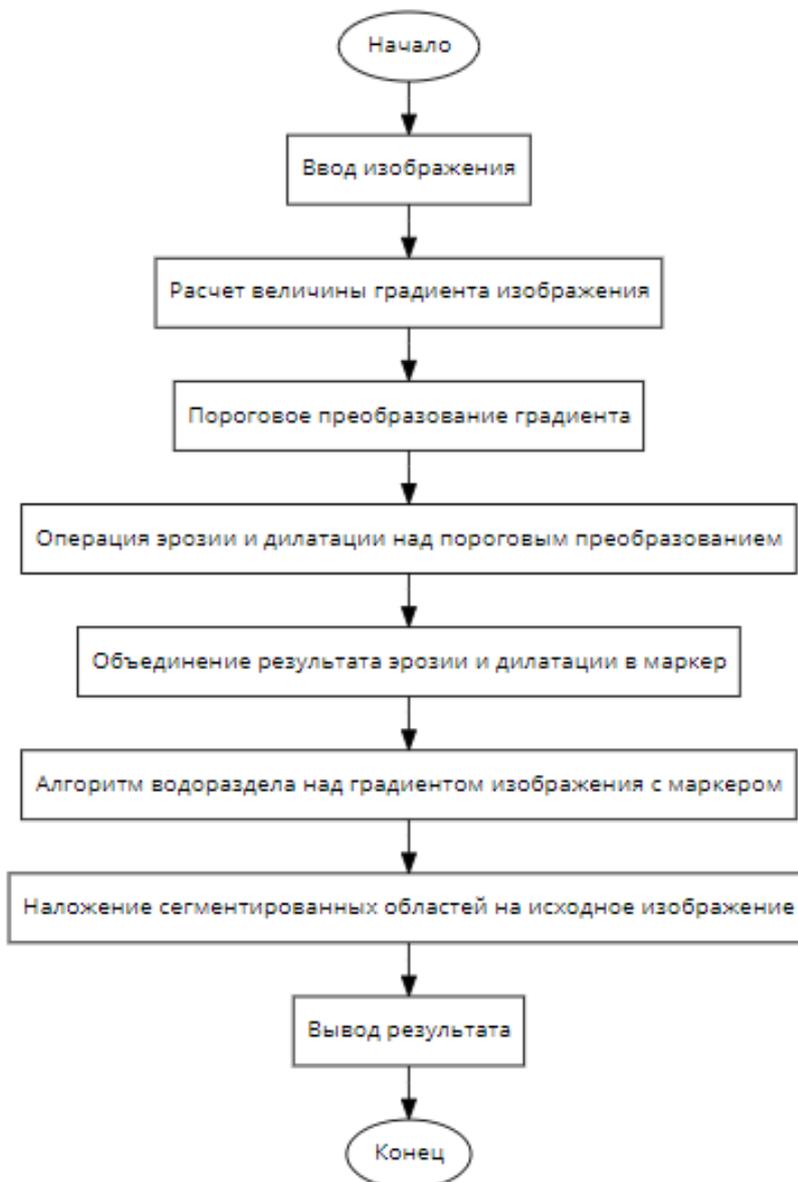


Рис. 2. Алгоритм маркерного водораздела

Сегментация водораздела на основе маркеров может сегментировать уникальные границы изображения. Для автоматического создания маркера используется эрозия и дилатация над пороговым преобразованием градиента изображения.

Алгоритм находит и извлекает точки, указывающие на наличие объектов или фона в определенных местах изображения. Затем устанавливаются местоположения маркеров фона и маркеров переднего плана, как региональные минимумы в пределах топологической поверхности (как правило, градиент исходного входного изображения), и только после этого применяется алгоритм собственно водораздела. Размещение внутренних и внешних маркеров в интересующих регионах легко справляется с проблемой чрезмерной сегментации. Иллюстрация метода и блок-схема алгоритма маркерного водораздела и показаны на рисунках 1 и 2 соответственно.

Основной недостаток реализации данного алгоритма – необходимость использования процедуры предварительной обработки для изображений с большим количеством локальных минимумов. Результаты работы алгоритма проверялись на спутниковых снимках, полученных с помощью программы SAS.Planet. На рисунках 3, 4, 5 приведен пример процесса сегментации одного из таких снимков, соответствующего участка города Смоленска, который включает оригинал спутникового снимка, результат преобразования изображения в градации серого и окончательную сегментацию.



Рис. 3. Оригинал спутникового снимка



Рис. 4. Результат преобразования изображения в градации серого



Рис. 5. Сегментация, полученная для спутникового снимка с помощью разработанной программы

Заключение. Метод маркерного водораздела показал свою высокую эффективность при решении задачи сегментации изображений высокого разрешения. Многочисленные эксперименты подтвердили высокую надежность работы алгоритма при правильной его реализации средствами OpenCV. Отметим, что оценка результатов сегментации

была проведена визуально, что делает выводы достаточно субъективными. В перспективе планируется выполнить количественную оценку результатов, что позволит улучшить качество исследуемого метода. Планируется также выполнить параллельную сегментацию объемных изображений методом их декомпозиции и разбить последовательный алгоритм на несколько потоков для параллельных вычислений.

Литература

1. Разработка алгоритма семантической сегментации аэрофотоснимков реального времени / Ю.Б. Блохинов [и др.] // Компьютерная оптика. 2018. Т. 42, №1. С. 141–148.
2. Соловьев Р.А., Тельпухов Д.В., Кустов А.Г. Автоматическая сегментация спутниковых снимков на базе модифицированной свёрточной нейронной сети UNET // Инженерный вестник Дона. 2017. №4. URL: http://ivdon.ru/uploads/article/pdf/IVD_56_soloviev_N.pdf_116222c2f5.pdf.
3. Цифровая обработка изображений / Л.И. Рубанов [и др.]. М.: Техносфера, 2012. 1104 с.
4. Gonzalez R.C., Woods R.E. Digital Image Processing using MATLAB. Upper Saddle River, NJ Prentice Hall, 2008.
5. Hamdi M.A. Modified Algorithm marker-controlled watershed transform for Image segmentation Based on Curvelet Threshold // Canadian Journal on Image Processing and Computer Vision. 2011. Vol. 2 No. 8. P. 88–91.

T.A. Samoylova, I.M. Strunin
Smolensk State University

IMAGE SEGMENTATION METHOD OF MARKER WATERSHED

Keywords: *image segmentation, marker watershed method, Python language, OpenCV library.*

Abstract. *In this article, we consider the solution of the segmentation problem by means of the OpenCV library and the Python language using a relatively new marker watershed method effective for high resolution images. The essence of the method is as follows. It is proposed to consider the image as a certain map of the area, where the brightness values represent the height values relative to a certain level. If this area is filled with water, pools are formed. With further water filling, the pools are combined. Pool pools are defined as lines depending on their brightness characteristics.*

**И.Н. Сеницын, Д.В. Жуков,
Э.Р. Корепанов, Т.Д. Конашенкова**
*Федеральный исследовательский центр «Информатика и
управление» Российской академии наук (ФИЦ ИУ РАН)*

УДК 621

РАЗВИТИЕ ПРЯМЫХ МЕТОДОВ ИНТЕРПОЛЯЦИОННОГО АНАЛИТИЧЕСКОГО МОДЕЛИРОВАНИЯ РАСПРЕДЕЛЕНИЙ В СТОХАСТИЧЕСКИХ СИСТЕМАХ

Ключевые слова: *вейвлет Хаара, интерполяционные методы аналитического моделирования, плотность вероятности, стохастическая система, уравнение Пугачёва для характеристической функции, характеристическая функция.*

Разработаны два новых прямых интерполяционных метода аналитического моделирования (МAM) одномерных (характеристической функции (х. ф.) и плотности вероятности (п. в.)) в нелинейных скалярных стохастических системах (СтС), основанные на вейвлетах Хаара. Проведенные вычислительные эксперименты в СтС с разрывными правыми частями показывают существенно более высокую точность вейвлет МAM, чем прямые интерполяционные на основе теоремы отсчетов В.А. Котельникова как в стационарных, так и в переходных режимах. Отмечается, что применение обоих рассматриваемых МAM к СтС высокой размерности встречает серьезные вычислительные трудности и оправдано, когда многомерная СтС может быть заменена системой эквивалентных одномерных СтС. Так, для высокочастотных СтС ударосеймозащиты после перехода к нормальным координатам удается создать эффективные программные средства анализа, моделирования и синтеза.

Постановка задачи. Как известно [1], среди существующих методов аналитического моделирования стохастических процессов (СтП) в дифференциальных СтС важное место занимают методы, основанные на прямом численном решении эволюционных уравнений для n -мерной х. ф. С.В. Мальчиковым в [2; 3] для гауссовских СтС на основе теоремы отсчетов В.А. Котельникова [4] разработан метод интерполяционного аналитического моделирования (МИAM) для одномерной х. ф. В [5] дано обобщение [2, 3] для негауссовских СтС на основе замены негауссовской СтС эквивалентной гауссовской (нормальной) СтС. В [6] разработаны МИAM для сложных непрерывных и дискретных скалярных и приводимых к ним векторных СтС. Выделены типовые классы сложных СтС. Методы основаны на интерполяционном решении эволюционного уравнения Пугачёва для одномерной х. ф. Получены уравнения чувствительности для оценки х.

ф. к параметрам СтС. В основу базовых алгоритмов МИАМ положена теорема отсчетов В.А. Котельникова. Рассмотрены практические вопросы выбора интерполяционных формул В.А. Котельникова и оценки числа интерполяционных отсчетов.

Рассмотрим развитие методов [5; 6] для off-line вейвлет МАМ на основе вейвлетов Хаара [7–9].

Основные допущения и определения. Пусть исходная дифференциальная СтС имеет вид [1]:

$$dY = a(Y, t)dt + b(Y, t)dW, \quad t \geq t_0, \quad Y(t_0) = Y_0, \quad (1)$$

где Y – переменная состояния СтС; Y_0 – случайная величина (СВ) с известной плотностью вероятности (п. в.); $W = W(t)$ – скалярный СтП с независимыми приращениями с одномерной х. ф. $h = h(\mu; t)$; $a = a(Y, t)$ и $b(Y, t)$ – известные детерминированные функции, допускающие точки разрыва первого рода. Если существует одномерная п. в. $f(y; t)$, то х. ф. СтП Y $g(\lambda; t)$ будет удовлетворять следующему уравнению Пугачёва [1]:

$$\frac{\partial g(\lambda; t)}{\partial t} = \int_{-\infty}^{\infty} [i\lambda a(y, t) + \chi(b(y, t)\lambda; t)] \cdot e^{i\lambda y} f(y; t) dy, \quad (2)$$

$$g(\lambda; t_0) = g_0(\lambda), \quad (3)$$

$$\chi(\mu; t) = \frac{\partial}{\partial t} h(\mu; t), \quad \mu = b(y, t)\lambda. \quad (4)$$

Будем считать, что $t \in (t_0, T)$, $\lambda \in [-\Lambda, \Lambda]$, и выполним замену переменных в (2), (3) согласно

$$\bar{t} = \frac{t - t_0}{T - t_0}, \quad \bar{\lambda} = \frac{\lambda + \Lambda}{2\Lambda}. \quad (5)$$

Тогда получим следующие нормализованные уравнения для х. ф.:

$$\frac{\partial \bar{g}(\bar{\lambda}; \bar{t})}{\partial \bar{t}} = (T - t_0) \int_{-\infty}^{\infty} [i\Lambda(2\bar{\lambda} - 1)\bar{a} + \bar{\chi}] e^{i\Lambda(2\bar{\lambda} - 1)y} \bar{f}(y; \bar{t}) dy, \quad (6)$$

$$\bar{g}(\bar{\lambda}; 0) = \bar{g}_0(\bar{\lambda}). \quad (7)$$

Здесь введены обозначения:

$$\begin{aligned} \bar{t}, \bar{\lambda} \in [0, 1]; \quad t(\bar{t}) = t_0 + (T - t_0)\bar{t}, \quad \lambda(\bar{\lambda}) = \Lambda(2\bar{\lambda} - 1), \\ \bar{g}(\bar{\lambda}; \bar{t}) = g(\lambda(\bar{\lambda}); t(\bar{t})), \quad \bar{g}_0(\bar{\lambda}) = g_0(\lambda(\bar{\lambda})), \quad \bar{a} = a(y, t(\bar{t})), \\ \bar{\chi} = \chi(b(y(t(\bar{t})), t(\bar{t})) \cdot \lambda(\bar{\lambda}); t(\bar{t})). \end{aligned} \quad (8)$$

Введем ортонормированный базис вейвлетов Хаара на $[0, 1]$ и интегралов от них:

$$w_1(x) = \begin{cases} 1 & \text{при } x \in [0, 1), \\ 0 & \text{при } x \notin [0, 1), \end{cases} \quad (9)$$

$$w_i(x) = \begin{cases} \sqrt{m} & \text{при } x \in \left[\frac{k}{m}, \frac{k+0,5}{m}\right), \\ -\sqrt{m} & \text{при } x \in \left[\frac{k+0,5}{m}, \frac{k+1}{m}\right), \end{cases} \quad (10)$$

$$p_1(x) = \int_0^x w_1(\tau) d\tau = \begin{cases} 1 & \text{при } x \in [0,1), \\ 0 & \text{при } x \notin [0,1), \end{cases} \quad (11)$$

$$p_l(x) = \int_0^x w_l(\tau) d\tau = \begin{cases} \sqrt{m} \left(x - \frac{k}{m} \right) & \text{при } x \in \left[\frac{k}{m}, \frac{k+0,5}{m} \right), \\ \sqrt{m} \left(\frac{k+1}{m} - x \right) & \text{при } x \in \left[\frac{k+0,5}{m}, \frac{k+1}{m} \right), \\ 0 & \text{при } x \notin \left[\frac{k}{m}, \frac{k+1}{m} \right), \end{cases} \quad (12)$$

Здесь $l = \overline{2, L}$; J — максимальный уровень разрешения; $j = 0, 1, \dots, J$; $m = 2^j$; $k = 0, 1, \dots, m-1$; $l = m + k + 1$; $L = \max l = 2 \cdot 2^J$.

Основные результаты. Выполним сначала предварительные преобразования с уравнением Пугачёва (6). Пусть двумерная х. ф. допускает вейвлет разложения по (9), (10)

$$\frac{\partial \bar{g}(\bar{\lambda}; \bar{t})}{\partial \bar{t}} = \sum_{n=1}^N \sum_{l=1}^L c_{nl} w_n(\bar{t}) w_l(\bar{\lambda}). \quad (13)$$

Здесь $N = 2 \cdot 2^{J_1}$; $L = 2 \cdot 2^{J_2}$; J_1 — максимальный уровень вейвлет разложения по переменной \bar{t} , а J_2 — по переменной $\bar{\lambda}$. Тогда для х. ф. $\bar{g}(\bar{\lambda}, \bar{t})$ по (11), (12) будет иметь место вейвлет разложение

$$\bar{g}(\bar{\lambda}; \bar{t}) = \bar{g}_0(\bar{\lambda}) + \sum_{n=1}^N \sum_{l=1}^L c_{nl} p_n(\bar{t}) w_l(\bar{\lambda}). \quad (14)$$

Учитывая, что п. в. $f(y; t)$ связана с х. ф. преобразованием Фурье

$$f(y; t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-i\lambda y} g(\lambda; t) d\lambda, \quad (15)$$

представим разложение п. в. по системе функций $\{w_l(\bar{\lambda})\}$ и $\{p_n(\bar{t})\}$ в форме:

$$f(y; t) = f(y; t(\bar{t})) = \bar{f}(y; \bar{t}) = f_0(y) + \frac{\Lambda}{\pi} e^{i\Lambda y} \sum_{n=1}^N \sum_{l=1}^L c_{nl} p_n(\bar{t}) \int_0^1 e^{-2i\Lambda \bar{\lambda} y} w_l(\bar{\lambda}) d\bar{\lambda}. \quad (16)$$

После введения обозначений

$$F_l(y) = F_{l1}(y) - iF_{l2}(y), \quad F_{l1}(y) = \frac{\Lambda}{\pi} \int_0^1 \cos[\Lambda(2\bar{\lambda} - 1)y] w_l(\bar{\lambda}) d\bar{\lambda},$$

$$F_{l2}(y) = \frac{\Lambda}{\pi} \int_0^1 \sin[\Lambda(2\bar{\lambda} - 1)y] w_l(\bar{\lambda}) d\bar{\lambda}, \quad c_{\Lambda l} = c_{n1} + ic_{n2} \quad (17)$$

и учёта свойства действительности п. в. придем к уравнению

$$\bar{f}(y; \bar{t}) = f_0(y) + \sum_{n=1}^N \sum_{l=1}^L p_n(\bar{t}) [c_{n1} F_{l1}(y) + c_{n2} F_{l2}(y)]. \quad (18)$$

Далее выполним с уравнением (18) следующие преобразования. Запишем (14) в виде

$$\sum_{n=1}^N \sum_{l=1}^L (c_{nl1} + ic_{nl2}) w_n(\bar{t}) w_l(\bar{\lambda}) = (T - t_0) \int_{-\infty}^{\infty} [i\Lambda(2\bar{\lambda} - 1)\bar{a} + \bar{\chi}] \times \\ \times \{ \cos[(2\bar{\lambda} - 1)\Lambda y] + i \sin[(2\bar{\lambda} - 1)\Lambda y] \} \cdot \bar{f}(y; \bar{t}) dy. \quad (19)$$

Если ввести в (19) обозначения

$$\alpha_1 = \alpha_1(y, \bar{\lambda}, \bar{t}) = \bar{\chi} \cdot \cos[(2\bar{\lambda} - 1)\Lambda y] - \Lambda \bar{a} (2\bar{\lambda} - 1) \sin[(2\bar{\lambda} - 1)\Lambda y] \\ \alpha_2 = \alpha_2(y, \bar{\lambda}, \bar{t}) = \Lambda \bar{a} (2\bar{\lambda} - 1) \cdot \cos[(2\bar{\lambda} - 1)\Lambda y] + \bar{\chi} \sin[(2\bar{\lambda} - 1)\Lambda y] \quad (20)$$

то видно, что оно является комплексным и распадается на два действительных уравнения:

$$\sum_{n=1}^N \sum_{l=1}^L c_{nl\varepsilon} w_n(\bar{t}) w_l(\bar{\lambda}) = (T - t_0) \int_{-\infty}^{\infty} \alpha_{\varepsilon} \bar{f} dy \quad (\varepsilon = 1, 2). \quad (21)$$

Все интегралы по y в (21) при фиксированных $\bar{\lambda}$ и \bar{t} могут быть вычислены заранее по следующим формулам:

$$\xi_{\varepsilon} = \xi_{\varepsilon}(\bar{\lambda}, \bar{t}) = \int_{-\infty}^{\infty} \alpha_{\varepsilon} f_0 dy \quad (\varepsilon = 1, 2), \quad (22)$$

$$\beta_{l\varepsilon} = \beta_{l\varepsilon}(\bar{\lambda}, \bar{t}) = \int_{-\infty}^{\infty} \alpha_{\varepsilon} F_{l1} dy \quad (\varepsilon = 1, 2), \quad (23)$$

$$\gamma_{l\varepsilon} = \gamma_{l\varepsilon}(\bar{\lambda}, \bar{t}) = \int_{-\infty}^{\infty} \alpha_{\varepsilon} F_{l2} dy \quad (\varepsilon = 1, 2). \quad (24)$$

В результате, для $\varepsilon = 1$ и 2 перепишем (21) в виде

$$\sum_{n=1}^N \sum_{l=1}^L c_{nl\varepsilon} w_n(\bar{t}) w_l(\bar{\lambda}) = (T - t_0) \left\{ \xi_{\varepsilon} + \sum_{n=1}^N \sum_{l=1}^L [c_{nl1} p_n(\bar{t}) \beta_{l\varepsilon} + c_{nl2} p_n(\bar{t}) \gamma_{l\varepsilon}] \right\}. \quad (25)$$

Теперь разобьем: 1) интервал $\bar{t} \in [0, 1]$ на N равных частей $\Delta \bar{t} = 1/N$ ($N = 2, 2^{J_1}$) и определим точки $t_r = (r - 0,5)\Delta \bar{t}$ ($r = \overline{1, N}$); 2) интервал $\bar{\lambda} \in [0, 1]$ на L равных частей $\Delta \bar{\lambda} = 1/L$ ($L = 2, 2^{J_2}$) и определим точки $\lambda_s = (s - 0,5)\Delta \bar{\lambda}$ ($s = \overline{1, L}$). Далее вычислим значения всех функций, зависящих от \bar{t} и $\bar{\lambda}$ в точках t_r и λ_s . В результате получим систему из двух ($\varepsilon = 1$ и 2) линейных алгебраических уравнений, определяющих $c_{nl\varepsilon}$:

$$\sum_{n=1}^N \sum_{l=1}^L c_{nl\varepsilon} w_n(\bar{t}_r) w_l(\bar{\lambda}_s) = (T - t_0) \left\{ \xi_{\varepsilon}(\bar{\lambda}_s, \bar{t}_r) + \sum_{n=1}^N \sum_{l=1}^L [c_{nl1} p_n(\bar{t}_r) \beta_{l\varepsilon}(\bar{\lambda}_s, \bar{t}_r) + c_{nl2} p_n(\bar{t}_r) \gamma_{l\varepsilon}(\bar{\lambda}_s, \bar{t}_r)] \right\}. \quad (26)$$

Следовательно, согласно первому МИАМ, имеем в силу (26) $2 \cdot N \cdot L$ линейных алгебраических уравнений для определения неизвестных коэффициентов $c_{nl\varepsilon}$ ($n = \overline{1, N}, l = \overline{1, L}$). Зная $c_{nl\varepsilon}$, получим разложение (14) по вейвлетам Хаара и интегралов от них. После перехода к переменным $\lambda = \Lambda(2\bar{\lambda} - 1)$ и $t = t_0 + (T - t_0)\bar{t}$ получим соответствующее разложение х. ф. $g(\lambda; t)$:

$$g(\lambda; t) = g_0(\lambda) + \sum_{n=1}^N \sum_{l=1}^L c_{nl} p_n \left(\frac{t - t_0}{T - t_0} \right) w_l \left(\frac{\lambda + \Lambda}{2\Lambda} \right). \quad (27)$$

Согласно второму МИАМ, аналогично получается разложение п. в. $f(y;t)$:

$$f(y;t) = f_0(y) + \sum_{n=1}^N \sum_{l=1}^L p_n \left(\frac{t-t_0}{T-t_0} \right) [c_{nl1} F_{l1}(y) + c_{nl2} F_{l2}(y)]. \quad (28)$$

Функции $F_{l\varepsilon} = F_{l\varepsilon}(y)$ в (28) при $(l > 1)$ вычисляются заранее по формулам

$$F_{11} = F_{11}(y) = \frac{1}{\pi y} \sin(\Lambda y), \quad F_{12} = 0, \quad (29)$$

$$F_{l1} = F_{l1}(y) = \frac{\sqrt{m}}{2\pi y} \left\{ 2 \sin \left[\left(2 \frac{k+0,5}{m} - 1 \right) \Lambda y \right] - \sin \left[\left(2 \frac{k}{m} - 1 \right) \Lambda y \right] - \sin \left[2 \left(2 \frac{k+1}{m} - 1 \right) \Lambda y \right] \right\} \quad (30)$$

$$F_{l2} = F_{l2}(y) = \frac{\sqrt{m}}{2\pi y} \left\{ \cos \left[\left(2 \frac{k}{m} - 1 \right) \Lambda y \right] + \cos \left[\left(2 \frac{k+1}{m} - 1 \right) \Lambda y \right] - 2 \cos \left[\left(2 \frac{k+0,5}{m} - 1 \right) \Lambda y \right] \right\}.$$

Литература

1. Пугачев В.С., Сеницын И.Н. Теория стохастических систем. М.: Логос, 2000; 2004. 1000 с.
2. Мальчиков С.В. Приближенный метод определения законов распределения фазовых координат нелинейных автоматических систем // Автоматика и телемеханика. 1970. № 5. С. 43-50.
3. Казаков И.Е., Мальчиков С.В. Анализ стохастических систем в пространстве состояний. М.: Наука, 1983. 348 с.
4. Котельников В.А. Теория потенциальной помехоустойчивости. М.: Госэнергоиздат. 1956. 412 с.
5. Сеницын И.Н. Метод интерполяционного аналитического моделирования одномерных распределений в стохастических системах // Информатика и ее применения. 2018. Т. 12, вып. 1. С. 55-61.
6. Сеницын И.Н. Интерполяционное аналитическое моделирование распределений в сложных стохастических системах // Информатика и ее применение. 2019. Т. 13, вып. 1 (в печати).
7. Lepik U. Numerical solution of evolution equations by the Haar wavelet methods. 2007. Appl. Math. Comput. 185:695-704.
8. Lepik U. Application of the Haar wavelet transform to solving integral and differential equations. 2007. Proc. Estonian Acad. Sci. Phys. Math. 56:28-46.
9. Сеницын И.Н., Корепанов Э.Р., Конашенкова Т.Д. Инструментальное программное обеспечение анализа и синтеза систем высокой доступности (VII) // Системы высокой доступности. 2019. Т. 15, № 1 (в печати).

I.N. Sinitsyn, D. V. Zhukov, E.R. Korepanov, T.D. Konashenkova
FRC CSC of RAS (Moscow)

DEVELOPMENT OF DIRECT METHODS FOR STOCHASTIC SYSTEMS INTERPOLATIONAL ANALYTICAL MODELING OF DISTRIBUTIONS

Keywords: Haar wavelet; Interpolation analytical modeling method; Probability density; Pugachev equation for characteristic function; Stochastic system.

Abstract. *The Haar wavelet interpolational methods of characteristic function and probability density in scalar nonlinear stochastic system (StS) are developed. Computer experiments for vibro and shock noises for discontinuous StS in comparison with interpolation methods based on Kotelnikov V.A. theorem are more effective.*

А.Е. Ширай

Федеральный исследовательский центр «Информатика и управление» Российской академии наук (ФИЦ ИУ РАН)

УДК 519.283

ТРЕБОВАНИЯ К УНИВЕРСАЛЬНОЙ МИКРОПРОЦЕССОРНОЙ АРХИТЕКТУРЕ ПЯТОГО ПОКОЛЕНИЯ

Ключевые слова: *универсальная микропроцессорная архитектура, язык высокого уровня, объект-сегмент, поток-безопасное программирование, контекстная защита, веб-сервер, виртуальная машина.*

В данной работе рассматриваются требования к универсальной микропроцессорной архитектуре нового поколения, реализующей модель вычислений, предназначенную для решения современных задач, соответствующую современным языкам и технологиям программирования, а также возможностям технологий изготовления аппаратуры.

Под архитектурой вычислительной системы подразумевается модель вычислений, реализованная в аппаратно-программной среде и предназначенная для выполнения программ. Если основные качества модели вычислений определяются программной реализацией, мы имеем дело с архитектурой виртуальной машины, соответственно, архитектуру аппаратной части также определяет заложенная в неё модель вычислений. При этом микроархитектура микропроцессоров, из которых строятся современные системы, влияет в основном на такие параметры, как быстродействие, пропускная способность и т.д.

В значительной мере архитектура микропроцессоров вычислительной системы определяется системой команд, наборами регистров, принципами организации адресации памяти, т.е. архитектурой набора команд (ISA Architecture). Производители выпускают семейства микропроцессоров различного назначения производительности, с одинаковой ISA, но с разной реализацией микроархитектуры, различными технологиями. Новые микроархитектуры в литературе часто называют поколениями микроархитектур, но, если они реализуют ту же модель вычислений, что

и прежние микроархитектуры, это нельзя считать переходом на новую архитектуру вычислительной системы.

В настоящее время нет определения термина «поколение универсальных архитектур микропроцессоров». Есть поколения реализаций известных архитектур, таких как X86 32/64 от INTEL и AMD, SPARC T/M от SUN/ORACLE, IBM POWER, а также ARM, MIPS и т.д., но самим архитектурам при этом не ставится в соответствие термин «поколение».

Если поколение графических процессоров определяет модель шейдеров, задающая структуру графического конвейера, то для универсальных архитектур в качестве критерия будем использовать количество ядер и модель управления вычислениями.

Таким образом, можно выделить четыре поколения универсальных микропроцессорных архитектур (без учёта поддержки графики).

1. Одноядерные однозадачные архитектуры.

В настоящее время такие микропроцессоры чаще всего используются в системах на кристалле (SOC), обычно в качестве контроллеров.

2. Одноядерные многозадачные архитектуры.

Эти системы отличаются аппаратной поддержкой виртуальной памяти и многоуровневой аппаратной защитой адресного пространства процессов, что позволило реализовать ОС с вытесняющей многозадачностью. Такие системы и сейчас используются в бюджетных решениях.

3. Многоядерные многозадачные архитектуры.

Эти архитектуры расширили возможности систем предыдущего поколения в части количества одновременно решаемых задач, выделения отдельных ядер для системной поддержки. Исчезла необходимость переключать процессор и выгружать состояние процессов при внешних событиях, что снизило накладные расходы на параллельные вычисления, повысилась пропускная способность систем массового обслуживания (интернет-серверов, серверов телекоммуникаций).

4. Многоядерные многопоточные (multithread) архитектуры.

С появлением новых ядер, исполняющих несколько потоков вычислений в одном ядре, с новыми суперскалярными вычислителями, новыми высокопроизводительными шинами между ядрами, поддерживающими режим когерентности кэш-памятей, добавлением атомарных операций для синхронизации доступа к общей памяти, универсальные микропроцессоры научились активно поддерживать многопоточный режим исполнения программ.

Несмотря на то, что в микропроцессорах с архитектурой четвёртого поколения реализована работа со множеством потоков, они являются прямыми наследниками архитектуры третьего поколения. Это

симметричные мультипроцессоры с общим доступом в оперативную память. Они сохранили преемственность с прежними поколениями в части системы команд. Общий доступ в память и необходимость поддержания когерентности кэш-памятей в ядрах накладывают ограничения на число ядер в процессоре и число потоков в приложениях. Три уровня кэш-памяти с эвристическими алгоритмами заполнения не всегда обеспечивают высокий процент попадания, что может приводить к большому времени ожидания поступления данных.

Несмотря на свои достоинства микропроцессоры с архитектурой 4-го поколения не позволили осуществить прорыв в области параллельного программирования.

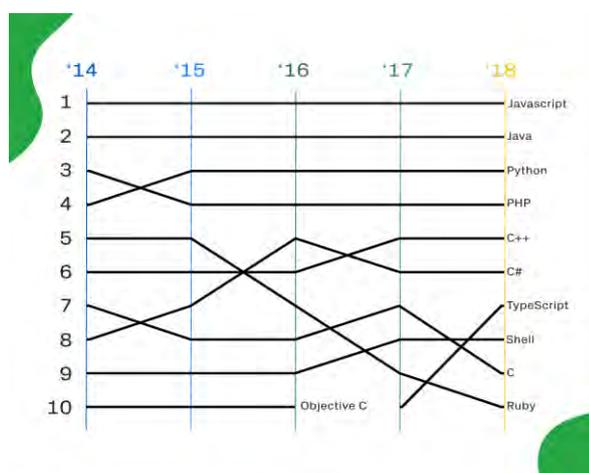


Рис. 1. Рейтинги популярности языков программирования по годам (по числу людей, которые вносили изменения в репозитории)

Создание, отладка и оптимизация параллельных программ, содержащих параллельный доступ многих потоков к общим данным, остались «искусством». Накладные расходы на организацию параллельного доступа к общим данным по-прежнему велики. Отладка и разработка таких программ занимает существенно больше времени, чем обычных однопоточных или параллельных программ, которые работают с независимыми данными, доступ к которым не требует синхронизации. В качестве примера можно привести идеологию языка Python, который для разрешения конфликтов доступа использует «глобальный лок» GIL, что приводит к кратному замедлению программ при использовании многопоточности.

Свойства современных языков программирования

Современными языками для универсальных систем будем считать те языки, которые имеют наивысший рейтинг в списках новых и редактируемых проектов (рис. 1) и те, которые показали наибольший прирост (рис 2) в процентах к предыдущим показателям (статистика GitHub).

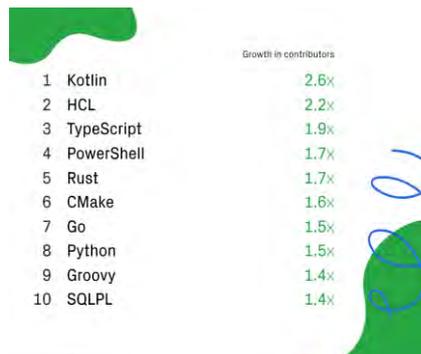


Рис. 2 Динамика роста популярности языков программирования

Сотрудники GitHub выразили мнение о критериях популярности языка программирования в 2018 году. Ими стали следующие.

- **Безопасность типов.** Во многих языках, за исключением языка Python, наблюдается рост популярности статической типизации. Специалисты компании считают, что это происходит благодаря эффективности и безопасности как для отдельных разработчиков, так и для компаний, работающих над большими проектами.

- **Интероперабельность.** Причина, по которой TypeScript взлетел вверх в рейтингах, – совместимость с JavaScript. То же касается Rust и Kotlin, которые взаимодействуют с JavaScript и Java. Если рассматривать Python, то Swift-разработчики могут напрямую работать с API на этом языке, не используя каких-либо обёрток.

- **Open Source.** Все упомянутые языки являются проектами с открытым исходным кодом. По мнению GitHub, их активная поддержка Open Source сообществами может помочь языкам стать популярнее у разработчиков.

Данная таблица демонстрирует, что рост производительности аппаратуры привёл к смене поколений в языках программирования. Языки, ориентированные на особенности архитектуры существующих микропроцессоров такие, как C/C++, постепенно теряют своё значение. Несмотря на то что сейчас они всё ещё являются де факто стандартными средствами системного программирования, и это наряду с огромным объёмом библиотек, позволяет им сохранять рейтинги. Роль этих языков в современных проектах для универсальных вычислительных систем снижается, а область применения смещается в область программирования для специализированных систем, где важны возможности «ручного» управления ресурсами, включая область высокопроизводительных вычислений. Сохраняющаяся популярность C++ обоснована ещё и тем, что здесь приведена суммарная статистика по всем платформам. Заметим также, что популярность C++ выросла только в тех странах, где создаётся новое системное ПО (США, Китай) [5]. Снижение же рейтинга C# связано с тем, что его, как правило,

используют совместно с продуктами от Microsoft, а не Open Source, хотя во всех странах, кроме США и Китая его рейтинг выше, чем у C++. Рейтинг языка C, также высок только в Азии, где, по-видимому, его часто используют в специализированных SOC, входящих в контроллеры для бытовой техники и т.д.

Современные языки программирования являются объектно-ориентированными, интерпретируемыми, платформ-независимыми языками. То, что они в основном Open Source, не означает, что они не могут исполняться под другими ОС. Рост ценности программных продуктов и развитие интернет-ресурсов приводят к размыванию межплатформенных барьеров.

Для интернет-приложений разработано множество различных средств, обеспечивающих межплатформенную совместимость, например, для веб-сервера IIS от Microsoft, изначально работающего с Asp.net, есть такое средство как Helicon Zoo, ядром которого является «native IIS модуль, по сути, играющий роль моста между веб-сервером IIS и фреймворками на Ruby, Python, Perl и др» [4]. Ещё пример: для МАК есть средство Parallel desktop, которое обеспечивает исполнение Windows программ под МАК ОС, так, что пользователь не видит вызова эмулятора OS Windows.

В настоящее время уже проделана большая работа в направлении кроссплатформенности на уровне современных компиляторов, которые порождают байт-код (LLVM, JVM, Dot-Net CIL), который может быть интерпретирован или преобразован JIT компиляторами в исполняемый код для целевой платформы. За последние годы производительность программ, написанных на интерпретируемых и скриптовых языках программирования, выросла на порядок.

Некоторые микропроцессоры имеют и аппаратную поддержку для ускорения исполнения байт-кода. Так, процессоры ARM используют технологию битовой компиляции байт-кода в команды процессора Jazelle RST, позволяющую компилировать байт-код для любой виртуальной машины.

Современные языки программирования позволяют создавать параллельные программы, использующие не только многопоточность, но и просто асинхронное выполнение участков кода. Это обеспечивают такие ключевые слова, как «async / await», «promise», «future». Таким образом, современная программа может представлять собой сеть асинхронно функционирующих компонентов, возможно многопоточных, в которых некоторые участки кода исполняются также независимо и параллельно. При этом может вставать проблема определения момента завершения параллельной работы, что может оказаться намного сложнее самого действия. Например, можно запустить асинхронное сложение двух векторов, которое может быть

выполнено довольно быстро, но потом придётся дождаться результата подсчёта выполненных операций, если нам необходимо в дальнейшем распорядиться результирующим вектором как единым целым и это может существенно снизить эффект распараллеливания.

В современных языках нет понятия «кусочек памяти», с которым можно делать всё что угодно, как это делается в языке «С», где разрешено свободное преобразование типа указателя и адресная арифметика над указателями, что может приводить к непредсказуемым ошибкам. Вместо этого предусмотрена возможность оперировать полями объектов, которые могут быть описаны как союзы структур, включать массивы, но при этом осуществляется как контроль типов, так и контроль индексов, что делает программы надёжнее.

Требования к архитектуре микропроцессора пятого поколения

Как было отмечено выше, понятие «архитектура» включает в себя аппаратно-программный комплекс, обеспечивающий определённый уровень системной поддержки задачам, исполняющимся в системе. Аппаратная часть архитектур нового поколения соответствует новым технологиям производства, обеспечивающим повышение уровня интеграции микросхем, что приводит к повышению производительности при сохранении норм тепловыделения. Сейчас ведущий производитель TSMC освоил производство кристаллов по нормам 7 и 5 нм. Это открывает перспективы для создания архитектур с расширенными возможностями.

1. Поддержка многоплатформенности.

Современные облачные вычислительные ресурсы, интернет порталы сталкиваются с необходимостью поддержки виртуальных машин для различных языков программирования. Сейчас чаще всего используют несколько виртуальных машин с различными ОС. Но, если такая необходимость возникла у клиента, то для него устанавливать дополнительные виртуальные машины – проблематично. Подход, когда система сама разбирается в том, какой системный интерфейс нужен задаче, – гораздо перспективнее. Конечному пользователю не важно, в каком окружении идёт программа, – ему важно чтобы она работала. Учитывая, что ценность программного обеспечения велика, необходимо, наконец, обеспечить исполнение любой задачи, которая когда-либо была написана и отлажена. Примером бережного отношения к коду может служить IBM AS400/System I – в течение 30 лет, после многократной замены элементной базы, расширения набора платформ, ни одна программа не потребовала изменения программного кода. Это было достигнуто за счёт продуманного разделения аппаратной и виртуальной части системы.

Новое поколение микропроцессоров за счёт повышения производительности, параллельной эмуляции и JIT компиляции сможет

обеспечить выполнение задач, созданных для различных платформ как минимум без замедления. При этом у вычислительной системы будет достаточно ресурсов, чтобы обеспечить системную поддержку любой задаче. Предполагается, что нужные компоненты будут автоматически устанавливаться из облачных хранилищ.

2. Всесторонняя поддержка параллельных вычислений.

Очевидно, что процессор пятого поколения должен быть многоядерным и многопоточным. В зависимости от области применения он должен будет поддерживать параллельное и квазипараллельное одновременное выполнение сотен потоков, что позволит реализовать параллельную компиляцию, эмуляцию программ, созданных для различных платформ.

В первую очередь должна быть реализована аппаратная поддержка поток-безопасного доступа к глобальным объектам.

Второе – аппаратная поддержка исполнения множества параллельных ветвей внутри объекта. Это включает необходимость аппаратной синхронизации и аппаратной поддержки определения момента завершения множества параллельно исполняемых подмножеств команд и ветвей. Реализация этих требований предполагает новые аппаратные решения, в частности другую организацию подсистемы памяти микропроцессора и алгоритмов доступа к ней. Организация вычислений должна предполагать конвейерность, возможность «управления данными», message driven и т.д. Так, в проекте системы КСИ, сегменты памяти являются глобальными объектами, способными раздавать данные связанным с ними потокам. Эти объекты, располагаясь ближе к памяти, могут реализовывать заданную функциональность, например быть очередями, словарями, системными объектами со своими интерфейсами. Альтернативная организация подсистемы памяти исключает необходимость аппаратной поддержки когерентности кэш-памятей, что также существенно снижает накладные расходы на организацию параллельных вычислений.

3. Аппаратная поддержка ООП.

Современные языки программирования не оперируют понятием «глобальная память». Они используют ссылки на объекты и методы, которые модифицируют поля объектов, но это не ассоциируется с глобальной памятью. Для этого не обязательно использовать реальную адресную информацию, которая указывает на местоположение в общей памяти. Переход на виртуальную адресацию, не связанную с физической, не так страшен, как кажется, если есть альтернативные методы организации сохранения объектов в памяти – для этого могут служить объекты в сервере памяти, которые используют интерфейсы, предоставленные той ОС, к которой относится процесс. Эти объекты сами обеспечивают наличие в буферах кэш-памяти актуальной

информации и способны сами раздавать её потребителям, если это предусмотрено алгоритмом. Например, такой объект-сегмент памяти может реализовать «кучу» для виртуальной машины языка C, который в данной системе не является языком системного программирования.

Наряду с глобальными объектами – сегментами памяти можно с ними связывать один или несколько вычислительных потоков, способных обрабатывать сообщения в своём локальном контексте. Это не медленнее, чем получать данные из кэш-памятей внешних уровней. Разница же в том, что сообщения содержат меньше лишней информации и не требуют синхронизации доступа, поскольку не являются общими данными. Для программ, использующих ООП, инкапсуляция данных характерна. Аппаратная поддержка поток-безопасности обеспечит корректный доступ к эксклюзивным методам объектов.

4. Аппаратная поддержка графики и поиска.

Реализация многоядерной параллельной системы, каковой будет процессор пятого поколения, не исключает возможности добавления ядер графического сопроцессора, но если в обычных вычислительных ядрах поддержать основные функции графического конвейера, то это позволит использовать их в системах, не требовательных к качеству 3D-графики, но зато позволит избежать использования сторонних SDK при параллельных вычислениях.

Помимо этого, в ядрах должны быть поддержаны дополнительные операции, например операция поиска по ключу в векторах. Это пригодится, в частности, при n-арном поиске ключей в массивах.

Заключение

Концепция архитектуры микропроцессора пятого поколения изложена в работах [1–3]. Время показывает, что они становятся всё более актуальными, а прогресс в области развития аппаратуры и системного ПО показывает, что эти идеи осуществимы. Возможно, мы ещё увидим новые процессоры, частично удовлетворяющие изложенным требованиям в недалёком будущем.

Литература

1. Ширай А.Е. Системная поддержка вычислений в многоядерной конвейерной серверной архитектуре // Системы компьютерной математики и их приложения: материалы междунар. конф. Смоленск: СмолГУ. 2014. Вып. 15. С. 106-109.

2. Ширай А.Е., Гайдаенко Т.И., Провоторова А.О. Аппаратная поддержка векторно-матричной модели данных // Системы компьютерной математики и их приложения: материалы междунар. конф. Смоленск: СмолГУ, 2012. Вып. 13. С. 135-138.

3. Ширай А.Е. Использование объектно-ориентированного подхода в архитектуре многоядерного микропроцессора для ускорения параллельных конвейерных вычислений // Системы компьютерной математики и их приложения: материалы междунар. конф. – Смоленск: СмолГУ, 2011. Вып. 12.

4. <https://www.helicontech.com/zoo/>.

5. <https://tproger.ru/news/github-languages-rating-2018/>.
6. http://www.spscicomp.org/ScicomP16/presentations/Power7_Performance_Overview.pdf.
7. <https://habrahabr.ru/post/328348/>.
8. Ранганатан Судхакар MIPS – это прекрасная архитектура
http://www.electronics.ru/files/article_pdf/4/article_4710_247.pdf.
9. Anton Shilov TSMC-teams-up-with-arm-and-cadence-to-build-7-nm-chip-in-q1-2018
<https://www.anandtech.com/show/11832/tsmc-teams-up-with-arm-and-cadence-to-build-7-nm-chip-in-q1-2018>.
10. Алексей Снастин. Как использовать виртуальную машину Parrot
https://www.ibm.com/developerworks/ru/library/l-parrot_1/index.html.
11. ECMAScript® 2017 Language Specification
<https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf>.

A.E. Shiray
FRC CSC of RAS (Moscow)

STRUCTURE FEATURES OF KERNELS IN THE PIPELINE ARCHITECTURE V GENERATION

Keywords: *computing model, high level programming language, universal microprocessor-based architecture, parallel executing, memory server, thread safety, multi thread.*

Abstract. *This paper examines the requirements for a universal microprocessor-based architecture of the new generation implementing the computation model, designed to meet today's challenges and the relevant modern languages technologies of programming and the opportunities technology manufacturing equipment.*

А.В. Игнатенков, А.М. Ольшанский, И.А. Довгерд, Г.А. Довгерд
*АО «ВТБ Капитал», АО «НИИАС», ФГБОУ ВО «Московский
автомобильно-дорожный государственный технический университет
(МАДИ)»*

УДК 517.9+004,852+656.22

К ВОПРОСУ О ПОВЫШЕНИИ КАЧЕСТВА ОБУЧЕНИЯ МНОГОСЛОЙНЫХ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ С ПЕРЕМЕННОЙ ПРОВОДИМОСТЬЮ СИГНАЛА¹

Ключевые слова: *искусственная нейронная сеть, расписание, постобучение, ошибка.*

¹ Исследование проведено при поддержке гранта РФФИ 17-20-01065 «Разработка теории нейросетевого управления железнодорожными транспортными системами».

В статье описывается один из способов обучения многослойных искусственных нейронных сетей с переменной проводимостью сигнала с использованием постобучения. Предлагается также способ динамического обучения подобной сети.

Развитие нейросетевых технологий в мире обусловило рост применения нейронных сетей и к специфическим задачам на транспорте. Помимо способов, при которых нейросетевое решение сочетается в дальнейшем с использованием жадных алгоритмов [1], в АО «НИИАС» разработана многослойная искусственная сеть с переменной проводимостью сигнала, которая позволяет построить нормативное расписание движения поездов и даже может быть использована при оперативном управлении [2].

Задана многослойная нейронная сеть с переменной проводимостью, состоящая из n слоев, и входной вектор $X = \{x_0, x_1, \dots, x_m\}$, который обозначает моменты начала передачи сигналов по слоям сети.

Задан желаемый вектор выхода сигналов с последнего слоя сети, в качестве которого принимается желаемое значение выхода нулевого слоя $X^{(0)}$.

Работа такой сети происходит по общему алгоритму обучения любой сети с учетом того, что в процедурах обучения используются некоторые специфические подходы [3].

Одной из проблем при обучении такой сети является квазипериодический характер поведения функции ошибки, а также значительная по времени продолжительность обучения, связанная с необходимостью преодолевать периодические выбросы сигнала ошибки (рис. 1.)

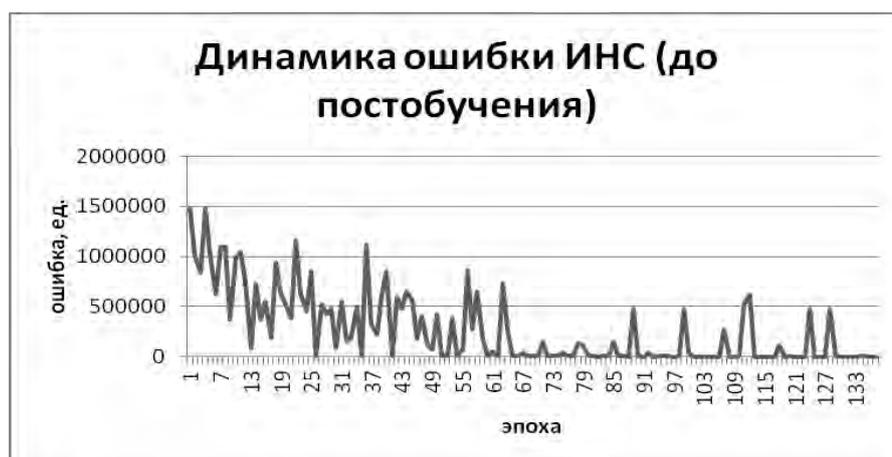


Рис. 1. Типичная динамика ошибки многослойной ИНС

Одним из приёмов решения вопроса избыточных периодически возникающих колебаний величины ошибки является введение контрольных ошибок, которые в дальнейшем применяются на процедуре

постобучения ИНС. Сеть начинает с больших величин ошибки, записывая их в контрольные ошибки. Это необходимо, чтобы в дальнейшем эти ошибки учитывались при последующих эпохах, чтобы ограничить динамику разброса ошибки. Во избежание сохранения большого массива ошибок и значений весов по нейронам и слоям для нейронной сети нами был ограничен буфер для записи только одного, лучшего значения ошибки. В дальнейшем при получении результата ошибки на текущей эпохе хуже, чем записанной контрольной, ИНС подвергается постобучению, пока не выйдет на результат контрольных ошибок или лучше.

После получения результатов ошибок 1250 эпох авторами был проведен их анализ, построен график, выведен тренд (рис. 2). Если сравнить графики динамики ошибки многослойной ИНС до и после постобучения, то можно сразу отметить уменьшение динамики разброса ошибки, переход к более стабильному положению сети, устойчивым точкам. Затем было проведено разложение графиков с использованием функции «STL» программной среды «RStudio» для более детального изучения, определения частоты встречаемости ошибки, выяснения причин резкого скачкообразного роста ошибки даже из относительно устойчивых положений.

Видно, что наблюдается значительная корреляция между выбросами сигнала ошибки в ряде остатков разложения и в периодической компоненте. При этом вклад тренда невысок (1/15 на начало обучения).

В связи с этим необходимо создать механизм управления состоянием ИНС, который, не фиксируя полный набор значений весов связей в каждую эпоху, позволил бы элиминировать влияние периодической и остаточной компонент. Такая задача может быть решена, например, с использованием подходов по синтезу оптимальных регуляторов.

В этих условиях предлагается альтернативный подход к обучению таких сетей, не требующий введения дополнительных сущностей, решения возникших новых задач и т.п.

О. Жерон в своей книге [4] описывает идею обучения ИНС посредством параллельного обучений двух сетей сразу, в которой вторая ИНС пробегает очень много эпох, вычисляя ошибки, сравнивая их с заданными параметрами. Если результаты получились удовлетворительными, тогда данные значения переносятся в первую ИНС. В этом случае вторая ИНС выступает в роли невидимых дублирующих нейронов основной сети, в которых проходят все вычисления и определение наилучшего результата.

Для представленной конструкции ИНС такой алгоритм изображен на рисунке 3.

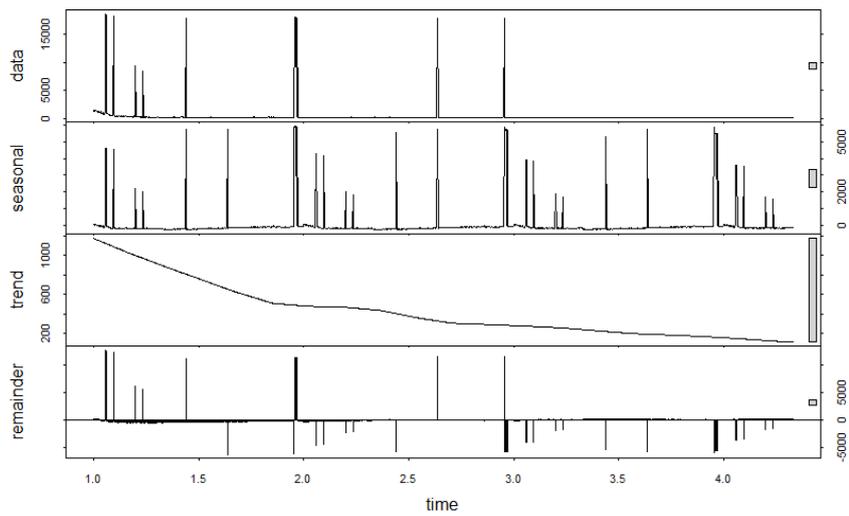


Рис. 2. STL-разложение сигнала ошибки ИНС с постобучением

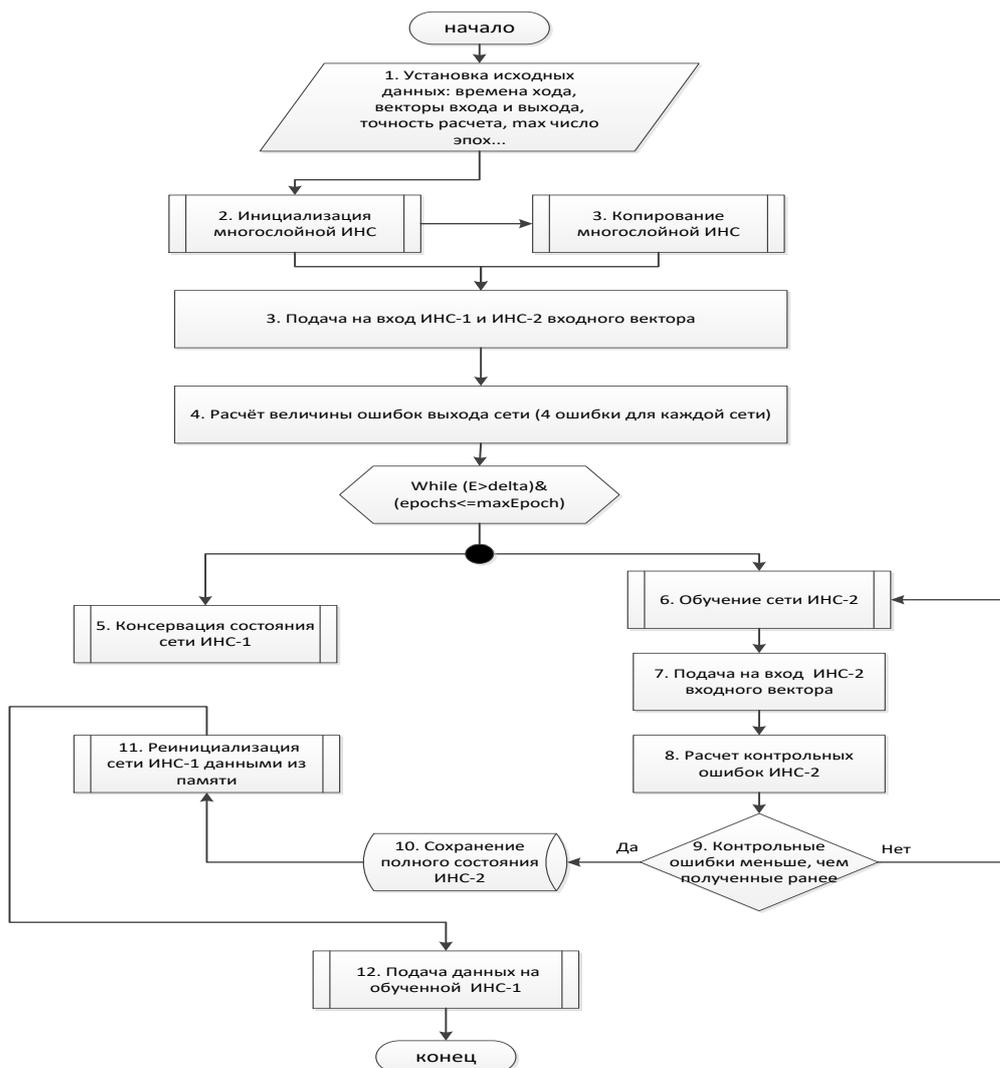


Рис. 3. Укрупненный алгоритм одновременного обучения двух сетей

Таким образом, в настоящей работе приведен ряд подходов, позволяющих повысить качество обучения многослойных искусственных сетей с переменной проводимостью сигнала без построения полной математической модели изменения величин ее весов.

При этом итоговое значение ошибок в конце обучения снижается до 3–50 единиц против 100–300, наблюдавшихся в предыдущих версиях нейросетевого ПО.

Литература

1. Костенко В.А., Винокуров А.В. Локально-оптимальные алгоритмы построения расписаний, основанные на использовании сетей Хопфилда // Программирование. 2003. № 4. С. 27–40.

2. Игнатенков, А.В., Ольшанский А.М. О применении искусственных нейронных сетей для оперативной корректировки расписаний // Системы компьютерной математики и их приложения материалы междунар. конф. Смоленск: СмолГУ. 2018. Вып. 19. С. 36-41.

3. Игнатенков А.В. Нейросетевое моделирование и упорядочение транспортных потоков на линиях железных дорог: автореф. дис.... канд. техн. наук. 05.13.18 – Математическое моделирование, численные методы и комплексы программ. Пенза, ПензГУ, 2018. 22 с.

4. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем: пер. с англ. СПб.: ООО «Альфа-книга», 2018. 688 с.

A.V. Ignatenkov, A. M. Olshansky, I.A. Dovgerd, G.A. Dovgerd
JSC “VTB Capital”, JSC “Railway Signalling Institute”, Moscow State
Automobile and Road Technical University

ABOUT IMPROVEMENT OF TRAINING QUALITY OF MULTILAYER NEURAL NETWORKS WITH VARIABLE SIGNAL CONDUCTIVITY

Keywords: *artificial neural network, schedule, post-training, control.*

Abstract. *The article is devoted to a new technique for multilayer artificial neural network’s training. The goal of the training is to reduce oscillations of the error signal using so-called post-training. Also an approach for dynamical training of such neural network is described.*

СВЕДЕНИЯ ОБ АВТОРАХ

Аветисов Юрий Гарриевич – ведущий инженер-программист ГУП «Литейно-прокатный завод», студент Смоленского государственного университета. E-mail: yury.avetisov@gmail.com.

Агасиев Талех Азер оглы – аспирант кафедры РК6 САПР МГТУ им. Н.Э. Баумана. E-mail: agtaleh@mail.ru.

Божинский Егор Андреевич – магистрант физико-математического факультета Смоленского государственного университета. E-mail: egor.bozhinsky@gmail.com

Борисов Вадим Владимирович – доктор технических наук, профессор, профессор кафедры «Вычислительная техника» филиала Национального исследовательского университета «МЭИ» в г. Смоленске. E-mail: vbor67@mail.ru.

Вишняков Рустам Рустамович – аспирант кафедры радиосистем передачи информации Московского технологического университета. E-mail: Vishn9-r@yandex.ru.

Гагарина Лариса Геннадьевна – профессор, доктор технических наук, директор института СПИНТех НИУ МИЭТ. E-mail: gagar@bk.ru.

Гайчиков Роман Леонидович – студент физико-математического факультета Смоленского государственного университета. E-mail: gaychikov2013@gmail.ru.

Гатауллин Табрис Талгатович – бакалавр Института математики и механики им. Н.И. Лобачевского Казанского федерального университета. E-mail: gataullin.tabris@gmail.com.

Грибер Юлия Александровна – доктор культурологии, профессор кафедры социологии, философии и работы с молодежью Смоленского государственного университета. E-mail: griber@gmail.com.

Григорьева Галина Михайловна – студентка бакалавриата физико-математического факультета Смоленского государственного университета. E-mail: grigoryevagal@gmail.com.

Голубенцов Александр Анатольевич – магистрант физико-математического факультета Смоленского государственного университета. E-mail: alexandr1295@mail.ru.

Гончаров Евгений Игоревич – студент Смоленского государственного университета. E-mail: drbenvey1996@mail.ru.

Городилов Алексей Владиславович – кандидат технических наук, ИП Каверина А.В. «Русская мода». E-mail: kaverina@mail.ru.

Демкович Александра Вячеславовна – студентка физико-математического факультета Смоленского государственного университета. E-mail: demkovich98@gmail.com.

Диев Олег Евгеньевич – Смоленский государственный университет, студент. E-mail: defarb@yandex.ru.

Довгерд Глеб Александрович – младший научный сотрудник сектора перспективных разработок Проектного конструкторско-технологического бюро АО «НИИАС», г. Москва; студент 4 курса факультета управления Московского автомобильно-дорожного государственного технического университета (МАДИ). E-mail: christmas1409@yandex.ru.

Довгерд Игнат Александрович – младший научный сотрудник сектора перспективных разработок Проектного конструкторско-технологического бюро АО «НИИАС», г. Москва; студент 4 курса факультета управления Московского автомобильно-дорожного государственного технического университета (МАДИ). E-mail: wolf1409@yandex.ru.

Евдокимова Галина Семёновна – доктор педагогических наук, профессор, заведующая кафедрой прикладной математики Смоленского государственного университета. E-mail: kaf-matem@smolgu.ru.

Емельченков Евгений Петрович – доцент, кандидат физико-математических наук, заведующий кафедрой информатики Смоленского государственного университета. E-mail: ypy1101@gmail.com.

Жарков Александр Андреевич – магистрант физико-математического факультета Смоленского государственного университета. E-mail: al.an.zharkov@ya.ru.

Жуков Денис Владимирович – начальник отдела в ОИП Федерального исследовательского центра «Информатика и управление» Российской академии наук» (ФИЦ ИУ РАН). E-mail: dzhukov@ipiran.ru.

Захаров Виктор Николаевич – доктор технических наук, доцент, ученый секретарь Федерального исследовательского центра

«Информатика и управление» Российской академии наук. E-mail: vzakharov@ipiran.ru.

Зуев Артем Сергеевич – магистрант физико-математического факультета Смоленского государственного университета. E-mail: edge95meloman@gmail.com.

Иванов Михаил Сергеевич – магистрант физико-математического факультета Смоленского государственного университета. E-mail: m_iv_1996@mail.ru.

Игнатенков Александр Владимирович – кандидат технических наук, дата-аналитик АО «ВТБ-капитал». E-mail: a.ignatenkov@gmail.com.

Ильин Павел Леонидович – студент второго курса физико-математического факультета Смоленского государственного университета. E-mail: zub.dayu@mail.ru.

Карпенко Анатолий Павлович – доктор физико-математических наук, профессор, заведующий кафедрой РК6 САПР МГТУ им. Н.Э. Баумана. E-mail: arkarpenko@mail.ru.

Кирикова Анастасия Викторовна – студентка физико-математического факультета Смоленского государственного университета. E-mail: anastasiakiricova52@gmail.com.

Кириллов Егор Васильевич – студент физико-математического факультета. E-mail: kiregr97@gmail.com.

Ковалева Наталья Юрьевна – магистрант физико-математического факультета Смоленского государственного университета. E-mail: nata_1997_01@mail.ru.

Козлов Сергей Валерьевич – кандидат педагогических наук, доцент, доцент кафедры информатики Смоленского государственного университета. E-mail: svkozlov1981@yandex.ru.

Колюсь Артур Николаевич – магистрант Смоленского государственного университета. E-mail: kolyus@mail.ru.

Конашенкова Татьяна Дмитриевна – ведущий программист Федерального исследовательского центра «Информатика и управление» Российской академии наук (ФИЦ ИУ РАН). E-mail: tkonzshenkova@ipiran.ru.

Кондратчикова Полина Игоревна – студент физико-математического факультета Смоленского государственного университета. E-mail: polinakondratchikova@gmail.com.

Кононова Александра Игоревна – кандидат технических наук, доцент кафедры ИПОВС МИЭТ. E-mail: illinc@bk.ru.

Корепанов Эдуард Рудольфович – кандидат технических наук, ведущий научный сотрудник Федерального исследовательского центра «Информатика и управление» Российской академии наук (ФИЦ ИУ РАН). E-mail: ekorepanov@ipiran.ru.

Кристалинский Владимир Романович – кандидат физико-математических наук, доцент, доцент кафедры информатики Смоленского государственного университета. E-mail: kristvr@rambler.ru.

Кристалинский Роман Ефимович – кандидат физико-математических наук, доцент, научный сотрудник научно-исследовательского центра Военной академии войсковой противовоздушной обороны Вооруженных Сил Российской Федерации. E-mail: kristalinskiyy-roman@rambler.ru.

Ланева Ирина Владимировна – кандидат экономических наук, доцент кафедры прикладной математики Санкт-Петербургского государственного университета морского и речного флота имени адмирала С.О. Макарова. E-mail: lanir@list.ru.

Макаров Александр Ильич – студент-магистр физико-математического факультета Смоленского государственного университета. E-mail: al.makarov8@gmail.com.

Мартышин Сергей Александрович – кандидат военных наук, начальник отдела научно-исследовательского центра Военной академии войсковой противовоздушной обороны Вооруженных Сил Российской Федерации.

Меженцев Игнат Олегович – студент физико-математического факультета Смоленского государственного университета. E-mail: ignarmez@gmail.com.

Мельник Клим Владимирович – студент физико-математического факультета. E-mail: blazerluna@gmail.com.

Миронов Артем Игоревич – студент бакалавриата физико-математического факультета Смоленского государственного университета. E-mail: 615153@mail.ru.

Мунерман Виктор Иосифович – кандидат технических наук, доцент кафедры информатики Смоленского государственного университета. E-mail: vimoona@gmail.com.

Мунерман Даниил Викторович – стажер-исследователь, Смоленский государственный университет. E-mail: danvmoon@gmail.com.

Нанкевич Алёна Анваровна – магистрант физико-математического факультета Смоленского государственного университета. E-mail: alyonka251p@gmail.com.

Недорезова Екатерина Андреевна – магистрант физико-математического факультета Смоленского государственного университета направления «Прикладные интернет-технологии». E-mail: e-nedorezova@mail.ru.

Нигмедзянова Айгуль Махмутовна – кандидат физико-математических наук, доцент кафедры высшей математики и математического моделирования Института математики и механики им. Н.И.Лобачевского Казанского Федерального университета. E-mail: aigmani@rambler.ru.

Ольшанский Алексей Михайлович – кандидат технических наук, начальник сектора перспективных разработок Проектного конструкторско-технологического бюро АО «НИИАС», г. Москва. E-mail: lexolshans@gmail.com.

Парамонов Алексей Анатольевич – доктор технических наук, профессор, заведующий кафедрой радиосистем передачи информации Московского технологического университета. E-mail: paramonov@mirea.ru.

Павлюков Алексей Игоревич – бакалавр физико-математического факультета Смоленского государственного университета. E-mail: pavluickov@yandex.ru.

Парфенова Елена Ивановна – студентка физико-математического факультета Смоленского государственного университета. E-mail: theirbis345@gmail.com.

Петров Евгений Николаевич – аспирант института СПИНТех НИУ МИЭТ. E-mail: fiddenmar@gmail.com.

Посыпкин Михаил Анатольевич – доктор физико-математических наук, главный научный сотрудник Федерального

исследовательского центра «Информатика и управление» Российской академии наук (ФИЦ ИУ РАН). E-mail: mposypkin@gmail.com.

Самойлова Татьяна Аркадьевна – кандидат технических наук, доцент кафедры информатики Смоленского государственного университета. E-mail: tatsamoilova@gmail.com.

Синицын Игорь Николаевич – доктор технических наук, профессор, главный научный сотрудник Федерального исследовательского центра «Информатика и управление» Российской академии наук (ФИЦ ИУ РАН). E-mail: sinitsin@dol.ru.

Солодченкова Татьяна Болеславовна – кандидат физико-математических наук, доцент кафедры физики и технических дисциплин Смоленского государственного университета. E-mail: tsolodchenkova@mail.ru.

Струнин Игорь Михайлович – бакалавр физико-математического факультета Смоленского государственного университета. E-mail: struninigor@mail.ru.

Суин Илья Алексеевич – магистрант физико-математического факультета Смоленского государственного университета. E-mail: suin224@yandex.ru.

Суханова Анна Геннадьевна – кандидат технических наук, доцент, доцент кафедры прикладной математики Смоленского государственного университета. E-mail: Ann-Sukhanova@yandex.ru.

Тарасевич Алёна Константиновна – магистрант физико-математического факультета Смоленского государственного университета. E-mail: tarasevichalena@mail.ru.

Тихонова Мария Ивановна – аспирант Высшей школы экономики факультета компьютерных наук, специалист по анализу данных Сбербанка. E-mail: m_tikhonova94@mail.ru.

Тихонова Ольга Вадимовна – доктор технических наук, старший научный сотрудник, профессор кафедры радиоэлектронных систем и комплексов РТУ МИРЭА. E-mail: o_tikhonova@inbox.ru.

Усачев Валерий Игнатьевич – доцент кафедры прикладной математики Смоленского государственного университета. E-mail: usa-vi@yandex.ru.

Федулов Александр Сергеевич – доктор технических наук, профессор, заведующий кафедрой «Вычислительная техника» филиала

Национального исследовательского университета «МЭИ» в г. Смоленске. E-mail: fedulov_a@mail.ru.

Фролов Дмитрий Сергеевич – программист, кафедра комплексного обеспечения информационной безопасности Санкт-Петербургского государственного университета морского и речного флота имени адмирала С.О. Макарова. E-mail: w.frolovds@gmail.com.

Хоанг Ван Зунг – аспирант кафедры радиоэлектронных систем и комплексов МИРЭА – Российского технологического университета. E-mail: tuandung@mail.ru.

Ходченков Валерий Юрьевич – студент бакалавриата физико-математического факультета Смоленского государственного университета. E-mail: tansdf@mail.ru.

Шалаби Набиль Камальэдинович – аспирант кафедры радиоэлектронных систем и комплексов Института радиотехнических и телекоммуникационных систем (РТС), РТУ – МИРЭА. E-mail: nabeel.shalaby@gmail.com

Ширай Александр Евгеньевич – Федерального исследовательского центра «Информатика и управление» Российской академии наук (ФИЦ ИУ РАН). E-mail: aesheerai@gmail.com.

Ястребов Михаил Юрьевич – кандидат экономических наук, профессор кафедры комплексного обеспечения информационной безопасности Санкт-Петербургского государственного университета морского и речного флота имени адмирала С.О. Макарова. E-mail: mikyast@gmail.com.

СОДЕРЖАНИЕ

СЕКЦИЯ 1. Системы компьютерной математики	3
<i>Аветисов Ю.Г., Кристалинский В.Р.</i> Использование нечетких методов при решении задач оптимизации раскроя	3
<i>Агасиев Т.А., Карпенко А.П.</i> Мета-оптимизация алгоритмов глобальной параметрической оптимизации	8
<i>Божинский Е.А., Евдокимова Г.С.</i> Статистический анализ рынка смартфонов за 2018 год	16
<i>Вишняков Р.Р.</i> Алгоритм структурно-параметрического синтеза радиотехнической системы	22
<i>Гатауллин Т.Т., Нигмедзянова А.М.</i> Исследование решений и построение фазовых портретов линейных автономных систем дифференциальных уравнений в СКМ Maple	28
<i>Городилов А.В., Кононова А.И.</i> Имитационное моделирование погрешности экспериментального исследования распределения длин путей между узлами в глобальной сети	34
<i>Евдокимова Г.С., Тарасевич А.К.</i> Применение нейронных сетей при анализе выживаемости больных раком желудка	42
<i>Кристалинский В.Р., Голубенцов А.А.</i> Нечеткая модель управления системой управления в умном доме	47
<i>Кристалинский В.Р., Колюсь А.Н.</i> О веб-сервисе для вычисления значений истинности нечетких высказываний	52
<i>Кристалинский Р.Е., Мартышин С.А.</i> Решение некоторых задач об оптимальном распределении ресурсов	57
<i>Солодченкова Т.Б., Гайчиков Р.Л.</i> Выбор критериев анализа качества изображения в методе цифровой голографии	61
<i>Суханова А.Г.</i> Моделирование нейронной сети для осуществления классификации инвестиционных портфелей в системе Mathcad	67
<i>Тихонова М.И.</i> Извлечение информации из резюме методами автоматической обработки текста	72
<i>Тихонова О.В.</i> Метод тестирования цифровой системы связи с помощью алгоритмического полунатурного моделирования	78
<i>Хоанг Ван Зунг, Парамонов А.А.</i> Эффективное использование частотно-энергетического ресурса в системах передачи информации с псевдослучайной перестройкой рабочей частоты при низкой скорости передачи	84
<i>Шалаби Н.К.</i> Многофункциональные групповые беспилотные авиационные	

системы с каналами радиосвязи миллиметрового диапазона	89
<i>Ястребов М.Ю., Ланева И.В., Фролов Д.С.</i> Показатели качества хеширования и анализ некоторых эвристических хеш-алгоритмов	98
СЕКЦИЯ 2. Параллельные вычисления и многоядерные процессоры	104
<i>Борисов В.В., Федулов А.С.</i> Модель и метод интеллектуального управления рисками нарушения информационной безопасности процессов в социотехнических системах	104
<i>Гончаров Е.И., Мунерман В.И., Самойлова Т.А.</i> Выбор параметров многомерных матриц для обобщенного алгоритма шифрования Хилла	111
<i>Грибер Ю.А., Самойлова Т.А.</i> Информационные технологии в социологическом исследовании цветоименований и цветовых предпочтений	116
<i>Григорьева Г.М., Миронов А.И., Мунерман В.И., Ходченков В.Ю.</i> Алгоритм сложения векторов на квантовом компьютере	124
<i>Григорьева Г.М., Ходченков В.Ю.</i> Реализация методов интеллектуального анализа данных в медицинской практике	129
<i>Диев О.Е., Мунерман В.И.</i> Контейнерный метод генерации элементов баз данных в распределённых системах	134
<i>Жарков А.А.</i> Один подход к повышению криптостойкости систем кодирования для интернета вещей	138
<i>Захаров В.Н.</i> О проблеме «P versus NP»	143
<i>Емельченков Е.П., Зуев А.С.</i> Об определении эквивалентности графических образов средствами мобильных ОС	152
<i>Иванов М.С., Мунерман В.И.</i> Генерация хранимых процедур для реализации алгоритма бустрофедона	157
<i>Ильин П.Л., Мунерман В.И.</i> Рекурсивное вычисление детерминанта многомерной матрицы	162
<i>Кирикова А.В., Мунерман В.И., Самойлова Т.А.</i> Реализация поиска изображений в базах данных	167
<i>Кириллов Е.В., Мельник К.В., Мунерман В.И.</i> Параллельная реализация вывода ассоциативных правил на основе NUMA-архитектуры	172
<i>Ковалева Н.Ю.</i> Модель мультиагентной системы логистического предприятия	177
<i>Козлов С.В., Демкович А.В.</i> Особенности разработки интернет-магазина продаж средствами MS Visual Studio	182
<i>Козлов С.В., Кондратчикова П.И.</i> Общие принципы разработки интернет-магазина товаров с использованием Java-технологий	189

<i>Козлов С.В., Меженцев И.О.</i> Взаимодействие языка программирования C# с межплатформенной средой разработки Unity	193
<i>Козлов С.В., Суин И.А.</i> О некоторых аспектах применения инвариантных методов функционального анализа данных в различных предметных областях	199
<i>Макаров А.И.</i> Проблема коэффициента информационной энтропии для показательного закона распределения	205
<i>Мунерман В.И., Мунерман Д.В.</i> Соответствие операций в многомерно-матричной и реляционной моделях данных	209
<i>Нанкевич А.А.</i> Оценка качества результатов систем машинного перевода	215
<i>Недорезова Е.А.</i> Обоснование выбора методов интеллектуальной обработки текстов с частично определенной семантикой	221
<i>Парфенова Е.И., Усачев В.И.</i> Разработка веб-сервиса для решения статистических задач с использованием электронных таблиц	228
<i>Петров Е.Н., Гагарина Л.Г.</i> Разработка методики классификации неструктурированной информации на примере библиографических данных с помощью условно-случайных полей	232
<i>Посыпкин М.А.</i> Алгоритмы аппроксимации неявно заданных областей	239
<i>Самойлова Т.А., Павлюков А.И.</i> Нейроуправление движением в реальном времени	244
<i>Самойлова Т.А., Струнин И.М.</i> Сегментация изображений методом маркерного водораздела	250
<i>Синицын И.Н., Жуков Д.В., Корепанов Э.Р., Конашенкова Т.Д.</i> Развитие прямых методов интерполяционного аналитического моделирования распределений в стохастических системах	256
<i>Ширай А.Е.</i> Требования к универсальной микропроцессорной архитектуре пятого поколения	261
<i>Игнатенков А.В., Ольшанский А.М., Довгерд И.А., Довгерд Г.А.</i> К вопросу о повышении качества обучения многослойных искусственных нейронных сетей с переменной проводимостью сигнала	269
Сведения об авторах	274

Системы компьютерной математики и их приложения

Выпуск 20

Часть 1

Издательство Смоленского государственного
университета

Редакторы *Л.В. Бушуева, О.В. Папко*

Подписано к печати 15.04.2019. Формат 60×84 1/16.

Бумага офсетная. Печать ризографическая.

Усл. п. л. 17,5. Уч.-изд. л. 17,5. Тираж 120 экз.

Заказ №

Отпечатано с оригинал-макета авторов в ИТЦ СмолГУ
214000, г. Смоленск, ул. Пржевальского, 4.