

ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ

ИУС  
МКМ 2019

**СБОРНИК МАТЕРИАЛОВ**

**Х МЕЖДУНАРОДНОЙ НАУЧНО-ТЕХНИЧЕСКОЙ КОНФЕРЕНЦИИ**

**“ИНФОРМАТИКА,  
УПРАВЛЯЮЩИЕ СИСТЕМЫ,  
МАТЕМАТИЧЕСКОЕ  
И КОМПЬЮТЕРНОЕ  
МОДЕЛИРОВАНИЕ”**

**В РАМКАХ V МЕЖДУНАРОДНОГО НАУЧНОГО ФОРУМА  
ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ**

**22-24 МАЯ 2019 Г.  
ДОНЕЦК**

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ**

**ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**ФАКУЛЬТЕТ  
КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ**

**ИНФОРМАТИКА, УПРАВЛЯЮЩИЕ СИСТЕМЫ,  
МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ  
МОДЕЛИРОВАНИЕ  
(ИУСМКМ-2019)**

**Материалы X Международной научно-технической  
конференции в рамках  
V Международного Научного форума  
Донецкой Народной Республики**

**22-24 мая 2019 г.**

Научное издание

**ИНФОРМАТИКА, УПРАВЛЯЮЩИЕ  
СИСТЕМЫ, МАТЕМАТИЧЕСКОЕ  
И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ  
(ИУСМКМ-2019)**

Материалы X Международной научно-технической  
конференции в рамках  
V Международного Научного форума  
Донецкой Народной Республики

22-24 мая 2019 г.

Web-сайт конференции: <http://www.iuskm.donntu.org>

УДК 004

Материалы X Международной научно-технической конференции «Информатика, управляющие системы, математическое и компьютерное моделирование» (ИУСМКМ-2019). – Донецк: ДОННТУ, 2019. – 146 с.

Рецензенты: Карабчевский В.В., к.т.н., доц., зав.каф. КМД; Павлыш В.Н., д.т.н., проф., зав.каф. ПМ; Николаенко Д.В., к.т.н., доц., доц. каф. КИ; Зори С.А., д.т.н., доц., проф. каф. ПИ; Павлий В.А., к.т.н., доц. каф. КМД; Орлов Ю.К., к.т.н., доц., доц. каф. ИИСА; Копытова О.М., к.ф-м.н., доц., доц. каф. ИИСА; Мальчева Р.В., к.т.н., доц., проф. каф. КИ, зам.декана по науке ФКНТ ГОУВПО ДонНТУ; Светличная В.А., к.т.н., доц., доц. каф. АСУ; Бельков Д.В., к.т.н., доц., доц. каф. ПМ; Григорьев А.В., к.т.н., доц., проф. каф. ПИ.

Сборник подготовлен по результатам X Международной научно-технической конференции «Информатика, управляющие системы, математическое и компьютерное моделирование», проведенной в рамках V Международного Научного форума Донецкой Народной Республики.

Организаторами конференции выступили Министерство образования и науки ДНР; ГОУ ВПО «Донецкий национальный технический университет» (ДонНТУ) факультет компьютерных наук и технологий (ФКНТ), кафедра компьютерного моделирования и дизайна (КМД); Полоцкий государственный университет (Республика Беларусь, г. Полоцк); Белгородский государственный национальный исследовательский университет (Российская Федерация, г. Белгород) и Технологический институт (филиал) Донского государственного технического университета (Российская Федерация, г. Азов).

Материалы, вошедшие в сборник, представлены научно-педагогическими сотрудниками, аспирантами, магистрантами и студентами высших учебных заведений из России, Беларуси, и ДНР.

Рекомендовано к публикации на заседании Ученого совета ДонНТУ.  
Протокол №5 от «28» июня 2019 г.

Адрес оргкомитета:

г. Донецк, ул. Кобозева, 17, Донецкий национальный технический университет, 4 учебный корпус, ФКНТ, кафедра КМД, ком. 4.41.

E-mail: [iuskm\\_19@mail.ru](mailto:iuskm_19@mail.ru)

## СОДЕРЖАНИЕ

стр.

<b>Интеллектуализация техносферы: закономерности и перспективы.</b> <i>Аноприенко А.Я.</i> .....	6
<b>Использование программных интерфейсов LibreOffice для вывода XML данных.</b> <i>Жданович П.Б.</i> .....	19
<b>Построение и свойства обобщенных тригонометрических систем нечетного порядка.</b> <i>Клово А.Г., Кузнецов А.Е., Кузнецов Р.Е., Чистякова Т.А.</i> .....	25
<b>Использование вейвлет-анализа для моделирования распространения информации в социальных сетях.</b> <i>Анохина И.Ю., Рощина Е.В.</i> .....	30
<b>Об одной задаче, связанной с булевыми матрицами</b> <i>Манилов Д.Ю., Решетников А.В.</i> .....	35
<b>Математическое моделирование техногенного месторождения с учетом развития горных работ.</b> <i>Прокопенко Е.В., Масло С.В., Гром А.В.</i> .....	38
<b>Свободные алгебры в классе полуабелевых n-групп.</b> <i>Щучкин Н.А.</i> .....	42
<b>Современные системы контроля версий ПО как инструмент разработки интеллектуальных САПР ПО.</b> <i>Гурин А.Г., Грищенко Д.А., Григорьев А.В.</i> .....	46
<b>Оценка эффективности методов и алгоритмов, используемых для решения задачи коммивояжера.</b> <i>Савкин В.Ю., Светличная В.А.</i> .....	52
<b>Анализ эффективности методов решения задачи о транспортировке груза.</b> <i>Бражник А.В., Секирин А.И., Валуева О.С.</i> .....	57
<b>Модифицированный алгоритма Виолы-Джонса для распознавания лиц в подсистеме идентификации личности посредством портретной экспертизы.</b> <i>Коношенко В.О., Секирин А.И., Валуева О.С.</i> .....	62
<b>Модификация генетического алгоритма для задачи составления перспективного и ежедневного меню в условиях младшего дошкольного воспитательного учреждения.</b> <i>Солоницын Л.П., Землянская С.Ю.</i> .....	67
<b>Особенности расчета и оптимизации учебной нагрузки в условиях АСУ вуза.</b> <i>Липова Э.Е., Секирин А.И.</i> .....	73
<b>Построение и использование твердотельной модели неправильной пирамиды в среде AutoCAD.</b> <i>Ермолаева А.А., Карабчевский В.В.</i> .....	79
<b>Компьютерное моделирование явления теплопроводности в ортотропной пластине при сосредоточенном воздействии источника тепла.</b> <i>Кравченко Т.М., Гольцев А.С.</i> .....	84
<b>Математическая модель плоских колебаний вибрационной машины с карданом Гука в приводе трансмиссии и ее реализация в среде математического моделирования MATLAB.</b> <i>Кухта С.С.</i> .....	89
<b>Использование адаптивного контроля величины шага при численном интегрировании быстро меняющихся функций вложенным методом Рунге-Кутты пятого порядка.</b> <i>Маглеванный И.И., Астахова Н.А., Карякина Т.И.</i> .....	94

<b>Обзор «Harmony» – инструмента для модификации C#-приложений в процессе исполнения.</b>	
<i>Вивденко В.С.</i> .....	99
<b>Разработка автоматической системы трансформации английских вставок в русских текстах с применением глубокого обучения.</b>	
<i>Пикалев Я.С., Ермоленко Т.В.</i> .....	102
<b>Моделирование случайной величины по экспоненциальному распределению в MATLAB и проверка по критерию хи-квадрат и критерию Колмогорова.</b>	
<i>Дубовик С.Ю., Бочаров Д.М.</i> .....	107
<b>Решение проблемы распознавания образов при помощи сверточных нейронных сетей.</b>	
<i>Колбасов С.Ю., Сорокин Р.А.</i> .....	111
<b>Использование методов интеллектуального анализа данных для оптимизации составления расписания в образовательных организациях.</b>	
<i>Пащенко Д.А., Маслова Е.А.</i> .....	115
<b>Разработка андроид-приложения учета и ведения заказов доставки еды.</b>	
<i>Радченко М.В., Сорокин Р.А.</i> .....	120
<b>Анализ эмоционального состояния человека по фотографическим изображениям.</b>	
<i>Миненко А.С., Семенова А.П.</i> .....	123
<b>Методологии тестирования программного обеспечения.</b>	
<i>Грабельников В.А.</i> .....	127
<b>Разработка системы диспетчеризации автоматизированной системы управления технологической линией.</b>	
<i>Дробитько А.В., Орлов Ю.К.</i> .....	131
<b>Разработка системы оценки результатов стрелковой тренировки.</b>	
<i>Тарасов Н.Ю., Тарасова И.А.</i> .....	136
<b>Моделирование и исследование задачи Стефана.</b>	
<i>Миненко А.С., Лёвкина А.В., Радевич Е.В.</i> .....	142

## ПРЕЗЕНТАЦИЯ ПЛЕНАРНОГО ДОКЛАДА

### ИНТЕЛЛЕКТУАЛИЗАЦИЯ ТЕХНОСФЕРЫ: ЗАКОНОМЕРНОСТИ И ПЕРСПЕКТИВЫ

**Анопrienко Александр Яковлевич**

Донецкий национальный технический университет,  
кафедра компьютерной инженерии

E-mail: [anoprien@ya.ru](mailto:anoprien@ya.ru)

#### *Аннотация*

*Анопrienко А. Я. Интеллектуализация техносферы: закономерности и перспективы. В докладе выполнен анализ системодинамики развития цифровых технологий в современную эпоху. Продемонстрирована тотальность экспоненциальных закономерностей не только в компьютерной сфере, но и в развитии человечества. Обозначены вызовы ближайших десятилетий.*

*Ключевые слова: Техносфера, закономерности, экспоненциальный рост, перспективы.*

#### *Annotation*

*Anoprienko A. Y. Intellectualization of the technosphere: regularities and prospects. The report analyzes the system dynamics of digital technologies in the modern era. The totality of exponential regularities not only in the computer sphere, but also in the development of mankind is demonstrated. The challenges of the coming decades are outlined.*

*Key words: Technosphere, regularities, exponential growth, prospects.*



«Информатика, управляющие системы, математическое  
и компьютерное моделирование – 2019» (ИУСМКМ-2019)

---

# Интеллектуализация техносферы: закономерности и перспективы

**Анопrienко Александр Яковлевич**

Заведующий кафедрой компьютерной инженерии

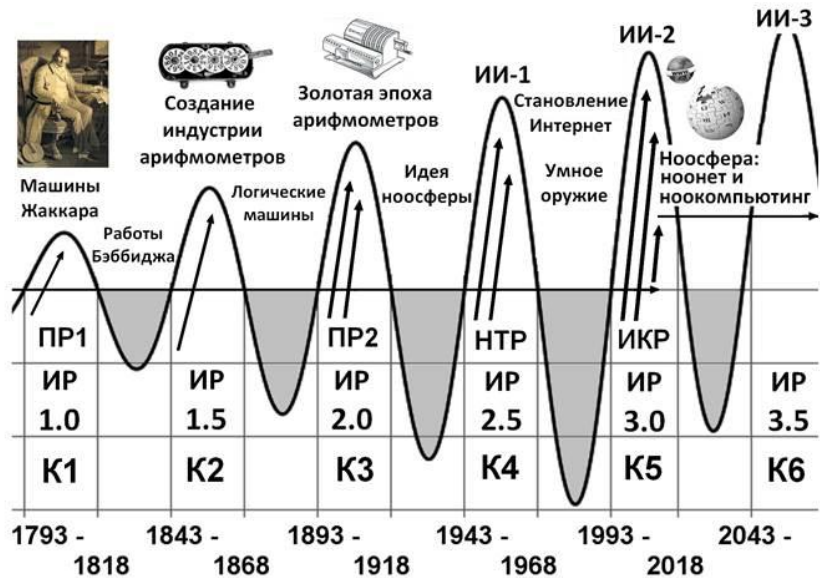
22.05.2019 БА3-1 ДонНТУ



## Интеллектуализация техносферы: закономерности и перспективы

### Системодинамика развития цифровых технологий в современную эпоху

Периодические закономерности на основе модифицированных циклов Кондратьева



2

Донецкий национальный технический университет

Кафедра компьютерной инженерии

Аноприенко Александр Яковлевич



## Интеллектуализация техносферы: закономерности и перспективы

### «Всеобщий закон Мура»: тотальность экспоненциальных закономерностей



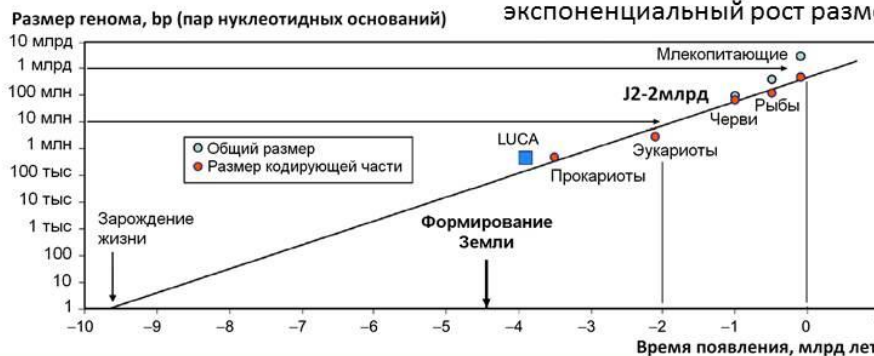
В основе экспоненциальных закономерностей лежит **кривая самообучения**

Единая система обозначения

темпов экспоненциальных процессов: **J2-2млрд**  
(количество порядков роста – базовый период)

Самый масштабный процесс:

экспоненциальный рост размера генома



3

Донецкий национальный технический университет

Кафедра компьютерной инженерии

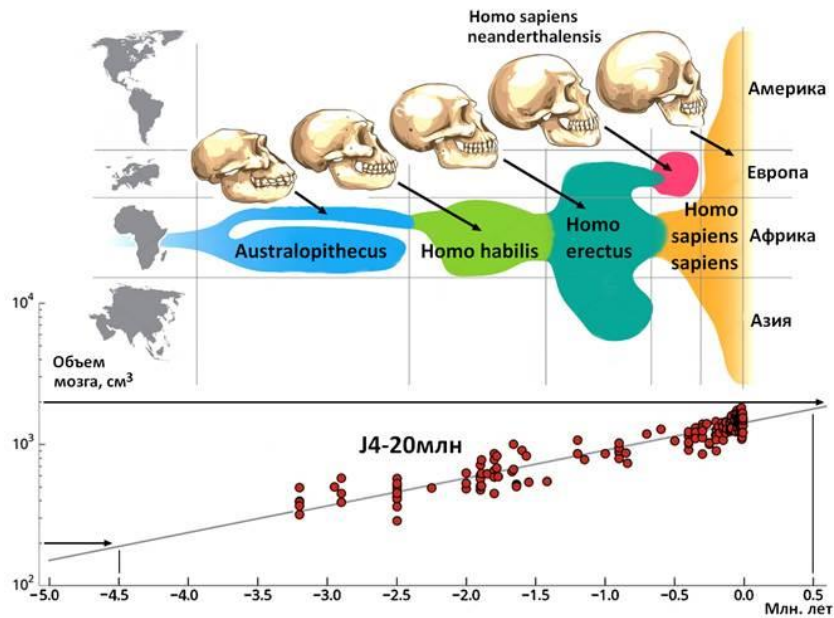
Аноприенко Александр Яковлевич





Интеллектуализация техносферы:  
закономерности и перспективы

Экспоненциальный рост размеров мозга человека: **J4-20млн**

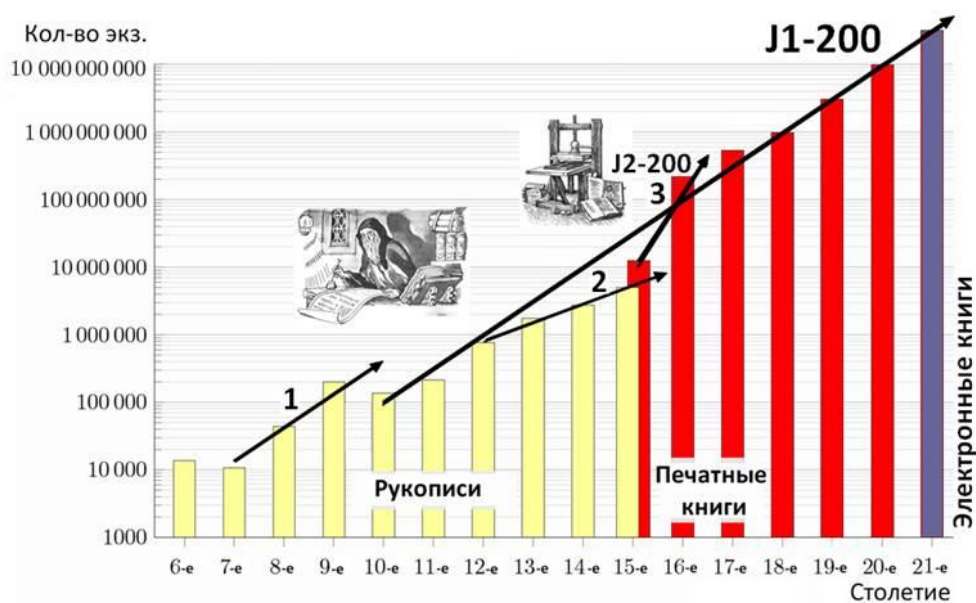


4



Интеллектуализация техносферы:  
закономерности и перспективы

Экспоненциальный рост тиражей книг: **J1-200**



5



## Интеллектуализация техносферы: закономерности и перспективы

Экспоненциальный рост грамотности: **J2-200**



Источник: Literate World Population (Our World In Data based on OECD and UNESCO)

6

Донецкий национальный технический университет

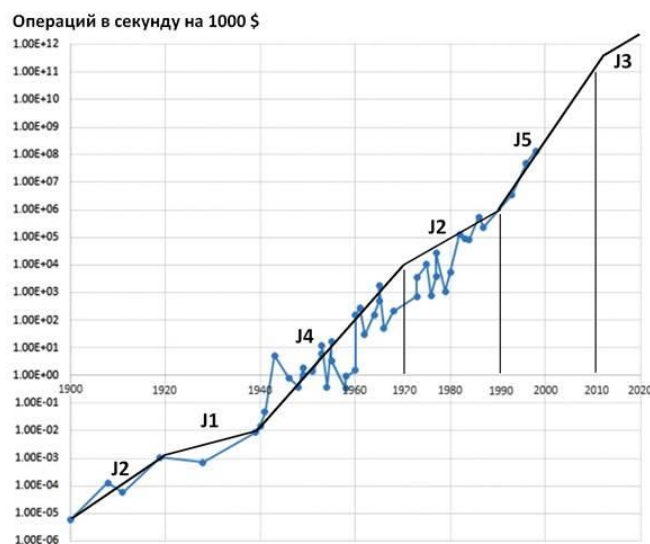
Кафедра компьютерной инженерии

Аноприенко Александр Яковлевич



## Интеллектуализация техносферы: закономерности и перспективы

Экспоненциальный рост производительности вычислений: **J1-J5**



7

Донецкий национальный технический университет

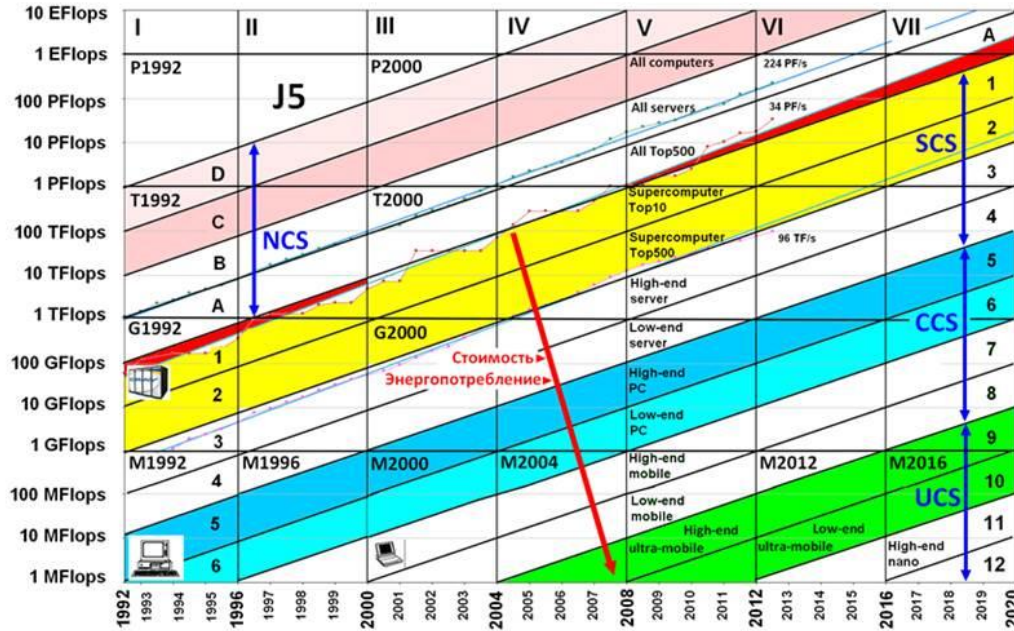
Кафедра компьютерной инженерии

Аноприенко Александр Яковлевич



### Интеллектуализация техносферы: закономерности и перспективы

Экспоненциальный рост производительности вычислений 1993-2014: **J5**



8

Донецкий национальный технический университет

Кафедра компьютерной инженерии

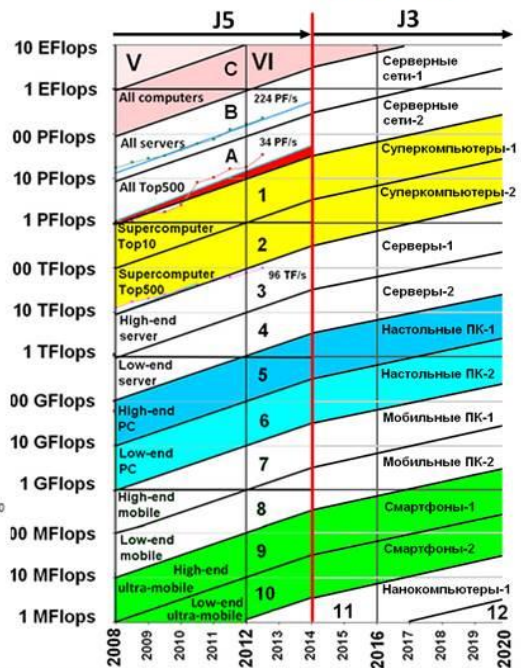
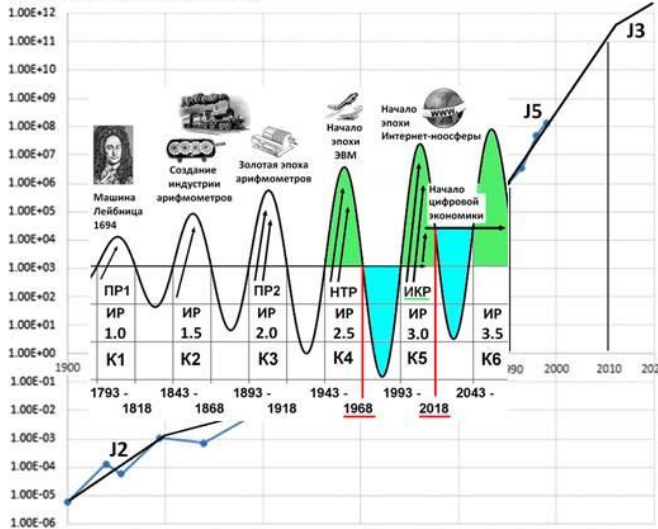
Аноприенко Александр Яковлевич



### Интеллектуализация техносферы: закономерности и перспективы

Экспоненциальный рост производительности вычислений 2014: **J5-J3**

Операций в секунду на 1000 \$



Донецкий национальный технический университет

Кафедра компьютерной инженерии

Аноприенко Александр Яковлевич



## Интеллектуализация техносферы: закономерности и перспективы

Экспоненциальный рост степени интеграции 1955-1968-2014: **J5-J4-J3**



10

Донецкий национальный технический университет

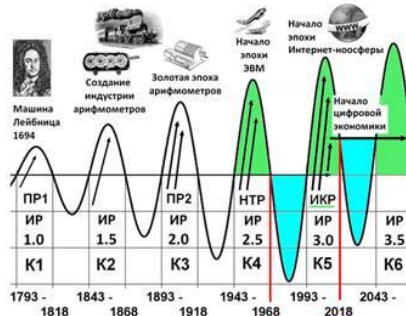
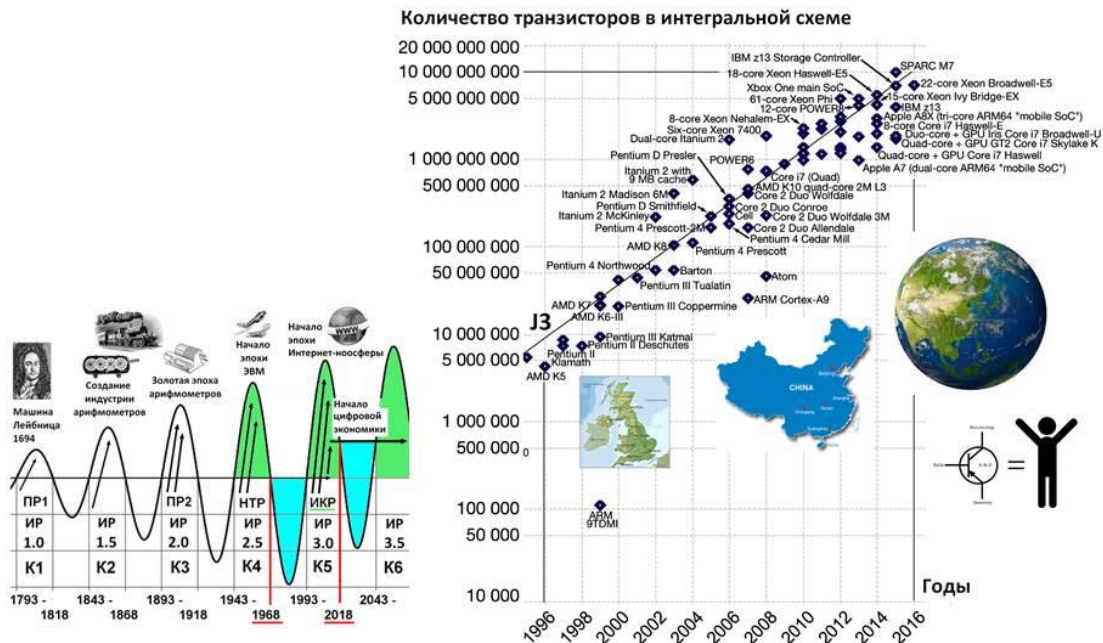
Кафедра компьютерной инженерии

Аноприенко Александр Яковлевич



## Интеллектуализация техносферы: закономерности и перспективы

Экспоненциальный рост степени интеграции 1993-2014: **J3**



Донецкий национальный технический университет

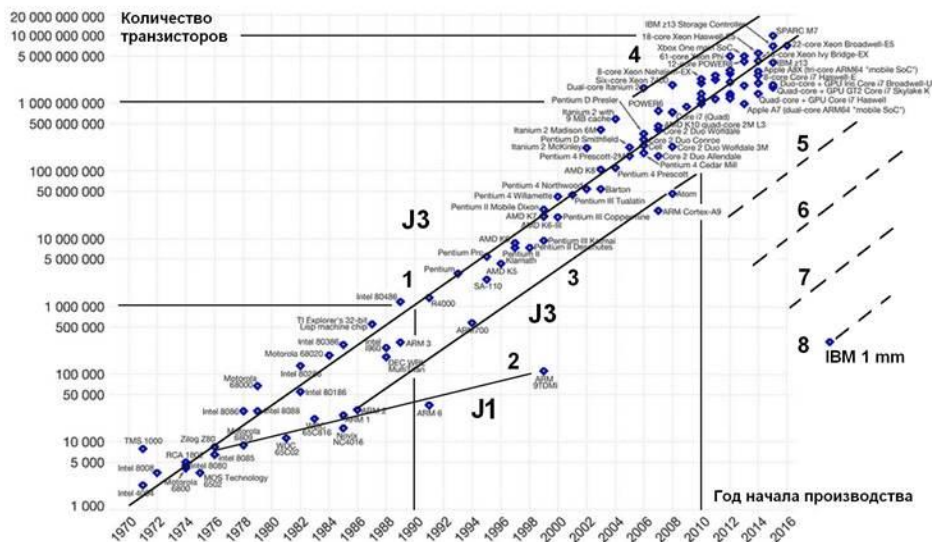
Кафедра компьютерной инженерии

Аноприенко Александр Яковлевич



## Интеллектуализация техносферы: закономерности и перспективы

Экспоненциальный рост степени интеграции 1993-2014: **J3 ???**

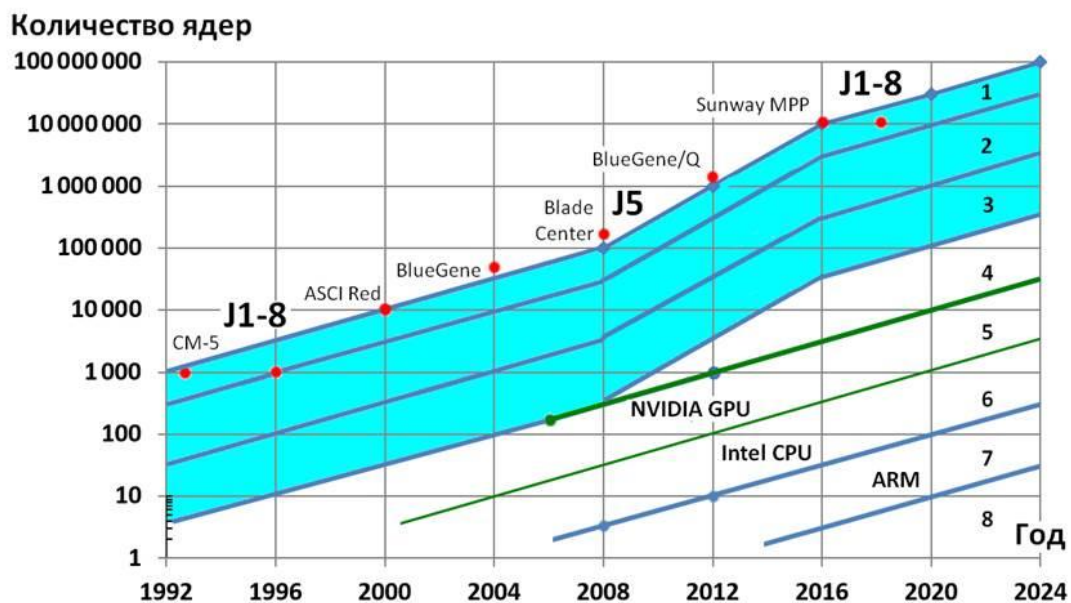


12



## Интеллектуализация техносферы: закономерности и перспективы

Экспоненциальный рост количества ядер в суперкомпьютерных системах

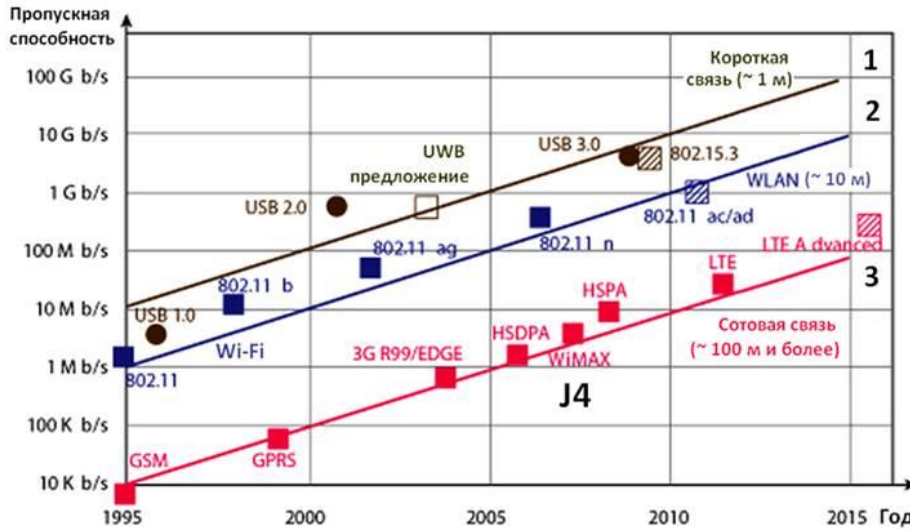


13



## Интеллектуализация техносферы: закономерности и перспективы

Экспоненциальный рост **связности глобальной системы 1993-2014: J4**



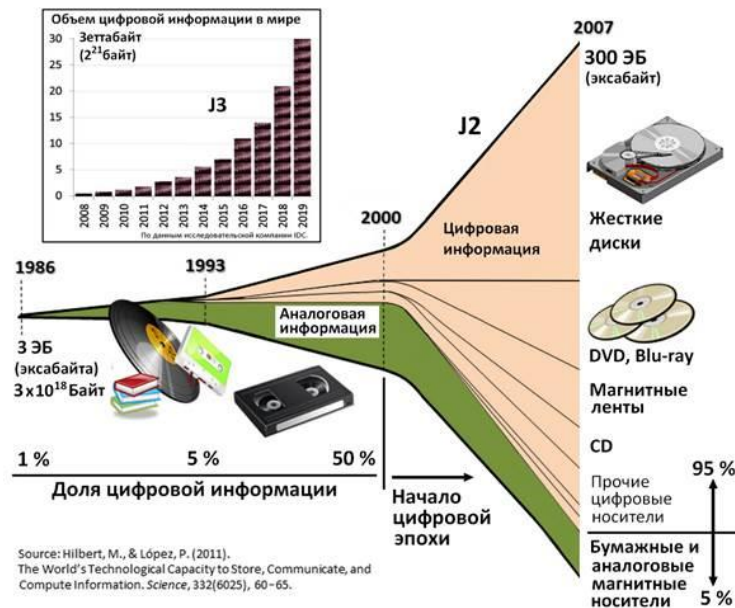
Сроки  
внедрения  
5G ???

14



## Интеллектуализация техносферы: закономерности и перспективы

Экспоненциальный рост **объемов информации 1993-2014: J2 (J3)**



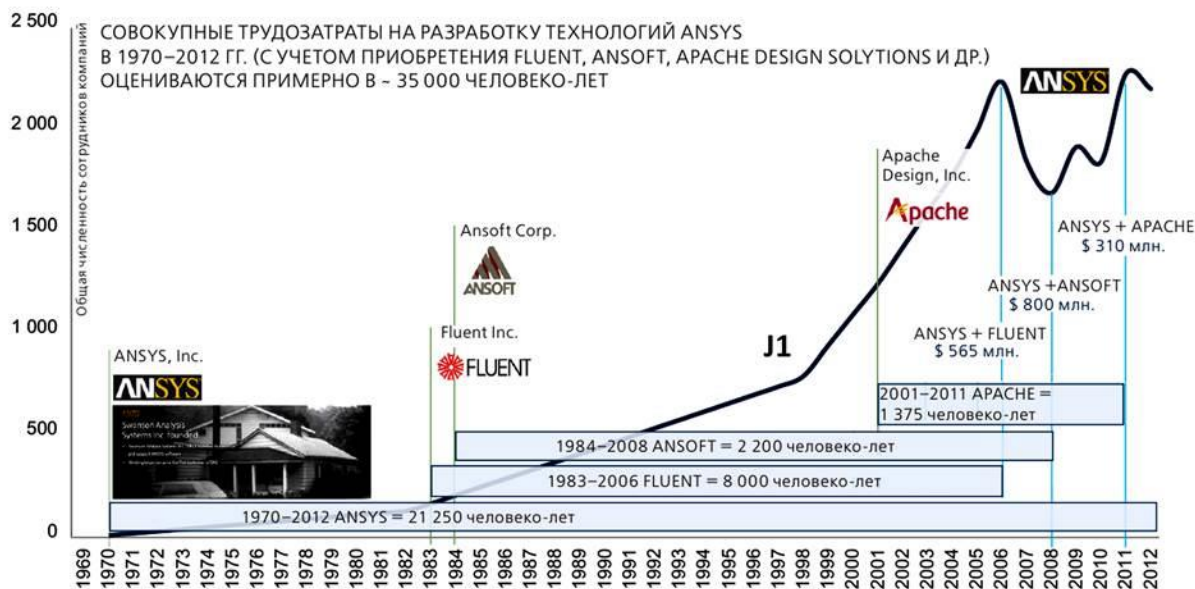
Source: Hilbert, M., & López, P. (2011).  
The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025), 60-65.

15



## Интеллектуализация техносферы: закономерности и перспективы

### Экспоненциальный рост концентрации знаний в САПР 1969-2014: J1



Донецкий национальный технический университет

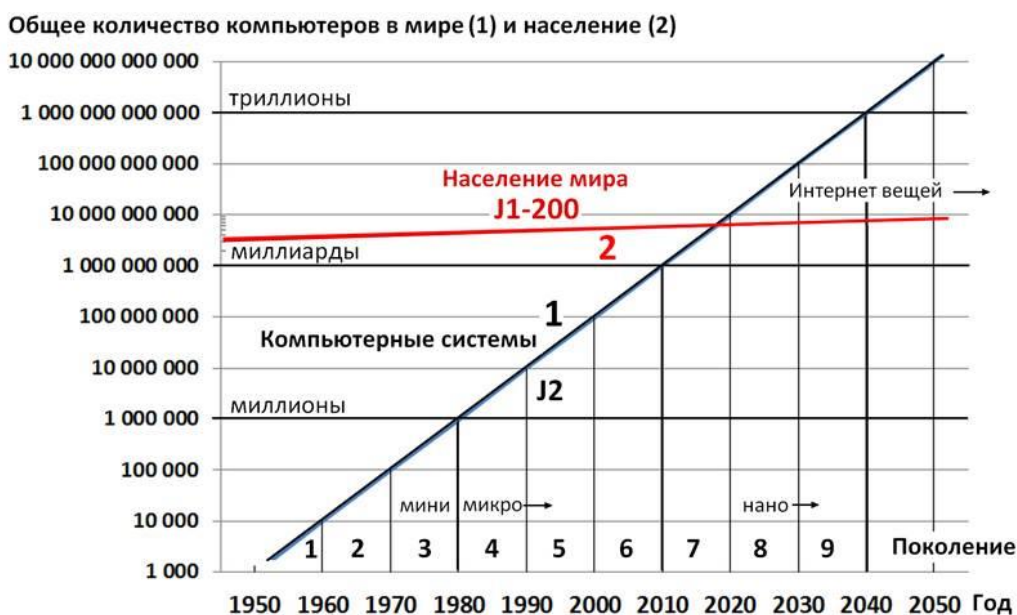
Кафедра компьютерной инженерии

Аноприенко Александр Яковлевич



## Интеллектуализация техносферы: закономерности и перспективы

### Экспоненциальный рост населения и компьютерных систем 1969-2014: J2



17

Донецкий национальный технический университет

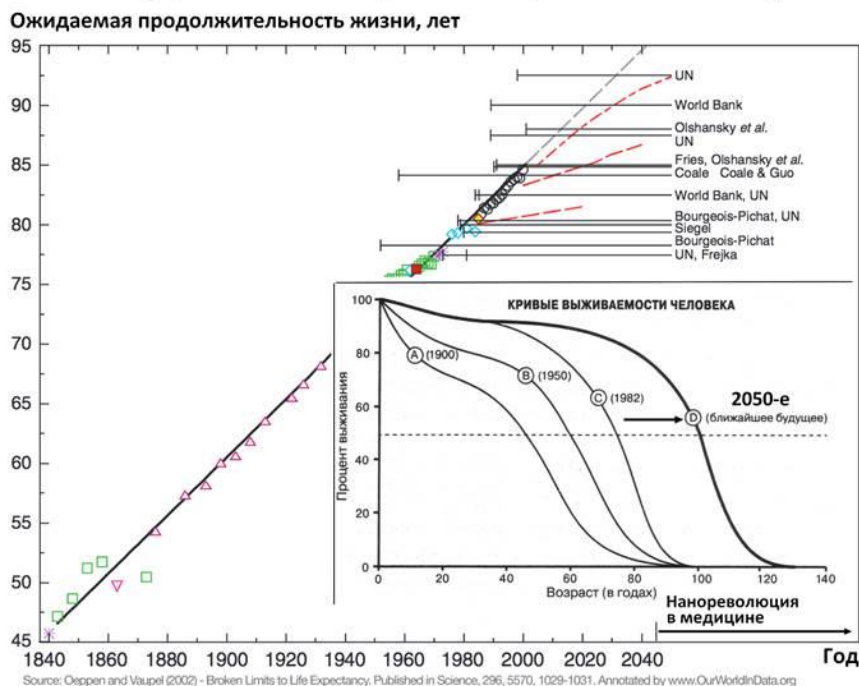
Кафедра компьютерной инженерии

Аноприенко Александр Яковлевич



## Интеллектуализация техносферы: закономерности и перспективы

### Социальный эффект экспоненциального роста компьютерных систем



18

Донецкий национальный технический университет

Кафедра компьютерной инженерии

Аноприенко Александр Яковлевич



## Интеллектуализация техносферы: закономерности и перспективы

Компьютерное расширение  
«Человеческого космоса»:

От Пикокосмоса  
до Теракосмоса

По аналогии со шкалой  
Кардашева-Кутца:

От докомпьютерной «Цивилизации 1»  
(зеленая)  
через современную компьютерную  
«Цивилизацию 2» (красная)  
к будущей «Цивилизации 3» (синяя)

Масштаб, м	Обозначение	1	2	3	Характерные объекты
1.E+27					Вселенная
1.E+26					
1.E+25	Иоттакосмос				
1.E+24	Им				
1.E+23					
1.E+22	Зеттакосмос				До ближайшей галактики Андромеды - 25 Зм
1.E+21	Зм				Диаметр галактики Млечный Путь - 1 Зм
1.E+20					
1.E+19	Энтакосмос				
1.E+18	Эм				Звезды Ориона (Ригель - 8 Зм)
1.E+17					Ближние звезды (Процион - 100 Пм)
1.E+16	Петтакосмос				Ближайшие звезды (Альфа Центавра - 40 Пм)
1.E+15	Пм				
1.E+14					
1.E+13	Теракосмос				Вояджер-1 (20 Тм) - солнечная система
1.E+12	Тм				Солнце-Сатурн (1,4 Тм)
1.E+11					Солнце-Земля (150 Гм)
1.E+10	Гигакосмос				Земля - Марс (55 Гм)
1.E+09	Гм				
1.E+08					Земля - Луна (376 Мм)
1.E+07	Мегакосмос				Высота геостационарной орбиты 35 Мм
1.E+06	Мм				Диаметр Земли 12 Мм
1.E+05					Орбиты пилотируемых полетов (300-500 Км)
1.E+04	Килокосмос				Стратосфера (10-50 Км)
1.E+03	Км				Тропосфера (1-10 Км) - авиация
1.E+02					
1.E+01	Макрокосмос				Макросреда
1.E+00	м				Человек
1.E-01	дм				BodyNet
1.E-02	см				
1.E-03	мм				
1.E-04					
1.E-05	Микрокосмос				Клеточный уровень
1.E-06	мкм				Бактерии
1.E-07					Вирусы
1.E-08	Нанокосмос				Молекулярный уровень
1.E-09	нм				Технологический предел 2020-х: 5-7 нм
1.E-10					Атомарный уровень (атом цезия - 450 пм)
1.E-11	Пикокосмос				Атомарный уровень (атом гелия - 62 пм)
1.E-12	пм				Субатомарный уровень
Диапазон порядков		10	20	30	

19

Донецкий национальный технический университет

Кафедра компьютерной инженерии

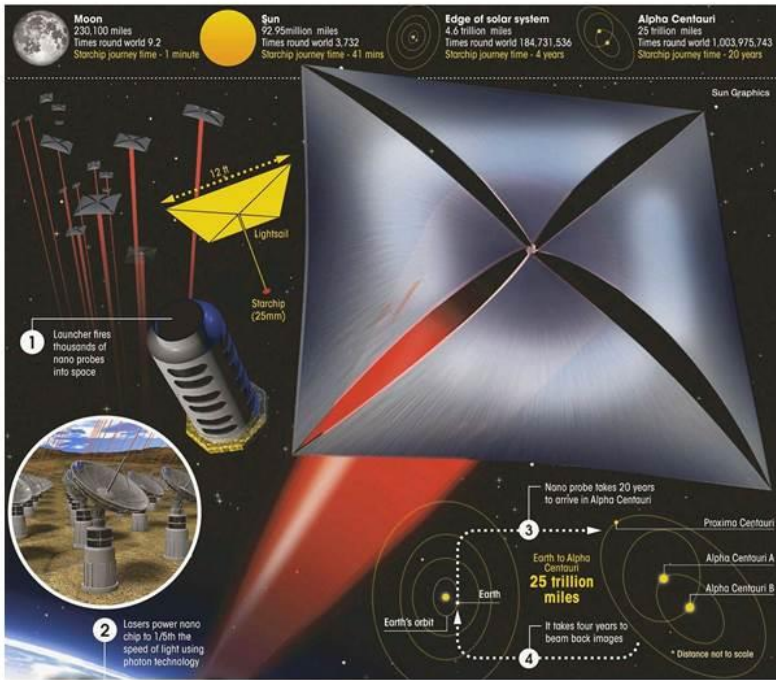
Аноприенко Александр Яковлевич





## Интеллектуализация техносферы: закономерности и перспективы

### Экспедиция микророботов к ближайшим звездам



значение	1	2	3	Характерные объекты
Иоттаносмос				Вселенная
Зеттаносмос				До ближайшей галактики Андромеды - 25 Зм Диаметр галактики Млечный Путь - 1 Зм
Экзаносмос				Звезды Ориона (Ригель - 8 Зм)
Петаносмос				Ближние звезды (Прокцион - 100 Пм) Ближайшие звезды (Альфа Центавра - 40 Пм)
Тераносмос				Вояджер-1 (20 Тм) - предел солнечной системы Солнце-Сатурн (1,4 Тм) Солнце-Земля (150 Гм), Солнце-Юпитер (780 Гм)
Гиганосмос				Солнце-Меркурий (57 Гм)
Меганосмос				Земля - Луна (376 Мм) Высота геостационарной орбиты 35 Мм Диаметр Земли 12 Мм
Килокосмос				Орбиты пилотируемых полетов (300-500 Км) Стратосфера (10-50 Км) Тропосфера (1-10 Км) - диапазон высот авиации
Макрокосмос				Макросреда
Миникосмос				Человек BodyNet
Микрокосмос				Клеточный уровень Бактерии Вирусы
Нанокосмос				Молекулярный уровень Технологический предел 2020-х: 5-7 нм Атомарный уровень (атом цезия - 450 пм) Атомарный уровень (атом гелия - 62 пм)
Пикокосмос				Субатомарный уровень
	10	20	30	

Донецкий национальный технический университет

Кафедра компьютерной инженерии

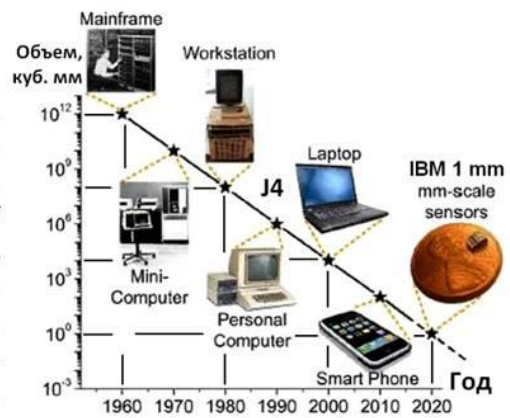
Аноприенко Александр Яковлевич



## Интеллектуализация техносферы: закономерности и перспективы

### Вызовы ближайших десятилетий:

### Компьютерное обеспечение **Интернета вещей**



21

Донецкий национальный технический университет

Кафедра компьютерной инженерии

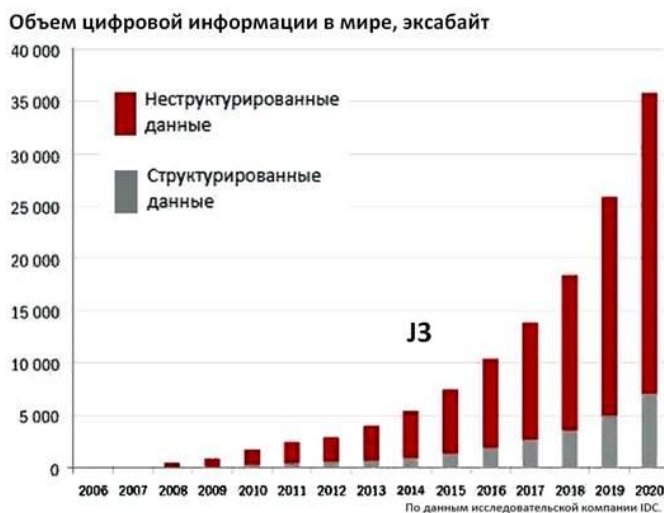
Аноприенко Александр Яковлевич



## Интеллектуализация техносферы: закономерности и перспективы

### Вызовы ближайших десятилетий:

Эффективная обработка **Больших данных** из миллиардов самых разнородных источников



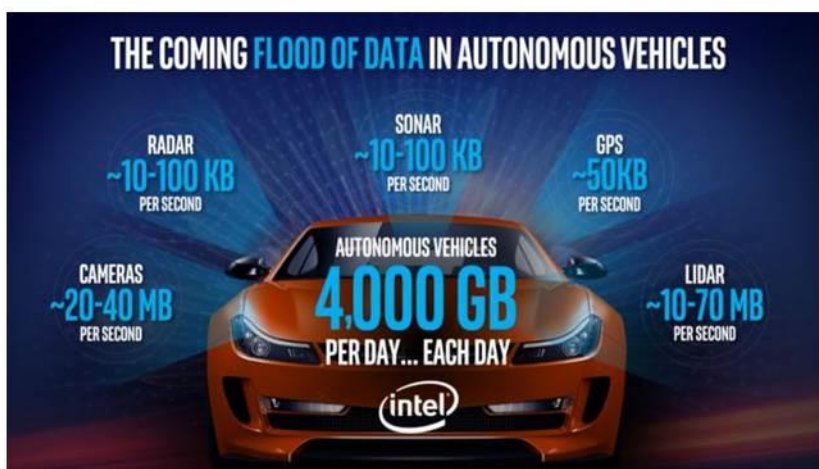
22



## Интеллектуализация техносферы: закономерности и перспективы

### Вызовы ближайших десятилетий:

Эффективная обработка **Больших данных** из миллиардов самых разнородных источников



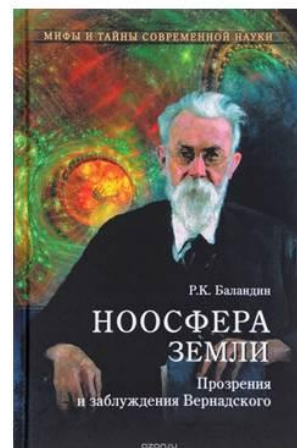
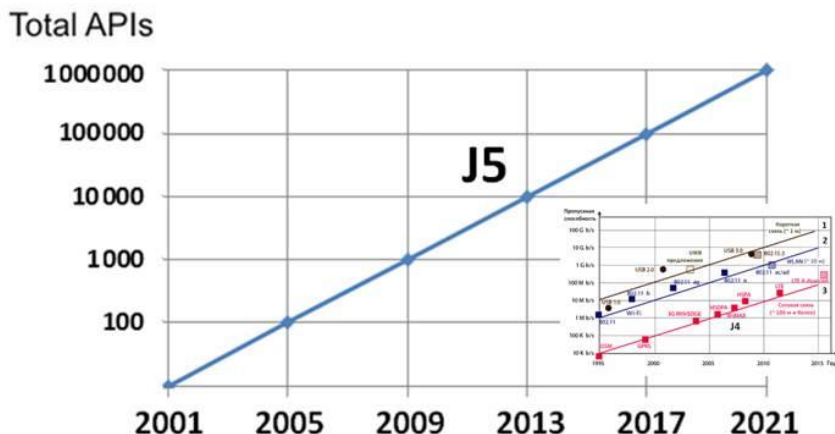
23



Интеллектуализация техносферы:  
закономерности и перспективы

Вызовы ближайших десятилетий:

**Ноокомпьютинг:** глобальная инфраструктура как единая платформа



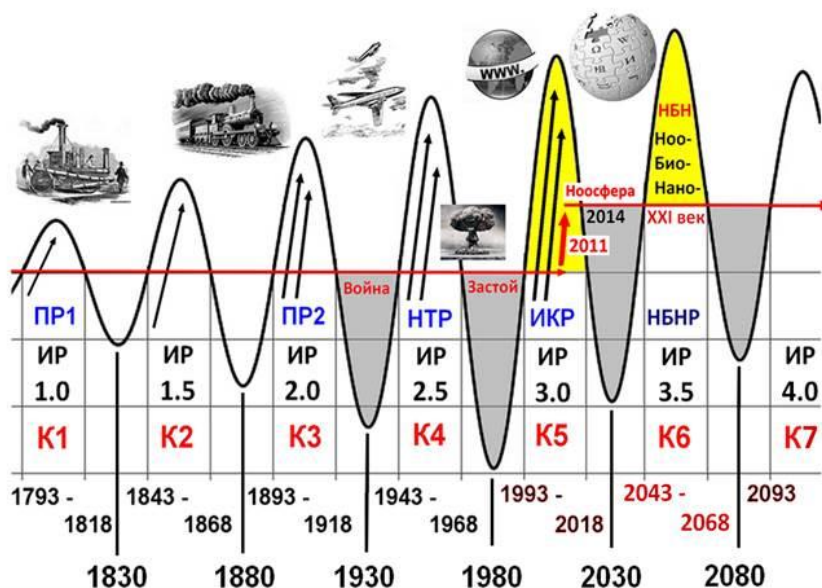
24



Интеллектуализация техносферы:  
закономерности и перспективы

Вызовы ближайших десятилетий:

модель мировой динамики на основе  
модифицированных волн Кондратьева (K1-K7)



25

## ПРЕЗЕНТАЦИЯ ДОКЛАДА

П.Б.Жданович, ВолгГМУ

### Использование программных интерфейсов LibreOffice для вывода XML-данных

ИУСМКМ-2019, Донецк

### Экосистема свободного ПО

- **Р.Столлман:**
  - Свободная операционная система
  - Свободный компилятор
- **Сегодня:**
  - Открытые протоколы
  - Свободные средства разработки
  - Свободные СУБД
  - Свободные генераторы отчетов

## OpenOffice/LibreOffice

- Развитая WYSIWYG-система
- Поддержка активного содержимого
- WYSIWYG
- Доступность
- Управляемость через UNO (аналогично COM для Microsoft Office)
- Headless Mode (для использования на стороне сервера)

## XML как универсальный способ представления данных

- ```
<test testdate="01.05.2017" testnum="345" testname="весенний семестр 2016-2017
уч.год" stages="2" stagename="2 семестр">
  <results>
    <result varnum="17" varname="Бег 100 м" value="16.10" interval="4"/>
    <result varnum="18" varname="Бег 2-3 км" value="11.59" interval="4"/>
    <result varnum="13" varname="Время 10 хлопков" value="6.00" interval="4"/>
    <result varnum="301" varname="Гибкость" value="20.00" interval="4"/>
    <result varnum="20" varname="Задержка дыхания" value="19.00" interval="2"/>
    <result varnum="14" varname="Прыжок в длину с места" value="180.00"
      interval="4"/>
    <result varnum="21" varname="Равновесие" value="25.00" interval="4"/>
    <result varnum="16" varname="Силовой норматив" value="50.00" interval="4"/>
    <result varnum="19" varname="Статическая выносливость" value="133.00"
      interval="4"/>
    <result varnum="15" varname="Челночный бег" value="25.30" interval="3"/>
  </results>
  +<grpresults></grpresults>
  +<streamresults></streamresults>
  <specresults></specresults>
</test>
```

## XPath

- ```
<test testdate="01.05.2017" testnum="345" testname="весенний семестр 2016-2017 уч.год" stages="2" stagename="2 семестр">
```

```
<results>
```

```
  <result varnum="17" varname="Бег 100 м" value="16.10" interval="4"/>
```

```
  <result varnum="18" varname="Бег 2-3 км" value="11.59" interval="4"/>
```

```
</test>
```
- /test/@testdate  
(равно 01.05.2017)
- /test/results/result[2]/@varnum (равно 17)
- /test/results/result[@varnum=17]/@value (равно 16.10)

## Экспорт в Calc

results    fx Σ = @varname

	A	B	C	D	E	F
1	<b>Результаты теста</b>					
2	<i>/test/@testname</i>					
3	<i>Дата проведения:</i>		<i>/test/@testdate</i>			
4	<b>Норматив</b>			<b>Результат</b>	<b>Коридор</b>	
5	<i>@varname</i>			<i>@value</i>	<i>@interval</i>	
6	<i>/test/results/result</i>					
7						

## Экспорт в Calc

A5:C14    fx    Σ    =    Бег 100 м

	A	B	C	D	E	F
1	<b>Результаты теста</b>					
2	<i>весенний семестр 2016-2017 уч.год</i>					
3	Дата проведения:	01.05.2017				
4	<b>Норматив</b>		<b>Результат</b>	<b>Коридор</b>		
5	Бег 100 м		16.1	4		
6	Бег 2-3 км		11.59	4		
7	Время 10 хлопков		6	4		
8	Гибкость		20	4		
9	Задержка дыхания		19	2		
10	Прыжок в длину с места		180	4		
11	Равновесие		25	4		
12	Силовой норматив		50	4		
13	Статическая выносливость		133	4		
14	Челночный бег		25.3	3		
15						

## Экспорт в Calc

D15    fx    Σ    =    =SUM(D5:D14)

	A	B	C	D	E	F
1	<b>Результаты теста</b>					
2	<i>весенний семестр 2016-2017 уч.год</i>					
3	Дата проведения:	01.05.2017				
4	<b>Норматив</b>		<b>Результат</b>	<b>Коридор</b>		
5	Бег 100 м		16.1	4		
6	Бег 2-3 км		11.59	4		
7	Время 10 хлопков		6	4		
8	Гибкость		20	4		
9	Задержка дыхания		19	2		
10	Прыжок в длину с места		180	4		
11	Равновесие		25	4		
12	Силовой норматив		50	4		
13	Статическая выносливость		133	4		
14	Челночный бег		25.3	3		
15		Итого	485.99			
16						

## Экспорт в Calc

```
protected static void t8_1()throws java.lang.Exception{
    Office office = new Office(new OfficeRuntime(),Office.VISIBLE);
    CalcDocument doc = office.openCalcDocument
        ("file:///home/zhd/tmp/testResults.ods",Boolean.FALSE);
    doc.save("file:///tmp/modified.ods");
    XML2ods c = new XML2ods(new XMLFileSource(
        new File("/home/zhd/tmp/testresults.xml")),doc);
    c.xml2Ranges();
    doc.save();
}
```

## Экспорт в Writer

### Результаты теста

от

<u>Норматив</u>	<u>Результат</u>	<u>Центильный коридор</u>

Преподаватель \_\_\_\_\_

Bookmark

Page	Name
1	}test}\$stestname
1	}test}\$stestdate



## Экспорт в Writer

Результаты теста  
“”  
от

Норматив	Результат	Центильный коридор

Преподаватель \_\_\_\_\_

Table Properties

Table Text Flow Columns Borders Background

Properties

Name /test/results/result#@varname@value@interval

Width 17.00 cm - +  Relative

Spacing

## Экспорт в Writer

Результаты теста  
“весенний семестр 2016-2017 учеб. год”  
от 01.05.2017

Норматив	Значение	Центильный коридор
Бег 100 м	16.10	4
Бег 2-3 км	11.59	4
Время 10 хлопков	6.00	4
Глубокость	20.00	4
Задержка дыхания	19.00	2
Прыжок в длину с места	180.00	4
Равновесие	25.00	4
Силовой норматив	50.00	4
Статическая выносливость	133.00	4
Челночный бег	25.30	3

Преподаватель \_\_\_\_\_

УДК 551.594, 51-7, 519.6

## ПОСТРОЕНИЕ И СВОЙСТВА ОБОБЩЕННЫХ ТРИГОНОМЕТРИЧЕСКИХ СИСТЕМ НЕЧЕТНОГО ПОРЯДКА

Клово А.Г., Кузнецов А.Е., Кузнецов Р.Е., Чистякова Т.А.

Южный федеральный университет,  
институт компьютерных технологий и информационной безопасности

E-mail: [klovo\\_ag@mail.ru](mailto:klovo_ag@mail.ru)

### **Аннотация:**

*Клово А.Г., Кузнецов А.Е., Кузнецов Р.Е., Чистякова Т.А. Построение и свойства обобщенных тригонометрических систем нечетного порядка. В работе рассматриваются некоторые вопросы, связанные с построением систем функций, обобщающих известные тригонометрические и гиперболические функции. В данной работе акцент делается на изучение свойств систем нечетного порядка.*

### **Annotation:**

*Klovo A. G., Kuznetsov A. E., Kuznetsov R. E., Chistyakova T. A. Construction and properties of generalized trigonometric systems of odd order. The paper deals with some issues related to the construction of systems of functions that generalize the known trigonometric and hyperbolic functions. In this paper, the emphasis is on the study of the properties of odd-order systems.*

### **Цель работы**

Ставится задача исследования свойств систем функций, обладающих тем свойством, что они при дифференцировании переходят друг в друга, причем производная последней функции набора переходит в первую функцию со знаком плюс или минус. Такими системами второго порядка являются тригонометрические и гиперболические функции. Функции из систем четвертого порядка использовались в работах по прочности конструкций, например, в работах [2], [4]- [6]. Общая теория таких систем четного порядка обсуждается в работах [1], [3]. В настоящей работе исследуются свойства таких систем нечетного порядка.

### **1. Преобразование ОТС первого типа во второй**

Функции  $K_{3,1}(x)$ ,  $K_{3,2}(x)$ ,  $K_{3,3}(x)$  образуют обобщенную тригонометрическую систему (о.т.с.[3;1]) третьего порядка первого типа, если выполнены однозначно

определяющие эту систему условия:  $(K_{3,1}(x))' = K_{3,2}(x)$ ,  $(K_{3,2}(x))' = K_{3,3}(x)$ ,  $(K_{3,3}(x))' = K_{3,1}(x)$ ,  $K_{3,1}(0) = K_{3,2}(0) = 0$ ,  $K_{3,3}(0) = 1$ .

Функции  $\Phi_{3,1}(x)$ ,  $\Phi_{3,2}(x)$ ,  $\Phi_{3,3}(x)$  образуют обобщенную тригонометрическую систему (о.т.с.[3;2]) третьего порядка второго типа, если выполнены однозначно

определяющие ее условия:  $(\Phi_{3,1}(x))' = \Phi_{3,2}(x)$ ,  $(\Phi_{3,2}(x))' = \Phi_{3,3}(x)$ ,  $(\Phi_{3,3}(x))' = -\Phi_{3,1}(x)$ ,  $\Phi_{3,1}(0) = \Phi_{3,2}(0) = 0$ ,  $\Phi_{3,3}(0) = 1$ .

Аналогично определяются, образующие обобщенную тригонометрическую систему (о.т.с.[5;1]) пятого порядка первого типа функции  $K_{5,1}(x)$ ,  $K_{5,1}(x)$ ,  $K_{5,1}(x)$ ,  $K_{5,1}(x)$ ,  $K_{5,1}(x)$ , последовательно переходящие друг в друга при взятии производных, причем производная последней функции равна первой из этой системы функции. В точке 0 последняя функция системы равна 1, остальные функции в точке 0 равны нулю.

Обобщенная тригонометрическая система (о.т.с.[5;2]) пятого порядка второго типа функции  $\Phi_{5,1}(x)$ ,  $\Phi_{5,1}(x)$ ,  $\Phi_{5,1}(x)$ ,  $\Phi_{5,1}(x)$ ,  $\Phi_{5,1}(x)$  отличается от первого типа системы условием  $(\Phi_{5,5}(x))' = -\Phi_{5,1}(x)$ .

Доказывается, что существуют квадратичные формы функций систем первого типа, образующие или порождающие системы второго типа того же порядка.

Для систем третьего порядка рассмотрим общий вид такой квадратичной формы

$$A(x) = a_{11}K_{3,1}^2(x) + a_{22}K_{3,2}^2(x) + a_{33}K_{3,3}^2(x) + 2a_{12}K_{3,1}(x)K_{3,2}(x) + 2a_{13}K_{3,1}(x)K_{3,3}(x) + 2a_{23}K_{3,2}(x)K_{3,3}(x) \quad (1)$$

и соответствующую матрицу

$$A_0 = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix}. \quad (2)$$

Производной  $A'(x)$  квадратичной формы (1) соответствует ее матрица

$$A_1 = \begin{pmatrix} 2a_{13} & a_{23} + a_{11} & a_{12} + a_{33} \\ a_{23} + a_{11} & 2a_{12} & a_{13} + a_{22} \\ a_{12} + a_{33} & a_{13} + a_{22} & 2a_{23} \end{pmatrix}, \quad (3)$$

производной  $A''(x)$  квадратичной формы (1) соответствует ее матрица  $A_2$ , а производной  $A'''(x)$  квадратичной формы (1) соответствует ее матрица  $A_3$ , задаваемая формулой

$$A_3 = \begin{pmatrix} 2(3a_{23} + a_{11}) & 5a_{12} + 3a_{33} & 5a_{13} + 3a_{22} \\ 5a_{12} + 3a_{33} & 2(3a_{13} + a_{22}) & 5a_{23} + 3a_{11} \\ 5a_{13} + 3a_{22} & 5a_{23} + 3a_{11} & 2(3a_{12} + a_{33}) \end{pmatrix}. \quad (4)$$

Очевидно, квадратичная форма (1) порождает обобщенную тригонометрическую систему тогда и только тогда, когда матрица (4) пропорциональна матрице (2), т.е. когда выполнены условия

$$\begin{aligned} \frac{2(3a_{23} + a_{11})}{a_{11}} &= \frac{2(3a_{13} + a_{22})}{a_{22}} = \frac{2(3a_{12} + a_{33})}{a_{33}} = \\ &= \frac{5a_{12} + 3a_{33}}{a_{12}} = \frac{5a_{13} + 3a_{22}}{a_{13}} = \frac{5a_{23} + 3a_{11}}{a_{23}} = \lambda. \end{aligned} \quad (5)$$

Система (5) распадается на 3 группы уравнений, имеющих относительно простые решения. Например, уравнение

$$\frac{2(3a_{23} + a_{11})}{a_{11}} = \frac{5a_{23} + 3a_{11}}{a_{23}} = \lambda$$

после замены  $\frac{a_{23}}{a_{11}} = t$  приводит нас к квадратному уравнению  $2t^2 - t - 1 = 0$  с корнями  $t_1 = 1$  и

$$t_2 = -\frac{1}{2}.$$

При  $t_1 = 1$  и, соответственно,  $\lambda_1 = 8$  мы еще раз получаем систему первого типа, записанную иным способом. Тем самым мы получаем тождества

$$\begin{aligned} K_{3,1}(x) &= K_{3,2}^2\left(\frac{x}{2}\right) + 2K_{3,1}\left(\frac{x}{2}\right)K_{3,3}\left(\frac{x}{2}\right) \\ K_{3,2}(x) &= K_{3,1}^2\left(\frac{x}{2}\right) + 2K_{3,2}\left(\frac{x}{2}\right)K_{3,3}\left(\frac{x}{2}\right). \\ K_{3,3}(x) &= K_{3,3}^2\left(\frac{x}{2}\right) + 2K_{3,1}\left(\frac{x}{2}\right)K_{3,2}\left(\frac{x}{2}\right). \end{aligned} \quad (6)$$

А вот при  $t_2 = -\frac{1}{2}$  и, соответственно,  $\lambda_2 = -1$  мы приходим к искомой системе второго типа, определенной равенствами

$$\begin{aligned} \Phi_{3,1}(x) &= K_{3,2}^2(x) - K_{3,1}(x)K_{3,3}(x) \\ \Phi_{3,2}(x) &= -K_{3,1}^2(x) + K_{3,2}(x)K_{3,3}(x). \\ \Phi_{3,3}(x) &= K_{3,3}^2(x) - K_{3,1}(x)K_{3,2}(x). \end{aligned} \quad (7)$$

Для систем пятого порядка запишем квадратичную форму

$$\begin{aligned} A(x) &= a_{11}K_{5,1}^2(x) + a_{22}K_{5,2}^2(x) + a_{33}K_{5,3}^2(x) + a_{44}K_{5,4}^2(x) + a_{55}K_{5,5}^2(x) + \\ &+ 2a_{12}K_{5,1}(x)K_{5,2}(x) + 2a_{13}K_{5,1}(x)K_{5,3}(x) + 2a_{14}K_{5,1}(x)K_{5,4}(x) + 2a_{15}K_{5,1}(x)K_{5,5}(x) + \\ &+ 2a_{23}K_{5,2}(x)K_{5,3}(x) + 2a_{24}K_{5,2}(x)K_{5,4}(x) + 2a_{25}K_{5,2}(x)K_{5,5}(x) + \\ &+ 2a_{34}K_{5,3}(x)K_{5,4}(x) + 2a_{35}K_{5,3}(x)K_{5,5}(x) + 2a_{45}K_{5,4}(x)K_{5,5}(x) \end{aligned} \quad (8)$$

и соответствующую матрицу

$$A_0 = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{12} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{13} & a_{23} & a_{33} & a_{34} & a_{35} \\ a_{14} & a_{24} & a_{34} & a_{44} & a_{45} \\ a_{15} & a_{25} & a_{35} & a_{45} & a_{55} \end{pmatrix}. \quad (9)$$

Производной  $A'(x)$  квадратичной формы (8) соответствует ее матрица

$$A_1 = \begin{pmatrix} 2a_{15} & a_{25} + a_{11} & a_{35} + a_{12} & a_{45} + a_{13} & a_{55} + a_{14} \\ a_{25} + a_{11} & 2a_{12} & a_{13} + a_{22} & a_{23} + a_{14} & a_{24} + a_{15} \\ a_{35} + a_{12} & a_{13} + a_{22} & 2a_{23} & a_{24} + a_{33} & a_{25} + a_{34} \\ a_{45} + a_{13} & a_{23} + a_{14} & a_{24} + a_{33} & 2a_{34} & a_{35} + a_{44} \\ a_{55} + a_{14} & a_{24} + a_{15} & a_{25} + a_{34} & a_{35} + a_{44} & 2a_{45} \end{pmatrix}. \quad (10)$$

Продолжая процесс, мы найдем матрицу, соответствующую пятой производной квадратичной формы (8)

$$\begin{pmatrix} 2(10a_{34} + 5a_{25} + a_{11}) & 15a_{35} + 7a_{12} + 10a_{44} & 15a_{45} + 12a_{13} + 5a_{22} & 15a_{23} + 12a_{14} + 5a_{55} & 15a_{24} + 7a_{15} + 10a_{33} \\ 15a_{35} + 7a_{12} + 10a_{44} & 2(10a_{45} + 5a_{13} + a_{22}) & 15a_{14} + 7a_{23} + 10a_{55} & 15a_{15} + 12a_{24} + 5a_{33} & 15a_{34} + 12a_{25} + 5a_{11} \\ 15a_{45} + 12a_{13} + 5a_{22} & 15a_{14} + 7a_{23} + 10a_{55} & 2(10a_{15} + 5a_{24} + a_{33}) & 15a_{25} + 7a_{34} + 10a_{11} & 15a_{12} + 12a_{35} + 5a_{44} \\ 15a_{23} + 12a_{14} + 5a_{55} & 15a_{15} + 12a_{24} + 5a_{33} & 15a_{25} + 7a_{34} + 10a_{11} & 2(10a_{12} + 5a_{35} + a_{44}) & 15a_{13} + 7a_{45} + 10a_{22} \\ 15a_{24} + 7a_{15} + 10a_{33} & 15a_{34} + 12a_{25} + 5a_{11} & 15a_{12} + 12a_{35} + 5a_{44} & 15a_{13} + 7a_{45} + 10a_{22} & 2(10a_{23} + 5a_{14} + a_{55}) \end{pmatrix}. \quad (11)$$

Условие пропорциональности матрицы (11) матрице (9) приведут нас к системе, которая распадается на автономные подсистемы, причем одна из них записывается в виде

$$\frac{2(10a_{34} + 5a_{25} + a_{11})}{a_{11}} = \frac{15a_{25} + 7a_{34} + 10a_{11}}{a_{34}} = \frac{15a_{34} + 12a_{25} + 5a_{11}}{a_{25}} = \lambda. \quad (12)$$

Эту систему удастся решить, обыграв тот факт, что сумма коэффициентов во всех числителях одинаковая. При этом для неизвестных  $u = \frac{a_{34}}{a_{11}}$ ,  $v = \frac{a_{25}}{a_{11}}$  получены 3 решения.

Первые два из них  $u_1 = v_1 = 1$ ,  $u_2 = \frac{-1 + \sqrt{5}}{4}$ ,  $v_2 = \frac{-1 - \sqrt{5}}{4}$  возвращают нас к системе первого типа, давая некоторые тождества, например

$$K_{5,1}^2(x) = K_{5,2}^2\left(\frac{x}{2}\right) + 2K_{5,1}\left(\frac{x}{2}\right)K_{5,3}\left(\frac{x}{2}\right) + 2K_{5,4}\left(\frac{x}{2}\right)K_{5,5}\left(\frac{x}{2}\right). \quad (13)$$

А вот третье решение системы (12)  $u_3 = \frac{-1 - \sqrt{5}}{4}$ ,  $v_3 = \frac{-1 + \sqrt{5}}{4}$  соответствует отрицательному  $\lambda$  и порождает *о.т.с.*[5; 2] с первым членом

$$\Phi_{5,1}(x) = \chi \left( K_{5,2}^2\left(\frac{x}{\mu^{1/5}}\right) + 0,5(\sqrt{5} - 1)K_{5,1}\left(\frac{x}{\mu^{1/5}}\right)K_{5,3}\left(\frac{x}{\mu^{1/5}}\right) - 0,5(\sqrt{5} + 1)K_{5,4}\left(\frac{x}{\mu^{1/5}}\right)K_{5,5}\left(\frac{x}{\mu^{1/5}}\right) \right), \quad (14)$$

где  $\mu = 2,5\sqrt{5} + 5,5$ ,  $\chi = \mu^{4/5}(3,5 - 1,5\sqrt{5})$ .

## 2. Тригонометрия систем нечетного порядка

Для систем второго порядка, т.е. в школьной тригонометрии существует большое число формул. Эти формулы обобщаются в работах [1], [3] на случай обобщенных тригонометрических систем четного порядка. Отметим некоторые установленные факты для систем нечетного порядка.

Рассмотрим функцию  $K_{3,1}(x+y)$ , зависящую от аргумента  $x$  и параметра  $y$ . Она удовлетворяет дифференциальному уравнению  $y''' = y$  вместе с функциями  $K_{3,1}(x)$ ,  $K_{3,2}(x)$ ,  $K_{3,3}(x)$ . В то же время вронсиан этих трех функций в точке 0 равен

$$\begin{vmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{vmatrix} = -1,$$

т.е. отличен от нуля. Поэтому

$$K_{3,1}(x+y) = C_1(y)K_{3,1}(x) + C_2(y)K_{3,2}(x) + C_3(y)K_{3,3}(x). \quad (14)$$

Подставляя в полученной равенство  $x=0$  сразу и дважды после последовательного дифференцирования (14), мы приходим к равенству

$$K_{3,1}(x+y) = K_{3,1}(x)K_{3,3}(y) + K_{3,2}(x)K_{3,2}(y) + K_{3,3}(x)K_{3,1}(y).$$

Этим и другими способами может быть получена большая серия формул. Приведем некоторые из них:

$$\Phi_{3,1}(x+y) = \Phi_{3,1}(x)\Phi_{3,3}(y) + \Phi_{3,2}(x)\Phi_{3,2}(y) + \Phi_{3,3}(x)\Phi_{3,1}(y),$$

$$K_{3,1}(-x) = \Phi_{3,1}(x), \quad K_{3,2}(-x) = -\Phi_{3,2}(x), \quad K_{3,3}(-x) = \Phi_{3,3}(x),$$

$$\int K_{3,1}^2(x)dx = K_{3,1}(x)K_{3,3}(x) - \frac{1}{2}K_{3,2}^2(x) + C.$$

$$\int \Phi_{3,1}^2(x)dx = -\Phi_{3,1}(x)\Phi_{3,3}(x) + \frac{1}{2}\Phi_{3,2}^2(x) + C.$$

### Литература

1. Клово А.Г. Задачи на собственные значения для одного линейного параметризованного оператора. Деп. ВИНТИ № 5428-80, 24 дек. 1980, 34с.
2. Клово А.Г. Некоторые вопросы колебаний кварцевых резонаторов с одним держателем. Физические основы микроэлектроники. Сб. трудов МИЭТа, М., 1979, стр.58-65.
3. Клово А.Г. Некоторые свойства обобщенных тригонометрических систем. Донецкие чтения 2018: образование, наука, инновации, культура и вызовы современности. Материалы III Международной научной конференции 25 октября 2018 года, г. Донецк. Том 1 Физико-математические и технические науки, с. 349-350.
4. Крылов А.Н. О расчете балок, лежащих на упругом основании. Л., изд. АН СССР, 1931.
5. Пузыревский Н.П. Расчеты фундаментов. Петроград, изд. И.И.П.С., 1923.
6. Филиппов А.П. Методы расчета сооружений на колебания. М., Л., Стройиздат, Наркомстром, 1940.

УДК 519.876.5

## ИСПОЛЬЗОВАНИЕ ВЕЙВЛЕТ-АНАЛИЗА ДЛЯ МОДЕЛИРОВАНИЯ РАСПРОСТРАНЕНИЯ ИНФОРМАЦИИ В СОЦИАЛЬНЫХ СЕТЯХ

**Анохина И.Ю., Рощина Е.В.**

Донецкий национальный технический университет  
кафедра прикладной математики  
E-mail: [ingatula@mail.ru](mailto:ingatula@mail.ru)

### *Аннотация:*

*Анохина И.Ю., Рощина Е.В. Использование вейвлет-анализа для моделирования распространения информации в социальных сетях. Обоснована актуальность исследования социальных сетей. Рассмотрены различные модели, описывающие процесс распространения информации в сети. Показана целесообразность предварительного применения вейвлет-преобразований для работы с временными рядами.*

### *Annotation:*

*Anokhina I.Yu., Roshchina E.V. Using wavelet analysis to model the dissemination of information in social networks. The relevance of the research of social networks is grounded. Considered various models that describe the process of disseminating information in the network. The expediency of preliminary use of wavelet transforms for working with time series is shown.*

**Введение.** Согласно [отчету](#) Digital in 2019, составленному аналитиками агентства We Are Social, Всемирная сеть насчитывает 4,39 миллиарда пользователей, что на 366 миллионов больше, чем в 2018 году. В среднем пользователи проводят в Интернете 6 часов 42 минуты в день, а 45% населения мира в настоящее время являются пользователями социальных сетей (более 3,5 миллиардов человек) [1].

По данным исследований, проведенных Mail.ru Group к 25-летию Рунета (апрель 2019г.), 81% россиян используют Интернет ежедневно и многократно в течение дня, 64% опрошенных используют интернет в основном для общения в социальных сетях [2]. Для сравнения, в 2000г. на вопрос о пользовании Интернетом положительно ответили только 3.6% респондентов.

Исследования, проведенные специалистами «Левада-Центр», показали, что 65% россиян, пользующихся соцсетями, предпочитают «ВКонтакте», 63% — «Одноклассники», 23% — Instagram [3].

Благодаря огромной аудитории пользователей социальные сети стали инструментом влияния. Анализ социальных сетей используется бизнес-аналитиками для оценки предпочтений клиентов, в политической сфере для продвижения идей. Оценивая групповые предпочтения, можно выявить тренды интересов и приоритетов для решения стратегических задач.

Основным инструментом массового донесения и распространения информации на сегодняшний день является сеть Интернет, в которой для этого используются социальные сети, сообщества, форумы, средства массовой информации, персональные блоги. Через них происходит формирование личного и общественного мнения.

Вся информация хранится в сети в открытом доступе, пользователи делятся мнениями в текстовом виде и через рейтинговые системы (отметки «нравится», «поделиться»). Процесс распространения информации происходит по двум направлениям: распространение через СМИ и через группы в социальных сетях.

**Постановка проблемы.** Распространение информации – процесс, посредством которого некоторый информационный объект распространяется по коммуникационным каналам во времени и в пространстве среди узлов сети[4].

Существует несколько различных подходов к моделированию процесса распространения информации.

Как правило, процесс распространения информации предполагает три этапа: формирование базы распространения (медленный рост), резкий рост и третий этап – насыщение и спад интереса к информации.

Новая информация поступает в социальную сеть через агентов изменений (gatekeepers), а затем постепенно воспринимается другими агентами, которые передают информацию дальше. На рис.1 показано, как происходит распространение информации. Внешнее воздействие, допустим, информация, поступившая из СМИ, приводит к активации узлов сети. Как отдельная группа, так и индивидuum, получают информацию и передают ее по сети, при этом образуется путь.

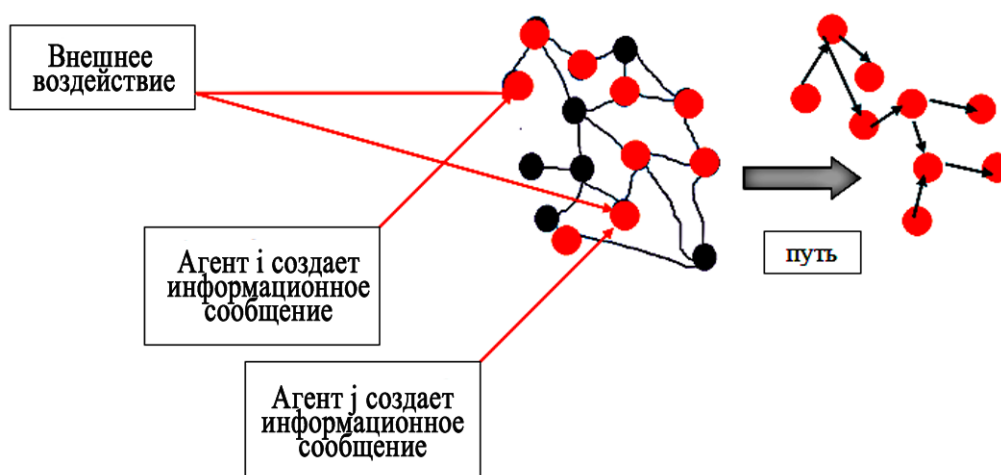


Рис. 1. Схема распространения информации в сети

Одним из методов моделирования процесса распространения информации является моделирование, базирующееся на аналогиях с физикой и медициной и получившее название модели просачивания и заражения или эпидемии [5, 6]. Скорости распространения информации высоки по аналогии с обычной эпидемией, при условии, что информация вызвала всеобщий интерес. Распространение начинается с небольшого числа групп и постепенно переходит на все большее их число, достигает пика, далее следует спад.

24 апреля 2019г. был подписан указ «Об определении в гуманитарных целях категорий лиц, имеющих право обратиться с заявлениями о приеме в гражданство Российской Федерации в упрощённом порядке». Информация соответствовала требованиям актуальности, важности, вызвала интерес не только на территориях ДНР и ЛНР, России, но и во многих странах мира [7].

Была поставлена задача смоделировать процесс распространения информации в социальных сетях, оценить временные характеристики, определить интенсивность распространения информации.

**Исследования.** При описании процессов распространения информации в сети используются несколько моделей:

- модель эпидемии **SIR** (susceptible – infected – removed) описывает передачу эпидемии (информации) от одного агента к другому с определенным параметром затухания. Состояние агента (это может быть как пользователь, так и сообщество в сети) описывают как уязвимое (предрасположенное к информации), зараженное (активное участие в обсуждении, распространении информации), невосприимчивое



(потеря интереса к новости и дальнейшее нежелание распространять её или отсутствие интереса вообще). Модель характеризуется частотой и интенсивностью заражения, параметром затухания (скорость «выздоровления»).

– Расширенная модель **SIRS** допускает, что выздоровевший вновь становится восприимчивым к болезни через некоторое время. Пример – заболевание гриппом. В социальной сети блоггер может прочитать статью (восприимчив), а затем написать об этой теме (инфицирован), и позже вернуться к ней еще раз (восприимчив); при обсуждении в группе один и тот же вопрос поднимается многократно, часть сообщества участвует в дискуссии повторно.

– Модель **клеточного автомата** используется для анализа распространения инноваций и новостей в Интернете. Здесь пользователь/группа соответствует одной клетке с двумя состояниями: 1 – новость принята, 0 – новость не принята. Группа принимает решение о принятии новости, ориентируясь на мнение ближайших соседей. В модели вводится понятие порогового значения, определяющего, принимает клетка/группа новость или нет [4]. Чтобы новость была принята и появилась на странице группы, необходимо выполнение условия

$$p \times m > R,$$

где  $m$  – количество групп, опубликовавших новость,  $p$  – вероятность принятия новости,  $R$  – пороговое значение.

Для разработки модели были собраны статистические данные за первые три дня после первой публикации информации об указе в СМИ. За это время новость была отображена на страницах 1298 групп ВКонтакте. Ее просмотрели около 140 тысяч пользователей Контакта. В табл.1 показано распределение активности пользователей за три дня. Учтено количество просмотров, лайки, репосты и число групп, опубликовавших новость.

Таблица 1 – Активность пользователей

День	Просмотры	Лайки	Репосты	Число групп
3	19.5%	15%	36%	43%
2	44.3%	49%	44%	34%
1	36.2%	36%	21%	23%

На рис.2 показано, как изменялась область распространения информации в течении трех суток, за 100% принят суммарный охват пользователей за эти три дня.

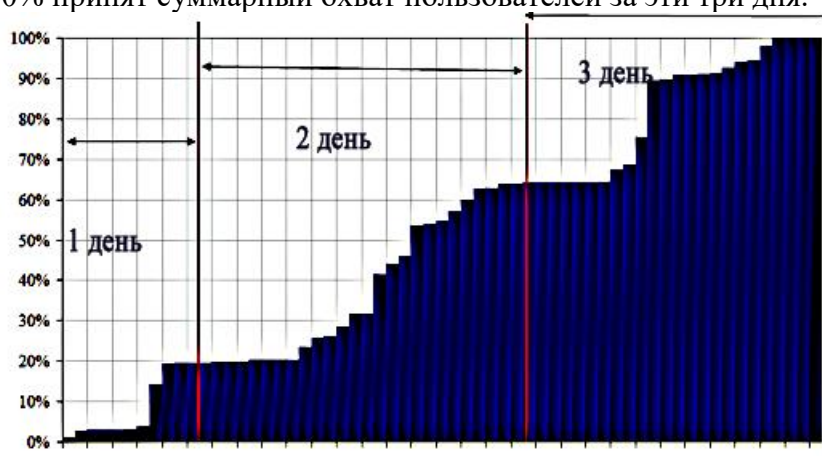


Рис. 2. Область распространения информации

Для анализа временных процессов распространения информации были использованы методы временного прогнозирования и модуль Time Series and Forecasting пакета Statistica. Однако, сначала данные были обработаны с помощью вейвлет-преобразований.

Вейвлет-анализ является одной из самых перспективных технологий анализа данных и находит применение в различных сферах интеллектуальной деятельности.

Термин «вейвлет» (wavelet) в переводе с английского означает «маленькая (короткая) волна». Вейвлеты - название семейств математических функций определенной формы, которые локальны во времени и по частоте, таким образом, при анализе мы рассматриваем временной ряд в терминах колебаний с характеристиками: время, частота, амплитуда[8].

Любая статистическая информация содержит не только полезную информацию, но и следы посторонних воздействий, помехи, шум. В таком случае аддитивную модель можно записать в виде:

$$s(t)=f(t)+\sigma e(t), \quad (1)$$

где  $f(t)$ – информация, очищенная от шума,  $s(t)$ – статистические данные,  $\sigma$ – уровень шума,  $e(t)$ – шум.

Удаление шума при помощи вейвлет-преобразования выполнялось в четыре этапа:

1. разложение сигнала по базису вейвлетов;
2. выбор порогового значения шума;
3. пороговая фильтрация;
4. реконструкция сигнала, т.е. получение информации, очищенной от шума.

На первом этапе было выполнено преобразование Хаара. При выборе порога шума использован универсальный критерий, по которому величина порога вычислялась по формуле

$$\theta=(2 \times \log(N))^{1/2}, \text{ где } N \text{ – количество данных в выборке.}$$

Пороговая фильтрация проводилась в соответствии с формулами:

$$y = \begin{cases} x + \theta, & \text{если } x < 0 \text{ и } |x| > \theta \\ x - \theta, & \text{если } x > 0 \text{ и } |x| > \theta \\ 0, & \text{если } |x| \leq \theta \end{cases}$$

На рис.3. даны результаты применения вейвлет-преобразований. По оси X отложены точки временного интервала, по Y – количество групп ВКонтакте, опубликовавших новость и присоединившихся к обсуждению. Пунктирной линией отображен исходный сигнал, т.е. статистические данные, красным цветом информация, очищенная от шума.

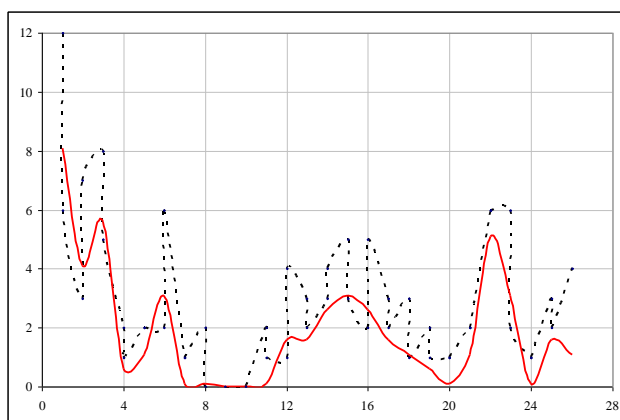
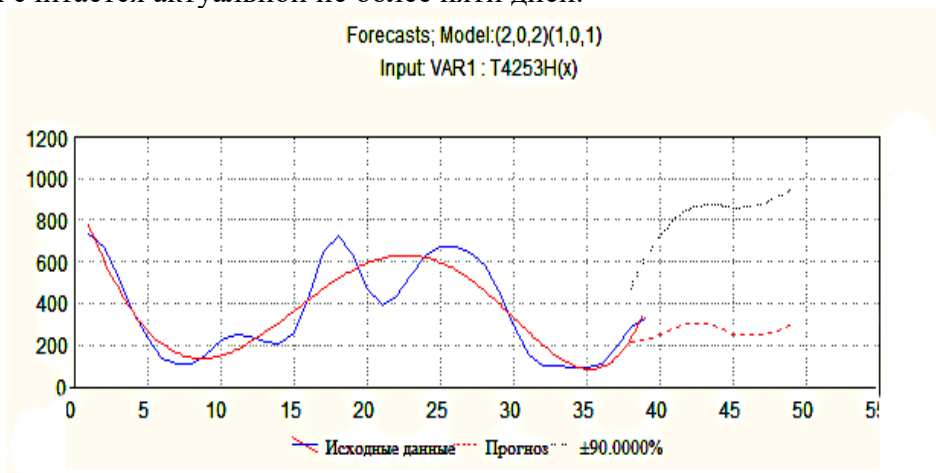


Рис. 3. Вейвлет-преобразования временного ряда

Аналогичные преобразования были применены к статистическим данным числа просмотров и репостов информации. Полученные данные обработаны в пакете Statistica с

целью прогнозирования длительности «интереса» к информации, см. рис.4. Синим цветом на рисунке отмечена кривая изменения количества просмотров, красным – интерполирующий полином, пунктирной линией – прогнозируемое количество просмотров в последующие часы. Как видно, интерес к новости будет сохраняться, однако ее уровень существенно снизится. Локальные минимумы соответствуют ночному времени. Максимальное число просмотров приходится на второй день, в последующие дни интерес начинает ослабевать. Однако количество групп, перепостивших новость, увеличивается, т.к. постепенно подключаются группы, не специализирующиеся на новостях.

Отметим, что это не противоречит имеющимся данным: актуальность практически любой темы в сети без дополнительных новых сообщений не превышает 11 дней, а информация считается актуальной не более пяти дней.



**Выводы.** Рассмотрены различные модели распространения информации в социальных сетях. Описано применение вейвлет-преобразований при моделировании временных рядов. Разработан метод прогнозирования распространения информационных сообщений в сетях и оценены возможные временные интервалы актуальности информации для пользователей социальных сетей.

#### Литература

1. Digital around the World in 2019. Digital in 2019 Режим доступа: URL: <https://wearesocial.com/global-digital-report-2019>.
2. Исследование Mail.ru Group к 25-летию Рунета. Mail.ru Group. Режим доступа: URL: <https://corp.mail.ru/ru/press/infograph/10391/>.
3. Социальные сети (рынок России). TADVISER. Государство. Бизнес. ИТ. Режим доступа: URL: <http://www.tadviser.ru/index.php/>.
4. Горковенко Д. К. Обзор моделей распространения информации в социальных сетях // Молодой ученый. — 2017. — №8. — С. 23-28. — URL <https://moluch.ru/archive/142/39946/>.
5. Губанов Д.А., Новиков Д.А., Чхартишвили А.Г. Социальные сети: модели информационного влияния, управления и противоборства. 3-е изд., перераб. и дополн. М.: МЦНМО, 2018. – 224 с.
6. Губанов Д.А., Новиков Д.А., Чхартишвили А.Г. Модели влияния в социальных сетях // Управление большими системами. 2009. № 27. С. 205-281.
7. Указ «Об определении в гуманитарных целях категорий лиц, имеющих право обратиться с заявлениями о приеме в гражданство Российской Федерации в упрощенном порядке». Режим доступа: URL: <http://www.kremlin.ru/acts/news/60358/> Сайт Президента России.
8. Добеши И. Десять лекций по вейвлетам. Перевод с англ. Е. Мищенко под редакцией А. Петухова. М.: НИИЦ, Регулярная и хаотическая динамика, 2004. - 464 с.

УДК 512.642

## ОБ ОДНОЙ ЗАДАЧЕ, СВЯЗАННОЙ С БУЛЕВЫМИ МАТРИЦАМИ

Манилов Д.Ю., Решетников А.В.

Национальный исследовательский университет "МИЭТ"

E-mail: 0-7@mail.ru

E-mail: [a\\_reshetnikov@hush.com](mailto:a_reshetnikov@hush.com)

### Аннотация:

**Манилов Д.Ю., Решетников А.В.** Доказано, что для матриц с чётным числом строк и столбцов матрицы, у которых в каких-либо строке и столбце стоят 1, а остальные элементы равны 0, образуют базис линейного пространства над двухэлементным полем.

### Annotation:

**Manilov D.Yu., Reshetnikov A.V.** On a task from Boolean matrices. We prove that, for the matrices having even numbers of rows and columns, the matrices whose some row and some columns consists of 1 and other elements equals to 0, is a basis of the linear space over the two-element field.

Для произвольной матрицы  $a \in A$  элемент, стоящий в её  $i$ -ой строке и  $j$ -ом столбце, будем обозначать через  $a[i, j]$ . Напомним определение символа Кронекера:

$$\delta_{ij} = \begin{cases} 1, & \text{если } i = j; \\ 0, & \text{иначе.} \end{cases}$$

Зафиксируем два чётных числа –  $n$  и  $m$ . Множество всех матриц над полем  $\mathbb{F}_2$  размера  $n \times m$  ( $n$  – число строк,  $m$  – число столбцов) обозначим через  $A$ . Рассмотрим матрицы  $e^{ij} \in A$  и  $f^{ij} \in A$ :

$$e^{ij}[x, y] = \delta_{ix} \& \delta_{jy}; \quad f^{ij}[x, y] = \delta_{ix} \vee \delta_{jy}, \quad (1)$$

где символы «&» и « $\vee$ » обозначают соответственно логическую конъюнкцию и логическую дизъюнкцию, а индексы  $i, j$  удовлетворяют условиям  $1 \leq i \leq n$ ,  $1 \leq j \leq m$ . Ясно, что матрицы  $e^{ij}$  образуют базис линейного пространства  $A$ . Для произвольной матрицы  $a \in A$  её разложение по матрицам  $e^{ij}$  как по базисным векторам имеет вид

$$a[x, y] = \sum_{i=1}^n \sum_{j=1}^m a[i, j] e^{ij}[x, y].$$

Оказывается, что в случае, когда оба числа  $n$  и  $m$  являются чётными, матрицы  $f^{ij}$  также образуют базис. Несложно описать разложение произвольной матрицы из пространства  $A$  по векторам данного базиса. Для этого будем использовать следующее обозначение: если  $a \in A$  – произвольная матрица, то через  $a^*$  обозначим матрицу

$$a^*[x, y] = \sum_{i=1}^n a[i, y] + \sum_{j=1}^m a[x, j] + a[x, y]. \quad (2)$$

**Теорема 1.** Пусть  $n$  и  $m$  – произвольные чётные натуральные числа,  $A$  – множество матриц над полем  $\mathbb{F}_2$ , состоящих из  $n$  строк и  $m$  столбцов. Тогда матрицы  $f^{ij}$ , определяемые по формуле (1), образуют базис линейного пространства  $A$ . При этом произвольная матрица  $a \in A$  раскладывается по матрицам  $f^{ij}$  следующим образом:

$$a[x, y] = \sum_{i=1}^n \sum_{j=1}^m a^*[i, j] f^{ij}[x, y], \quad (*)$$

где матрица  $a^*$  определяется формулой (2). Разложение матрицы  $a^*$  по матрицам  $f^{ij}$  имеет вид:

$$a^*[x, y] = \sum_{i=1}^n \sum_{j=1}^m a[i, j] f^{ij}[x, y].$$

Данный результат был получен в процессе решения следующей задачи (мы называем её задачей о крестах). Рассмотрим отображения  $\varphi^{ij} : A \rightarrow A$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq m$ , определённые формулой

$$a\varphi^{ij}[x, y] = \begin{cases} -a[x, y], & \text{если } (x=i) \text{ или } (y=j); \\ a[x, y], & \text{если } (x \neq i) \text{ и } (y \neq j), \end{cases}$$

здесь символ « $-$ » обозначает операцию логического отрицания:  $-0=1$ ,  $-1=0$ . Множество всех таких отображений обозначим через  $\Phi$ :  $\Phi = \{\varphi^{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m\}$ .

**Определение 1.** Пусть  $a \in A$  – произвольная матрица. Набор отображений  $\varphi_1, \varphi_2, \dots, \varphi_N \in \Phi$  назовём решением задачи о крестах для матрицы  $a$ , если композиция данных отображений переводит матрицу  $a$  в нулевую матрицу:

$$a\varphi_1\varphi_2\dots\varphi_N = 0.$$

Таким образом, задача о крестах состоит в том, чтобы для заданной матрицы  $a$  подобрать решение  $\varphi_1, \varphi_2, \dots, \varphi_N \in \Phi$  в смысле определения 1.

Идея получения общего решения задачи о крестах заключается в следующем. Рассмотрим какое-нибудь взаимно однозначное линейное отображение  $\lambda : A \rightarrow A$ . Обратное к  $\lambda$  отображение будем обозначать через  $\lambda^{-1}$ . Для каждого отображения  $\varphi^{ij} \in \Phi$ , определим отображение  $\psi^{ij} : A \rightarrow A$ :

$$a\psi^{ij} = a\lambda^{-1}\varphi^{ij}\lambda. \quad (3)$$

Множество всех таких отображений обозначим через  $\Psi$ :

$$\Psi = \{\psi^{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m\}.$$

Заметим, что для любых отображений  $\psi_1, \psi_2, \dots, \psi_N \in \Psi$ , если через  $\varphi_1, \varphi_2, \dots, \varphi_N$  обозначить соответствующие им отображения из множества  $\Phi$ :

$$\varphi_k = \lambda\psi_k\lambda^{-1} \quad \text{для всех } k \in \{1, \dots, n\}, \quad (4)$$

то будет выполнено следующее равенство:

$$a\psi_1\psi_2\dots\psi_N = a\lambda^{-1}\varphi_1\varphi_2\dots\varphi_N\lambda$$

(его легко проверить индукцией по  $N$ , используя утверждение (3) в качестве базиса индукции). Если для какой-либо матрицы  $a \in A$  найден набор отображений  $\psi_1, \psi_2, \dots, \psi_N \in \Psi$ , композиция которых переводит матрицу  $a$  в нулевую матрицу, то есть если выполнено условие

$$a\psi_1\psi_2\dots\psi_N = 0, \quad (5)$$

то композиция отображений  $\varphi_1, \varphi_2, \dots, \varphi_N$  переведёт матрицу  $a\lambda^{-1}$  в нулевую матрицу:

$$\begin{array}{ccccccccccc} a & \xrightarrow{\psi_1} & a\psi_1 & \xrightarrow{\psi_2} & a\psi_1\psi_2 & \xrightarrow{\psi_3} & \dots & \xrightarrow{\psi_n} & 0 \\ \uparrow & & \uparrow & & \uparrow & & & & \uparrow \\ a\lambda^{-1} & \xrightarrow{\varphi_1} & a\lambda^{-1}\varphi_1 & \xrightarrow{\varphi_2} & a\lambda^{-1}\varphi_1\varphi_2 & \xrightarrow{\varphi_3} & \dots & \xrightarrow{\varphi_n} & a\lambda^{-1}\varphi_1\dots\varphi_N \end{array}$$

(вертикальными стрелками обозначены отображения  $\lambda$ ; при составлении данной коммутативной диаграммы мы использовали равенство (4)). Ввиду того, что  $0\lambda = 0$ , мы получим

$$a\lambda^{-1}\varphi_1\varphi_2\dots\varphi_N = 0. \quad (6)$$

Пусть теперь  $b \in A$  – произвольная матрица. Подберём отображение  $\lambda$  таким образом, чтобы для любой матрицы  $a \in A$  заведомо существовал набор отображений  $\psi_1, \psi_2, \dots, \psi_N \in \Psi$ , обеспечивающий выполнение условия (5). Тогда матрицу  $b$  можно будет представить в виде  $b = a\lambda^{-1}$  для некоторой матрицы  $a \in A$  ввиду взаимной однозначности отображения  $\lambda$ . После этого с помощью формулы (4) мы получим набор отображений  $\varphi_1, \varphi_2, \dots, \varphi_N \in \Phi$ , композиция которых переведёт матрицу  $b$  в нулевую матрицу (см. (6)). Тем самым для матрицы  $b$ , выбранной произвольным образом, будет получено решение задачи о крестах в смысле определения 1.

Оказывается, чтобы решить задачу о крестах для произвольной матрицы из множества  $A$ , можно в качестве  $\lambda$  рассмотреть отображение

$$a\lambda = a^* \quad \text{для всех } a \in A. \quad (7)$$

Матрицу  $a^*$  мы далее называем *матрицей переключений* для матрицы  $a$ .

Формула (\*) позволяет для произвольной матрицы  $a \in A$  находить самое короткое решение  $\varphi_1, \varphi_2, \dots, \varphi_N \in \Phi$  задачи о крестах. Из теоремы 1 следует, что такое решение является единственным с точностью до порядка следования отображений, и оно содержит те и только те отображения  $\varphi^{ij} \in \Phi$ , для которых  $a^*[i, j] = 1$ . Данный метод решения требует предварительного вычисления матрицы переключений  $a^*$ ; это может быть сделано, например, по её определению (2).

Другой метод решения задачи о крестах состоит из двух этапов: на первом этапе из заданной матрицы  $a$  требуется получить матрицу  $b$  такую, что  $b^* = b$ ; на втором этапе из матрицы  $b$  требуется получить нулевую матрицу:

$$\begin{array}{ccccccc} a & \xrightarrow{\varphi_1} & a\varphi_1 & \xrightarrow{\varphi_2} & a\varphi_1\varphi_2 & \xrightarrow{\varphi_3} & \dots & \xrightarrow{\varphi_k} & b; \\ b & \xrightarrow{\varphi_{k+1}} & b\varphi_1 & \xrightarrow{\varphi_{k+2}} & b\varphi_1\varphi_2 & \xrightarrow{\varphi_{k+3}} & \dots & \xrightarrow{\varphi_n} & 0. \end{array}$$

На первом этапе в качестве  $\varphi_1, \varphi_2, \dots, \varphi_k$  выбираем те и только те отображения  $\varphi^{ij}$ , для которых  $a[i, j] = 1$ , на втором этапе в качестве  $\varphi_{k+1}, \varphi_{k+2}, \dots, \varphi_n$  берём те и только те  $\varphi^{ij}$ , для которых  $b[i, j] = 1$ .

Определения матриц  $e^{ij}$  и  $f^{ij}$  отличаются лишь тем, что над одними и тем же объектами в одном случае выполняется операция конъюнкции, а в другом – операция дизъюнкции. В связи с этим отметим, что в определении произведения булевых матриц аналогичная замена конъюнкции на дизъюнкцию приводит к понятию *дизъюнктного произведения*. Некоторые свойства дизъюнктного произведения и некоторые его связи с обычным произведением булевых матриц найдены в работе [1].

## Литература

1. Поплавский В. Б. О приложениях ассоциативности дуальных произведений алгебры булевых матриц. // *Фундаментальная и прикладная математика*. 2011/2012. Том 17, № 4. С. 181 – 192.

УДК 621.271.4:004.422.833

## МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ТЕХНОГЕННОГО МЕСТОРОЖДЕНИЯ С УЧЕТОМ РАЗВИТИЯ ГОРНЫХ РАБОТ

**Е.В.Прокопенко<sup>1</sup>, С.В.Масло<sup>2</sup>, А.В.Гром**

Донецкий национальный технический университет, г. Донецк

кафедра прикладной математики

доцент кафедры прикладной математики<sup>1</sup>, преподаватель кафедры компьютерных информационных технологий ГОУ ВПО «ДонАУиГС»<sup>2</sup>, студентка группы ПМК-15

E-mail: [prokopenko1515@rambler.ru](mailto:prokopenko1515@rambler.ru)

### **Аннотация:**

**Е.В.Прокопенко<sup>1</sup>, С.В.Масло<sup>2</sup>, А.В.Гром.** *Математическое моделирование техногенного месторождения с учетом развития горных работ.* В данной статье рассматривается концепция взаимосвязи формирования техногенного месторождения (породного отвала) с развитием горных работ.

### **Annotation:**

**Prokopenko E.V., Maslo S.V., Grom A.V.** *Mathematical modeling of technogenic deposits, taking into account the development of mining* This article discusses the concept of the relationship of the formation of technogenic deposits (waste dump) with the development of mining.

### **Общая постановка проблемы**

В настоящее время внимание промышленности, как в дальнем, так и в ближнем зарубежье вновь обращается к породным отвалам угольных шахт, т.к. отвальные породы могут представлять интерес в качестве источника вторичного минерального сырья. Технология формирования породного отвала предусмотрена с выполнением мероприятий по предупреждению самовозгорания отвальной массы. При формировании ярусов учитываются следующие факторы: время работы шахты, максимальная высота отвала. Данная технология формирования породного отвала предполагает проектное наращивание отдельного яруса в случае, если предусмотрены все технологические мероприятия, что не всегда выполняется при отсыпке пород на отвал. В основном это связано с тем, что породы в отвал отсыплются в различных соотношениях литологических разностей, которые определить или прогнозировать достаточно сложно.

Проектирование отвалообразования необходимо осуществлять с учетом всех возможных факторов и особенностей вскрышного массива месторождения, состояния основания, которые могут оказать влияние на устойчивость отвала, степень его воздействия на окружающую среду [1,2].

По технологии отвалообразования порода насыпается на отвал последовательно небольшими порциями (дискретно), т.е. за определенное время и в определенном месте, в соответствии с планограммой развития горных работ (Рис. 1).

Особое место в этой цепочке занимает плановое выполнение маркшейдерских съемок породного отвала. В период между съемками можно составить схему отсыпки пород на отвал, а также предусмотреть порядок насыпки породы из конкретного проходческого забоя, который соответствует обрабатываемому в данный период времени пласту. Зная литологический состав пород обрабатываемого пласта, а также технологию и порядок отправки породы на отвал, можно составить хронолитологическую карту породного отвала.

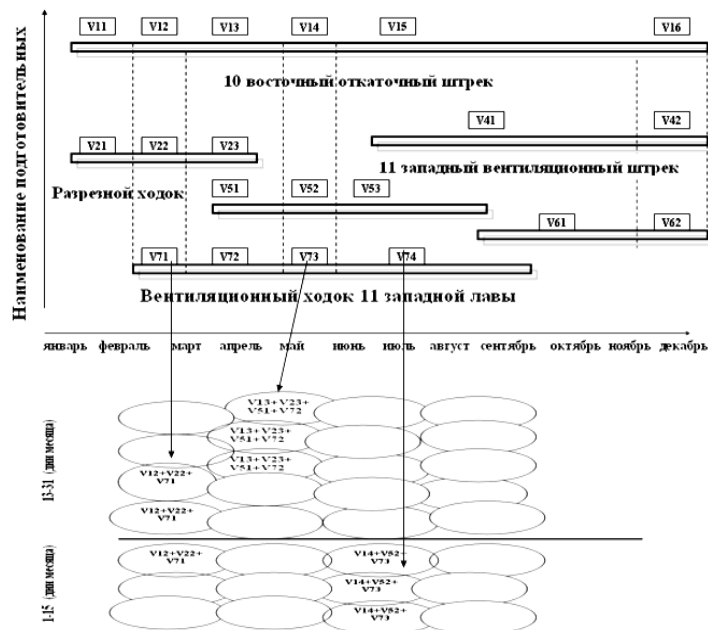


Рис. 1 Технология и порядок отправки породы на отвал из проходческих забоев

Следовательно, систематизированная отсыпка породы по определенной схеме позволяет найти местоположение породной массы с конкретной горнопроходческого участка с одновременной характеристикой состава пород, отсыпаемых на отвал. [3]

Этот технологический процесс представлен на рисунке 2, где параметры  $V_1, V_2, \dots, V_i$  определяют единичный объем породы, отсыпаемой на отвал. Под единичным объемом подразумевается такое количество породы, которое осыпается на отвал в период, равный декаде или одному месяцу. Период отсыпки породы на отвал можно регулировать в зависимости от постановки той или иной задачи. Цифрами 1...15 (работали К забоев) и 16...31 (работали К1 забоев) показано, что данный период насыпки рассматривается в течение одного месяца с интервалом в 15 дней.

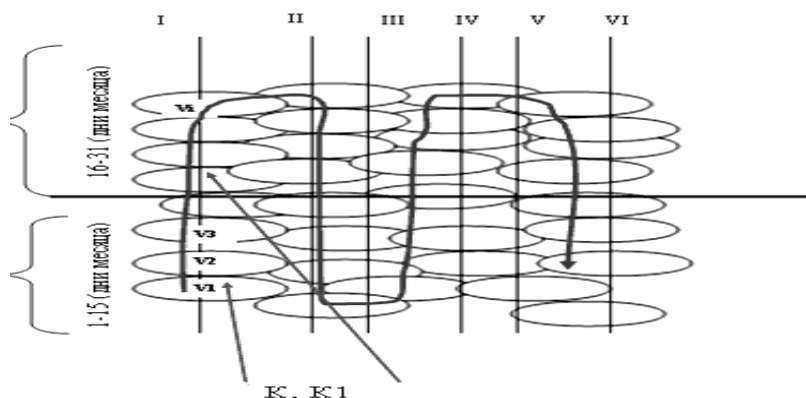


Рис. 2. Схема систематизированной отсыпки породы на отвал

I, II, III...VI – номер месяца; К, К1 – количество забоев, работающих в данный период времени. Так как порода поступает с конкретного забоя в определенное место отвала, то данную технологическую цепочку можно представить с математической точки зрения в виде следующей зависимости:



$$B = \sum_{i=T1}^{T2} \sum_{j=1}^k (t_i, f_j, x_j, y_j, z_j, \gamma_j), \quad (1)$$

где  $B$  – общая характеристика пород отвала;

$T1$  и  $T2$  – соответственно начало и конец интервала передачи породы из конкретного проходческого забоя на отвал;  $k$  – количество работающих забоев;  $t_j$  – момент передачи и складирования породы из забоя на отвал в определенное место;

$f_j$  – свойства единичных объемов пород, извлекаемых из забоя (химические, физические);

$x_j, y_j, z_j$  – координаты точек, указывающие расположение единичных объемов в определенном месте забоя;  $\gamma_j$  – случайная величина, зависящая от режима работы конкретного забоя (цикличность работы, аварийные ситуации, круглосуточный режим), а также от технологии ведения горных работ (БВР, с помощью комбайнов).

Как отмечалось выше, современное складирование шахтных пород в отвалах производится без учета их качественных и физико-механических характеристик, что затрудняет использование пород с определенными свойствами, хранящихся в этих отвалах, а также их утилизацию. Технологические схемы отсыпки отвалов не учитывают сегрегацию пород, что не позволяет более полно использовать земельный отвод под отвалы. Одной из задач исследования является разработка комплексного подхода к изучению процесса формирования породного отвала с учетом технологии проведения горных подготовительных выработок и технологической транспортной цепочки доставки породы на отвал для рационального размещения этой породы в отвале для дальнейшего хранения и утилизации.

Данная задача предполагает составление технологической цепочки, предусматривающей порядок выдачи породы на отвал, то есть рассмотрение календарного плана горных работ, планирование горных работ; транспортирование пород под землей и на поверхности; складирование на отвале.

Как отмечалось выше, по технологии отвалообразования, порода насыпается на отвал дискретно. Определенное место в этой цепочке занимает периодичность выполнения маркшейдерских съемок породного отвала. Период между съемками  $\Delta t$  определяет местоположение того или иного участка горных работ в зависимости от планограммы развития работ, и также определяет состав пород в данном месте отвала, то есть "химию" пород. Следовательно, технологическая схема данной цепочки позволяет найти местоположение породы с конкретного горнопроходческого участка с конкретной характеристикой литологического состава породы, отсыпаемой на отвал.

Технология формирования породного отвала заключается в насыпке породы на отвал отдельными ярусами. Так как отдельный ярус породного отвала насыпается определенное количество лет, то можно составить карту насыпки отвала за определенный период времени.

**Выводы.** Применительно к полученной выше информации можно сказать, что зная свойства горных пород, извлекаемых за определенный промежуток времени из ряда одновременно работающих проходческих забоев, можно формировать отвал как техногенное месторождение путем отсыпки извлеченной породы в определенное место яруса по определенной схеме с учетом его последующей рекультивации при непрерывном геомеханическом контроле устойчивости, несущей способности и осадок. Исходя из вышеизложенного, можно сделать вывод, что маркшейдерский план это хронология деятельности "живого организма", и данную хронологию можно использовать для построения динамической модели формирования породного отвала, так как:

1. Горные работы ведутся по определенной системе и планированию, используя планограмму развития горных работ.

2. Планограмма отображается на планах горных работ по каждому пласту, на котором ведутся работы.

3. Развитие горных работ осуществляется за определенный интервал времени ( $t$ ) и в определенном месте полезного ископаемого, т. е, осуществляется во времени и в пространстве.

4. Данная информация отображается на маркшейдерских планах горных работ в виде подвигания забоя каждой выработки за определенное время. [4,5,6].

В связи с тем, что каждый пласт имеет свое геологическое строение, можно составить прогноз химических реакций, происходящих при соприкосновении тех или иных элементов, содержащихся в различных пластах, то есть заранее выявить неблагоприятные зоны на отдельном ярусе и в целом на отвале.

Таким образом, в результате построения трехмерной и двумерной модели участков яруса породного отвала за определенный период времени можно получить пространственную модель, которая показывает структуру данного отвала. На основе данной модели можно решить следующие задачи:

1. Установить время отсыпки того или иного участка проходческих работ. Определить состав пород на участках.

2. По составу пород и их свойствам оценить возможные очаги самовозгорания отдельных участков и отвала в целом.

3. Установить участки, на которых сложены породы, пригодные для дальнейшего использования в хозяйственных целях.

4. Установить участки, на которых возможна разработка с целью доизвлечения полезного компонента (сформировано техногенное месторождение).

### Литература

1. Вахмянин И.С. Разработка модели алгоритма управления информационными потоками в ситуационных центрах органов государственной власти / Вахмянин И.С., Ильин Н.И., Новикова Е.В.—Бизнес-Информатика №1(15)- 2011.—С. 3—9.

2. Прокопенко Е.В. Построение пространственной модели отвала с использованием программного пакета Surfer /С.В.Борщевский, Е.В.Прокопенко// Збірник наукових праць НГУ. — Національний гірничий університет, 2010. —№34, т.1.— С. 82—87.

3. Разработка динамической модели породных отвалов угольных шахт/ Прокопенко Е.В., Борщевский С.В.//Наукові праці УкрНДМІ НАН України. Випуск 6 /Під заг.ред. А.В.Анциферова.-Донецьк, УкрНДМІ НАН України, 2010.-14с.

4. Матлак Е. С. Безотходное производство - основное направление использования природных ресурсов и охраны окружающей среды / Е. С. Матлак / Общегосударственный научно-технический журнал. —Донецк: ДонГТУ, 1998.— С. 10—14.

5. Методические указания по проектированию рекультивации нарушенных земель на действующих и проектируемых предприятиях угольной промышленности.— Пермь: ВНИИОСуголь— 1991.

6. Охрана природы. Земли. Классификация нарушенных земель для рекультивации. ГОСТ 17.5.3.02-84

УДК 512.572

## СВОБОДНЫЕ АЛГЕБРЫ В КЛАССЕ ПОЛУАБЕЛЕВЫХ $n$ -ГРУПП

Щучкин Н.А.

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Волгоградский государственный социально-педагогический университет»

Кафедра алгебры, геометрии и математического анализа

E-mail: [nikolaj\\_shchuchkin@mail.ru](mailto:nikolaj_shchuchkin@mail.ru)

### Аннотация:

**Щучкин Н.А.** Свободные алгебры в классе полуабелевых  $n$ -групп. В данной статье описывается строение свободных алгебр в классе полуабелевых  $n$ -групп с обобщением коммутативности.

### Annotation:

**Schuchkin N.A.** Free algebras in the class of semi-abelian  $n$ -groups. This article describes the structure of free algebras in the class of semi-Abelian  $n$ -groups with a generalization of commutativity.

### Введение

Алгебру  $\langle G, f \rangle$  с  $n$ -арной операцией  $f$  ( $n \geq 2$ ) называют  $n$ - группой [1], если в ней выполняется обобщенный закон ассоциативности

$$f(f(a_1, \dots, a_n), a_{n+1}, \dots, a_{2n-1}) = f(a_1, \dots, a_i, f(a_{i+1}, \dots, a_{i+n}), a_{i+n+1}, \dots, a_{2n-1}) \quad (1)$$

для всех  $i = 1, \dots, n-1$  и для любых элементов  $a_1, \dots, a_{j-1}, a_{j+1}, \dots, a_n, b$  из  $G$  разрешимо и имеет единственное решение каждое из уравнений

$$f(a_1, \dots, a_{j-1}, x_j, a_{j+1}, \dots, a_n) = b$$

для всех  $j = 1, \dots, n$ . При  $n = 2$  имеем обычную (бинарную) группу. Таким образом,  $n$ - группа является обобщением понятия группы на  $n$ -арный случай при  $n > 2$ . Здесь и далее мы будем использовать стандартные обозначения:

$$f(x_1, \dots, x_1, x_2, \dots, x_2, \dots, x_n, \dots, x_n) = f(x_1^{(s)}, x_2^{(s)}, \dots, x_n^{(s)}) \quad \square$$

всякий раз, когда  $x_{k+1} = \dots = x_{k+s} = x$  ( $x$  и  $x_i^j$  при  $i > j$  – пустые символы). В  $n$ - группе  $\langle G, f \rangle$ , используя закон (1), определяют новую  $(k(n-1) + 1)$ -арную операцию  $f_{(k)}$  по правилу:

$$f_{(k)}(x_1^{k(n-1)+1}) = \underbrace{f(f(\dots f(f(x_1^n), x_{n+1}^{2n-1}), \dots))}_{k \text{ раз}}, x_{(k-1)(n-1)+2}^{k(n-1)+1}.$$

Нас интересует коммутативность (или перестановочность элементов) в теории  $n$ - групп, которая имеет несколько обобщений групповой коммутативности. Мы будем изучать класс  $n$ - групп с обобщением коммутативности, которую предложил Э. Пост в своей работе [2]. С учётом свойств эквивалентных последовательностей в  $n$ - группе, обобщение

коммутативности, указанное в [2], можно сформулировать следующим образом:  $n$ -группа называется  $m$ -полуабелевой, если  $m - 1$  делит  $n - 1$  и верно тождество

$$f(a, a_1^{m-2}, b, c_1^{n-m}) = f(b, a_1^{m-2}, a, c_1^{n-m}).$$

Если  $m = 2$ , то  $n$ -группу называют абелевой, т.е. верно тождество

$$f(a, b, c_1^{n-2}) = f(b, a, c_1^{n-2}).$$

Если  $m = n$ , то  $n$ -группу называют полуабелевой, т.е. верно тождество

$$f(a, c_1^{n-2}, b) = f(b, c_1^{n-2}, a)$$

Очевидно, любая абелева  $n$ -группа является  $m$ -полуабелевой, а последняя, в свою очередь, является полуабелевой. Обратная цепочка рассуждений неверна.

В любом классе алгебр имеются основополагающие объекты – это свободные алгебры. Описание строения свободных алгебр в заданном классе играет важную роль в изучении этого класса алгебр. Свободные  $n$ -группы в классе всех  $n$ -групп изучались В.А. Артамоновым в работе [3]. В работах [4] изучались свободные  $n$ -группы в классе абелевых  $n$ -групп. В работе [7] изучено строение свободных  $n$ -групп в классе  $m$ -полуабелевых  $n$ -групп.

### 1. Некоторые сведения из теории $m$ -полуабелевых $n$ -групп

Имеется тесная связь между группами и  $n$ -группами. Отметим основной результат работ [5], [6]. Теорема Глускина-Хоссу утверждает, что на любой  $n$ -группе  $\langle G, f \rangle$  можно определить группу  $\langle G, + \rangle$ , где сложение действует по правилу:  $a + b = f(a, c^{(n-3)}, \bar{c}, b)$  для

фиксированного элемента  $c$  из  $G$ . Тогда для элемента  $d = f(c^{(n)})$  и отображения

$\varphi(x) = f(c, x, c^{(n-3)}, \bar{c})$ , которое является автоморфизмом группы  $\langle G, + \rangle$ , верны равенства

$$\varphi(d) = d, \quad \varphi^{n-1}(x) + d = d + x \quad \text{для любого } x \in G, \quad (2)$$

$$f(a_1^n) = a_1 + \varphi(a_2) + \dots + \varphi^{n-2}(a_{n-1}) + \varphi^{n-1}(a_n) + d \quad \text{для любых } a_i^n \in G \quad (3)$$

Элемент  $c$  является нулем в группе  $\langle G, + \rangle$ . Группу  $\langle G, + \rangle$  называют ретрактом  $n$ -группы  $\langle G, f \rangle$  и обозначают  $ret_c \langle G, f \rangle$ . Известно [5],[6], что любые два ретракта одной и той же  $n$ -группы изоморфны.

Верно и обратно (обратная теорема Глускина-Хоссу): в любой группе  $\langle G, + \rangle$  для выбранных автоморфизма  $\varphi$  и элемента  $d$  с условиями (2) задается  $n$ -группа  $\langle G, f \rangle$ , где  $f$  действует по правилу (3).  $n$ -Группу  $\langle G, f \rangle$  в этом случае называют  $(\varphi, d)$ -определенной на группе  $\langle G, + \rangle$  и обозначают  $der_{\varphi, d} \langle G, + \rangle$ .

$n$ -Группа  $\langle G, f \rangle$ ,  $(1_G, 0)$ -определенная на группе  $\langle G, + \rangle$ , называется производной от этой группы.

## 2. Строение свободных $m$ -полуабелевых $n$ -арных групп

Приступаем к изучению свободных  $n$ -групп в классе  $m$ -полуабелевых  $n$ -групп. Рассмотрим множество  $\{x_\alpha \mid \alpha \in I\}$ . Для каждого элемента  $x_\alpha$  определим прямую сумму  $\langle A_\alpha, + \rangle = \sum_{j=1}^{m-1} \langle (x_{\alpha_j}), + \rangle$  бесконечных циклических групп  $\langle (x_{\alpha_j}), + \rangle$ . Рассмотрим прямую сумму  $\langle F, + \rangle = \langle (a), + \rangle + \sum_{\alpha \in I} \langle A_\alpha, + \rangle$ , где  $\langle (a), + \rangle$  – бесконечная циклическая группа. На каждой группе  $\langle A_\alpha, + \rangle$  выбираем автоморфизм  $\varphi_\alpha$ , действующий по правилу: для любого  $t_1 x_{\alpha 1} + t_2 x_{\alpha 2} + \dots + t_{m-1} x_{\alpha m-1} \in A_\alpha$  имеем  $\varphi_\alpha(t_1 x_{\alpha 1} + t_2 x_{\alpha 2} + \dots + t_{m-1} x_{\alpha m-1}) = t_{m-1} x_{\alpha 1} + t_1 x_{\alpha 2} + \dots + t_{m-2} x_{\alpha m-1}$ . Тогда на группе  $\langle F, + \rangle$  имеем автоморфизм  $\varphi$ , действующий по правилу: для любого  $sa + \sum_{i=1}^r z_{\alpha_i} \in \langle F, + \rangle$  получим  $\varphi(sa + \sum_{i=1}^r z_{\alpha_i}) = sa + \sum_{i=1}^r \varphi_{\alpha_i}(z_{\alpha_i})$ . Очевидно, элемент  $a$  и автоморфизм  $\varphi$  группы  $\langle F, + \rangle$  удовлетворяют (11) (при  $d = a$ ), тогда на группе  $\langle F, + \rangle$  определяем  $m$ -полуабелеву  $n$ -группу  $\langle F, f \rangle = \text{der}_{\varphi, a} \langle F, + \rangle$ , где  $n$ -арная операция действует по правилу (3).

**Теорема 1 [7].**  $n$ -Группа  $\langle F, f \rangle$  является свободной в классе  $m$ -полуабелевых  $n$ -групп со свободным порождающим множеством  $X = \{-a + x_{\alpha 1} \mid \alpha \in I\} \cup \{0\}$ .

## 3. Строение свободных абелевых и полуабелевых $n$ - групп

Рассмотрим построение свободной  $n$ - группы в классе полуабелевых  $n$ - групп. Поступаем, как и выше, полагая  $m = n$ . Для каждого элемента  $x_\alpha$  из множества  $\{x_\alpha \mid \alpha \in I\}$  определим прямую сумму  $\langle A'_\alpha, + \rangle = \sum_{j=1}^{n-1} \langle (x_{\alpha_j}), + \rangle$  бесконечных циклических групп  $\langle (x_{\alpha_j}), + \rangle$ . Затем строим прямую сумму  $\langle F', + \rangle = \langle (a), + \rangle + \sum_{\alpha \in I} \langle A'_\alpha, + \rangle$ , где  $\langle (a), + \rangle$  – бесконечная циклическая группа. На каждой группе  $\langle A'_\alpha, + \rangle$  выбираем автоморфизм  $\varphi'_\alpha$ , действующий по правилу: для любого  $\sum_{j=1}^{n-1} t_j x_{\alpha_j} \in A'_\alpha$  имеем  $\varphi'_\alpha(\sum_{j=1}^{n-1} t_j x_{\alpha_j}) = t_{n-1} x_{\alpha 1} + t_1 x_{\alpha 2} + \dots + t_{n-2} x_{\alpha n-1}$ . Тогда на группе  $\langle F', + \rangle$  имеем автоморфизм  $\varphi'$ , действующий по правилу: для любого  $sa + \sum_{i=1}^r z_{\alpha_i} \in \langle F', + \rangle$  получим  $\varphi'(sa + \sum_{i=1}^r z_{\alpha_i}) = sa + \sum_{i=1}^r \varphi'_{\alpha_i}(z_{\alpha_i})$ . Очевидно, элемент  $d = a$  и автоморфизм  $\varphi'$  группы  $\langle F', + \rangle$  удовлетворяют (2), значит, на группе  $\langle F', + \rangle$  определяем полуабелеву  $n$ -группу  $\langle F', f' \rangle = \text{der}_{\varphi', a} \langle F', + \rangle$ , где  $n$ -арная операция  $f'$  действует по правилу (3).

**Следствие 1 [7].**  $n$ -Группа  $\langle F', f' \rangle$  является свободной в классе полуабелевых  $n$ - групп со свободным порождающим множеством  $X = \{-a + x_{\alpha 1} \mid \alpha \in I\} \cup \{0\}$ . Любая свободная  $n$ - группа  $\langle H, g \rangle$  в классе полуабелевых  $n$ -групп со свободным порождающим множеством  $W = \{x_\alpha \mid \alpha \in I\} \cup \{c\}$  изоморфна  $n$ -группе  $\langle F', f' \rangle$ .

Рассмотрим теперь построение свободной  $n$ -группы в классе абелевых  $n$ -групп. Поступаем как и выше, полагая  $m = 2$ . Строим прямую сумму  $\langle F'', + \rangle = \langle (a), + \rangle + \sum_{\alpha \in I} \langle (x_\alpha), + \rangle$ , где  $\langle (a), + \rangle$  и  $\langle (x_\alpha), + \rangle$  ( $\alpha \in I$ ) – бесконечные циклические группы. Очевидно, элемент  $d = a$  и тождественный автоморфизм  $1_{F''}$  группы  $\langle F'', + \rangle$  удовлетворяют (2), значит, на группе  $\langle F'', + \rangle$  определяем абелеву  $n$ -группу  $\langle F'', f'' \rangle = \text{der}_{1_{F''}, a} \langle F'', + \rangle$ , где  $n$ -арная операция  $f''$  действует по правилу (3).

**Следствие 2 [7].**  $n$ -Группа  $\langle F'', f'' \rangle$  является свободной в классе абелевых  $n$ - групп со свободным порождающим множеством  $X = \{-a + x_\alpha \mid \alpha \in I\} \cup \{0\}$ . Любая свободная  $n$ - группа  $\langle H, g \rangle$  в классе абелевых  $n$ -групп со свободным порождающим множеством  $W = \{y_\alpha \mid \alpha \in I\} \cup \{c\}$  изоморфна  $n$ -группе  $\langle F'', f'' \rangle$ .

### Литература

1. Dornst W., Untersuchungen über einen verallgemeinerten Gruppenbegriff, *Math. Zeitschr.* 29 (1928), 1-19.
2. Post E. L., Polyadic Groups, *Trans. Amer. Math. Soc.* 48 (1940), 208-350.
3. Артамонов В. А., Свободные  $n$ -группы, *Матем. заметки*, Т.8, №4 (1970), 499-507.
4. Щучкин Н. А., Свободные абелевы  $n$ -арные группы, *Чебышевский сборник*, Т. XII, Выпуск 2(38) (2011), 163-170.
5. Глускин Л. М., Позиционные оперативы, *Мат. сборник*, Т.68 (110), №3 (1965), 444– 472.
6. Hosszu M., On the explicit form of  $n$ -group operations, *Publ. Math.* 10 (1963), 88-92.
7. Щучкин Н. А. Свободные  $m$ -полуабелевы  $n$ -арные группы // *Электронные информационные системы*. 2018. №4, Выпуск 19. С. 97–109.

УДК 004.4'242

## СОВРЕМЕННЫЕ СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ ПО КАК ИНСТРУМЕНТ РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНЫХ САПР ПО

Гурин А.Г., Грищенко Д.А., Григорьев А.В.  
Донецкий национальный технический университет  
Кафедра программной инженерии

E-mail: [alexandergurin1996@gmail.com](mailto:alexandergurin1996@gmail.com), [grigorievalvl@gmail.com](mailto:grigorievalvl@gmail.com)

### **Аннотация:**

*Гурин А.Г., Грищенко Д.А., Григорьев А.В. Современные системы контроля версий ПО как инструмент разработки интеллектуальных САПР ПО. В данной статье выполнен анализ этапов развития систем контроля версий ПО. Выделен наиболее перспективный класс систем контроля версий с расширенным функционалом. Подвергнута детальному анализу система контроля версий такого типа – система git. На основе специфики системы была поставлена задача создания интеллектуальной надстройки над git, задачей которой является интеллектуальный анализ и синтез программ.*

### **Annotation:**

*Gurin A.G., Grischenko D.A., Grigoriev A.V. Modern version control systems software as a tool for developing intelligent CAD software. This article analyzes the stages of development of software version control systems. The most promising class of version control systems with enhanced functionality is highlighted. A detailed analysis of this type of version control system has been analyzed - the git system. Based on the specifics of the system, the task was to create an intellectual add-on over git, the task of which is the intellectual analysis and synthesis of programs.*

### **Введение**

На текущий момент развития информационных технологий существенную роль играют системы контроля версий (СКВ), которые обеспечивают контроль изменения каких-либо документов. Они могут быть использованы как в программной инженерии и в САПР, так и в других направлениях информационных технологий.

На данном этапе развития программирования тяжело представить проект, в котором будут не важны резервные копии и контроль разных версий файлов. Важным моментом является то, что СКВ способна обеспечивать совместную разработку программных продуктов коллективом программистов.

Совместная разработка программных продуктов определенного класса программ в рамках СКВ позволяет накапливать разные варианты рабочих программных решений. Главное достоинство современных СКВ – это возможность накопления и повторного использования экспертных знаний об апробированных программных решениях в форме, удобной для хранения, передачи и редактирования [1].

В связи с широким использованием систем контроля версий актуальной является задача развития их возможностей и поиск новых областей применения.

СКВ с расширенным функционалом позволяет создавать надстройки, которые и смогут выполнять интеллектуальный анализ и синтез программ, опираясь на накопленные программные решения.

Целью данной статьи является:

- анализ назначений и возможностей систем контроля версий;
- анализ СКВ с расширенным функционалом;
- поиск путей создания интеллектуальной надстройки над СКВ.

### 1. Назначение, возможности систем контроля версий

Система контроля версий — это система, которая ведет контроль за файлом или за группой файлов на протяжении жизни всего проекта. Также СКВ позволяет вернуться к любым изменениям совершивших во время разработки его.

Первые системы контроля версий были локальными, то есть проект хранился только на локальном компьютере. Возникла проблема совместной разработки программных проектов. На смену им пришли централизованные системы контроля версий, которые позволяют работать в команде, так как проект находится на центральном сервере, и каждый участник команды имеет к нему доступ. Но и в таких системах есть недостаток - если центральный сервер выходит из строя, то теряется весь прогресс по проекту, затем - предельные системы контроля версий. Их особенностью стало то, что у каждого участника команды была копия репозитория с центрального сервера [2].

Ранее авторами был проведен сравнительный анализ систем контроля версий, результаты которого представлены в табл. 1. [3].

Таблица 1 – Сравнение программных реализаций системы управления версиями

Название программной реализации	Преимущества	Недостатки
CSV	- Некоторые отдают ей предпочтение, за то, что она проверена временем	- История не содержит переименованные или перемещённые файлы - Символические ссылки на файлы создают уязвимости в безопасности - Атомарные операции отсутствуют, что есть вероятность, что приведет к повреждениям кода - Использование операций с ветками дорого обходится
SVN	- Может работать с атомарными операциями - Использование операций с ветвлением кода не так дорого обходится - Не малый ассортимент плагинов IDE - Не использует пиринговую модель	- Ошибки, которые связаны с переименованием файлов и директорий, до сих пор присутствуют - Проблемы с набором команд для работы с репозиторием - Скорость относительно меньше конкурента
Git	- Значительное увеличение в скорости работы - Использование операций с ветвлением кода являются дешевыми - Можно работать без подключения к интернету - Распределенная, пиринговая модель - Хранятся измененные файлы целиком	- Присутствуют ограниченная поддержка Windows
Mercurial	- Легкий в освоение при старте - Присутствует документация подробная - Распределенная модель	- Нет возможности слияния двух родительских веток



Следует отметить, что данный анализ был выполнен с точки зрения полноты функционала систем контроля версий в рамках обычных САПР ПО типа Делфи.

Однако, среди данного набора систем контроля версий существует подмножество систем, обладающих расширенным функционалом, который позволяет использовать данные системы контроля версий в новых целях для выполнения новых задач. Среди них можно назвать git. Особенность функционала состоит в том, что данная система способна хранить весь текст программного комплекса, структурированный в соответствии с его архитектурой. Прочие системы лишены такой возможности и отслеживают только точки ветвления изменения программного кода. Особенность данного класса позволяет поставить задачу разработки интеллектуального САПР ПО на их основе.

Выполним детальнее анализ системы такого типа, а затем определим принципы решения задачи создания интеллектуального САПР ПО на их основе.

## 2. Анализ системы контроля версий с расширенным функционалом

### 2.1 Назначения возможностей системы контроля версий - git

Рассмотрим систему расширенного функционала, которая является системой контроля версий git.

Git — это распределённая система контроля версий. Основное отличие Git'a от любой другой системы контроля версий — это подход Git'a к работе со своими данными. Концептуально, большинство других систем хранят информацию в виде списка изменений в файлах. Эти системы представляют хранимую информацию в виде набора файлов и изменений, сделанных в каждом файле, по времени.

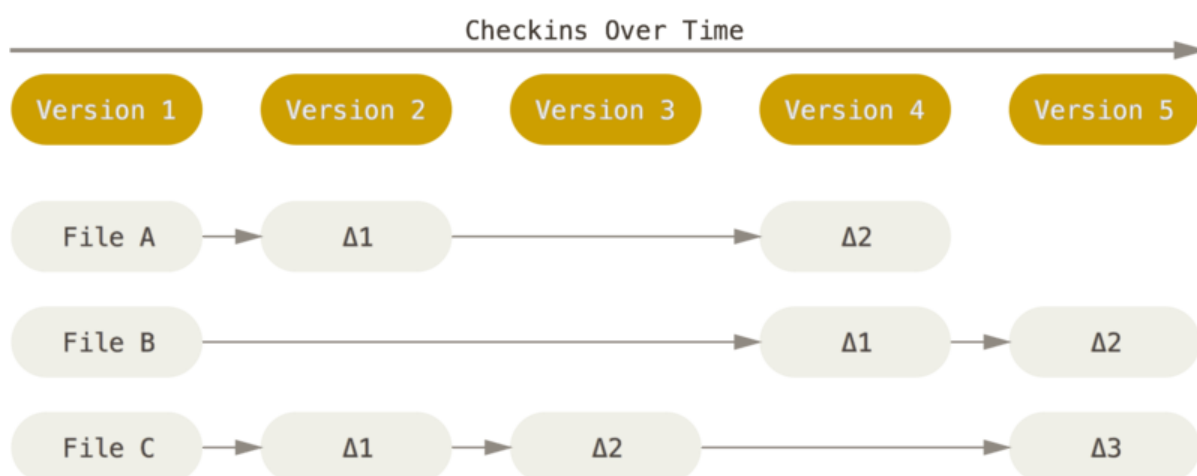


Рисунок 1 - Хранение данных как набора изменений относительно первоначальной версии каждого из файлов.

Git не хранит и не обрабатывает данные таким способом. Вместо этого, подход Git'a к хранению данных больше похож на набор снимков файловой системы. Каждый раз, когда вы делаете коммит (сохранение состояния своего проекта в Git'e), система запоминает, как выглядит каждый файл в этот момент и сохраняет ссылку на этот снимок. Для увеличения эффективности, если файлы не были изменены, Git не запоминает эти файлы вновь, а только создаёт ссылку на предыдущую версию идентичного файла, который уже сохранён. Git представляет свои данные как, скажем, поток снимков.

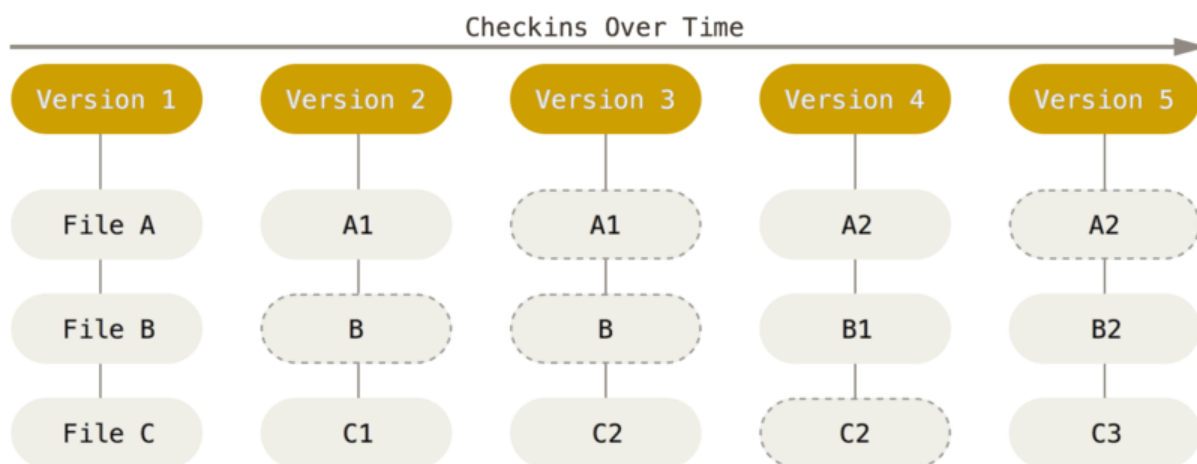


Рисунок 2 - Хранение данных как снимков проекта во времени.

Это очень важное отличие между Git и почти любой другой СКВ. Git переосмысливает практически все аспекты контроля версий, которые были скопированы из предыдущего поколения большинством других систем [4].

## 2.2. Анализ особенностей архитектуры системы контроля версий – git

Нижний уровень git является так называемой контекстно-адресуемой файловой системой. Для каждого объекта в репозитории вычисляется SHA-1-хеш, и именно он становится именем файла, содержащего данный объект в каталоге `.git/objects`.

Для оптимизации работы с файловыми системами, не использующими деревья для каталогов, первый байт хеша становится именем подкаталога, а остальные — именем файла в ней, что снижает количество файлов в одном каталоге.

Кроме того, в репозитории существует каталог `refs`, который позволяет задать читаемые человеком имена для каких-либо объектов Git. В командах Git оба вида ссылок — читаемые человеком из `refs`, и нижележащие SHA-1 — полностью взаимозаменяемы.

В репозитории иногда производится сборка мусора, во время которой устаревшие файлы заменяются на «дельты» между ними и актуальными файлами, где актуальная версия файла хранится неинкрементально, инкременты используются только для возврата к предыдущим версиям, после чего данные «дельты» складываются в один большой файл, к которому строится индекс.

Это снижает требования по ёмкости хранения.

Инструмент командной строки git содержит ряд команд по непосредственной манипуляции этим репозиторием на низком уровне.

Эти команды не нужны при нормальной работе с git как с системой контроля версий, но нужны для реализации сложных операций, а также дают возможность создать на базе репозитория git своё приложение [5].

Данный ряд возможностей позволяет создать свою надстройку над git, которая будет выполнять интеллектуальный анализ и синтез программ, используя накапливаемые программные решения программистом.

## 3. Предлагаемый путь создания интеллектуальной надстройки

Надстройка (плагин) - независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения

и/или использования её возможностей. Плагины обычно выполняются в виде библиотек общего пользования.

Для решения задачи предлагается использовать путь создания интеллектуальной надстройки над git. Надстройка на базе git, как конкретный проблемно-ориентированный САПР, должна иметь следующие характерные черты:

- надстройка может обеспечить для обучения базы знаний совокупность апробированных решений, представленных в виде текстов,
- надстройка не имеет явно выраженного языка задания ТЗ.

Соответственно, возможны следующие пути построения интеллектуальной надстройки, являющиеся вариантом модели САПР решения типичной задачи проектирования.

Создание целевого пространства систем (ЦПС), т.е. пространства возможных решений-прототипов.

Имеется два пути:

- надстройка может быть источником решений-прототипов; далее ЦПС в нужной форме представления создается инструментальной оболочкой автоматически;
- эксперт в данной предметной области может самостоятельно ввести ЦПС в нужной форме представления [6].

Уровни квалификации пользователя при избранном подходе к созданию экспертной системы отличаются его возможностью построить ЦПС (множество возможных решений-прототипов) и ПОС (множество ТЗ).

Возможны варианты, когда пользователь может построить:

- ЦПС и ПОС;
- только ЦПС;
- только ПОС;
- ничего, т.е. ни ЦПС, ни ПОС [1].

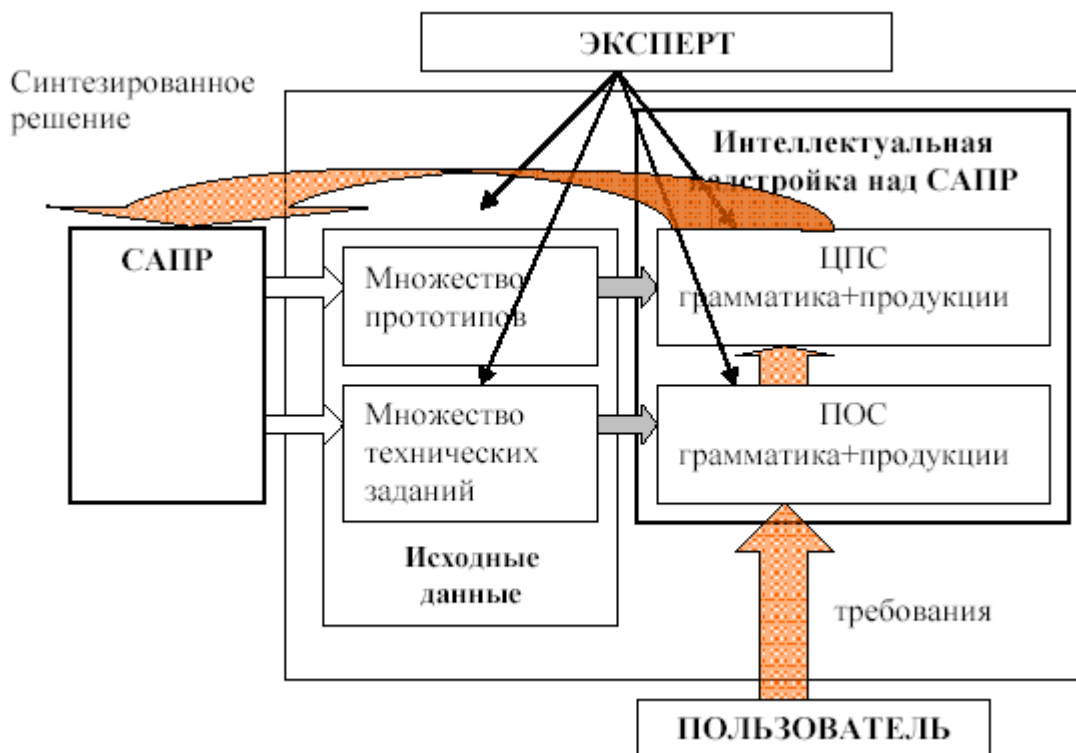


Рисунок 3 - Порядок создания и использования надстройки

### Заключение

Широкое применение СКВ открывают дальнейшие перспективы в их развитии, поиске новых решений и направлений их использования.

В данной работе проведён общий анализ возможностей и назначений СКВ, частный анализ систем контроля версий с расширенным функционалом, а также предложены пути создания интеллектуальной надстройки на базе СКВ такого класса.

Т.о., в данной работе предложено одно из новых возможных новых сфер применения СКВ, состоящее в использовании его функционала как базы для создания на их основе интеллектуального САПР ПО.

Суть предложенного пути состоит в создании интеллектуальной надстройки, которая позволит выполнять интеллектуальный анализ и синтез программ, используя накопленный в рамках СКВ опыт программистов в разработке программного продукта в конкретной программной области.

Как перспективную работу следует определить создание комплекса конкретных методов и механизмов реализации избранного пути.

### Список литературы

1. Григорьев А.В., Грищенко Д.А. - Интеллектуализация процесса проектирования аппаратуры средствами языка VHDL [Электронный ресурс] – Режим доступа: [http://masters.donntu.org/2007/fvti/tsatsenkina/library/stat\\_kosh.htm](http://masters.donntu.org/2007/fvti/tsatsenkina/library/stat_kosh.htm)
2. Git - О системе контроля версий [Электронный ресурс] – Режим доступа: <https://git-scm.com/book/ru/v2/Введение-О-системе-контроля-версий>
3. Анализ состояний методов, алгоритмов и программных средств в поддержке управления версиями [Текст] / Григорьев А.В., Гурин А.Г // Программная инженерия: методы и технологии разработки информационно вычислительных систем (ПТИВС-2018): сб. статей. – Донецк, 2018. – С. 43-47.
4. Git - Основы Git [Электронный ресурс] – Режим доступа: <https://git-scm.com/book/ru/v2/Введение-Основы-Git>
5. Git — Википедия [Электронный ресурс] – Режим доступа: <https://ru.m.wikipedia.org/wiki/Git>
6. Григорьев А.В. - Пути создания интеллектуальных САПР при различных уровнях квалификации экспертов [Электронный ресурс] – Режим доступа: <http://masters.donntu.org/2010/fknt/shaydt/library/article4.htm>

УДК 004.23, 004.891.2

## ОЦЕНКА ЭФФЕКТИВНОСТИ МЕТОДОВ И АЛГОРИТМОВ, ИСПОЛЬЗУЕМЫХ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЁРА

Савкин В.Ю., Светличная В.А.

Донецкий национальный технический университет  
кафедра автоматизированных систем управления  
E-mail: [vladislav1936@gmail.com](mailto:vladislav1936@gmail.com)

### *Аннотация.*

*Савкин В.Ю., Светличная В.А. Оценка эффективности методов и алгоритмов, используемых для решения задачи коммивояжёра. В данной статье исследуется решение задачи коммивояжёра. Приведена аналитическая и математическая постановка задачи коммивояжёра. В качестве методов решения задачи исследованы метод ветвей и границ, генетический и муравьиный алгоритмы, их распараллеленные версии. Описаны принципы реализации этих методов. Выполнен сравнительный анализ и оценена эффективность их использования.*

### *Annotation.*

*Vladislav Savkin, Victoria Svetlichnaya. Evaluating the effectiveness of methods and algorithms used to solve the traveling salesman problem. This article explores the solution of the traveling salesman problem. The analytical and mathematical formulation of the traveling salesman problem is given. As methods for solving the problem, the branches and bounds method, genetic algorithm, ant colony optimization algorithm and their parallelized versions are investigated. The principles of implementation of these methods are described. A comparative analysis was performed and the effectiveness of their using was evaluated.*

### **Введение.**

Задача коммивояжёра представляет собой задачу поиска кратчайшего Гамильтонова пути в полном конечном графе с  $N$  вершинами. Коммивояжёр, выходящий из какого-нибудь пункта, желает посетить  $N-1$  других пунктов и вернуться к исходному пункту. Известны расстояния между всеми этими пунктами. Требуется установить, в каком порядке он должен посещать эти пункты, чтобы маршрут можно было считать оптимальным (или субоптимальным) исходя из выбранного критерия [1].

### **Цель исследования.**

Целью данного исследования является выявление таких исходных параметров и настроек алгоритмов, при которых можно было бы достичь наиболее эффективного решения задачи коммивояжёра для заданного количества вершин. Эффективность решения задачи конкретным методом может быть оценена при помощи таких критериев, как относительная погрешность вычислений, временные затраты и ресурсоёмкость.

При этом предполагается исследовать метод ветвей и границ, генетический алгоритм и алгоритм муравьиной колонии, а также их распараллеленные версии:

### **Математическая постановка задачи.**

Имеется  $n$  пунктов, стоимость между которыми задана матрицей  $C$ . Коммивояжёр должен побывать в каждом пункте один раз и вернуться в исходный пункт маршрута, затратив при этом минимум средств. Математически это может быть представлено следующей формулой:

$$Z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min, \quad (1)$$

где  $n$  – количество пунктов на карте;

$c_{ij}$  – матрица стоимости между пунктами;

$x_{ij}$  – матрица переходов с компонентами:

$x_{ij} = 1$ , если маршрут включает переезд из точки  $i$  непосредственно в точку  $j$ ;

$x_{ij} = 0$ , в противном случае [2].

Постановка задачи представляет собой поиск минимума функции  $Z$  при выполнении ограничений и неотрицательности значений матрицы  $X$ . Система ограничений для данной задачи выглядит следующим образом:

$$\begin{cases} \sum_{j=1}^n x_{ij}, i = 1..n \\ \sum_{i=1}^n x_{ij}, j = 1..n \\ u_i - u_j + nx \leq n - 1, i, j = 1..n, i \neq j \end{cases} \quad (2)$$

где  $n$  – количество пунктов на карте;

$x_{ij}$  – матрица переходов с компонентами;

$u_i, u_j$  – произвольные целые неотрицательные числа.

При этом каждое из ограничений выражает следующие условия:

- коммивояжёр выезжает из каждого пункта один раз;
- коммивояжёр въезжает один раз в каждый пункт;
- маршрут не содержит в себе замкнутые маршруты (т.н. условие незамкнутости), кроме одного, включающего все пункты; кроме того, маршрут не содержит петель [3].

#### **Описание методов решения задачи и принципов их реализации.**

В качестве методов решения задачи коммивояжёра использованы метод ветвей и границ, генетический алгоритм, алгоритм муравьиной колонии (или муравьиный алгоритм), а также распараллеленная версия каждого из этих методов.

**Метод ветвей и границ.** Данный метод относится к точным методам и алгоритмам, которые не могут иметь временную сложность меньше экспоненциальной (при однопоточной реализации). Метод ветвей и границ является модификацией алгоритма полного перебора (поиск в пространстве  $N!$  решений посредством перебора всех вариантов). Он сохраняет все положительные свойства алгоритма полного перебора, но тем не менее мало пригоден для задач, где  $N$  не является значительно малым. Его суть заключается в добавлении проверки критерия ограничивающей функции, исходящего из знания задачи, по которому на определённом уровне можно приостановить построение данной ветви дерева перестановок [4]. В основе метода ветвей и границ лежат следующие этапы:

- вычисление нижней границы (оценки);
- разбиение на подмножества, т.е. ветвление;
- расчёт оценок;
- нахождение решений;
- определение признака оптимальности;
- оценка точности приближённого решения [2].

В ходе исследования был реализован классический метод ветвей и границ. В распараллеленном варианте данного метода параллельные вычисления использовались на этапах редукции матрицы расстояний, ветвления множества маршрутов и оценки путей маршрута.

Стоит отметить, что в некоторых случаях задачи с относительно большим количеством вершин были решены за приемлемое время с применением параллелизма. Однако, в целом наблюдался небольшой выигрыш во времени, и задачи с относительно большим количеством вершин всё так же требовали значительных временных затрат, пусть и не таких больших, как при однопоточной реализации.

С учётом данного фактора, при распараллеливании метода ветвей и границ требуется использовать крупнозернистый параллелизм, в основе которого лежит параллельное решение подзадач с обменом информацией о найденных (в ходе решения подзадач) значениях целевой функции на допустимых решениях (так называемых рекордах). При этом несколько подзадач могут обрабатываться одновременно отдельными процессами. Если один из процессов находит допустимое решение, то соответствующее рекордное значение целевой функции может быть разослано остальным процессам, позволяя им существенно ускорить работу, за счёт отбрасывания ветвей дерева обхода, заведомо не содержащих оптимального решения [5].

**Генетический алгоритм.** Генетические алгоритмы – это процедуры поиска, основанные на механизмах естественного отбора и наследования. В них используется эволюционный принцип выживания наиболее приспособленных особей. Они отличаются от традиционных методов оптимизации несколькими базовыми элементами. В частности, генетические алгоритмы обладают рядом отличительных свойств:

- кодирование параметров – генетические алгоритмы обрабатывают не значения параметров самой задачи, а их закодированную форму;
- операции на популяции – генетические алгоритмы осуществляют поиск решения исходя не из единственной точки (начальное приближение), а из некоторой популяции;
- использование минимума информации о функции – генетические алгоритмы используют только целевую функцию, а не производные либо иную дополнительную информацию;
- рандомизация операций – генетические алгоритмы применяют вероятностные, а не детерминированные правила выбора [6].

В ходе исследования был реализован классический генетический алгоритм для задачи коммивояжёра. Эксперимент проводился при различных входных параметрах алгоритма, таких как: размер популяции, максимальное число эпох, численность турнира, тип кроссовера, вероятность кроссовера, вероятность мутации. В качестве оператора селекции использовался турнирный метод отбора, а в качестве типа кроссовера использовались кроссоверы со следующими названиями: альтернативные рёбра, обмен подтурами, эвристический кроссовер. Лучший результат был достигнут с использованием следующих входных параметров:

- численность турнира - 2-3;
- тип кроссовера – эвристический кроссовер;
- вероятность кроссовера – 0,8;
- вероятность мутации – 0,007.

При этом размер популяции и максимальное число эпох настраивались пропорционально количеству вершин.

В распараллеленном варианте использовалась модель «клиент-сервер» (иногда её называют «рабочий-хозяин»), суть которой заключается в том, что сервер (основной поток) выполняет основные операции генетического алгоритма, а клиенты (потoki) выполняют оценку значений фитнес-функции всех особей популяции.

**Муравьиный алгоритм.** Идея муравьиных алгоритмов – моделирование поведения колонии муравьёв, связанного с их способностью быстро находить кратчайший путь от муравейника к источнику пищи и адаптироваться к изменяющимся условиям, находя новый

кратчайший путь. При своём движении муравей оставляет за собой дорожку феромонов, и эта информация используется другими муравьями для выбора пути. Это элементарное правило поведения и определяет способность муравьёв находить новый путь, если старый оказывается недоступным.

Если промоделировать процесс такого поведения на некотором графе, рёбра которого представляют собой возможные пути перемещения муравьёв в течение определённого времени, то наиболее обогащённый феромоном путь по рёбрам этого графа и будет являться решением задачи, полученным с помощью муравьиного алгоритма [7].

В ходе исследования был реализован муравьиный алгоритм с применением элитарной муравьиной системы. Эксперимент проводился при различных входных параметрах алгоритма, таких как: размер колонии муравьёв, количество итераций, коэффициент испарения феромона, коэффициент важности предыдущего опыта поиска, коэффициент важности расстояния между городами. Лучший результат был достигнут с использованием следующих входных параметров:

- коэффициент испарения феромона – 0,3;
- коэффициент важности предыдущего опыта поиска – 2;
- коэффициент важности расстояния между городами – 3;
- количество итераций – 10000.

При этом размер колонии муравьёв приравнивался к количеству вершин в задаче.

При параллельной реализации алгоритма на каждой итерации муравьи действовали независимо друг от друга, т.е. каждый из них находил собственный маршрут, при этом неявный обмен информацией выполнялся с помощью матрицы феромонов. Эта матрица представляет собой глобальную структуру данных, используемую всеми муравьями.

#### **Сравнительный анализ эффективности исследуемых методов.**

Программная реализация исследуемых методов и алгоритмов была выполнена на языке программирования C#. Для распараллеливания использовалась библиотека Task Parallel Library и технология Parallel LINQ (PLINQ). Тестирование проводилось на Intel Core i7-3632QM 3,2GHz. В ходе эксперимента исследовалась задача коммивояжёра с 20, 40, 60, 80 и 100 вершинами. Координаты вершин были сгенерированы псевдослучайным образом.

Сравнение эффективности исследуемых методов представлено в таблицах 1-2 [3].

Таблица 1 – Относительная погрешность методов, %

Кол-во вершин, N	Метод ветвей и границ	Генетический алгоритм	Муравьиный алгоритм
20	0,09	2,85	0,78
40	0,34	3,54	2,04
60	0,68	3,81	1,89
80	0,81	5,27	1,81
100	1,09	5,05	2,03

Таблица 2 – Временные затраты методов, мс

Кол-во вершин, N	Метод ветвей и границ		Генетический алгоритм		Муравьиный алгоритм	
	1 поток	Многопоточн.	1 поток	Многопоточн.	1 поток	Многопоточн.
20	5881	2156	245	165	104	86
40	557798	194538	740	368	924	689
60	19451803	5328936	1954	854	3603	1427
80	-	13939668	4130	1541	10342	4126
100	-	25194744	7139	2983	22230	9651



*Примечание.* «-» – решение не было найдено за приемлемое (8ч) время.

## **Выводы**

С учётом результатов исследования [5] [8] можно сказать, что выбор алгоритма решения задачи коммивояжёра во многом может зависеть от временного интервала, за который должно быть найдено решение. Если время работы алгоритма является определяющим фактором (т.е. решение нужно получить как можно быстрее), то в этом случае наиболее эффективным является генетический алгоритм, т.к. он быстрее всех находит решение, но при этом имеет и большую погрешность. Если время решения задачи условно не ограничено, то наиболее эффективным является метод ветвей и границ, т.к. точность вычислений этого метода наиболее высока по сравнению с другими представленными методами. Однако, стоит учесть, что в случаях, когда задача коммивояжёра содержит большое количество вершин, приемлемых временных затрат при использовании метода ветвей и границ можно достичь только с использованием крупнозернистого параллелизма. Наиболее же эффективным методом решения с точки зрения соотношения временных затрат и точности вычислений является муравьиный алгоритм, который наиболее широко используется при решении задачи коммивояжёра.

## **Литература**

1. Товстик, Т.М. Алгоритм приближённого решения задачи коммивояжёра / Т.М. Товстик, Е.В. Жукова // Вестник СПбГУ. – 2013. – №1. – С. 101-109.
2. Борознов, В.О. Исследование решения задачи коммивояжёра // Вестник АГТУ. Сер: Управление, вычислительная техника и информатика. – 2009. – №2. – С.147-151.
3. Савкин, В.Ю. Анализ методов решения задачи коммивояжера / В.Ю. Савкин, В.А. Светличная // Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ – 2018) / Материалы IX международной научно-технической конференции. – 2018. – С.6-10.
4. Костюк, Ю.Л. Эффективная реализация алгоритма решения задачи коммивояжёра методом ветвей и границ // Прикладная дискретная математика: Научный журнал – Томск : Национальный исследовательский Томский государственный университет. – 2013. – №2. – С.78–90.
5. Волошинов, В.В. Оценка производительности крупноблочного алгоритма метода ветвей и границ в вычислительной среде Everest / В. В. Волошинов, С. А. Смирнов // Программные системы: Теория и приложения. – 2017. – №1. – С.105-119.
6. Гараба, И.В. Сравнительный анализ методов решения задачи коммивояжёра для выбора маршрута прокладки кабеля кольцевой сети кольцевой архитектуры // Молодёжный научно-технический вестник. – 2013. – №1. – С.173-188.
7. Мурзин, Б.П. Использование алгоритма муравьиной колонии для определения оптимального маршрута доставки грузов / Б.П. Мурзин, В.А. Светличная // Інформаційні управляючі системи та комп'ютерний моніторинг (ІУС КМ-2011) / Збірка матеріалів II всеукраїнської науково-технічної конференції студентів, аспірантів та молодих вчених. – 2011. – С.183-186.
8. Семенкина, О.Е. Effectiveness Comparison of Ant Colony and Genetic algorithms for solving combinatorial optimization problems // Вестник Сибирского государственного аэрокосмического университета имени академика М. Ф. Решетнева. – 2012. – №1. – С.96-97.

УДК 007.51+004.021

## АНАЛИЗ ЭФФЕКТИВНОСТИ МЕТОДОВ РЕШЕНИЯ ЗАДАЧИ О ТРАНСПОРТИРОВКЕ ГРУЗА

**Бражник А.В.\*, Секирин А.И.\*, Валуева О.С.\*\***

Донецкий национальный технический университет

\*кафедра автоматизированных систем управления,

\*\*кафедра автоматики и телекоммуникаций

E-mail: [alexeybrazhnik1996@gmail.com](mailto:alexeybrazhnik1996@gmail.com)

### *Аннотация*

*Бражник А.В., Секирин А.И., Валуева О.С. Анализ эффективности методов решения задачи о транспортировке груза. В данной статье описаны основные особенности задачи о транспортировке груза, частью которой является задача коммивояжёра. Приведена математическая постановка задачи о транспортировке груза. Определены критерии оптимальности маршрута доставки и ограничения, влияющие на формирование маршрута и загрузку транспортных средств. Исследованы методы и алгоритмы, которые применялись для решения этой задачи.*

### *Annotation:*

*Brazhnik A.V., Sekirin A.I., Valueva O.S. Analysis of the effectiveness of methods for solving the problem of cargo transportation. This article describes the main features of the problem of cargo transportation, part of which is the traveling salesman problem. The mathematical formulation of the problem of cargo transportation is given. Criteria for optimizing the delivery route and restrictions affecting the formation of the route and the loading of vehicles are defined. The methods and algorithms that were used to solve this problem were investigated.*

### **Введение**

Задачу о транспортировке грузов целесообразно рассматривать в рамках прикладной задачи об оптимизации работы транспортной компании. Транспортная компания выполняет перевозки груза парком транспортных средств различной грузоподъемности и различного объема. Парк транспортных средств компании может быть разбит на несколько пунктов погрузки, следовательно, расстояния от различных пунктов погрузки для одного и того же пункта выгрузки будут различны. При этом каждый автомобиль может перевозить несколько грузов, при этом не менее одного и не более грузоподъемности этого транспортного средства.

Маршрут определяется порядком обхода точек выгрузки груза при условии, что начальной и конечной точкой маршрута является место погрузки, а точки выгрузки груза могут идти в любом порядке, однако необходимо принимать во внимание длину маршрутов между пунктами выгрузки и ограничение на сроки доставки груза [1].

Исходя из выше приведенных ограничений требуется определить такую загрузку транспортных средств и такие маршруты, при которых суммарные затраты на преодоление пути всеми автомобилями с грузом будут минимальны. Таким образом, в данной работе решается задача коммивояжера и частично задача о загрузке.

Задача коммивояжёра – одна из самых известных задач комбинаторной оптимизации, заключающаяся в поиске самого выгодного маршрута, проходящего через указанные города хотя бы по одному разу с последующим возвратом в исходный город [2]. В условиях задачи указываются критерий выгодности маршрута (кратчайший, самый дешёвый, совокупный критерий и тому подобное) и соответствующие матрицы расстояний, стоимости и тому

подобного. Как правило, указывается, что маршрут должен проходить через каждый город только один раз - в таком случае выбор осуществляется среди гамильтоновых циклов.

Оптимизационная постановка задачи коммивояжера относится к классу NP-трудных задач, впрочем, как и большинство её частных случаев. Время, необходимое для решения задачи, имеет экспоненциальную зависимость от количества пунктов, которые необходимо посетить коммивояжеру [3].

**Целью исследования** методов решения задачи о транспортировке груза является определение такого метода, который имеет наиболее оптимальное соотношение показателей по критериям минимизации относительной погрешности решения задачи и минимизации времени решения задачи. Для этого необходимо решить ряд следующих задач:

- формализовать постановку задачи;
- определить ограничения, влияющие на маршрут;
- определить критерии оптимальности решения задачи;
- определить наиболее эффективный алгоритм для решения задачи по вышеуказанным параметрам.

#### **Постановка задачи**

Рассматриваемая задача о транспортировке груза является частным случаем задачи коммивояжера, в котором присутствует разбиение на несколько начальных пунктов, в данном случае это пункты загрузки транспортных средств. Эти пункты расположены на различном удалении от пунктов выгрузки транспортных средств, таким образом, для каждого из пунктов загрузки необходимо составить свою собственную матрицу стоимости перевозок.

Целевая функция для задачи определяется по формуле 1.

$$\sum_{i=1}^N C_i L_i \rightarrow \min, \quad (1)$$

где  $C_i$  – стоимость перевозки груза  $i$ -м автомобилем на единицу расстояния,  $L_i$  – суммарная длина маршрута, по которому едет  $i$ -й автомобиль,  $N$  – количество автомобилей.

При этом необходимо учесть следующие ограничения, определяемые формулой 2.

$$\left\{ \begin{array}{l} 1. \sum_{j=1}^N M_j \leq MA_i, \\ 2. \sum_{j=1}^N V_j \leq VA_i, \\ 3. T_j \leq T_{jk}, \\ 4. K_{omnp} > 1, \end{array} \right. \quad (2)$$

где  $N$  – количество грузов в одном автомобиле,  $M_j$  – масса  $j$ -го груза в автомобиле,  $MA_i$  – грузоподъемность  $i$ -го автомобиля,  $V_j$  – объем  $j$ -го груза в автомобиле,  $VA_i$  – объем кузова  $i$ -го автомобиля,  $T_j$  – фактическое время разгрузки  $j$ -го груза,  $T_{jk}$  – критическое время разгрузки  $j$ -го груза,  $K_{omnp}$  – количество точек отправления автомобилей.

При этом ограничение 1 определяет, что суммарная масса груза, перевозимого одним автомобилем, не может быть больше его грузоподъемности. Ограничение 2 определяет, что суммарный объем груза, перевозимого одним автомобилем, не может быть больше его объема. Ограничение 3 определяет, что время доставки груза в точку выгрузки должно быть не позднее некоторого  $T_{критического}$ , которое задано в исходных данных. Ограничение 4 определяет, что пунктов отправки автомобилей несколько (более 1-го).

Также необходимо учесть следующие дополнительные ограничения:

1. грузоподъемность автомобилей различна
2. каждый груз может быть перевезен только одним автомобилем
3. пункты выгрузки различны для каждого груза

Критерием оптимальности маршрута согласно формуле 1 является минимальная суммарная стоимость перевозки всех грузов транспортными средствами.

#### **Обзор методов и алгоритмов**

Для решения задач коммивояжера применяются алгоритмы, которые можно отнести к трем группам: точные алгоритмы, эвристические алгоритмы и алгоритмы поиска. Далее приводится рассмотрение каждой из групп в отдельности.

**Точные алгоритмы** позволяют находить точное решение задачи, однако временные затраты на поиск решения данными алгоритмами в большой степени зависят от размерности задачи, и для задач коммивояжера большой размерности такие алгоритмы не могут найти решение за приемлемое для решения время. Наиболее полно описанными в научной литературе точными методами решения задачи коммивояжера являются:

- алгоритм полного перебора;
- метод ветвей и границ;
- алгоритм Литтла [2];
- метод последовательной сепарации.

**Эвристические алгоритмы** основаны на решении частных случаев задачи коммивояжера, вследствие чего имеют относительно большую погрешность при изменении исходных данных задачи, однако имеют наибольшую эффективность с позиций анализа временных затрат на выполнение алгоритма. Наиболее полно описанными в научной литературе эвристическими методами решения задачи коммивояжера являются:

- BV-метод [4];
- метод включения дальнего;
- метод имитации отжига;
- метод ближайшего соседа;
- метод Кларка-Райта [5];
- нейронные сети.

**Алгоритмы поиска** имеют наиболее сбалансированные точность решения и временные затраты среди алгоритмов для решения задачи коммивояжера. Наиболее полно описаны в научной литературе генетические алгоритмы и алгоритмы колонии муравьев, или муравьиные алгоритмы.

Генетические алгоритмы вдохновлены эволюционной теорией, согласно которой в природе непригодные и слабые виды в окружающей среде сталкиваются с исчезновением посредством естественного отбора. Самые сильные виды имеют больше возможностей для передачи их генов для будущих поколений путем воспроизводства. Этот процесс занимает много времени. В долгосрочной перспективе виды, несущие правильную комбинацию генов, становятся все более доминирующими [6].

Иногда, во время этого медленного процесса эволюции, случайные изменения могут происходить с генами. Эти непреднамеренные изменения могут предложить дополнительные преимущества для процесса естественного отбора через диверсификацию. В этом бесконечном испытании выживания, новые виды развиваются из старых, неудачные изменения и комбинации автоматически исключаются естественным отбором [6].

В терминологии генетических алгоритмов вектор решения называется особью или хромосомой. Хромосомы состоят из структурных единиц, называемых генами. Каждый ген представляет из себя один или несколько элементов хромосомы. В оригинальной реализации генетического алгоритма решения кодировались как вектора двоичных чисел, однако впоследствии были придуманы другие варианты кодировки решений.

Обычно хромосома является уникальным решением в пространстве решений. Генетический алгоритм работает с набором хромосом, называемых популяцией. Популяция обычно инициализируется случайным образом. В ходе работы алгоритма, происходит постепенное изменение состава популяции на особей с более высокой приспособленностью; у каждого кандидата есть набор свойств (генотип), который может быть видоизменен, что повышает вероятность поиска лучших решений. Алгоритм завершается, когда критерий был выполнен (максимальное количество итераций или достигнут удовлетворительный уровень фитнес-функции).

Муравьиные алгоритмы основаны на метаэвристических агентах, которые используются для поиска решений различных проблем оптимизации. Агенты называются искусственными муравьями. Эти алгоритмы были вдохновлены поведением муравьев и составляют вместе с другими вдохновленными рою алгоритмами большое семейство оптимизационной метаэвристики [7].

Алгоритм является результатом тщательного наблюдения за поведением социальных насекомых вообще и муравьев в частности. Муравьи являются социальными насекомыми, и они ведут коллективную деятельность, направленную на благо колонии. Каждый муравей независим и общается с другими муравьями через летучие химические вещества под названием феромоны.

Муравьи очень чувствительны к этим веществам и используют их чтобы обозначить их путь, чтобы быть обнаруженным их товарищами по гнезду. Наиболее распространенным примером является путешествие от гнезда до источника пищи. При обнаружении источника муравей откладывает феромоны на земле через свои брюшные железы. Каждый новый муравей может легко следовать по этому пахучему следу, который приведет его к источнику еды, а затем в гнездо. Таким образом, каждый муравей косвенно и коллективно помогает сообществу.

Алгоритм представляет собой последовательные проходы искусственных муравьев по графу, являющемуся представлением задачи коммивояжера, от начальной вершины до конечной. При этом муравьи откладывают по пути своего следования некоторое количество феромона. В свою очередь количество феромона на определенной дуге графа влияет на выбор этой дуги в качестве пути для очередного муравья. Поскольку феромон имеет свойство испаряться с течением времени, муравьи стремятся идти по более коротким маршрутам, на которых остается больше феромона. По истечении неопределенного количества итераций происходит схождение алгоритма до одного субоптимального пути, что является завершением работы алгоритма.

### Результаты сравнения

Анализ научной литературы, посвященной методам и алгоритмам решения задачи коммивояжера, показал очевидное превосходство поисковых алгоритмов над другими группами алгоритмов. В ходе проведения научного поиска были выделены два наиболее пригодных для решения поставленной задачи вида алгоритмов: генетический и муравьиный алгоритм. Результаты сравнения двух вышеуказанных видов алгоритмов на 3-х эталонных примерах приведены в таблице 1 [7].

Таблица 1. Результаты сравнения генетического и муравьиного алгоритмов

Пример		Генетический алгоритм			Муравьиный алгоритм		
Название	Лучшее решение	Решение	Погрешность, %	Время, с	Решение	Погрешность, %	Время, с
Berlin52	7542	7542	0.0	3.673	7575	0.4375	17.727
Eil76	538	570	5.9479	13.519	551	2.4163	73.906
A280	2579	3218	24.770	531.087	3092	19.8914	767.71

Оба алгоритма обеспечивают удовлетворительные результаты на четырех тестах; они показывают адекватную производительность на небольших тестах и устойчивость на средних и больших.

Генетический алгоритм отлично подходит для решения задачи коммивояжера малой и средней размерности; без каких-либо предварительных знаний о пространстве решения или помощи любой локальной процедуры поиска, он обеспечивает быструю сходимость и поиск оптимальных решений. Во всех случаях алгоритм является надежным и поддерживает постоянное улучшение поведения, он всегда дает решения и решения становятся лучше с течением времени.

Муравьиный алгоритм дает лучшие результаты в больших случаях. Однако он значительно проигрывает в производительности по сравнению с генетическим алгоритмом. Склонен к сходимости в локальном оптимуме, потому что он обновляет феромон в соответствии с текущим лучшим решением [8]. Но при достаточно большом времени для испарения феромонов муравьиный алгоритм обеспечивает поиск лучших результатов, чем генетический.

### **Выводы**

В этой работе представлен сравнительный анализ эффективности алгоритмов для решения задачи о транспортировке груза и основной из ее подзадач – задачи коммивояжера. Были исследованы алгоритмы, относящиеся к трем группам: точные, эвристические и алгоритмы поиска. Было проведено подробное сравнение качественных и временных характеристик двух известных алгоритмов поиска: генетических и муравьиных алгоритмов. Хотя оба алгоритма обеспечили удовлетворительные результаты по всем тестовым задачам, результаты их работы имеют видимые отличия. Для решения поставленной задачи с учетом приведенных ограничений был выбран генетический алгоритм, как быстрый, простой в реализации и эффективный с точки зрения вычислительных ресурсов.

### **Литература**

1. Гарнов, А. П. Инструментарий логистики: учебник / А. П. Гарнов, Н. С. Киреева – М.: Креативная экономика, 2009. – 310 с.
2. Товстик Т.М. Алгоритм приближенного решения задачи коммивояжера / Т.М. Товстик, Е.В. Жукова // Вестник СПбГУ. Серия 1. Математика. Механика. Астрономия. – СПб:СПбГУ, 2013. – с. 101-106.
3. Курейчик В.М. Генетический алгоритм решения логистической задачи / В.М. Курейчик, А.А. Рокотянский // Известия ЮФУ. Технические науки. – Таганрог: ТРТУ, 2012. – №136 – с. 245-251.
4. Борознов В.О. Исследование эвристического метода решения задачи коммивояжера / В.О. Борознов // Электронный научный журнал ИССЛЕДОВАНО В РОССИИ. – Москва: МФТИ, 2008. – с. 322-328.
5. Александрова О.А. Оптимизация грузовых перевозок с использованием генетических алгоритмов / О.А. Александрова, А.И. Секирин // Информатика и компьютерные технологии – 2009 №5, – Донецк: ДонНТУ, с. 237–244.
6. Скобцов Ю.А. Основы эволюционных вычислений. Донецк: ДонНТУ, 2008. – 326 с.
7. Haroun S.A. A Performance Comparison of GA and ACO Applied to TSP / S.A. Haroun, B. Jamal, E.H. Nisham // International Journal of Computer Applications. – 2015. – vol.117 – pp. 28–35.
8. Семенкина, О.Е. Effectiveness Comparison of Ant Colony and Genetic algorithms for solving combinatorial optimization problems // Вестник Сибирского государственного аэрокосмического университета имени академика М. Ф. Решетнева. – 2012. – №1. – С.96-97.

УДК 004.02

## МОДИФИЦИРОВАННЫЙ АЛГОРИТМ ВИОЛЫ-ДЖОНСА ДЛЯ РАСПОЗНАВАНИЯ ЛИЦ В ПОДСИСТЕМЕ ИДЕНТИФИКАЦИИ ЛИЧНОСТИ ПОСРЕДСТВОМ ПОРТРЕТНОЙ ЭКСПЕРИЗЫ

**Коношенко В.О.\*, Секирин А.И.\*, Валуева О.С.\*\***

\*Донецкий национальный технический университет  
кафедра автоматизированных систем управления

\*\*Донецкий национальный технический университет  
кафедра автоматизации и телекоммуникаций

E-mail: [konoshenko.vl@gmail.com](mailto:konoshenko.vl@gmail.com)

### **Аннотация:**

*Коношенко В.О., Секирин А.И., Валуева О.С. Модифицированный алгоритм Виолы-Джонса для распознавания лиц в подсистеме идентификации личности посредством портретной экспертизы. Рассмотрен общий математический подход данного алгоритма и его особенности применения. Определены параметры и инструменты, позволяющие улучшить показатели распознавания. Предложены параметры для оптимизации поиска лица в кадре.*

### **Annotation:**

*Konoshenko V., Sekirin A., Valueva O. Viola-Jones modified algorithm for face recognition in the subsystem of personal identification through portrait examination. The general mathematical approach of this algorithm and its application features are considered. Defined parameters and tools to improve recognition performance. The proposed parameters to optimize the search for a person in the frame.*

**Постановка проблемы.** На текущий момент существует множество различных методов и алгоритмов, которые позволяют обработать изображение и получить информацию о присутствующих на ней лицах. Виола и Джонс предложили алгоритм для распознавания лиц, который стал прорывом в области распознавания лиц. Однако модификация параметров данного алгоритма позволяет получить результаты, превосходящие исходное решение. Распознавание лиц – задача, которую человек выполняет без особых усилий несколько раз в день. Это одна из ключевых биометрических технологий. Распознавание лиц имеет ряд преимуществ перед другими биометрическими технологиями: это естественность, доступность и простота в использовании. [1]

**Цель работы.** Изучение методов и средств оптимизации практического применения алгоритма Виола-Джонса для дальнейшего его использования в подсистеме идентификации личности.

Алгоритм основан на принципе скользящего окна. Принцип скользящего окна основан на прохождении исходного изображения рамкой меньшего размера, которая поэтапно смещается на определенное расстояние. При прохождении данного окна по изображению на область применяется каскад слабых классификаторов (рис 1), который определяет присутствует ли лицо на изображении.

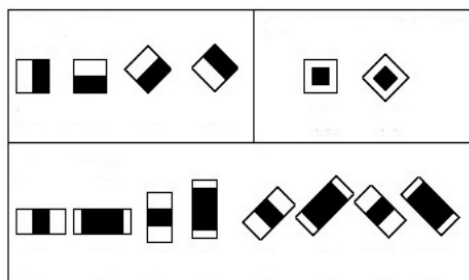


Рис 1. Примитивы Хаара

**Общая схема работы алгоритма.** Перед началом работы алгоритма распознавания необходимо обучить на основе тестовой выборки изображений базу данных, которая состоит из признаков, их приоритета и границ области. Алгоритм распознавания ищет обученный объект на разных масштабах изображения, используя базу данных. Критерием остановки поиска является порог количества кандидатов в прямоугольнике лица, наложенных друг на друга в одной области. Результатом работы алгоритма является множество найденных необъединенных объектов на разных масштабах. Общая схема работы алгоритма представлена на изображении ниже (рис.2).

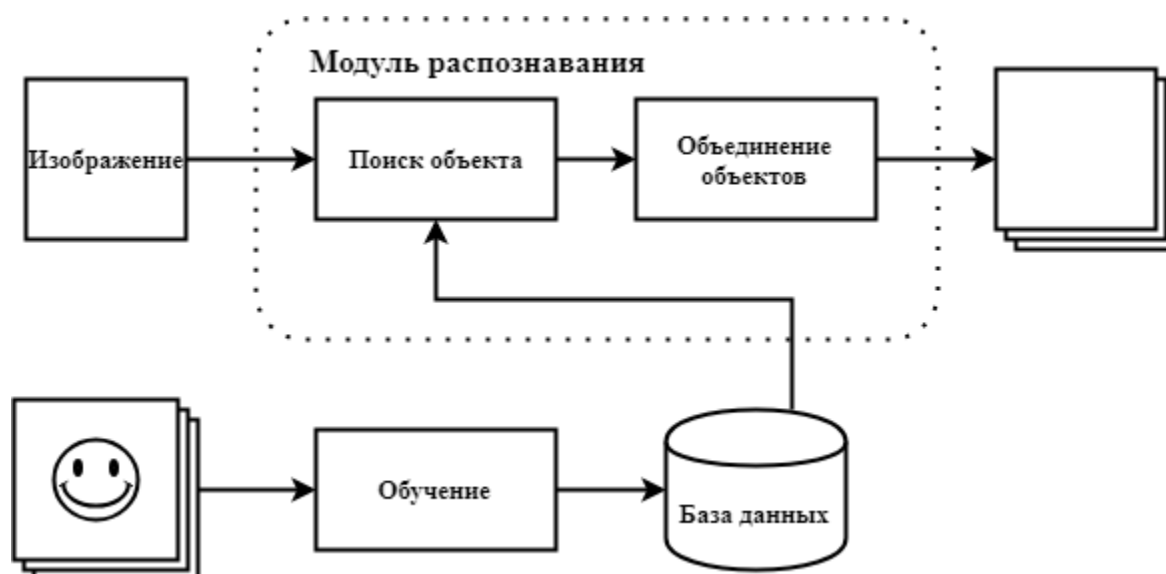


Рис 2. Общая схема работы алгоритма

### Математическая постановка задачи

Общая работа алгоритма разделена на два подалгоритма: алгоритм обучения и алгоритм распознавания. Признаками для алгоритма распознавания являются классификаторы Хаара[2]. Каждая из данных областей имеет свой размер светлой и темной области, пропорции, и минимальный размер. Следовательно, были предложены следующие признаки Хаара, как пространство признаков для класса лиц.

Признаки позволяют получить точное значение перепада яркости по оси X и Y. Общий признак Хаара для распознавания лиц представляет набор двух смежных прямоугольников, которые лежат выше глаз и на щеках. Значение признака определяется по формуле:



$$F = X - Y$$

где  $X$  – сумма значений яркостей точек закрываемых светлой частью признака,  $Y$  – сумма значений яркостей точек закрываемых темной частью признака.

Однако, при вычислении суммы значений интенсивностей для каждого признака необходимы значительные вычислительные мощности. Виолой и Джонсом было предложено использовать интегральное представление изображения.

Интегральное изображение  $Iimg(x,y)$ , построенное по исходному изображению  $K(i,j)$  это матрица, каждый элемент которой равен сумме значений яркости соответствующих пикселей на исходном изображении, расположенных слева и сверху от него.

$$Iimg(x,y) = \sum_{\substack{i \leq x, j \leq y \\ i=0, j=0}} K(i,j)$$

где  $K(i, j)$  — яркость пикселя исходного изображения, а каждый элемент матрицы  $Iimg[x, y]$  представляет собой сумму яркостей пикселей в прямоугольнике от  $(0,0)$  до  $(x, y)$ .

Расчет суммарной яркости пикселей произвольного участка изображения для анализа признака:

$$FIimg(x,y) = Iimg(x,y) - Iimg(x-1,y-1) + Iimg(x,y-1) + Iimg(x-1,y);$$

Интегральное представление позволяет быстро рассчитать суммарную яркость произвольной прямоугольной области. Вычисляются признаки, которые являются мерой схожести фрагмента изображения с классификаторами Хаара:

$$ImgAttribute = BrightPart - DarkPart$$

BPart – сумма значений яркостей точек закрываемых *светлой частью признака*,

DPart – сумма значений яркостей точек закрываемых *темной*.

Проверка наличия признака в фрагменте изображения:

$$D_f = \begin{cases} 1, \text{ объект существует на изображении } FragmentImg \\ 0, \text{ объект не обнаружен} \end{cases}$$
$$FragmentImg \rightarrow D_f,$$

где  $D_f$  — множество допустимых значений признака, а  $FragmentImg$  - характеризуемый признаками объект. Присутствующие признаки формируют вектор признакового описания объекта:

$$AttVector = \{f1, \dots, fn\}$$

Чтобы узнать есть ли в данной области лицо, необходимо отправить извлеченный вектор признаков на классификатор.

Классификатор – функция, определяющая принадлежность объекта к одному из известных классов. Сильный классификатор – функция от слабых классификаторов [3]. В зависимости от ответов слабых классификаторов, возвращается ответ такого вида: точно не лицо; или возможно, лицо.

Сильные классификаторы применяются последовательно в следующем порядке:

- 1) уменьшения «подозрительности»;
- 2) увеличения количества используемых слабых классификаторов.

Результатом работы классификатора является множество прямоугольников:

$$FaceImg(x, y, w, h)$$

При достижении порога количества прямоугольников на лице кандидата, наложенных друг на друга в одной области, предполагаемый фрагмент изображения является областью с лицом.

Полученную область изображения  $FaceImg(x,y,w,h)$  будет использоваться в следующей задаче - распознавания лиц.

Необходимо минимизировать время захвата и распознавания лица, а также количество ошибочных определений.

$$t_{обр} \rightarrow \min$$

$$\Delta DetectionError \rightarrow \min$$

DetectionError – ошибка распознавания

### Схема обучения

Современный алгоритм использует около 20 масок. Для каждой маски и каждой конфигурации тренируется классификатор, который дает наименьшую ошибку на всей тренировочной выборке, а после добавляется в базу данных признаков. Этим способом алгоритм обучается и результатом его работы является база данных из N слабых классификаторов. Условием остановки является достижение заранее заданного количества слабых классификаторов в базе (рис. 3).

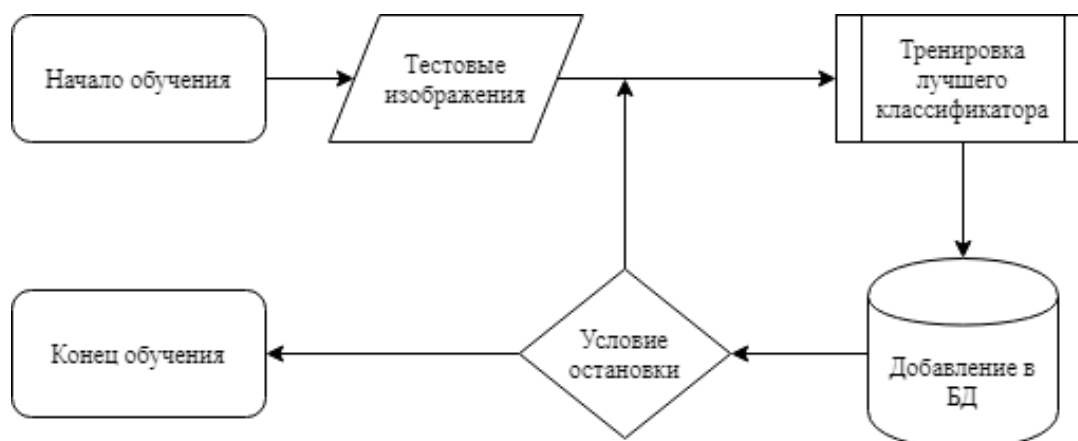


Рис 3. Схема обучения алгоритма.

Для реализация данного алгоритма будет использована OpenCV - библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом.

На ее основе будет проведена оптимизация работы алгоритма путем изменения ключевых параметров [4]:

scaleFactor - масштабный коэффициент, используемый для создания масштабной пирамиды. Масштабирование входящего изображения позволяет изменить размер более крупного лица на меньший, сделав его детектируемым по алгоритму. Оптимальное значение, полученное экспериментальным путем 1.1 - означает, что используется небольшой шаг для изменения размера, то есть уменьшение размера на 10% увеличивает вероятность совпадения размера с моделью. Это также означает, что алгоритм работает медленнее, поскольку он более тщателен.

minNeighbors - параметр, определяющий количество схождений, которые должен иметь каждый прямоугольник-кандидат. Данный параметр влияет на качество обнаруженных

лиц. Более высокое значение приводит к меньшему количеству обнаружений, но с более высоким качеством. 4~6 - хорошее значение для него.

minSize - минимальный возможный размер объекта. Объекты, меньшие, чем заданный размер игнорируются. Оптимально [30, 30] хорошие размеры для начала обнаружения лиц.

maxSize - максимально возможный размер объекта. Объекты, большие, чем заданный размер, игнорируются. Значение по умолчанию предполагает, что Вы хотите обнаружить без верхнего предела размер лица.

Преимущества данного метода:

1. Позволяет обнаружить более одного лица на изображении.
2. Простые классификаторы позволяют продемонстрировать хорошую скорость работы. Это позволяет использовать их в работе с потоком видео.

Недостатки данного метода:

1. Длительное время работы алгоритма обучения. В ходе обучения алгоритму необходимо проанализировать большое количество тестовых изображений.
2. Большое количество близко расположенных друг к другу результатов из-за применения различных масштабов и алгоритма скользящего окна.

### **Выводы**

В данной работе изложена математическая постановка задачи распознавания лица, а также предложен способ её решения при помощи алгоритма Виола-Джонса. Был предложен способ его модификации и оптимальные параметры для начальной работы распознавания лица. В дальнейшем планируется реализовать предложенный алгоритм с модификациями и проанализировать его эффективность. Дальнейшее развитие алгоритма планируется путем последующих модификаций для повышения эффективности работы алгоритма.

### **Литература**

1. Коношенко В. О., Привалов М. В., Пашкова Ю. И Сравнительный анализ методов распознавания лиц для использования в подсистеме идентификации личности // Материалы IX Международной научно-технической конференции «Информатика, управляющие системы, математическое и компьютерное моделирование» (ИУСМКМ-2018). – Донецк: ДонНТУ, 2018. – с. 245
2. Вейвлет Хаара [электронный ресурс] Википедия – свободная энциклопедия: [сайт]. [2016]. URL: [https://ru.wikipedia.org/wiki/Вейвлет\\_Хаара](https://ru.wikipedia.org/wiki/Вейвлет_Хаара)
3. Копелиович Михаил, аспирант Института математики, механики и компьютерных наук Алгоритм детектирования лица Виолы-Джонса.
4. Официальная документация библиотеки OpenCV [электронный ресурс] [https://docs.opencv.org/2.4/modules/objdetect/doc/cascade\\_classification.html](https://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html)

УДК 004.23, 004.891.2

## МОДИФИКАЦИЯ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ ЗАДАЧИ СОСТАВЛЕНИЯ ПЕРСПЕКТИВНОГО И ЕЖЕДНЕВНОГО МЕНЮ В УСЛОВИЯХ МЛАДШЕГО ДОШКОЛЬНОГО ВОСПИТАТЕЛЬНОГО УЧРЕЖДЕНИЯ

**Солоницын Л.П., Землянская С.Ю.**

Донецкий национальный технический университет  
кафедра автоматизированных систем управления

E-mail: [solleo55@gmail.com](mailto:solleo55@gmail.com)

### **Аннотация:**

*Солоницын Л.П., Землянская С.Ю. Модификация генетического алгоритма для задачи составления перспективного и ежедневного меню в условиях младшего дошкольного воспитательного учреждения. Рассмотрены принципы составления меню в младших воспитательных учреждениях. Определен подход оптимизации составления меню в младших воспитательных учреждениях. Предложен модифицированный генетический алгоритм для решения данной задачи.*

### **Annotation:**

*Solonitsyn L.P., Zemlyanskaya S.Y. Genetic algorithm modification for the task of perspective and daily menu compiling in the conditions of a junior pre-school educational institution. The principles of making menus in junior educational institutions are considered. The approach to optimizing the menu in junior educational institutions is defined. The genetic algorithm modification for solving this problem is proposed.*

### **Введение**

В каждом детском саду выполняются задачи составления перспективного и ежедневного меню, которые, обычно, выполняются медицинским работником. Однако, в части детских садов эту позицию достаточно часто занимают люди с не медицинской квалификацией, что положительно на общее здоровье детей не влияет. Поэтому существует необходимость создания интеллектуальной системы, которая на основе определённых параметров будет составлять перспективные и ежедневные меню, которые будут удовлетворять нормам СанПиН [1]. В данной статье предложен способ её решения используя модифицированный генетический алгоритм (ГА). [2]

### **Описание задачи**

Данная задача является задачей оптимизации. Входными данными является список продуктов со склада детского сада, технологические карточки блюд и нормы питания; выходными – перспективное меню за определённый период или меню на день.

Составление каждого из меню выполняется в определённые промежутки времени. Перспективное меню – один раз на определённый период времени, например, на 2 недели. Ежедневное меню или меню на день – лишь в том случае, если план поставок был нарушен, и в результате этого на складе отсутствуют необходимые продукты для следования составленному ранее перспективному меню на данный период.

Общий алгоритм составления меню на день можно представить следующим образом:

1. Получение списка имеющихся на складе продуктов.
2. Выбор блюда для каждого из приемов пищи.
3. Проверка, соответствует ли готовое меню установленным нормам и требованиям.
4. Пункты 2-3 повторяются до достижения необходимого результата.

Данный алгоритм повторяется необходимое количество раз, когда составляется перспективное меню, затем подводятся общие итоги и утверждается план поставки продуктов.

Описанная процедура усложняется тем, что рецептов блюд, утверждённых для приготовления в дошкольных детских учреждениях, существует большое количество, что приводит к возникновению следующих проблем:

1. Процесс составления меню может занять продолжительное время, так как даже опытный человек редко сразу выберет оптимальную комбинацию блюд.
2. Субъективность при выборе блюд, так как у всех есть свои предпочтения в еде.
3. Погрешность при расчёте необходимого количества продуктов и подведении итогов.

Эти проблемы являются частью любого процесса, в котором участвует человек. Однако есть возможность их практически полностью устранить при использовании автоматизированного программного обеспечения [1].

### Математическая постановка

Исходными данными для задачи составления меню на день являются:

- количество детей в яслях  $K_y$  и остальных группах  $K_o$ ;
- множество продуктов на складе  $P$ ;
- множество их стоимости за единицу измерения  $Q$ ;
- список норм для меню на день для яслей, таких как количество калорий  $Cal_{dy}$ , жиров  $Fat_{dy}$ , белков  $Pr_{dy}$  и углеводов  $Carb_{dy}$ , и список аналогичных норм для остальных групп  $Cal_{do}$ ,  $Fat_{do}$ ,  $Pr_{do}$  и  $Carb_{do}$ ;
- их возможные отклонения от нормы  $\Delta Cal_{dy}$ ,  $\Delta Fat_{dy}$ ,  $\Delta Pr_{dy}$ ,  $\Delta Carb_{dy}$ ,  $\Delta Cal_{do}$ ,  $\Delta Fat_{do}$ ,  $\Delta Pr_{do}$  и  $\Delta Carb_{do}$ , соответственно);
- множество блюд  $B$  с входящими в них продуктами  $R_y$  и  $R_o$ , т.е. рецептурой;
- стоимость блюд  $C_{by}$  и  $C_{bo}$ ;
- содержание калорий  $Cal_{by}$ , жиров  $Fat_{by}$ , белков  $Pr_{by}$  и углеводов  $Carb_{by}$  в одной порции блюда на детей из яслей и аналогичные  $Cal_{bo}$ ,  $Fat_{bo}$ ,  $Pr_{bo}$  и  $Carb_{bo}$  для детей из прочих групп.

Требуется найти такую комбинацию блюд (меню), которая является допустимой и наиболее оптимальной, при этом её параметры – стоимость меню на одного ребёнка  $C$ , количество калорий  $Cal$ , углеводов  $Carb$ , жиров  $Fat$  и белков  $Pr$  с индексами  $y$  для яслей и с индексами  $o$  для детей прочих групп – должны стремиться к заданным нормам. Допустимой является комбинация блюд, для которой есть продукты на складе для приготовления её на  $K$  детей, и значения её параметров, обозначенных выше, лежат в пределах соответствующих норм с учётом допустимых отклонений:

$$\left\{ \begin{array}{l} C_y = C_{dy} \pm \Delta C_{dy}; C_o = C_{do} \pm \Delta C_{do} \\ Cal_y = Cal_{dy} \pm \Delta Cal_{dy}; Cal_o = Cal_{do} \pm \Delta Cal_{do} \\ Fat_y = Fat_{dy} \pm \Delta Fat_{dy}; Fat_o = Fat_{do} \pm \Delta Fat_{do} \\ Pr_y = Pr_{dy} \pm \Delta Pr_{dy}; Pr_o = Pr_{do} \pm \Delta Pr_{do} \\ Carb_y = Carb_{dy} \pm \Delta Carb_{dy}; Carb_o = Carb_{do} \pm \Delta Carb_{do} \end{array} \right. , \quad (1)$$

Наиболее оптимальной является комбинация блюд, которая допустима, и выше обозначенные параметры которой максимально близки к заданным нормам.

Ограничениями в данной задаче служат:

- Количество продукта в комбинации блюд  $p_i$  не должно превышать количество продукта на складе  $P_i$ ,  $i = 1 \dots n$ .
- Количество калорий в комбинации блюд  $Cal_o$  и  $Cal_y$  должны быть близки к норме  $Cal_{do}$  и  $Cal_{dy}$  с погрешностью в  $\Delta Cal_{do}$  и  $\Delta Cal_{dy}$ , соответственно.
- Аналогичные ограничения существуют для количества жиров  $Fat$ , белков  $Pr$ , углеводов  $Carb$  и стоимости  $C$ .

С учётом приведенных выше ограничений была составлена следующая целевая функция:

$$F = g_1 * G_1(C_y) + g_2 * G_2(Cal_y) + g_3 * G_3(Carb_y) + g_4 * G_4(Fat_y) + g_5 * G_5(Pr_y) + g_6 * G_6(C_o) + g_7 * G_7(Cal_o) + g_8 * G_8(Carb_o) + g_9 * G_9(Fat_o) + g_{10} * G_{10}(Pr_o), \quad (2)$$

где  $G_i(X)$  – условная функция, определяет близость  $X$  к оптимальному значению;  
 $g_i$  – коэффициент приоритета соответствующего параметра.

Приоритет, у всех параметров равный, но может отличаться, если, например, необходимо восстановить их баланс, который был нарушен в прошлые дни.

Так как нам важно, чтобы все параметры лежали в установленных нами промежутках, и в зависимости от близости к желаемым значениям менялось и значение целевой функции в лучшую сторону, то была составлена условная функция, которая выполняет данную задачу:

$$G_i(X_i) = \begin{cases} -w_{>ub} + d_{>ub}, & \infty \geq X > ub \\ (1 - w_{ia}) + d_{<ub}, & ub \geq X > ia \\ w_{ia} + d_{>lb}, & ia \geq X > lb \\ w_{<lb} + d_{<lb}, & lb \geq X \geq 0 \end{cases} \quad (3),$$

где  $ia$  – идеальное значение параметра  $X$ ;

$ub$  – верхняя граница параметра  $X$ ;

$lb$  – нижняя граница параметра  $X$ ;

$w$  – отношение кол-ва параметра  $X$  к  $ub$ ,  $ia$  или  $lb$ , зависит от текущей функции;

$d$  – значение функции для параметра по умолчанию, можно изменять его для каждого параметра отдельно, для того чтобы управлять оценкой параметра.

Исходя из этого, итоговая математическая постановка выглядит следующим образом:

$$F = g_1 * G_1(C_y) + g_2 * G_2(Cal_y) + g_3 * G_3(Carb_y) + g_4 * G_4(Fat_y) + g_5 * G_5(Pr_y) + g_6 * G_6(C_o) + g_7 * G_7(Cal_o) + g_8 * G_8(Carb_o) + g_9 * G_9(Fat_o) + g_{10} * G_{10}(Pr_o) \rightarrow \max$$

$$\left\{ \begin{array}{l} K_o * p_i + K_y * p_i \leq P_i, p_i \geq 0, i = 1 \dots n \\ C_o = C_{do} \pm \Delta C_{do}; C_y = C_{dy} \pm \Delta C_{dy} \\ Cal_o = Cal_{do} \pm \Delta Cal_{do}; Cal_y = Cal_{dy} \pm \Delta Cal_{dy} \\ Carb_o = Carb_{do} \pm \Delta Carb_{do}; Carb_y = Carb_{dy} \pm \Delta Carb_{dy} \\ Pr_o = Pr_{do} \pm \Delta Pr_{do}; Pr_y = Pr_{dy} \pm \Delta Pr_{dy} \\ Fat_o = Fat_{do} \pm \Delta Fat_{do}; Fat_y = Fat_{dy} \pm \Delta Fat_{dy} \\ Cal_y, Carb_y, Fat_y, Pr_y, C_y, K_y, Cal_o, Carb_o, Fat_o, Pr_o, C_o, K_o > 0 \end{array} \right. , \quad (4)$$

где  $p_i = \sum_{j=1}^N R_{ji}$ ;

$$Cal = 9Fat + 4Carb + 4Pr.$$

Однако данная математическая модель не отображает следующие условия и ограничения, которые также необходимо учитывать при составлении меню и принимать во внимание при реализации алгоритма:

1. Разнообразии блюд. Блюда могут повторяться лишь раз в  $v$  дней, так как детям от 1 года до 7 лет требуется разнообразие в пище.
2. Некоторые блюда и продукты нежелательно совмещать в одном приёме пищи, по тем или иным причинам.
3. Приоритеты продуктов. Часть блюд включает в себя скоропортящиеся продукты, которые необходимо реализовать как можно скорее.
4. Период, на который составляется перспективное меню.

Все эти условия и ограничения являются важными для процесса составления меню и были приняты во внимание при составлении модели генетического алгоритма.

#### **Адаптация и модификация ГА для составления ежедневного и перспективного меню в условиях детского сада**

**Структура хромосомы.** При анализе структуры меню, было замечено, что во всех меню присутствуют «плавающая» и фиксированная части. Например, для обеда фиксированными элементами являются хлеб, первое блюдо (суп, борщ и т.д.) и напиток, а остальные элементы приёма пищи могут варьироваться. Т.е. в один день это могут быть вареники с капустой, а в другой – плов. Следует заметить, что хотя и количество элементов в этой «плавающей» части приёма пищи может отличаться, её вариаций для каждого приёма пищи существует ограниченное количество. Исходя из этого, была разработана следующая структура хромосомы, которая может применяться для решения любой подобной задачи:

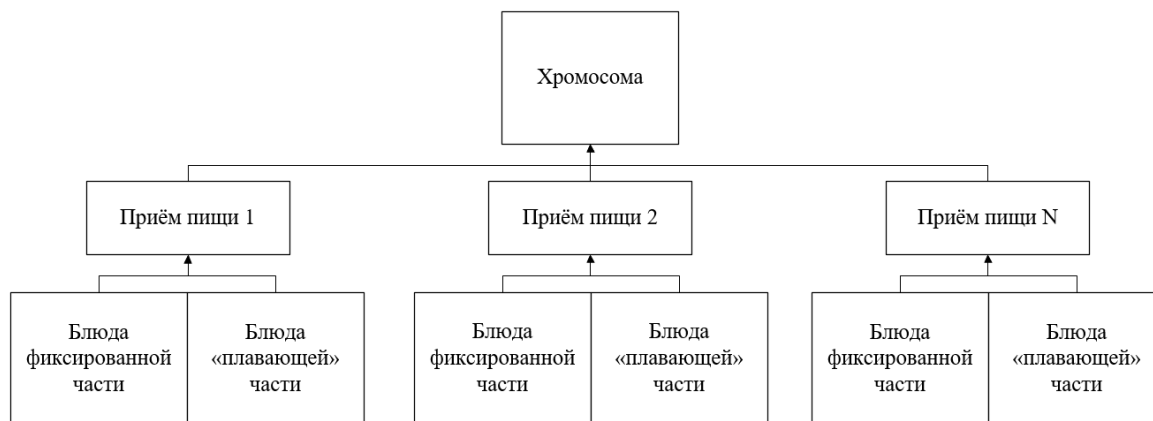


Рисунок 1 – Обобщённое представление хромосомы ГА

**Валидация особи.** В ходе работы алгоритма могут получаться решения, которые нельзя использовать по различным причинам. Для определения пригодности особи в ГА был добавлен этап валидации особи по различным факторам – наличие продуктов для приготовления меню на складе, присутствие несовместимых блюд, присутствие дубликатов в меню и т.д.

**Разнообразие при составлении перспективного меню.** Для сохранения разнообразия блюд при составлении перспективного меню на  $D$  дней с окном разнообразия  $v$  была придуман алгоритм, который создаёт множество уникальных блюд определённых типов и затем элементы этого множества удаляются из пула возможных блюд при составлении меню на  $i$ -ый день. При составлении меню на  $i+1$  день окно сдвигается,

множество исключаемых блюд пересобирается, а затем опять происходит их удаление из текущего пула возможных блюд.

**Селекция и редукция.** В качестве операторов селекции и редукции было решено использовать следующие способы отбора:

1. Турнирный отбор размером 2. Особь может участвовать в отборе несколько раз.
2. Турнирный отбор размером 2 с выбыванием, т.е. после победы особь больше не участвует в отборе.
3. Ранговая рулетка. Особь может участвовать в отборе несколько раз.

В операторах селекции и редукции было добавлено использование стратегии элитизма. В случае оператора селекции, перед проведением список родителей перемешивается методом Фишера-Йетса для снижения вероятности ситуации, в которой обмен информацией будет производиться только между очень близкими по значению фитнес-функции особями.

**Кроссовер.** Были разработаны 4 варианта оператора кроссовера, которые будут пригодны для структуры хромосомы, приведённой выше.

1. Обмен блюдами. Обмен блюд соответствующих типов между особями в соответствующих приёмах пищи. Например, гарниров в завтраке.
2. Обмен приёмами пищи. Обмен соответствующих приёмов пищи. Например, обмен всего состава обеда между особями.
3. Обмен «плавающих» частей приёмов пищи.
4. Обмен фиксированных частей приёмов пищи.

Для предотвращения 100%-го обмена информацией, что может привести в некоторых случаях к отсутствию новой информации в популяции, были введены по две вероятности для каждого варианта операции кроссовера – вероятность, что будет проводиться обмен в приёме пищи, и вероятность, что будет проводиться сам обмен.

**Мутация.** Были разработаны 5 вариантов оператора мутации:

1. Случайные блюда в приёмах пищи меняются на другие соответствующего типа.
2. Замена случайных блюд лишь в одном случайном приёме пищи.
3. Замена лишь одного случайного блюда в случайных приёмах пищи.
4. Замена «плавающей» части приёмов пищи на другую из списка возможных.
5. Замена блюд фиксированной части случайных приёмов пищи.

Для предотвращения 100%-го изменения информации были использованы вероятности, подобные тем, которые описанные выше для оператора кроссовера.

**Критерии остановки.** Для использования были выбраны следующие критерии:

1. Достижение максимального числа итераций.
2. В зале славы не сменяется лучшая хромосома заданное количество итераций.

Дополнительные критерии остановки рассматривались, однако не были продуманы до конца на данный момент.

## **Вывод**

В данной работе изложена математическая постановка задачи составления ежедневного и перспективного меню, а также предложен способ её решения при помощи модифицированного генетического алгоритма. Предложенный способ является отличным от существующих на базе генетического алгоритма [3, 4], муравьиных алгоритмов [5] и метода ветвей и границ [6], так как они не учитывают специфики составлению плана питания в детском саду. В дальнейшем планируется реализовать предложенный алгоритм и проанализировать его эффективность.



## Литература

5. Солоницын Л.П., Землянская С.Ю. Гримута А.В., Смирнов И.В. Интеллектуальная система составления перспективного и ежедневного меню в условиях младшего дошкольного воспитательного учреждения // Материалы IX Международной научно-технической конференции «Информатика, управляющие системы, математическое и компьютерное моделирование» (ИУСМКМ-2018). – Донецк: ДонНТУ, 2018. – с. 46-51
6. Скобцов Ю.А. Основы эволюционных вычислений / Ю.А. Скобцов. – Донецк: ДонНТУ, 2008.- 326с.
7. Gaál B., Vassányi I., Kozmann G. A novel artificial intelligence method for weekly dietary menu planning // Methods Archive 44.5 (2005): 655-664
8. Catalan-Salgado E-A, Zagal-Flores R., Torres-Fernandez Y., Paz-Nieves A. A Diet Generator Using Genetic Algorithms // Research in Computer Science (2014) (75):71-77
9. Rusin M., Zaitseva E. Hierarchical heterogeneous ant colony optimization // Computer Science and Information Systems (FedCSIS), 2012 Federated Conference on. IEEE, 2012
10. Sundmark N. Design and implementation of a constraint satisfaction algorithm for meal planning

УДК 004.021

## ОСОБЕННОСТИ РАСЧЕТА И ОПТИМИЗАЦИИ УЧЕБНОЙ НАГРУЗКИ В УСЛОВИЯХ АСУ ВУЗ

**Липова Э.Е., Секирин А.И.**

Донецкий национальный технический университет  
кафедра автоматизированных систем управления

E-mail: [elvira.lipova@yandex.ru](mailto:elvira.lipova@yandex.ru)

### **Аннотация:**

*Липова Э.Е., Секирин А.И. Особенности расчета и оптимизации учебной нагрузки в условиях АСУ вуза. Рассмотрены особенности расчета и оптимизации учебной нагрузки, существующие программные приложения и примененные ранее методы. Проведена математическая постановка задачи, учитывающая специфику учебной нагрузки вуза.*

### **Annotation:**

*Lipova E.E., Sekirin A.I. Features of the calculation and optimization of the study load in the conditions of the university automated control system. The features of the calculation and optimization of the study load, existing software applications and previously applied methods are considered. A mathematical formulation of the problem, taking into account the specifics of the academic load of the university.*

### **Общая постановка проблемы**

Педагогический процесс направлен на передачу знаний и умений преподавателя учащимся. Основным элементом организации такого процесса выступает планирование, в соответствии с законодательством и локальными актами учебной организации [1]. В планирование входит составление документации, основной целью создания которой является качественная подготовка учащихся по соответствующему профилю. К такой документации относится учебная нагрузка (УН). УН в первую очередь формируется по кафедре (УНК), затем распределяется между профессорско-преподавательским составом (УНППС). УНППС, в первую очередь, демонстрирует взаимосвязь преподавателя, дисциплин и групп, основанную на многочисленных факторах, влияющих на качество обучения в вузе. Первая часть факторов оговорена в соответствующих деловых бумагах, вторая часть, выражается в индивидуальных особенностях педагога: опыт работы со студентами, уровень знания, предпочтение конкретной дисциплины; предпочтение определённого вида занятий и др. Составление УНППС проводится на основании УНК заведующим кафедрой, затем проверяется и подписывается учебным отделом (УО). Длительность и трудоемкость данного процесса зависит от установленных ограничений, фактического штата и объёма читаемых предметов. Таким образом, чем больше профилей подготовки и, соответственно, предметов и выпускающих кафедр, тем сложнее процесс, что требует от ответственного сотрудника высокого профессионализма и значительных временных затрат. С целью повышения эффективности и производительности труда научно-преподавательского состава и, как следствие, качества образовательных услуг вуза предлагается рассмотреть возможные методы автоматизации распределения УНППС.

### **Специфика объекта автоматизации**

Распределение УНППС является эффективным распределением ресурсов для достижения поставленной цели, то есть планированием, что предполагает выполнение следующих этапов: постановка цели и задач; составление программы действий; вариантное составление

программы (вариантное проектирование); выявление необходимых ресурсов и их источников; определение непосредственных исполнителей и доведение планов до них; фиксация результатов планирования в материальном виде, например, в виде приказа в письменной форме. Качественная УНППС дает эффективное распределение учебных часов между ППС в соответствии с установленными ограничениями, для чего необходимо: соблюдать ограничения, накладываемые нормативными актами; обеспечить высокое качество образования; свести к минимуму предпосылки к синдрому эмоционального выгорания.

Нормативными актами выступают штатное расписание (ШР) и положение о нормах времени УН вуза. ШР – документ, устанавливающий структуру, штаты и должностной оклад сотрудников организации. Нормы времени УН определяют общий объем нагрузки по видам работ преподавателей (учебная, методическая, научная) согласно группам предметов, должностям и научным званиям. Эти документы составляются руководительским составом вуза и являются первостепенно важными ограничениями, накладываемыми на систему.

Сотрудник эффективней работает, когда проявляет интерес к порученной ему работе, поэтому руководство кафедрой учитывает их предпочтения по дисциплинам и учебным работам (аудиторным; практическим; лабораторным; консультациям; итоговому контролю).

УНППС является частью документооборота вуза и влияет на образовательный процесс, как показано на рисунке 1.



Рисунок 1 – Взаимосвязь документооборота и образовательного процесса вуза

Учебный план (УП) - нормативный документ, описывающий процесс обучения студентов, поступивших в текущем году. УНК формируется на основании утвержденных УП и включает потоки по предметам с учетом ограничений, установленных УО, только затем составляется УНППС, служащая основой для составления расписания учебных занятий.

Заведующий кафедрой определяют конкретные объемы отдельных видов нагрузки ППС с учётом численности кафедр, установленного для кафедр объёма работ, индивидуальных возможностей педагогических работников, а также затрат времени на повышение уровня их профессиональной подготовки [2]. Сотрудник эффективней работает, когда проявляет интерес к порученной ему работе, поэтому руководство кафедрой учитывает их предпочтения по дисциплинам и учебным работам, обеспечивает максимально возможную равномерность распределения часов между педагогами.

### Обзор инструментальных средств

Рассматриваемая в данной статье система уже реализована в ряде программных приложений. Проанализированы функциональные возможности двух систем, описанных в нескольких модулях: Magellan.Модуль «Кафедра», обеспечивающий формирование УНК и УНППС, и

БИТ.ВУЗ.Учет нагрузки преподавателей, отвечающий за составление УНППС (см. табл. 1) [3, 4].

Таблица 1 – Сравнительная характеристика инструментальных средств

№	Функционал	Magellan. Модуль «Кафедра»	БИТ.ВУЗ.Учет нагрузки преподавателей
1	Планирование УНК	+	+
2	Распределение УНППС	+	+
3	Гибкая настройка параметров расчета в соответствии с требованиями вуза	-	+
4	Импорт-экспорт УП	-	+
5	Учет выполнения УНППС преподавателями кафедры	-	+
6	Использование системы на компьютере, ноутбуке, планшете или смартфоне	+	+
7	Система электронного документооборота	+	+
8	Формирование отчетности	+	+
9	Автоматизированное формирование УНППС	-	-
10	Высокая стоимость программного обеспечения	+	+
11	Данные УП подгружаются только из дополнительного платного модуля	+	-

Существенным недостатком систем является не автоматизированное распределение нагрузки, из-за чего требуется наличие диспетчера. На основании постановки задачи видно, что передача такой задачи человеку - неэффективное решение.

### Обзор алгоритмов

Автоматизация формирования УНППС подразумевает регулирование процесса обучения с использованием методов оптимизации и ранжирования: поиска наилучших альтернатив и их упорядочивание по степени соответствия заданным критериям эффективности. Сотрудник имеет возможность перед запуском процедуры оптимизации задать значения, которые не будут изменяться.

С учетом вышеизложенного в качестве автоматизированной системы распределения УНППС использованы системы поддержки принятия решений (СППР). СППР представляет класс интеллектуальных систем, предлагающих множество альтернативных вариантов в условиях затрудненного объективного анализа предметной деятельности на основании имеющихся данных, построенных математических моделей и методов дискретной оптимизации [5]. Рассмотрены эволюционные методы сокращенного перебора дискретного математического программирования: искусственные нейронные сети (ИНС) [6]; генетический алгоритм (ГА) [7]; жадный алгоритм (ЖА) [8].

ИНС характеризуются высокой скоростью получения результата и возможностью распараллеливания вычислительного процесса, но они являются методами поиска локального минимума, поэтому для нахождения глобального решения соответствующий алгоритм необходимо запускать многократно из различных начальных состояний и выбирать наилучшие по качеству альтернативы [9]. Следовательно, использовать данный алгоритм может только на основании уже существующих рабочих планов и УНППС, что невозможно для сформированной в текущем году кафедры. Удовлетворительное качество результата будет достигнута только при наличии базы документов по кафедре за несколько лет. ИНС

сама формирует связи, вследствие чего возникает трудность формализации алгоритма выбора. В условиях постоянно изменяющейся нормативной базы данный недостаток критический.

ГА дает минимальную погрешность при нелинейных ограничениях. Трудоемкость вычисления функции зависит от размерности и сложности поставленной задачи [10, 11, 12]. Использование данного алгоритма для эффективного распределения УНППС требует качественной структурной оптимизации и больших вычислительных ресурсов, иначе единственный запуск функциональной оценки может потребовать значительных временных затрат для произведения необходимых расчетов.

ЖА обеспечивает сравнительно небольшой расход вычислительных ресурсов, поэтому в условиях ограниченных вычислительных мощностей вуза целесообразно использовать. В большинстве случаев, ЖА применяется к задачам, отвечающим двум условиям применимости: применимость принципа жадного алгоритма; обладание свойством оптимальности для подзадач [13]. Применимость алгоритма для эффективного формирования УНППС не доказана, при этом утверждается, что на практике метод дает наилучшее решение.

### Математическая постановка

Решение задачи распределения УНППС требует математической постановки. Ограничения системы составляются на основании решений, принимающих ЛПР при распределении УНППС.

Введем обозначение  $x_{ij}^t$  для назначения  $j$ -го вида учебной нагрузки  $i$ -й дисциплины  $t$ -му преподавателю [14].

$$x_{ij}^t = \begin{cases} 1, & \text{если } t - \text{му преподавателю назначается} \\ & j - \text{й вид учебного занятия} \\ & i - \text{й дисциплины,} \\ 0, & \text{в противном случае} \end{cases} \quad (1)$$

Значение  $x_{ij}^t$  при ручном решении поставленной задачи задается на основании следующих ограничений: соответствие учебных часов диапазонам норм времени, установленных трудовым законодательством и нормативам вуза; компетентность педагога в назначенной ему дисциплине; преемственность дисциплин; равномерность распределения нагрузки по видам занятий; предпочтения ППС в предметах и видах учебной нагрузки.

Выдвинуты следующие факторы для рецензирования знаний педагога в перечне кафедральных дисциплин: методические работы; общий и недавний опыт ведения  $t$ -ого курса. Следует отдавать предпочтение недавнему опыту, учитывая динамику развития преподаваемых наук. Удобство разработки УНППС включает в себя гибкую настройку, поэтому окончательное решение о важности факторов принимает ЛПР.

Согласно проведенному анализу, компетентность  $t$ -го преподавателя в  $i$ -ой дисциплине вычисляется по формуле:

$$C_i^t = \begin{cases} m_i^t * K_m + \frac{s_{\text{cont},i}^t}{s_{\text{total},i}^t} * K_s, & \text{при } s_{\text{total},i}^t \neq 0 \\ m_i^t * K_m, & \text{при } s_{\text{total},i}^t = 0 \end{cases}, \quad (2)$$

где  $m$  – количество методических работ;  $s_{\text{cont}}$  – непрерывный опыт;  $s_{\text{total}}$  – общий опыт;  $K_m$  – коэффициент важности методических работ;  $K_s$  – коэффициент важности опыта

Способность сотрудника вести  $i$ -й предмет напрямую зависит от его ученых степени и звания, должности. Данный комплекс формирует квалификацию  $t$ -го преподавателя:

$$w^t = d^t * r^t * p^t, \quad (3)$$

где  $d$  – ученая степень;  $r$  – ученое звание;  $p$  – должность.

Таким образом, оценка компетентности  $t$ -го преподавателя в  $u$ -ом виде нагрузки  $i$ -ой дисциплине:

$$R_{iu}^t = C_i^t * w^t. \quad (4)$$

Первой целевой функцией является максимизация качества обучения:

$$F_1(\bar{X}) = \sum_{t=1}^n \sum_{i=1}^s R_{ij}^t x_{ij}^t \rightarrow \max, \quad j = \overline{1, k}, \quad (5)$$

Равномерное распределение нагрузки выводится через минимизацию среднего квадратичного отклонения учебной нагрузки преподавателя:

$$F_2(\bar{X}) = \sqrt{\sum_{t=1}^n \left( \left( \frac{Y_t}{2} - Y_{осен, t} \right)^2 + \left( \frac{Y_t}{2} - Y_{весен, t} \right)^2 \right)} \rightarrow \min, \quad (6)$$

где  $Y_t$  - нагрузка  $t$ -го преподавателя на учебный год;  $Y_{осен, t}$  - нагрузка  $t$ -го преподавателя в осенний семестр;  $Y_{весен, t}$  - нагрузка  $t$ -го преподавателя в весенний семестр.

Данная формула может использоваться как для семестрового обучения, так и для модульного. Например, для трех периодов обучения получим:

$$F_2(\bar{X}) = \sqrt{\sum_{t=1}^n \left( \left( \frac{Y_t}{3} - Y_{осен, t} \right)^2 + \left( \frac{Y_t}{3} - Y_{зим, t} \right)^2 + \left( \frac{Y_t}{3} - Y_{весен, t} \right)^2 \right)} \rightarrow \min, \quad (6)$$

где  $Y_{зим, t}$  - нагрузка  $t$ -го преподавателя в зимний период [15].

Повышение качества обучения включает в себя повышение преемственности:

$$F_3(\bar{X}) = \sum_{i=1}^s f_{ij} \rightarrow \max, \quad (7)$$

где  $f_{ij}$  - показатель преемственности:

$$f_{ij} = \begin{cases} 1, & \text{если } x_{ij}^{\text{прошл}} = x_{ij}^{\text{буд}} \\ 0, & \text{в противном случае} \end{cases} \quad (8)$$

Максимизация желаний педагога выражена следующим образом:

$$F_4(\bar{X}) = \sum_{t=1}^n \sum_{i=1}^s \sum_{j=1}^k l_{ij}^t x_{ij}^t \rightarrow \max, \quad (9)$$

где  $l_{ij}^t$  - показатель желания вести  $j$ -й вид учебного занятия  $i$ -й дисциплины.

На вычисления накладываются ряд ограничений.

Нагрузка, допустимая трудовым законодательством вычисляется по формуле:

$$\sum_{i=1}^s \sum_{j=1}^k x_{ij}^t * \alpha_{ij} \leq \gamma^{\max}, \quad t = \overline{1, n} \quad (10)$$

где  $\alpha_{ij}$  - норматив соответственно для лекций, практических занятий и др.  $i$ -й дисциплины;  $\gamma^{\max}$  - рекомендуемая максимально допустимая нагрузка сотрудника по трудовому законодательству.

На нагрузку  $t$ -го преподавателя  $r$ -й должности исходя из диапазонов минимальной и максимальной нагрузки по должности сотрудника:

$$\gamma_p^{\min} \leq \sum_{i=1}^s \sum_{j=1}^k \sum_{\eta=1}^{\eta_0} Y_{ij}^{\eta \text{тп}} * x_{ij}^t \leq \gamma_p^{\max}, \quad t = \overline{1, n}, \quad Y_{ij}^{\eta \text{тп}} \geq 0 \quad (11)$$

где  $\gamma_p^{\min}$  - рекомендуемая минимальная нагрузка преподавателя  $r$ -й должности;  $Y_{ij}^{\eta \text{тп}}$  - доля нагрузки  $t$ -го преподавателя  $r$ -й должности;  $\gamma_p^{\max}$  - рекомендуемая максимально допустимая нагрузка преподавателя  $r$ -й должности.

На основании сформулированной математической постановки и сравнительной характеристики применяемых ранее методов эффективного составления УНПС, предлагается использование эволюционных методов.

## Выводы

Рассмотрены особенности формирования и эффективного распределения УНПЭС, ранее использованные для автоматизации этого процесса эволюционные методы: ИНС, ГА, ЖА. Сформулирована математическая постановка для анализируемой задачи, данная постановка в дальнейшем может быть использована вместе с эволюционным методом для реализации соответствующего программного обеспечения.

## Литература

1. Порядок организации учебного процесса в образовательных организациях высшего профессионального образования донецкой народной республики / Приказ Министерства образования и науки Донецкой Народной Республики от 10 ноября 2017 г. № 1171.
2. С. Н. Султанова, С. В. Тархов Модели и алгоритмы поддержки принятия решений при распределении учебной нагрузки преподавателей // Уфа: Вестник УГАТУ. 2006г. Т.7, № 3(16). С.107-114.
3. Модуль Кафедра. URL: <https://magellanius.ru/kafedra/> (Дата обращения: 19.01.2019).
4. Бит. ВУЗ URL: <https://www.pulsar.ru/progs/1903/> (Дата обращения: 21.01.2019).
5. И.П. Норенков Эволюционные методы в задачах выбора проектных решений / Норенков И.П., Арутюнян Н.М. // Научное издание МГТУ им. Н. Э. Баумана «Наука и образование». 2007г. №9.
6. В. В. Прищепа Перспективы применения технологий искусственного интеллекта при разработке системы поддержки принятия решений в составе программного комплекса автоматизации формирования нагрузки преподавателей кафедры / Прищепа В.В., Краснокутский В.А., Меренкова Л.Л. // Международная научно-техническая конференция «Информатика, управляющие системы, математическое и компьютерное моделирование»
7. А. Н. Шушура Синтез автоматизированной системы распределения учебной нагрузки преподавателей вузов на основе методов искусственного интеллекта / Шушура А. Н., Боровцова Е. В., Золотухина О. А. // Вестник ХНУ. 2012г. №1 (185). С. 151-155.
8. А. А. Сеньковская Моделирование процесса распределения учебной нагрузки кафедры с использованием жадного алгоритма / Сеньковская А. А., И.И. Фураева // Математические структуры и моделирование. 2017 г. № 4(44). С. 101-108
9. Н. А. Игнатъев О синтезе факторов в искусственных нейронных сетях // Вычислительные технологии ИВТ СО РАН. 2005г. Том 10. №3. С. 32-38.
10. В.М. Курейчик Генетические алгоритмы: Монография. Т: Изд. ТРТУ, 1998г. 242с.
11. С.В. Лаздынь, А. И. Секирин Оптимизация расписаний работы автоматизированных технологических комплексов механообработки с использованием генетических алгоритмов // ДонНТУ: Международный сборник научных трудов «Прогрессивные технологии и системы машиностроения». 2003г. № 25. С. 198-203
12. М.А. Безуглый, А.И. Секирин Методы повышения эффективности составления расписания в условиях учебного заведения // ДонНТУ: Международная научно-техническая конференция студентов, аспирантов и молодых ученых Компьютерная и программная инженерия. 2015г.
13. Т. Кормен Алгоритмы: построение и анализ / Кормен Т, Лейзерсон Ч., Ривест Р., Штайн К. // 2-е изд. М.: Вильямс, 2005г. 1296с.
14. С.В. Тархов, С.Н. Султанова Математическая модель распределения учебной нагрузки между преподавателями кафедры // Воронеж: Информационные технологии моделирования и управления. 2005г. №5 (23). С. 676-681.

УДК004.92

## ПОСТРОЕНИЕ И ИСПОЛЬЗОВАНИЕ ТВЕРДОТЕЛЬНОЙ МОДЕЛИ НЕПРАВИЛЬНОЙ ПИРАМИДЫ В СРЕДЕ AUTOCAD

**А.А. Ермолаева, В.В. Карабчевский**

Донецкий национальный технический университет

Кафедра компьютерного моделирования и дизайна

E-mail: [alka.ermolaeva@mail.ru](mailto:alka.ermolaeva@mail.ru)

### **Аннотация:**

*А.А. Ермолаева, В.В. Карабчевский. Построение и использование твердотельной модели неправильной пирамиды в среде AutoCAD. Рассмотрен алгоритм построения неправильной пирамиды средствами твердотельного моделирования в среде AutoCAD с помощью команды «Loft», а также применение полученной модели – нахождение натуральной величины сечения неправильной пирамиды.*

### **Annotation:**

*A.A. Ermolaeva, V.V. Karabchevsky. Constructing and using of a solid model of the irregular pyramid in AutoCAD. The algorithm of constructing an irregular pyramid by means of solid-state modeling in AutoCAD using the “Loft” command is considered and the application of the resulting model — finding the actual size of the section of an irregular pyramid.*

### **Общая постановка проблемы**

Освоение 3D моделирования предполагает решение задач, изученных с применением построений на комплексном чертеже, в трехмерном пространстве [1]. Для этого комплексный чертеж следует перевести в трехмерное представление путем поворота фронтальной плоскости вокруг оси абсцисс на 90 градусов.

### **Получение и использование каркасной модели**

На примере пирамиды покажем, как по комплексному чертежу в двух проекциях может быть получена каркасная модель пирамиды и ее сечения фронтально-проецирующей плоскостью.

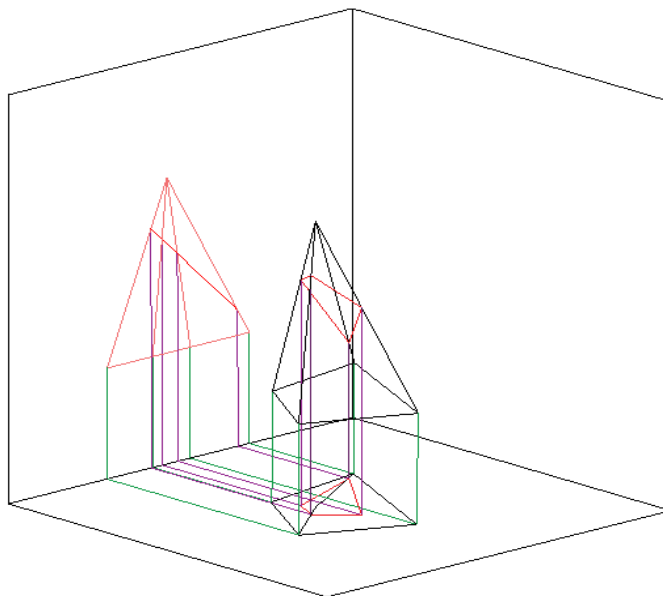


Рис.1. Каркасная модель пирамиды и сечения



После чего найдем натуральную величину сечения. Для этого воспользуемся способом замены плоскостей. На горизонтальной плоскости строим линию из одного из углов проекции сечения, параллельную фронтальной плоскости и находим расстояние из каждого угла горизонтальной проекции сечения до этой линии:

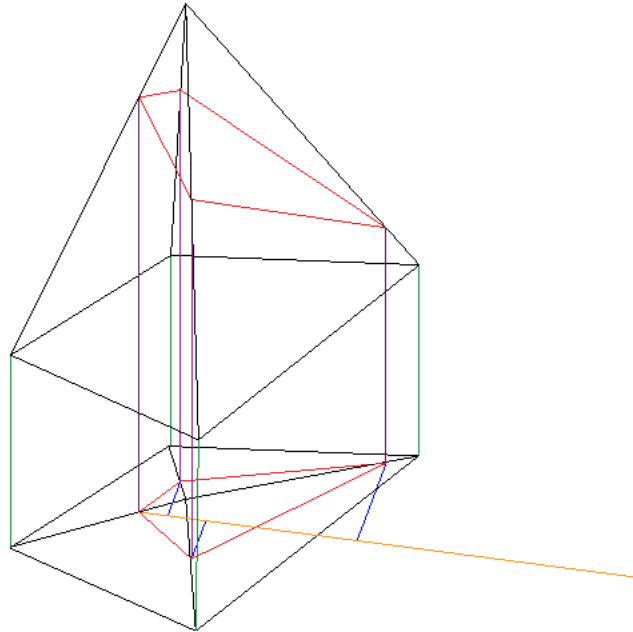


Рис.2. Нахождение расстояния

Далее на фронтальной плоскости строим проекцию полученной линии, на которой с помощью команды «Align» откладываем ранее найденные расстояния, соединяя полученные точки найдем натуральную величину сечения:

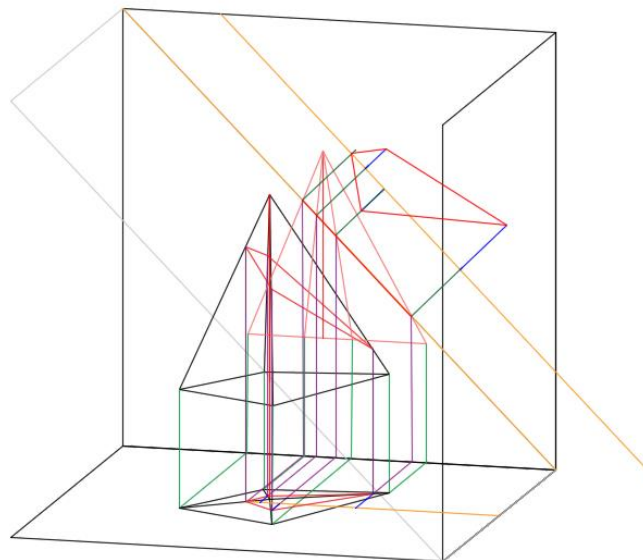


Рис.3. Натуральная величина сечения

Для построения пирамиды, основание которой представляет собой правильный многоугольник, средствами твердотельного моделирования в среде AutoCAD может быть использована команда «Pyramid». Однако при возникновении необходимости построения неправильной пирамиды этот способ не подходит. В таком случае мы можем воспользоваться командой «Loft» [2]. Принцип работы этой команды состоит в построении объекта по сечениям. В нашем случае мы будем использовать в качестве сечений основание пирамиды – многоугольник и вершину – точку.

Рассмотрим алгоритм построения:

1. С помощью команды «Polygon» или «Line» построим основание пирамиды. При использовании команды «Polygon» строим многоугольник, задав количество сторон, центр и режим вписанного или описанного вокруг окружности многоугольника, радиус этой окружности, а затем модифицируем длины сторон так, как нам необходимо. При использовании команды «Line» строим с помощью отрезков многоугольник, а затем с помощью команды «Join» объединяем в одну полилинию:

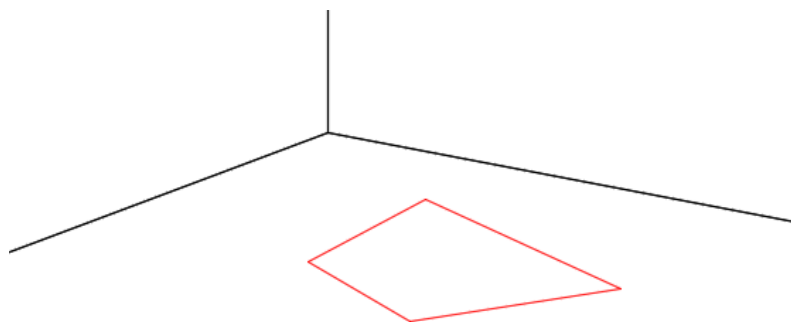


Рис.4. Основание неправильной пирамиды

2. С помощью команды «Point» строим точку, задав её координаты:

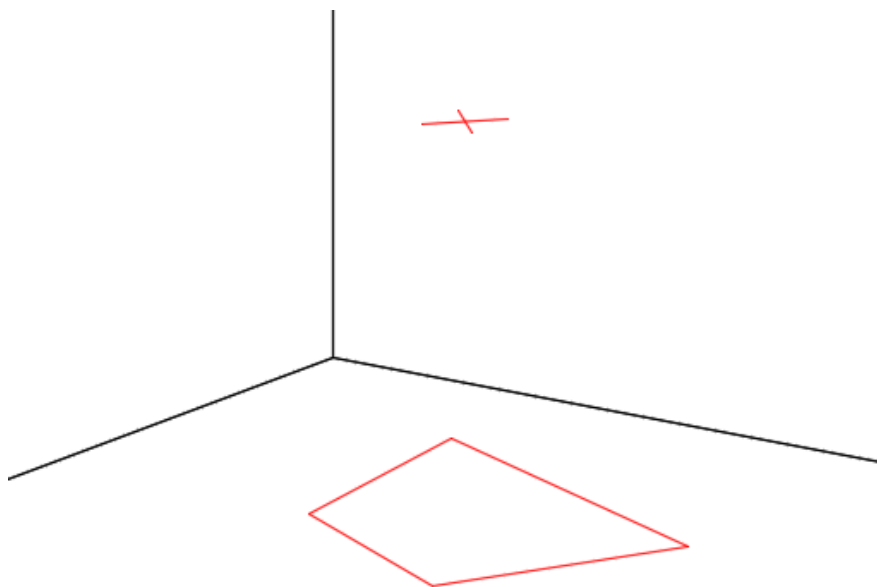


Рис.5. Вершина неправильной пирамиды

3. С помощью команды «Loft» строим неправильную пирамиду: задаем в порядке, требуемом для лофтинга, поперечные сечения – многоугольник и точку и переходим в режим «Точка», где в качестве временной точки объектного отслеживания выбираем вершину многоугольника и нажимаем Enter, после чего переходим в режим «Только поперечные сечения» и нажимаем Enter:

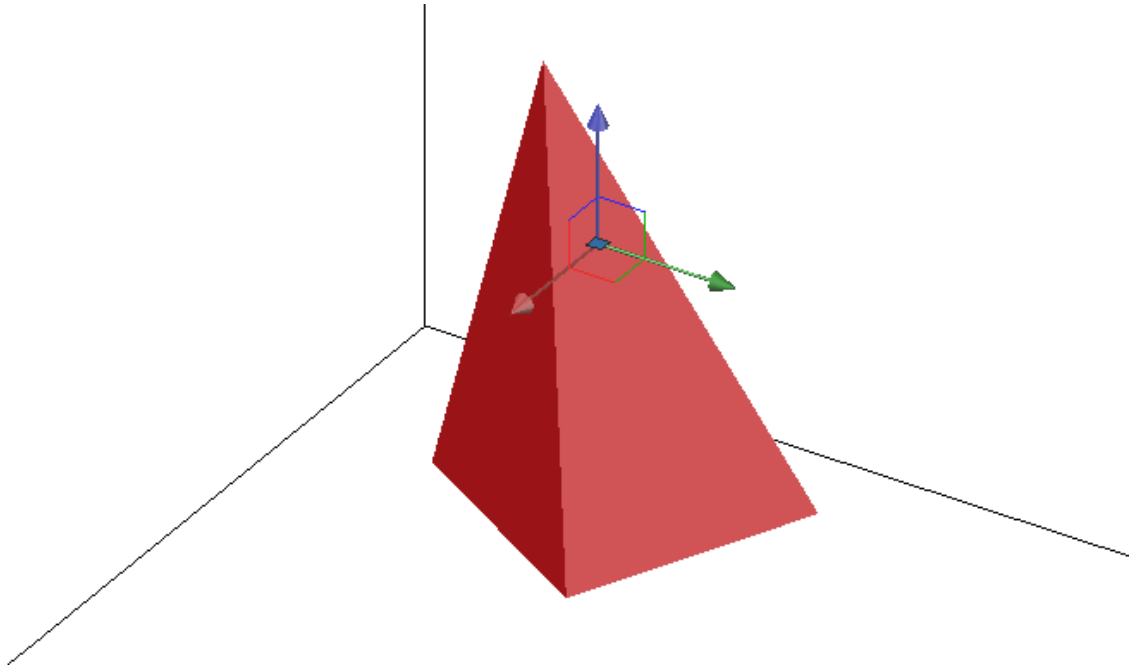


Рис.6. Неправильная пирамида

Таким образом, с помощью команды «Loft» была построена неправильная пирамида. Далее найдем натуральную величину сечения данной неправильной пирамиды.

4. Зададим секущую плоскость и с помощью команды «Section» построим сечение:

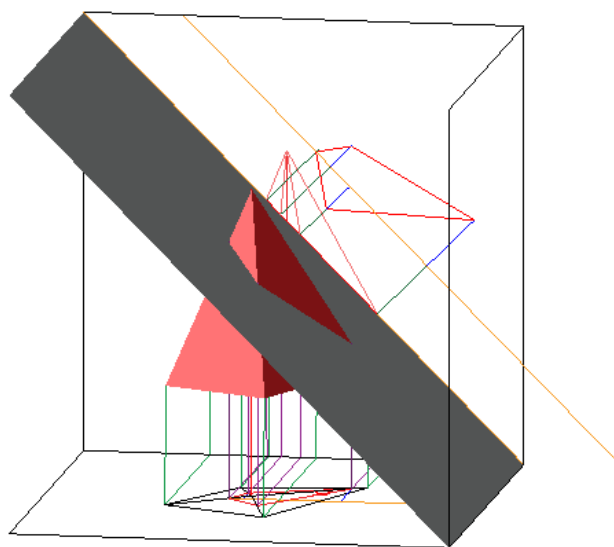


Рис.7. Сечение неправильной пирамиды

Сравним сечение, полученное на фронтальной плоскости и в результате использования команды «Section».

```
найдено: 2
REGION   Слой: "0"
          Пространство: Пространство модели
          Цвет: 237,31,36   Тип линий: "ПОСЛОУ"
          Вес линий: 0.53 мм
          Метка = 3f5
          Площадь: 31146.8586
          Периметр: 891.7549
Ограничивающая рамка: Нижняя граница X = 326.1059 , Y = 344.6820 , Z = 506.5231
                   Верхняя граница X = 584.4047 , Y = 576.5752 , Z = 506.5231
ПОЛИЛИНИЯ Слой: "0"
          Пространство: Пространство модели
          Цвет: 100,33,101  Тип линий: "ПОСЛОУ"
          Метка = 652
Замкнуто пространство
          площадь 31145.9999
          периметр 891.3523
ВЕРШИНА Слой: "0"
          Пространство: Пространство модели
```

Рис.8. Сравнение найденных сечений

Видно, что геометрические характеристики полученных сечений не совпадают, хотя и не значительно, геометрически точным является сечение, полученное по каркасной модели. Применение команды «Rugamid» для построения правильной пирамиды позволяет получить твердотельный объект, сечение которого совпадает с сечением, полученным с применением инструментов начертательной геометрии [3].

### Выводы

Полученная с использованием команды «Loft» модель неправильной пирамиды не в полной мере соответствует геометрии моделируемого объекта, поэтому целесообразность применения такой модели зависит от требуемой точности результатов.

### Литература

1. Карабчевский В.В. Трехмерное моделирование при решении позиционных и метрических задач в учебном процессе // Прикладна геометрія та інженерна графіка. Праці. – Таврійський державний агротехнологічний університет – Вип. 4, т.56. – Мелітополь: ТДАТУ, 2013, С. 176-186.
2. Операция ЛофтAutocad [Электронный ресурс]. – Режим доступа: <https://knowledge.autodesk.com/ru/support/autocad>. – Дата обращения: 26.04.2019. – Загл. с экрана.
3. Карабчевский В.В. Методы компьютерной геометрии. Донецк: ГВУЗ «ДонНТУ», Технопарк ДонНТУ «УНИТЕХ», 2010. – 179 с.

УДК 004.942+536.2

## КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ЯВЛЕНИЯ ТЕПЛОПРОВОДНОСТИ В ОРТОТРОПНОЙ ПЛАСТИНЕ ПРИ СОСРЕДОТОЧЕННОМ ВОЗДЕЙСТВИИ ИСТОЧНИКА ТЕПЛА

Кравченко Т.М., Гольцев А.С.

Донецкий национальный университет

кафедра прикладной механики и компьютерных технологий

E-mail: [tatuanakravchenko1997@gmail.com](mailto:tatuanakravchenko1997@gmail.com)

### **Аннотация:**

*Кравченко Т.М., Гольцев А.С. Компьютерное моделирование явления теплопроводности в ортотропной пластине при сосредоточенном воздействии источника тепла. Рассмотрено явление теплопроводности в тонких ортотропных пластинах, на которые воздействуют сосредоточенные источники тепла. Построена компьютерная модель, позволяющая наглядно представить результаты исследований и сравнить с температурным полем в изотропной пластине. При разработке использовались: среда разработки Microsoft Visual Studio 2012, язык программирования C++ и математический пакет Maple.*

### **Annotation:**

*Kravchenko T.M., Goltcev A.S. Computer simulation of the phenomenon of heat conduction in an orthotropic plate with concentrated exposure to a heat source. The phenomenon of heat conduction in thin orthotropic plates, which are affected by concentrated heat sources, is considered. A computer model was built, which allows visually presenting the results of the research and comparing it with the temperature field in an isotropic plate. The following were used in the development: Microsoft Visual Studio 2012 development environment, C++ programming language and Maple math package.*

### **Введение**

Исследование различных проблем теплопередачи играет важную роль при проектировании газотурбинных двигателей и установок, авиационной и ракетной техники. При этом помимо эффективного увеличения термического сопротивления по толщине стенок несущих и теплоотводящих конструкций, требуется обеспечение достаточной прочности и жесткости при ограничении веса.

При рассмотрении процесса сварки нужно также учитывать процесс теплопереноса в конкретном веществе, так как без этого невозможно выбрать правильный способ и режим сварки. Таким образом, применение результатов подобных исследований может быть использовано в промышленности и строительстве.

Основной задачей теории теплопроводности является определение и изучение пространственно-временного изменения температуры  $T = f(x, y, z, \tau)$ , где  $x, y, z$  – пространственные прямоугольные координаты, а  $\tau$  – время [1]. Наиболее простыми задачами являются расчеты стационарной теплопроводности в таких случаях, когда температура меняется только по одной координате, т. е. когда температурное поле одномерно.

Но на практике приходится сталкиваться с ситуациями, когда тепло распространяется от сосредоточенного источника в разных направлениях, причем нужно учитывать возможные различия свойств материалов в разных направлениях (анизотропия). В работе рассматривается задача распространения тепла в ортотропной пластине, свойства которой изменяются по взаимно перпендикулярным направлениям.

### Постановка задачи

Математическая постановка и метод решения данной задачи предложены в работе [2]. Рассмотрим тонкую ортотропную пластину толщиной  $2h$ , которая находится в тепловом контакте с внешней средой нулевой температуры. Через лицевые поверхности пластины осуществляется конвективный обмен с внешней средой по закону Ньютона.

Пусть оси координат расположены следующим образом: оси  $x$ ,  $y$  – в срединной плоскости пластины, а ось  $z$  – перпендикулярно срединной плоскости.

Объемные источники тепла распределены равномерно по толщине пластины в фиксированной точке  $(x, y)$ .

Для того чтобы перейти от решения трехмерной задачи к двумерной, рассматриваемой на плоскости, необходимо проинтегрировать по  $z$  объемную плотность источников тепла  $W_0$  ( $\frac{Дж}{м^3 \cdot с}$ ). Таким образом, получим  $W_1$  – интегральный источник тепла,

источник средней температуры [3].

$$W_1 = \frac{h}{2\lambda_{33}} w_1, \quad w_1 = \int_{-h}^h W_0 dz,$$

где  $\lambda_{33}$  – коэффициент теплопроводности в направлении координаты  $z$ .

Рассмотрим случай симметричного теплообмена, когда параметры теплообмена ( $Bi^+$  и  $Bi^-$ , где  $Bi$  – критерий Био [1]) на верхней и нижней лицевых поверхностях пластины одинаковы ( $Bi^+ = Bi^- = Bi$ ). Действие сосредоточенного источника тепла моделируем с помощью дельта-функции Дирака, стоящей в правой части уравнения теплопроводности. Плотность источников средней температуры в этом случае должна иметь вид

$$W_1(x_1, x_2) = W_1^* \delta(x_1, x_2)$$

где  $\delta(x_1, x_2)$  – двумерная дельта-функция Дирака;  $W_1^*$  – интенсивность сосредоточенного "плоского" источника тепла.

Будем учитывать линейное распределение температуры  $T(x, y, z)$  по толщине пластины, которое представимо в виде [2, 3]

$$T(x, y, z) = T_1(x, y) + \frac{z}{h} T_2(x, y)$$

где  $T_1$ ,  $T_2$  – интегральные характеристики температуры, средняя температура и температурный момент, соответственно. Эти величины определяются следующими формулами:

$$T_1(x, y) = \frac{1}{2h} \int_{-h}^h T(x, y, z) dz; \quad T_2(x, y) = \frac{3}{2h^2} \int_{-h}^h z T(x, y, z) dz.$$

В случае действия лишь источника средней температуры в условиях симметричного теплообмена температурный момент во всей области пластины не возникает, т.е. равен нулю. Тогда температура пластины будет определяться по формуле

$$T(x, y, z) = T_1(x, y).$$

Уравнения теплопроводности в этом случае в безразмерной системе координат

$$x_1 = x/h; \quad x_2 = y/h; \quad x_3 = z/h$$

будут иметь следующий вид:

$$\Delta_\lambda T_1(x_1, x_2) - \mu_1 T_1(x_1, x_2) = -W_1^* \delta(x_1, x_2), \quad (1)$$

где 
$$\Delta_{\lambda} = \lambda_1 \frac{\partial^2}{\partial x^2} + \lambda_2 \frac{\partial^2}{\partial y^2}, \quad \lambda_1 = \frac{\lambda_{11}}{\lambda_{33}}, \quad \lambda_2 = \frac{\lambda_{22}}{\lambda_{33}};$$

$\lambda_{11}$ ,  $\lambda_{22}$ ,  $\lambda_{33}$  – главные коэффициенты теплопроводности;  $\lambda_1$ ,  $\lambda_2$  – относительные коэффициенты теплопроводности;  $\mu_1 = \frac{1}{2}(Bi^+ + Bi^-)$ .

Будем считать, что край пластинки находится на значительном удалении от места приложения температурной нагрузки. Поскольку рассматривается локальное температурное поле, вызванное сосредоточенным источником тепла, будем полагать, что температура затухает при приближении к краям пластины. Поэтому пластину можно считать бесконечной и не учитывать граничные условия при решении задачи.

### Методика решения

Предложенный метод решения основан на применении двумерного интегрального преобразования Фурье к уравнению теплопроводности (1).

Уравнение теплопроводности в пространстве трансформант имеет следующий вид:

$$-(\lambda_1 \xi_1^2 + \lambda_2 \xi_2^2) \tilde{T}_1 - \mu_1 \tilde{T}_1 = -W_1^* \frac{1}{2\pi}. \quad (2)$$

Решая его, найдем трансформанту температуры:

$$\tilde{T}_1 = \frac{W_1^*}{2\pi} \frac{1}{\lambda_1 \xi_1^2 + \lambda_2 \xi_2^2 + \mu_1}.$$

Применяя обратное интегральное преобразование, получим

$$T(x_1, x_2) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{W_1^*}{2\pi} \frac{1}{\lambda_1 \xi_1^2 + \lambda_2 \xi_2^2 + \mu_1} e^{-i(\xi_1 x_1 + \xi_2 x_2)} d\xi_1 d\xi_2.$$

Применив формулы Эйлера и после перехода к полярным координатам:  $x_1 = r \cos \varphi$ ;  $x_2 = r \sin \varphi$ ;  $\xi_1 = \rho \cos \theta$ ;  $\xi_2 = \rho \sin \theta$ , и, заменив  $\lambda_1 \cos^2 \theta + \lambda_2 \sin^2 \theta = \lambda^2(\theta)$ , получим следующее:

$$T_1(r, \varphi) = \frac{W_1^*}{\pi^2} \int_0^{+\infty} \int_0^{\frac{\pi}{2}} \frac{1}{\rho^2 \lambda^2(\theta) + \mu_1} \cos(r\rho \cos \varphi \cos \theta) \cos(r\rho \sin \varphi \sin \theta) \rho d\rho d\theta.$$

Применяя разложение из соотношения Якоби [4], получим

$$T_1(r, \varphi) = \frac{W_1^*}{4\pi^2} \sum_{n=0}^{\infty} (-1)^n \varepsilon_n \cos 2n\varphi \int_0^{\frac{\pi}{2}} \frac{\cos 2n\theta}{\lambda^2(\theta)} d\theta \int_0^{+\infty} \frac{\rho J_{2n}(\rho r)}{\rho^2 + a^2(\theta)} d\rho, \quad (3)$$

где  $a^2(\theta) = \mu_1 / \lambda^2(\theta)$ .

С учётом интегрального представления специальной функции  $G_{n,\nu}(z)$  [4], получим выражение для средней температуры (4):

$$T_1(r, \varphi) = \frac{W_1^*}{4\pi^2} \sum_{n=0}^{\infty} \varepsilon_n \cos 2n\varphi \int_0^{\frac{\pi}{2}} \frac{\cos 2n\theta}{\lambda^2(\theta)} G_{n,n}(a(\theta)r) d\theta. \quad (4)$$

Для вычисления средней температуры по формуле (4) необходимо подпрограмма численного интегрирования, для реализации которой был выбран метод Гаусса.

Подпрограмма расчёта интегралов с заданной точностью 5, 10 или 20 узлов по методу Гаусса была написана на языке C++ и использована при расчёте температурного поля ортотропной пластины по формуле (1).

Для лучшего анализа полученных результатов представляет интерес визуализация полученной картины температурного поля, а также сравнение полученной картины температурного поля для ортотропной пластины с полем для изотропной (рис. 1, 2).

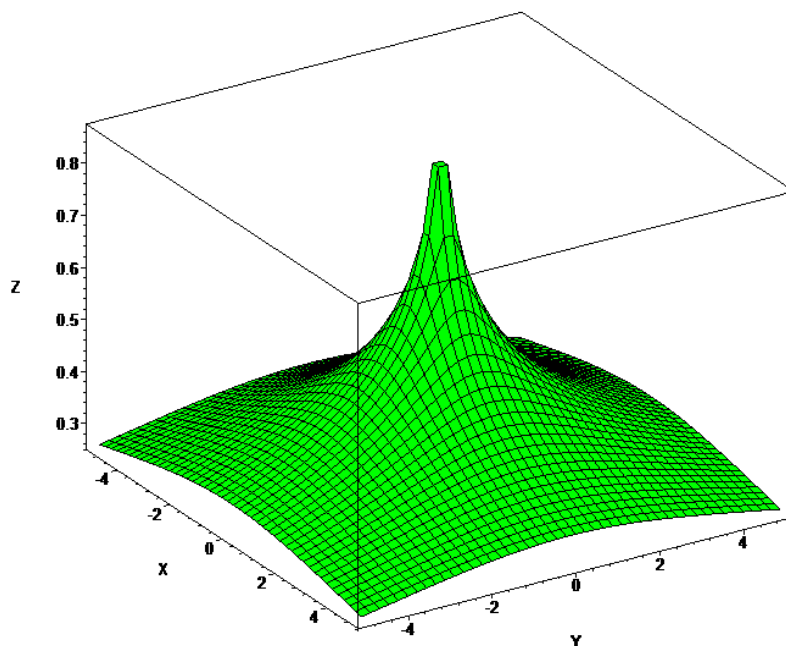


Рис. 1. Визуализация поля температур для изотропной пластины ( $Bi=0.001$ ,  $W_1^* = 1$ )

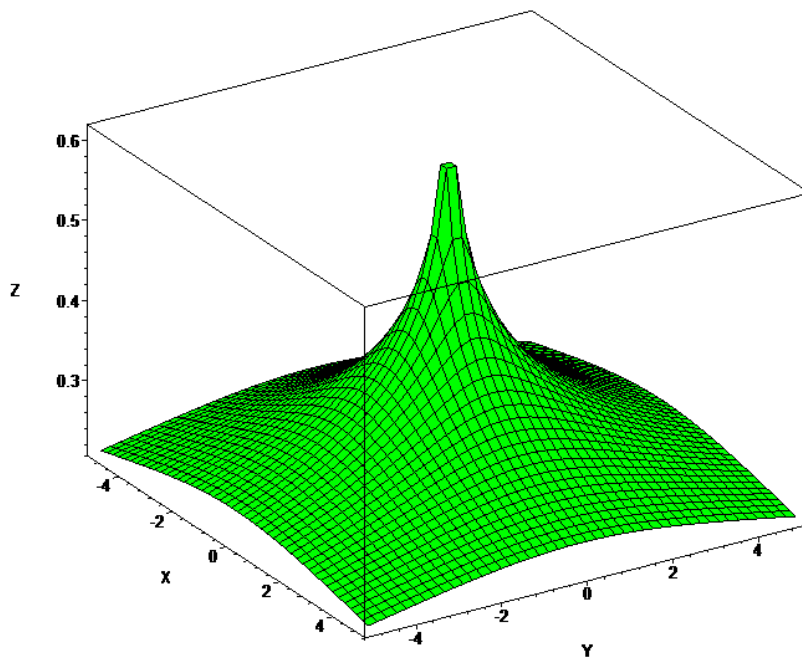


Рис. 2. Визуализация поля температур для ортотропной пластины ( $Bi=0.001$ ,  $W_1^* = 1$ ,  $\lambda_1=2.31$ ,  $\lambda_2=1$ )



### **Выводы**

В работе была рассмотрена задача теплопроводности в тонкой ортотропной пластине, на которую воздействует сосредоточенный источник тепла, и предложен способ её решения путём преобразования этой задачи к двумерной. Для сведения задачи к более простой была выбрана линейная аппроксимация изменения температуры в направлении нормали к плоскости пластины.

Далее поставленная задача решается с помощью двумерного интегрального преобразования Фурье, что позволяет свести исходное уравнение для распределения температуры к более простому, а затем перейти к пространству оригиналов.

Полученное решение в интегральной форме рассчитывается численно по методу Гаусса. Исходя из полученных результатов, можно сделать вывод, что при слабом теплообмене температура в изотропной пластине выше чем в ортотропной. Из распределения температуры видно, что температура в ортотропной пластине в направлении большей теплопроводности спадает быстрее, чем в направлении другой оси.

### **Литература**

1. Беляев Н.М. Методы теории теплопроводности. – Учебное пособие для вузов. В 2-х частях. Ч. 1. / Н.М. Беляев, А.А. Рядно. – М.: Высшая школа, 1982. – 327 с.
2. Гольцев А.С. Фундаментальное решение уравнений теплопроводности ортотропных пластин в общем случае теплообмена / А.С. Гольцев // Современные проблемы механики сплошной среды. Тр. III межд. конф., г. Ростов-на-Дону, 7-9 октября 1997 г. Т. 1. – Ростов-на-Дону: МП «Книга», 1997. – С. 97-101.
3. Подстригач Я.С. Неустановившиеся температурные поля и напряжения в тонких пластинках / Я.С. Подстригач, Ю.М. Коляно. – К.: Наук. думка, 1972. – 308 с.
4. Хижняк В.К. Смешанные задачи теории пластин и оболочек: учеб. пособие // В.К. Хижняк, В.П. Шевченко. – Донецк: ДонГУ, 1980. – 128 с.

УДК 004.942

## МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ПЛОСКИХ КОЛЕБАНИЙ ВИБРАЦИОННОЙ МАШИНЫ С КАРДАНОМ ГУКА В ПРИВОДЕ ТРАНСМИССИИ И ЕЕ РЕАЛИЗАЦИЯ В СРЕДЕ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ MATLAB

Кухта С.С.

Кафедра Компьютерного моделирования и дизайна,  
Донецкий национальный технический университет

E-mail: [kukhtass@mail.ru](mailto:kukhtass@mail.ru)

### **Аннотация:**

*Кухта С.С. Математическая модель плоских колебаний вибрационной машины с карданом Гука в приводе трансмиссии и ее реализация в среде математического моделирования MATLAB. В статье предложено использовать шарнир Гука с несоосным расположением ведущего и ведомого валов в приводе трансмиссии вибромашин для формирования полигармонических вибраций рабочего органа. В этом случае даже при равномерном вращении ведущего вала вращения ведомого вала становятся неравномерными. Это приводит к тому, что возмущающая сила, генерируемая неравномерно вращающимися дебалансными валами, становится негармонической, что и является причиной формирования почти периодических колебаний рабочего органа. В работе рассматривается математическая модель одномассовой вибромашины такого типа с плоскими перемещениями рабочего органа и формируется программное обеспечение в виде приложения MATLAB для ее анализа. Основная задача этого приложения – дать пользователю удобный инструмент исследования, позволяющий проводить определение режимов движения рабочего органа таких вибромашин при различных значениях ее параметров и апробировать различные подходы к выполнению их спектрального анализа.*

### **Annotation:**

*Kukhta S.S. A mathematical model of flat vibrations of a vibration machine with Hooke's cardan drive in a transmission and its implementation in the MATLAB mathematical modeling environment. In article, it was proposed to use the Hooke hinge with non-axial arrangement of the drive and driven shafts in the transmission drive of vibrating machines for the formation of polyharmonic vibrations of the working member. In this case, even with a uniform rotation of the drive shaft, the rotation of the driven shaft becomes uneven. This leads to the fact that the disturbing force generated by non-uniformly rotating unbalanced shafts becomes non-harmonic, which is the reason for the formation of almost periodic oscillations of the working body. The paper discusses a mathematical model of a single-mass vibratory machine of this type with flat movements of the working member, and software is formed as a MATLAB application for its analysis. The main objective of this application is to provide the user with a convenient research tool that allows determining the modes of movement of the working body of such vibratory machines for various values of its parameters and testing various approaches to performing their spectral analysis.*

### **1 Введение**

Для горнодобывающей промышленности характерно активное применение грохотов - вибрационных машин, предназначенных для сортировки (грохочения) сыпучих материалов путём их просеивания через сита (или решёта). В промышленности данные устройства применяются для классификации горных пород, сыпучих строительных материалов, в лабораториях для фракционного анализа сыпучих материалов и др. Также применяются для обезвоживания различных материалов (обогащенных углей, промытых руд).

Распространение получили инерционные грохоты, в которых возмущающие усилия формируются за счет вращения неуравновешенных валов. Принципиальная схема такого грохота в одномассовом исполнении представлена на рисунке 1 [2].

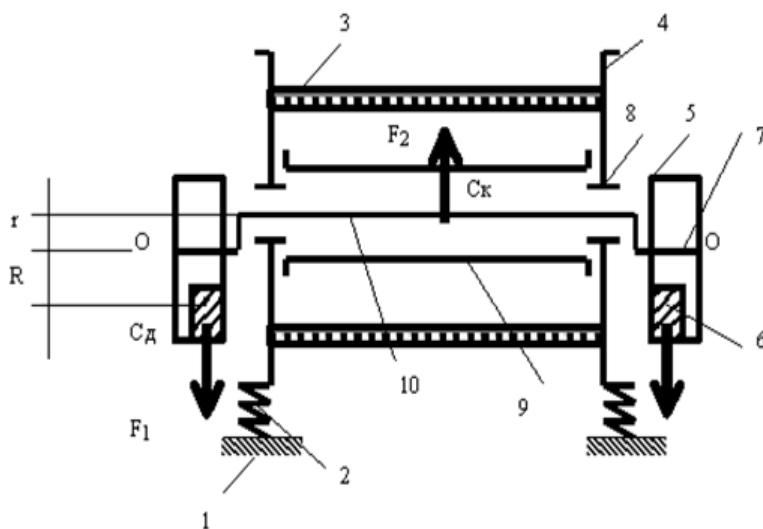


Рисунок 1 – Принципиальная схема инерционного грохота: 1 – опорная рама, 2 – амортизаторы, 3 – сита, 4 – короб, 5 – шкив, 6 – дебаланс, 7 – эксцентриковые концы вала, 8 – подшипники, 9 – труба вибровозбудителя, 10 – вал, Сд, Ск – центры тяжести дебаланса и короба, соответственно, О – геометрический центр шкива

В данном случае предполагается, что вращение передается через кардан Гука - универсальный пространственный шарнирный механизм в виде креста. Принципиальная схема кардана Гука представлена на рисунке 2 [3].

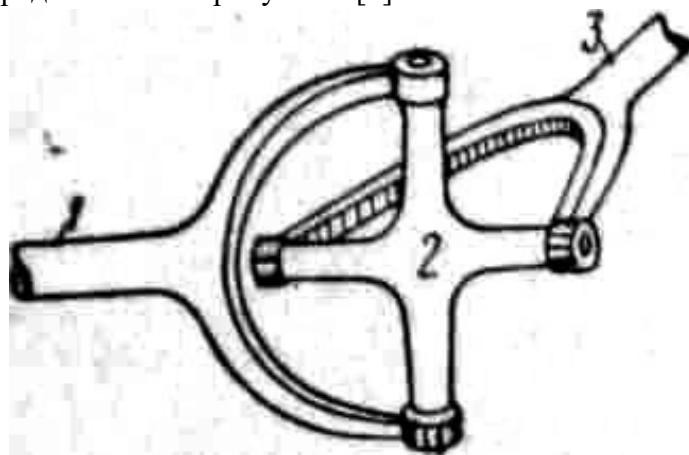


Рисунок 2 - Принципиальная схема кардана Гука

Данный механизм связывает при помощи промежуточного звена 2 два вала 1 и 3, каждый из которых оканчивается вилкой, вращающейся на противоположных концах креста. При передаче вращения между валами 1 и 3 передаточное число получается непостоянным; за время одного оборота оно меняется от  $\cos \alpha$  до  $1/\cos \alpha$ , где  $\alpha$  — угол между осями валов.

## 2 Цель работы

Цель работы заключается в разработке программного продукта, позволяющего проводить моделирование динамики такой вибромашины при различных значениях ее параметров. Предполагается сохранение результатов моделирования в виде изображений и обеспечение возможности последующего обращения к ним для дополнительного анализа.

Помимо этого, в приложении предусматриваются функции сохранения исходных данных, заполнение данных модели из сохраненного файла, редактирование любого параметра.

### 3 Основная часть

К исходным данным модели относятся следующие характеристики: масса колеблющихся частей вибромашины  $m$ , коэффициент сопротивления  $\mu$ , коэффициенты жесткости упругих элементов вдоль осей  $OX$  и  $OZ$   $c_x, c_z$ , масса неуравновешенных частей возбудителя  $m_0$ , их эксцентриситет  $r$ , шаг дифференцирования  $h$ , отношение реальной частоты возбуждений к собственной частоте системы  $\eta$  (принимает значения 0.5, 1, 2, 3), угол излома валов  $\beta$  (принимает значения 0, 15, 30 или 45 градусов). Для указываемых пользователем значений предусмотрен контроль корректности данных, например, для всех вводимых непосредственно с клавиатуры параметров предусмотрена проверка на неотрицательность, а угол излома и отношение частот можно выбрать только из указанного диапазона.

Интерфейс разрабатываемого программного продукта должен быть максимально простым, поскольку программа будет использоваться для ввода исходных данных модели и выполнения основных функций. Внешний вид приложения представлен на рисунке 3.

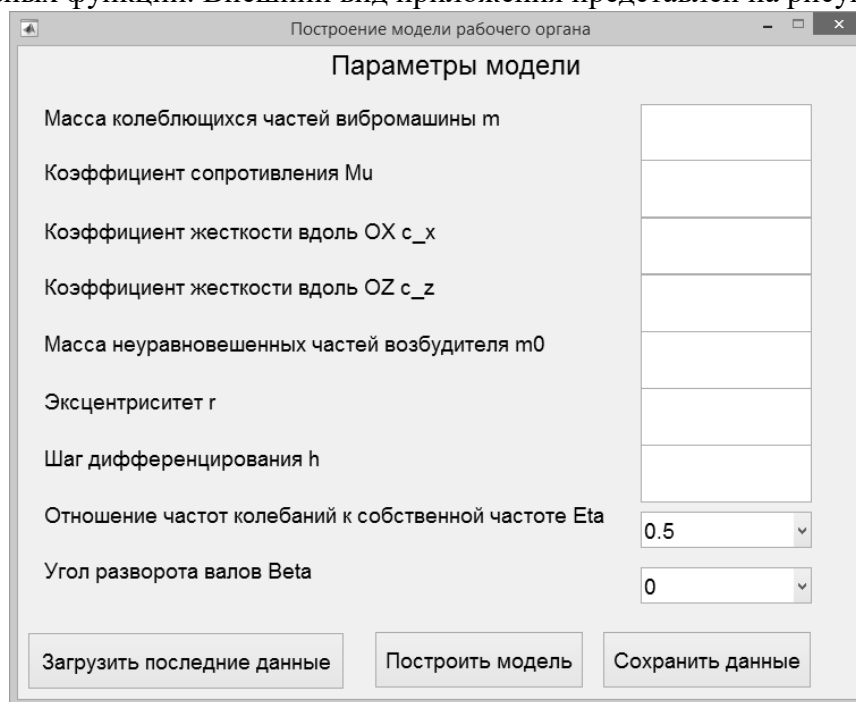


Рисунок 3 – Интерфейс приложения

При составлении уравнений движений принимаются традиционные для вибрационной техники допущения: рабочий орган является абсолютно твердым телом и рассматривается как материальная точка; характеристики упругих связей линейны; диссипация в системе описывается потерями в упругих связях и подчиняется гипотезе вязкого трения; коэффициент сопротивления пропорционален жесткости упругих элементов; внешнее воздействие описывается центробежной силой неуравновешенного ротора; двигатель вибрационной машины является идеальным и вращение его вала происходит с постоянной угловой скоростью. В общей форме уравнения движения имеют вид:

$$\overline{F_{инер}} + \overline{F_{внеш}} + \overline{F_{сопр}} + \overline{F_{упр}} = 0$$

где  $F_{внеш}$  – внешние силы, в данном случае – центробежная сила  $\overline{F_{внеш}} = \overline{F_{центр}} = m * \omega^2 * \vec{r}$ , где  $r = \begin{pmatrix} x \\ z \end{pmatrix}$ ,  $x, y$  – координаты центра масс короба (рисунок 3.2)

$F_{инерц}$  – сила инерции,  $\overline{F_{инерц}} = -m * \ddot{r}$ , где  $m$  – масса,  $\ddot{r}$  – ускорение центра масс короба

$F_{сопр}$  – сила сопротивления,  $\overline{F_{сопр}} = -\mu * \begin{pmatrix} c_x \dot{x} \\ c_z \dot{z} \end{pmatrix}$ , где  $c_x, c_z$  – коэффициенты жесткости опоры вдоль оси X и Z

$F_{упр}$  – сила упругости,  $\overline{F_{упр}} = \begin{pmatrix} c_x x \\ c_z z \end{pmatrix}$ , где  $c_x, c_z$  – коэффициенты жесткости опоры вдоль оси X и Y.

Поскольку рассматриваемые колебания рабочего органа являются плоскими, то его закон движения описывается системой двух уравнений:

$$\begin{cases} m\ddot{x} + b_x \dot{x} + c_x x = m_0 r \dot{\phi}^2 \sin(\phi t) \\ m\ddot{z} + b_z \dot{z} + c_z z = -m_0 r \dot{\phi}^2 \cos(\phi t) \end{cases}$$

где  $b_x, b_z$  – коэффициенты диссипации ( $b = \mu * c$ )

$\dot{\phi}$  – зависимость скорости вращения ведомого вала от ведущего ( $\dot{\phi}(t) = \frac{\omega * \cos \beta}{1 - \sin^2 \beta * \cos^2(\omega t)}$ ) [3]

Для решения системы переобозначим переменные:  $x_1 = x, x_2 = \dot{x}, x_3 = z, x_4 = \dot{z}$ . Решая данную систему, мы вводим дополнительные переменные:  $\dot{x}_1 = x_2, \dot{x}_3 = x_4$ . Таким образом, система из 2 уравнений превращается в систему из 4 уравнений:

$$\begin{cases} \dot{x}_1 = x_2 \\ m\dot{x}_2 + b_x x_2 + c_x x_1 - P(t) = 0 \\ \dot{x}_3 = x_4 \\ m\dot{x}_4 + b_z x_4 + c_z x_3 - P(t) = 0 \end{cases}$$

Начальные условия  $x_0 = [0, 0, 0, 0]$ , промежуток дифференцирования  $t \in [0, 400]$  с шагом интегрирования  $h$  (по умолчанию  $h = 0.01$ ). Для решения системы можно воспользоваться встроенными функциями MATLAB – решение системы методом Рунге-Кутты 4-5 порядков (ode45) или методом Адамса(ode113). Различие результатов решения разными методами незначительна (рисунок 4), однако метод Рунге-Кутта работает быстрее, поэтому в дальнейшем при расчетах используется функция ode45.

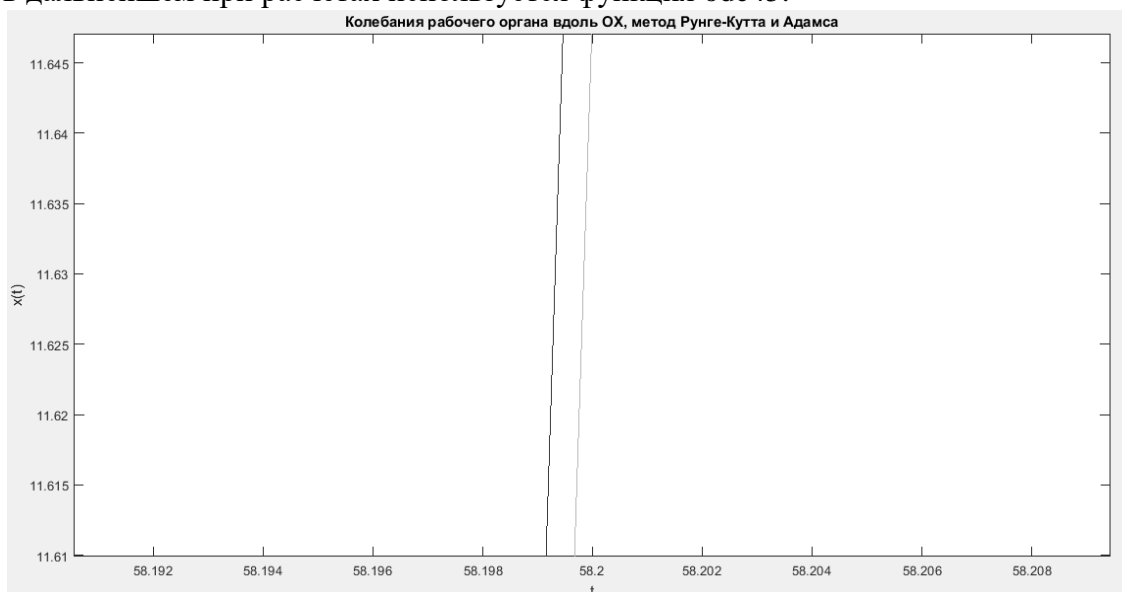


Рисунок 4 – Фрагмент решения системы уравнений методами Рунге-Кутта и Адамса

Выходными данными для данного приложения являются различные графики: график колебания рабочего органа вдоль оси OX, вдоль оси OZ, график движения рабочего органа

вдоль обеих осей, вывод фазовых графиков (графиков зависимостей  $\dot{x}(x)$  и  $\dot{z}(z)$ ). На рисунке 5 представлен график движения рабочего органа вдоль двух осей.

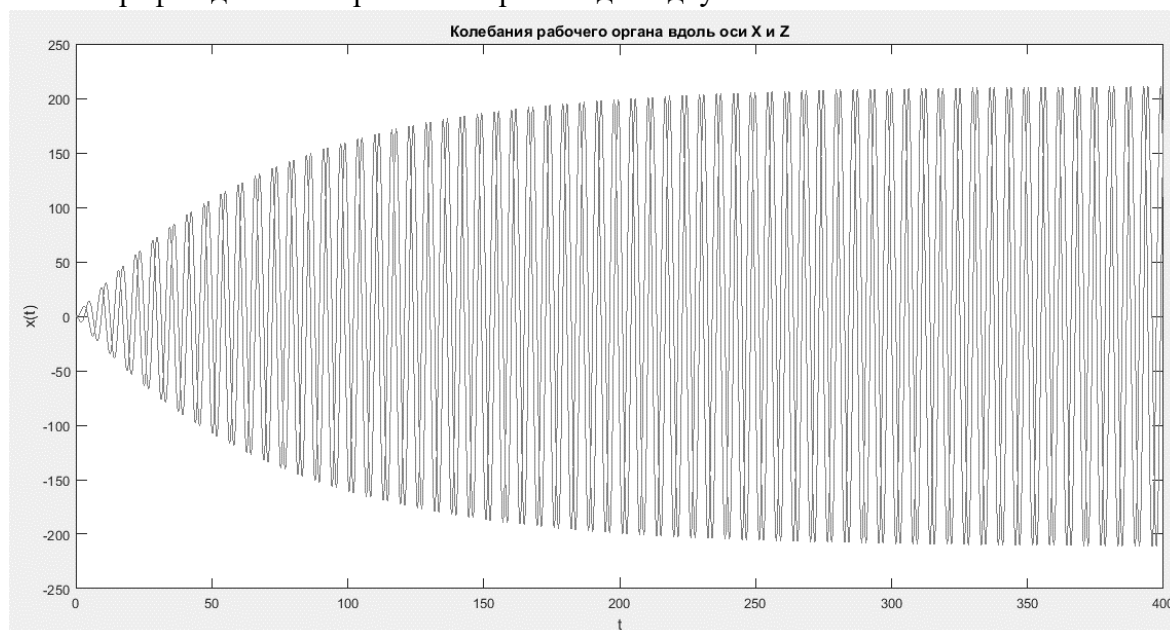


Рисунок 5 – Вывод графика колебаний рабочего органа вдоль оси X и Z

#### 4 Выводы

Данный программный продукт имеет практическую ценность, заключающуюся в том, что появится возможность проводить анализ динамики вибромашины, всего лишь указав ее параметры. Меняя угол излома и режим работы (отношение частоты колебаний к собственной частоте), можно будет определять стационарные режимы движения, проводить их спектральный анализ на различных временных участках, формулировать практические рекомендации. В настоящее время разработка приложения находится на начальной стадии

*Работа выполняется под руководством доцента кафедры КМД ДонНТУ В.Беловодского.*

#### Литература

1. Букин С.Л., Букина А.С., Селиверстов В.В. Динамическая модель одномассовой вибромашины с карданным валом в трансмиссии дебалансного возбудителя колебаний / «Прогрессивные технологии и системы машиностроения: Международ. сб. научных трудов». - Донецк: ДонНТУ. - 2014. Вып. 4 (50). - С.65-73.
2. Конспект лекций по курсу «Подготовительные процессы обогащения», Назимко Е.И. – ДонНТУ, 2008
3. Кожевников С.Н. Теория механизмов и машин: Учеб. пособие для вузов. Изд. 4-е, исп. – М.: Машиностроение, 1973. – 592 с.

УДК 519.63

## ИСПОЛЬЗОВАНИЕ АДАПТИВНОГО КОНТРОЛЯ ВЕЛИЧИНЫ ШАГА ПРИ ЧИСЛЕННОМ ИНТЕГРИРОВАНИИ БЫСТРО МЕНЯЮЩИХСЯ ФУНКЦИЙ ВЛОЖЕННЫМ МЕТОДОМ РУНГЕ-КУТТЫ ПЯТОГО ПОРЯДКА

Маглеванный И.И., Астахова Н.А., Карякина Т.И.

Волгоградский государственный социально-педагогический университет, 27 пр. Ленина,  
Волгоград, 400066

E-mail: [sianko@list.ru](mailto:sianko@list.ru)

### Аннотация:

*Маглеванный И.И., Астахова Н.А., Карякина Т.И. Использование адаптивного контроля величины шага при численном интегрировании быстро меняющихся функций вложенным методом Рунге-Кутты пятого порядка. В данной работе представлен квадратурный алгоритм, основанный на явном вложенном методе Рунге-Кутты с адаптивным контролем величины шага. Использование схемы Дормана-Принса позволяет значительно уменьшить объем вычислений при интегрировании быстро меняющихся функций. Представлен компактный программный код описанного алгоритма на языке С. Численный пример показывает, что предложенный алгоритм является эффективным и надежным.*

### Annotation:

*Maglevanny I.I., Astachova N.A., Karyakina T.I. The use of adaptive step-size control for numerical integration of rapidly varying functions by embedded fifth order Runge-Kutta rule. We present here the quadrature algorithm which is based on the explicit form of fifth-order embedded Runge-Kutta rule with adaptive step-size control. The use of Dormand-Prince scheme helped to reduce considerably the amount of computations while integrating the rapidly varying functions. A compact C-code of described algorithm is presented. The simulation results validate the efficiency and robustness of proposed algorithm.*

### Общая постановка проблемы

Вопросам разработки, исследования, тестирования и применения численных методов приближенного вычисления определенных интегралов посвящено огромное количество литературы, включая обширные монографии, справочники и множество журнальных статей. Однако это не снимает проблему разработки эффективных и экономичных в плане вычислительных затрат методов решения, специализированных для определенных классов задач. Одной из таких проблем является задача приближенного вычисления интегралов от быстро меняющихся функций. Такие функции могут иметь четко выраженные пики или же сильно колебаться от отрицательных значений к положительным [1,2].

### Численная оценка интеграла методом Рунге-Кутта

Основная идея использования методов Рунге-Кутты заключается в сведении задачи вычисления определенного интеграла к задаче Коши для вспомогательной функции  $u(t)$

$$I = \int_a^b f(t)dt \Leftrightarrow \frac{du}{dt} = f(t), u(t=a) = 0, t \in [a,b] \Rightarrow I = u(t=b), \quad (1)$$

т.е. является частным случаем задачи интегрирования системы обыкновенных дифференциальных уравнений (ОДУ) с начальными данными. Задача (1) решается вложенным методом Рунге-Кутты пятого порядка с адаптивным контролем величины шага

[3,4]. Наиболее эффективное управление величиной шага основано на вложенных формулах Рунге-Кутты, первоначально предложенных Мерсоном и приобретших популярность после работ Фелберга [5], обнаружившего метод пятого порядка, требующий вычисления шести значений правой части, который при другой комбинации полученных значений дает метод четвертого порядка. Предлагаемый нами алгоритм работает быстрее, чем стандартные программы для решения систем ОДУ [3,4]. Именно, так как правая часть дифференциального уравнения (2) не содержит величины  $u$ , мы можем исключить один функциональный вызов на каждом шаге интегрирования. Так мы получаем эффективный метод пятого порядка, требующий только пять вычислений правой части на каждом шаге, который описывается следующим алгоритмом.

Начиная с  $t = a$  и для данного шага  $h$  мы накапливаем приращения с соответствующими весами и оцениваем ошибку  $\Delta$  как разность между методами четвертого и пятого порядков

$$u(t+h) = u(t) + h \sum_{i=1}^5 b_i k_i, \quad \Delta = \sum_{i=1}^5 e_i k_i \approx h^5, \quad k_i = f(t + c_i h). \quad (2)$$

Здесь весовые коэффициенты  $b_i$ ,  $c_i$ ,  $e_i$  зависят от реализации вложенной схемы. Мы используем схему Дормана и Принса [3,4]. Используя информацию, приведенную в [4], мы получили необходимые константы, которые представлены в таблице 1.

Таблица 1 – Константы схемы Дормана-Принса.

$i$	1	2	3	4	5
$b_i$	35/384	500/1113	125/192	-2187/6784	11/84
$c_i$	0	0.3	0.8	8/9	1
$e_i$	71/57600	-71/16695	71/1920	-17253/339200	71/4200

Отметим, что, т.к.  $c_1 = 0$  и  $c_5 = 1$ , мы получаем  $k_5 = f(t+h)$ . Так как  $f(t+h)$  должен быть вычислен для того, чтобы начать следующий шаг, поэтому последний этап текущего шага используется в качестве первого этапа на следующем шаге (если этот шаг не был отвергнут из-за слишком большой погрешности). Такая технология получила название «Первый как последний» (First Same as Last [4]). Таким образом, при использовании неотклоненного шага только четыре вычисления подинтегральной функции необходимо для вычислений по формулам (2). Таким образом, по сравнению со стандартными программами для решения систем ОДУ [3,4], предложенный алгоритм работает в среднем в  $6/4=1.5$  раз быстрее.

После вычислений (2) мы знаем, по крайней мере, приближенно, погрешность  $\Delta$ , и следует решить, как удержать ее в необходимых границах. Вычисляем

$$err_{\max} = \frac{|\Delta|}{scale}, \quad scale = atol + rtol \cdot \max(u(t), u(t-h)). \quad (3)$$

Здесь  $atol$  и  $rtol$  являются допусками для абсолютной и относительной погрешностей. Мы используем  $atol = rtol = tol$ , где  $tol$  - оценка погрешности интегрирования, задаваемая пользователем. Такой выбор является наиболее безопасным, однако может быть изменен, при необходимости.



Алгоритм управления длиной шага на основе оценки контрольного члена погрешности. Новый шаг вычисляется по формуле

$$h_{new} = S \cdot (err_{max})^{-0.2}, \quad (4)$$

где  $S = 0.9$  - множитель, который вводится для контроля вследствие того, что наши оценки погрешности являются только приближенными [4].

Теперь нужно решить, является ли используемый шаг  $h$  допустимым. Если  $err_{max} \leq 1$ , мы принимаем шаг  $h$  и продолжаем расчет (3) для новой точки  $t \rightarrow t+h$  и с новым шагом  $h \rightarrow h_{new} \geq h$ , иначе отвергаем  $h$  и повторяем расчет (3) в той же точке  $t$ , но с меньшим шагом  $h \rightarrow h_{new} < h$ . Процесс продолжается то тех пор, пока не будет достигнута точка  $t+h=b$ . При этом не допускается слишком сильное увеличение (не больше чем в десять раз) и уменьшение (не больше чем в пять). Обоснование этого есть, например, в [3,4].

### Программный код

Практическая реализация описанного алгоритма была осуществлена на алгоритмическом языке С. Программный код представлен ниже.

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
static inline double max1(double x,double y){return x<y?y:x;}
static inline double min1(double x,double y){return x<y?x:y;}
static const int MAXSTP=32000;
static const double minscale=0.2,maxscale=10,alpha=0.2,safe=0.9;
static const double // Константы Дормана-Принса
c2=0.3,c3=0.8,c4=8./9.,b1=35./384.,b2=500./1113.,
b3=125./192.,b4=-2187./6784.,b5=11./84.,e1=71./57600.,
e2=-71./16695.,e3=71./1920.,e4=-17253./339200.,e5=71./4200.;
//////////
static double Stepper(double U,double dUdt,double t,double h,
double (*f)(double t),double *Delta,double *Knew)
{double k2,k3,k4,k5,Uout;
k2=f(t+c2*h);k3=f(t+c3*h);k4=f(t+c4*h);k5=f(t+h);*Knew=k5;
Uout=U+h*(b1*dUdt+b2*k2+b3*k3+b4*k4+b5*k5);
*Delta=h*(e1*dUdt+e2*k2+e3*k3+e4*k4+e5*k5);return Uout;
}
////////// Прототип функции, осуществляющей интегрирование
double INTEGRATOR(double (* f)(double t),//интегрируемая функция
double a,double b,//Пределы интегрирования
double h,//Начальный шаг
double TOL//Допустимая погрешность
)
{if(a==b)return 0;int reject=0,nstp=0;
double hmin=1.e-15,U,t,hnext,scale,Kt,errmax,htemp,Delta,Utemp,Knew,
atol=TOL,rtol=TOL;// Может быть изменено при необходимости
t=a;U=0;Kt=f(t);
while(++nstp<=MAXSTP){//Не более MAXSTP шагов.
for(;;){Utemp=Stepper(U,Kt,t,h,f,&Delta,&Knew);//Сделать шаг
scale=atol+rtol*max1(fabs(U),fabs(Utemp));
errmax=fabs(Delta/scale);//Оценить точность
if(errmax<=1.0)break;//Шаг успешен. Начать со следующей точки.
reject=1;//Шаг неудачен. Попробовать меньший шаг
```

```

scale=max1 (safe*pow(errmax, -alpha) , minscale) ; h*=scale;
if (t+h==t) {printf("Stepsize underflow in INTEGRATOR\n"); exit(0); }
} //end for(;;)
t+=h; U=Utemp; Kt=Knew; // Шаг успешен. Найти следующий шаг
if ((t-b) * (b-a) >= 0.0) return U; // Нормальный выход.
if (errmax==0.) scale=maxscale;
else {
scale=safe*pow(errmax, -alpha);
//Гарантировать minscale < hnext/h < maxscale.
if (scale < minscale) scale=minscale; if (scale > maxscale) scale=maxscale;
}
//Не позволить увеличить шаг, если последний шаг был отвергнут
if (reject) hnext=h*min1(scale, 1.); else hnext=h*scale;
if (fabs(hnext) <= hmin) {printf("Step size too small in NTEGRATOR\n"); exit(0); }
h=hnext; reject=0;
if ((t+h-b) * (t+h-a) > 0.0) h=b-t;
} //end while
printf("Too many steps in routine INTEGRATOR\n"); exit(0);
}

```

### Численный пример

Рассмотрим интеграл

$$I = \int_{-1}^2 ([L(0, \Gamma, t) + L(1, \Gamma, t)] + t^2 \cdot 10 \sin \Omega t) dt, \quad (5)$$

где  $L(c, \Gamma, t)$  - функция Лоренца

$$L(c, \Gamma, t) = \frac{1}{\pi} \frac{0.5\Gamma}{(x-c)^2 + (0.5\Gamma)^2}. \quad (6)$$

Здесь параметр  $\Gamma$  - полуширина пика, а высота пика равна  $L_{\max} = 2/(\pi\Gamma)$ .

График типичной подинтегральной функции показан на рис.1.

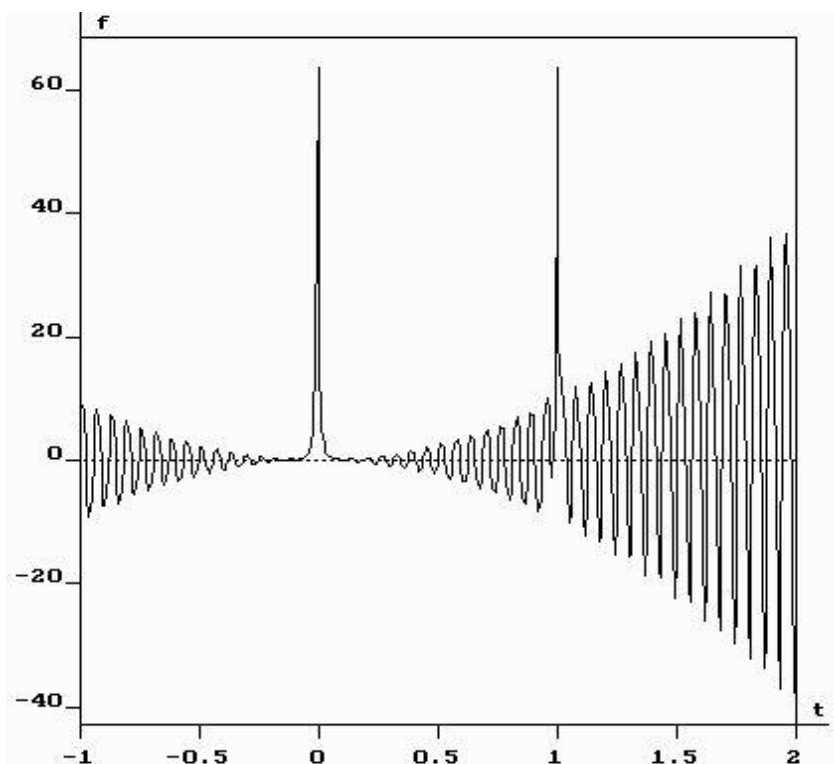


Рис. 1. График подинтегральной функции для интеграла (6) при  $\Gamma = 0.01$ ,  $\Omega = 100$ .

В таблице 2 показаны численные оценки интеграла (5) для различных  $\Gamma$  и  $\Omega$ , которые определяют функции с очень острыми пиками.

Таблица 2 – Численные результаты

$\Omega$	$\Gamma$	$L_{\max}$	$I_{\text{exact}}$	$I_{\text{appr}}$	$N$
100	0.01	63.662	1.884094	1.884094	4186
200	0.001	636.62	2.128524	2.128524	7522
300	0.0001	6366.2	2.13266	2.132661	10562
400	1e-5	63662	2.032005	2.032006	13218
500	1e-6	6366620	1.937502	1.937502	15962

В данной таблице  $I_{\text{exact}}$  - точные значения интеграла,  $I_{\text{appr}}$  - численные значения интеграла,  $N$  - число обращений к процедуре вычисления подинтегральной функции. Вычисления производились с точностью  $tol = 1.e - 7$ .

### Выводы

Один из достаточно сложных, но весьма важных этапов решения любой практической задачи связан с проблемой выбора алгоритма из нескольких известных. Очень часто скорость вычислений и точность решения задачи зависят от управляющих параметров алгоритма, задаваемых пользователем. Однако способ выбора оптимальных значений этих параметров, которые соответствовали бы индивидуальным особенностям решаемой задачи, бывает трудно формализовать или он вообще неизвестен. Поэтому особую ценность представляют алгоритмы, обладающие свойствами адаптивности, то есть умеющие учитывать индивидуальные характеристики конкретной задачи из области определения данного алгоритма. Практическая реализация одного из таких алгоритмов численного интегрирования представлена в данной работе.

Проблема оптимизации (модификации, модернизации) вычислительных методов по-прежнему сохраняет свою актуальность и определяет перспективу дальнейшего развития численного анализа.

### Литература

1. Maglevanny, I. I. Robust sampling-sourced numerical retrieval algorithm for optical energy loss function based on log–log mesh optimization and local monotonicity preserving Steffen spline, [Текст] / I. I. Maglevanny, V. A. Smolar // Nuclear Instruments and Methods in Physics Research B, V. 367 - 2016. — P. 26-36
2. Cash, J. R. A Variable Order Runge-Kutta Method for Initial Value Problems with Rapidly Varying Right-Hand Sides [Текст] / J. R. Cash, A. H. Karp // ACM Transactions on Mathematical Software V. 16 – 1990. — P. 201-222.
3. Hairer, E. Solving Ordinary Differential Equations I. Nonstiff Problems, 2nd ed. [Текст] / E. Hairer, S. P. Nørsett, G. Wanner // New York: Springer, 1993.
4. Press, W. H. Numerical Recipes: The Art of Scientific Computing (3rd ed.) [Текст] / W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery // New York: Cambridge University Press, 2007.
5. Fehlberg, E. Klassische Runge-Kutta-Formeln funfter und siebenter Ordnung mit Schrittweitenkontrolle [Текст] / Fehlberg E. // Computing, V. 4 – 1969. — S. 93-106.

УДК 004.424.32

## ОБЗОР «HARMONY» – ИНСТРУМЕНТА ДЛЯ МОДИФИКАЦИИ C#-ПРИЛОЖЕНИЙ В ПРОЦЕССЕ ИСПОЛНЕНИЯ

**В.С. Вивденко**

Донецкий национальный технический университет, г. Донецк  
кафедра искусственного интеллекта и системного анализа

E-mail: [hectovlad@yandex.ru](mailto:hectovlad@yandex.ru)

### *Аннотация*

**Вивденко В.С. Обзор «Harmony» – инструмента для модификации C#-приложений в процессе исполнения.** В статье предоставлен обзор основных возможностей и способов применения открытого инструмента для модификации C# приложений «Harmony». Описаны цели и задачи разработки игровых модификаций и трудности, сопутствующие процессу их создания и внедрения.

### *Abstract*

**Vivdenko V.S. «Harmony» review – the tool for modifying C# applications during runtime.** The article provides an overview of the main features and methods of using the open tool «Harmony» for modifying C# applications. It describes objectives and tasks of the development of game modifications and the difficulties associated with the process of their creation and implementation.

### **Введение**

Внесение изменений в ход работы программы является популярной задачей программирования. Целью таких изменений могут быть тестирование, логирование, исправление ошибок, доработка функций, внедрение новых возможностей, адаптация программного продукта под новые требования и задачи. Нередки случаи, когда отсутствует возможность внести изменения такого рода в исходный код продукта. В таких ситуациях решением может стать изменение хода программы во время выполнения – после ее запуска, путем редактирования тела программы, загруженного в память устройства.

### **Игровые модификации**

Особое место задачи изменения хода выполнения программ занимают в сфере компьютерных игр. Появление игровых модификаций – «модов» – обычное явление для популярных игр. Их создают энтузиасты-игроки, по своему усмотрению изменяя принципы игрового процесса, внося изменения в ход работы программы. Свои наработки в виде патчей они могут выкладывать в открытый доступ. Другие игроки могут применить их к своим версиям игр, если им понравятся вносимые изменения.

Целью модификации игр могут стать исправление ошибок в оригинальном продукте, доработка некоторых аспектов игрового процесса, изменение сложности игры, внедрение новых механик и ресурсов. Модификации способны изменять игру до полной неузнаваемости, а в последствии даже перерасти в самостоятельный проект, как это случилось с Counter-Strike, DotA, Portal, DayZ, PUBG.

Однако при попытке установить более одного мода для одной версии игры возникают сложности. В таких ситуациях могут обнаруживаться конфликты частей исходного кода либо ресурсов, в случае, когда их задействуют разные модификации. Это может привести к полной неработоспособности программы, либо к частичной потере функциональности одного из модов.

Для решения этих проблем, разработчики игр могут предусматривать поддержку модификаций от сторонних авторов, которая, как правило, включает в себя загрузку и управление сторонними библиотеками и ресурсами. В редких случаях создание такого инструментария могут взять на себя и сами игроки. Однако такой подход, хоть и значительно облегчает создание модификаций, все еще не исключает возможность появления конфликтов между разными модами.

### **Описание «Harmony»**

«Harmony» - является open-source библиотекой, разработанной Andreas Pardeike. Ее исходный код доступен на сайте GitHub.com [1], и распространяется под лицензией MIT. Первый тестовый выпуск состоялся 04.02.2017, и библиотека активно развивается по сей день. Целью создания Harmony было упрощение разработки модификаций для игр, написанных на C#, и предоставление разработчикам инструментов для избегания и решения конфликтов между разными модами. На сегодняшний день Harmony используется авторами модификаций для таких игр как RimWorld, BattleTech, The Long Dark, Oxygen Not Included, Subnautica, 7 Days To Die, Cities: Skylines, Kerbal Space Program, Besiege, Sheltered, Stardew Valley, Staxel, Total Miner, Ravenfield, The Ultimate Nerd Game, Unturned.

Принцип работы Harmony полагается на особенности языка C#, платформы .NET и JIT-компиляции. Они позволяют работать с исходным кодом приложения, получая информацию о его структуре через пространство имен Reflection, и работать с промежуточным кодом (IL-code) [2].

Для использования Harmony необходимо, чтобы модифицируемое приложение поддерживало загрузку сторонних библиотек как модулей. Загруженная таким образом Harmony способна задать альтернативное исполнение для любого из методов приложения. Среди особенностей Harmony можно выделить следующие:

- возможность вносить изменения, оставляя изначальный метод нетронутым;
- исполнение внедренного кода до или после изначального метода;
- точечная модификация изначального метода на уровне промежуточного кода;
- к одному методу может быть применено несколько модификаций, которые не будут конфликтовать между собой.

### **Модифицирование с использованием «Harmony»**

Целью для модификации может быть выбран почти любой метод, который содержится в оригинальном приложении. Им может быть метод класса, конструктор, деструктор, геттер или сеттер, и даже анонимный метод. Для задания целевого метода необходимо лишь точно указать его сигнатуру, которая включает в себя класс-родитель, название метода и набор типов параметров (для перегруженных методов). Исключением являются лишь методы, которые маркируются компилятором как “inline” – встроенные. Это очень малые по объему методы, к вызову которых нельзя применить модификацию, поскольку сам их вызов опускается в пользу ускорения выполнения программы. Для определения точной сигнатуры метода и изучения логики его исполнения могут быть использованы программы-декомпиляторы.

Модификация изначального метода в Harmony может осуществляться с помощью одного или нескольких способов [3]. Первый из них – префиксный. Сторонний метод, обозначенный как префикс для изначального метода, будет вызываться каждый раз перед обращением программы к изначальному методу. Префиксный метод может возвращать логическое значение, если оно окажется ложным – то выполнение оригинального метода на этом шаге будет отменено.

Следующий способ – постфиксный. Он определяет метод, который будет вызван сразу после завершения выполнения оригинального метода. Особенностью использования

постфикса является то, что в качестве параметра он может получить результат выполнения оригинального метода и даже изменить его. Таким образом этот способ позволяет редактировать значения, возвращаемые функциями.

Еще один способ модификации называется «транспайлер». Он позволяет вносить корректировки в строго заданные сегменты оригинального метода. В качестве параметра такой метод получает массив значений, который хранит каждую команду оригинального метода и ее аргумент в порядке их выполнения. Эти команды и аргументы могут быть прочитаны и заменены на альтернативные. Для успешного применения такого способа необходимо знать принципы работы промежуточного (IL) кода.

Каждый из методов, используемых в качестве модификации, может принимать в качестве параметров ряд значений, которые позволяют отследить текущее состояние программы, а сами значения тоже могут быть изменены. Среди них находятся информация об объекте, который хранит вызываемый метод, его поля, ссылка на изначальный метод, а также каждый из параметров, который был передан в изначальный метод в момент его вызова. Кроме этого, префиксный и постфиксный метод способны передавать информацию между собой, используя для этого специальную переменную «\_\_state». Комплексное использование этих способов модификации методов позволяет применять всевозможные исправления, которые могут потребоваться при создании мода.

Harmony предоставляет ряд инструментов для устранения возможных конфликтов между модификациями от разных авторов. Среди них есть получение информации об уже примененных к методу модификациях и управление очередностью исполнения модифицированных методов. Получив информацию о примененных модификациях, автор может скорректировать поведение собственного метода, чтобы воспрепятствовать их некорректному взаимодействию.

Очередность исполнения модификаций является важным элементом корректной работы. Поскольку модификации способны менять возвращаемый результат, передаваемые значения и даже отменять выполнение изначального метода, в зависимости от очередности их исполнения итоговый результат может различаться. Для контроля над этим авторы модификаций могут задать им приоритет исполнения, чем он будет выше – тем раньше выполнится метод.

### **Вывод**

«Harmony» является удобным и гибким инструментом для создания игровых модификаций на языке C#. Набор его возможностей предоставляет разработчикам широкий доступ для внесения изменений в игровые приложения, Кроме этого, он позволяет корректировать взаимодействие модификаций от разных авторов и предотвращает их конфликты. «Harmony» может быть использован как основной инструмент для разработки модификации для C# приложений.

## **Литература**

1. Github pardeike/Harmony – A library for patching, replacing and decorating .NET and Mono methods during runtime / Интернет-ресурс. - Режим доступа : [www/ URL: https://github.com/pardeike/Harmony](http://www.github.com/pardeike/Harmony). - Загл. с экрана.
2. Руководство по языку C# | Microsoft docs / Интернет-ресурс. - Режим доступа : [www/ URL: https://docs.microsoft.com/ru-ru/dotnet/csharp/](http://www.docs.microsoft.com/ru-ru/dotnet/csharp/). - Загл. с экрана.
3. Github pardeike/Harmony Wiki / Интернет-ресурс. - Режим доступа : [www/ URL: https://github.com/pardeike/Harmony/wiki](http://www.github.com/pardeike/Harmony/wiki). - Загл. с экрана.

УДК 004.912

## РАЗРАБОТКА АВТОМАТИЧЕСКОЙ СИСТЕМЫ ТРАНСФОРМАЦИИ АНГЛИЙСКИХ ВСТАВОК В РУССКИХ ТЕКСТАХ С ПРИМЕНЕНИЕМ ГЛУБОКОГО ОБУЧЕНИЯ

Пикалёв Я.С.<sup>1</sup>, Ермоленко Т.В.<sup>2</sup>

Государственное учреждение

«Институт проблем искусственного интеллекта»<sup>1</sup>, Донецкий национальный  
университет<sup>2</sup>

E-mail: [pikaliov@gmail.com](mailto:pikaliov@gmail.com), [naturewild71@gmail.com](mailto:naturewild71@gmail.com)

### *Аннотация*

*Пикалёв Я.С., Ермоленко Т.В. Разработка автоматической системы трансформации английских вставок в русских текстах с применением глубокого обучения. Изложен декларативно-процедурный подход, использующий как словарь, так и правила англо-русской практической транскрипции, для трансформации английских вставок, встречающихся в русских текстах. Подготовлен словарь англо-русской практической транскрипции с использованием механизма конечных автоматов. Обучена нейронная сеть для классификации языка текста с применением многослойных свёрточных нейронных сетей. Обучена нейронная сеть для трансформации слов на латинице, не найденных в словаре с использованием нейросетевой архитектуры типа энкодер-декодер.*

*Ключевые слова:* обработка естественного языка, классификация текста, практическая транскрипция.

### *Abstract*

*Pikaliov Y.S., Ermolenko T.V. Development of an automatic system for the transformation of English inserts in Russian texts using deep learning. A declarative-procedural approach is described, it is used both the dictionary and the rules of the English-Russian practical transcription, for the transformation of English inserts found in Russian texts. An English-Russian practical transcription dictionary has been prepared using the state machine mechanism. Trained neural network for classifying text language with multi-layer convolutional neural networks. Neural network for transformation of the dictionary in the Latin alphabet, not found in words using a neural network architecture such as an encoder-decoder.*

*Keywords:* natural language processing, text classification, practical transcription.

### **Введение**

На сегодняшний день в любом тексте на русском языке (статья, книга, новостная лента и т.п.) можно встретить большое количество вставок на латинице, чаще всего – на английском языке, которые, в основном, представлены названиями компаний и организаций, масс-медиа и т.п. («Apple», «Manchester United»). Реже встречаются цитаты или фразы на английском языке («to be or not to be»). Данные вставки осложняют процесс сбора данных для задач, связанных с обработкой естественного языка (например, классификация текста), а также с задачами распознавания и синтеза речи (сбор данных для формирования языковой модели; текстовая разметка аудиоданных). Т. е. для того, чтобы подобная система могла «работать» со словами, написанными на латинице, их необходимо трансформировать в ту же фонетическую систему, что и для русских слов. Для решения этой проблемы систему синтеза/расознавания речи, как правило, дополняют отдельным модулем для английского языка, что влечёт за собой дополнительную комплексность. Например, для системы

распознавания речи необходим набор обучающих данных для дополнительного языка со своей транскрипцией и языковой моделью, а также модуль для классификации языка.

Проблеме трансформации английских слов в кириллицу посвящены работы, в которых упор делается на использование фонетических правил [1-2] или на классическое машинное обучение [3-4]. Стоит выделить тот факт, что данные работы, в основном, направлены на трансформацию имён собственных, написанных на английском языке.

Таким образом, данная работа направлена на создание гибкой автоматической системы формирования практической транскрипции из слов, написанных на латинице, с использованием лингвистических знаний совместно с глубоким обучением, без необходимости в дополнительном языковом модуле.

### **Постановка задачи**

Английские слова, написанные в виде латиницы, должны быть преобразованы в формат кириллицы, используя практическую транслитерацию. В отличие от обычной транслитерации, практическая основывается на произношении слов, что позволяет распознавать английские звуки, используя единую лингвистическую и акустическую базу. После того как английские слова будут преобразованы в последовательность символов кириллицы, на всех этапах их дальнейшей обработки они ничем не будут отличаться от русских. Данная методика может быть использована для любой системы синтеза или распознавания речи. Одним из преимуществ предлагаемого способа обработки слов на латинице является его оптимальность: вместо написания дополнительного языкового модуля используется система англо-русской практической транскрипции для перевода английских слов на кириллицу. Кроме того, предлагаемый метод универсален, то есть может быть использован в любой русскоязычной системе синтеза/распознавания речи: после того, как вставка на латинице переведена в кириллическую графическую систему, она может обрабатываться на всех дальнейших этапах синтеза речи по тем же правилам, что и обычные (нормализованные) русские слова на кириллице.

Разрабатываемый подход является декларативно-процедурным, он использует как словарь, так и правила англо-русской практической транскрипции, изложенные в [6-9]. Для уменьшения комплексности используется нейросетевой (НС) подход и механизм конечных автоматов [10]. В качестве основных нейросетевых архитектур используются нейросетевая архитектура типа энкодер-декодер – Transformer [11]; а также архитектура многослойных свёрточных нейронных сетей (Convolutional Neural Network, CNN) [12].

Предложенный метод получения практической транскрипции использует две НС-модели.

1) НС для определения языка текста, используя CNN (чтобы отсеять тексты на иноязычном тексте); В качестве обучающего материала были использованы текстовые расшифровки для видеовыступлений из проекта Ted-Talks [13] для языков, содержащих символы кириллицы (азербайджанский, белорусский, болгарский, киргизский, казахский, русский, сербский, таджикский, македонский, украинский). А также для наиболее распространённых языков, содержащих символы латиницы (английский, немецкий, испанский, французский). Для тестирования использовалось по 100 случайно выбранных предложений для каждого языка. Для обучения использовалось от 2000 до 10000 предложений для каждого языка. Стоит отметить, что точность классификации английского языка составляет 95.7 %, русского – 98.3 %. На рис.1 отображены графики точности (accuracy) и потерь (loss) для данной НС-модели.

2) НС для трансформации английских слов формат кириллицы, используя архитектуру Transformer. Подготовлен набор обучающих данных, собранный из информации, извлечённой из работ [6-9], также была использована часть слов из CMU-Dict [14]. В общей сложности количество элементов обучающей выборки составляет около 50 тыс. пар слов. Для оценки результатов использовалась метрика: WER – отношение



количества неверно трансформированных слов к общему количеству слов (Word Error Rate) и PER – отношение количества неверно трансформированных символов к общему количеству символов (Phoneme Error Rate). На рис.2 отображены графики PER и WER для данной НС-модели.

Стоит отметить, что при сборе данных для обучения составлялся словарь, использующий направленные ациклические графы слов (directed acyclic word graphs, DAWG) [15], основанные на конечных состояниях автоматов. Данный словарь далее в тексте обозначается как dawg-словарь, состоящий из пар слов («window» – «виндоу»; «tooth» – «тус»).

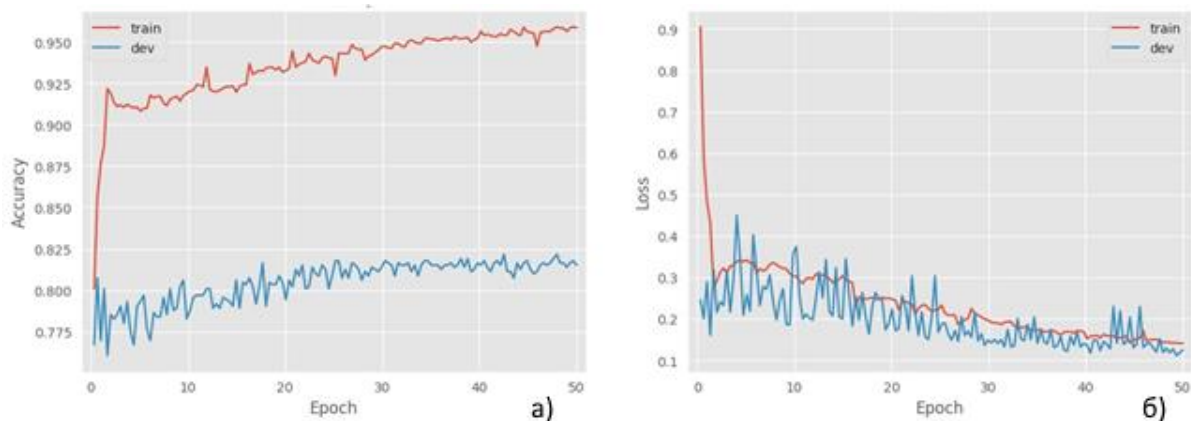


Рис. 1. Графики точности (а) и потерь (б) для НС-модели классификации языка

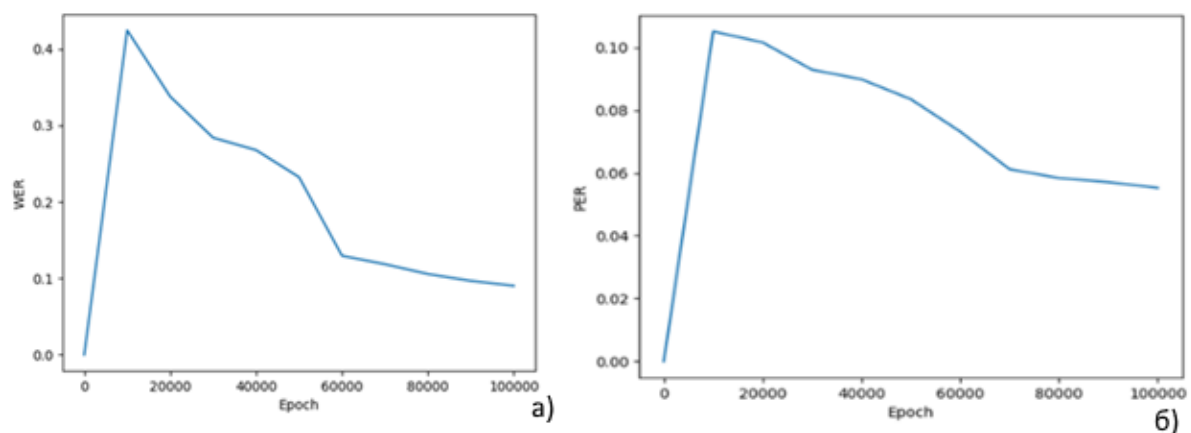


Рис. 2. Графики WER и PER для НС-модели трансформации английских вставок в слова на кириллице

#### Описание общего алгоритма

Алгоритм построения практической англо-русской транскрипции, схема которого изображена на рис.3, состоит в следующем.

- 1) Считывается предварительно нормализованный документ (отсутствуют цифробуквенные комплексы, сокращения, не контекстные токены и т.п.).
- 2) Документ разделяется на предложения.
- 3) Используется модель для определения языка, определяем язык текста для каждого предложения, если модель детектирует не русский язык, то берётся следующее предложение.
- 4) Предложение разделяется на токены.
- 5) Используя набор символов для латиницы, кириллицы и знаков препинания, определяется принадлежность токена (в случае, если токен не принадлежит никакой группе символов – токен определяется как „<unk>“). В случае, если токен состоит из латинских

символов переходим к следующему шагу. В случае, если токен состоит из символов кириллицы – переходим к шагу 8.

б) Вхождение токена ищется в dawg-словаре и извлекается его пара. В случае, если вхождение токена отсутствует в словаре -переходим к следующему шагу.

7) Используется НС для трансформации токена.

8) Получаем изменённое слово.

9) Из изменённых слов составляется изменённое предложение.

10) Из изменённых предложений составляется изменённый документ.

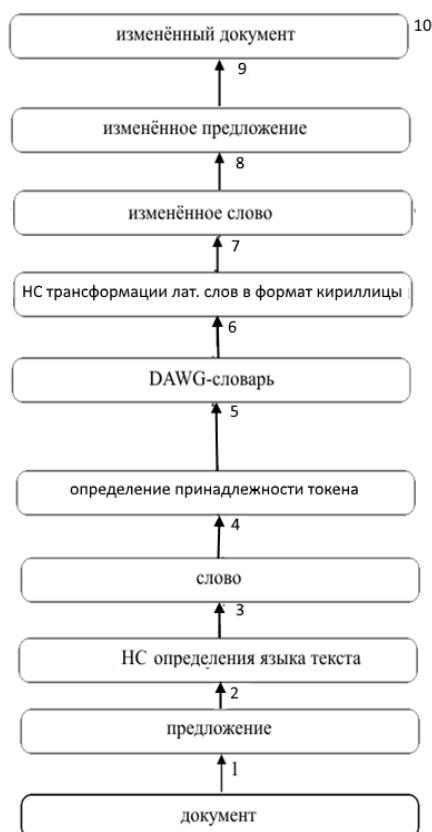


Рис. 3. Общая схема метода трансформации английских вставок

### Вывод

Предложенный подход для создания автоматической системы формирования практической транскрипции слов английского языка позволяет получать англо-русскую практическую транскрипцию с точностью более 90% для слов и более 95% для символов. Такое высокое качество обеспечивает учет орфоэпических норм русского языка не только на основе фонетических правил, но и с использованием глубокого обучения.

Предложенная технология получения практической транскрипции слов английского языка может быть использована в системах синтеза/распознавания русской речи с целью адаптации англоязычных слов на этапе формирования «нормализованных» слов на кириллице для их дальнейшей обработки системой по правилам, применяемым для слов русского языка. Универсальность данного метода заключается в том, что на выходе блока нормализации формируются «нормализованные» слова на кириллице, которые далее обрабатываются по тем же правилам, что и обычные слова русского языка. Благодаря этому одни и те же правила практической транскрипции можно включить в любую русскоязычную систему синтеза/распознавания речи, независимо от фонетической транскрипции, которая в ней используется.

### Литература

1. Бондаренко А. В. Формальный метод транскрипции иностранных имен собственных на русский язык [Текст] / В. А. Бондаренко, Ю. В. Визильтер, В. И. Горемычкин, Э. С. Клышинский // Программные продукты и системы. 2010. №1. URL: <https://cyberleninka.ru/article/n/formalnyy-metod-transkripsii-inostrannyh-imen-sobstvennyh-na-russkiy-yazyk>.
2. Черепанова О. Д. Озвучивание англоязычных употреблений в системе русскоязычного синтеза речи «ТЕКСТ-РЕЧЬ» с помощью практической транскрипции [Текст] / О. Д. Черепанова // Проблемы компьютерной лингвистики и типологии: сборник. – 2016. – С. 226.
3. Rabiner L.R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition [Text] / L.R. Rabiner // Proceedings of the IEEE. 1989. № 77, pp. 257-286.
4. Hermjakob U. Name translation in statistical machine translation-learning when to transliterate [Text] / U. Hermjakob, K. Knight, III H. Daumé // Proceedings of ACL-08: HLT. – 2008. – С. 389-397.
5. Суперанская. А.В. Теоретические основы практической транскрипции. [Текст] / А.В. Суперанская // М.: Наука, 1978, 283 с.
6. Гитляревский Р. С. Иностранные имена и названия в русском тексте: справочник, 3-е изд. [Текст] / Р. С. Гитляревский, Б.А. Старостин // М.: Высш. шк., 1985, 304 с.
7. Ермолович Д. И. Имена собственные на стыке языков и культур [Текст] / Д. И. Ермолович // М.: Р. Валент. – 2001. – Т. 200. – С. 3.
8. Рыбакин А.И. Словарь английских фамилий [Текст] / А.И. Рыбакин // М.: Астрель: АСТ, 2000. – 576 с.
9. Лидин Р. А. Иностранные фамилии и личные имена: Практика транскрипции на русский язык: Справочник. [Текст] / Р. А. Лидин // М., ООО «Издательство Толмач», 2006. – 480 с.
10. Mohri M., Nederhof M. J. Systems and methods for generating weighted finite-state automata representing grammars : пат. 7181386 США. – 2007.
11. Vaswani A. Tensor2tensor for neural machine translation [Text] / A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. Gomez // arXiv preprint arXiv:1803.07416. – 2018. URL: <https://arxiv.org/pdf/1803.07416.pdf>.
12. Kim Y. Convolutional neural networks for sentence classification [Text] / Y. Kim // arXiv preprint arXiv:1408.5882. – 2014. URL: <https://arxiv.org/pdf/1408.5882.pdf> (дата обращения: 19.02.2019).
13. TED Talks [Electronic resource] / Интернет-ресурс. - Режим доступа : [www/](http://www.ted.com/talks) URL: <https://www.ted.com/talks>. - Загл. с экрана.
14. Carnegie Mellon University [Electronic resource] / Интернет-ресурс. - Режим доступа : [www/](http://www.speech.cs.cmu.edu) URL: <http://www.speech.cs.cmu.edu> . - Загл. с экрана.
15. Yata S. Fast string matching with space-efficient word graphs [Text] / S. Yata, K. Morita, M. Fuketa, J. Aoe // 2008 International Conference on Innovations in Information Technology. – IEEE, 2008. – С. 79-83.

УДК 004.622

## МОДЕЛИРОВАНИЕ СЛУЧАЙНОЙ ВЕЛИЧИНЫ ПО ЭКСПОНЕНЦИАЛЬНОМУ РАСПРЕДЕЛЕНИЮ В MATLAB И ПРОВЕРКА ПО КРИТЕРИЮ ХИ-КВАДРАТ И КРИТЕРИЮ КОЛМОГОРОВА

Дубовик С.Ю., Бочаров Д.М.

Донецкий национальный технический университет  
кафедра искусственного интеллекта и системного анализа

E-mail: [seregadubovik@rambler.ru](mailto:seregadubovik@rambler.ru)

### *Аннотация:*

*Дубовик С.Ю., Бочаров Д.М. Моделирование случайной величины по экспоненциальному распределению в MATLAB и проверка по критерию хи-квадрат и критерию Колмогорова. При помощи программного пакета Matlab моделируются случайные величины по экспоненциальному закону распределения. Полученные результаты отображаются в виде гистограммы и проверяются по критериям хи-квадрат и Колмогорова.*

### *Annotation:*

*Dubovik S.Y., Bocharov D.M. Simulation of a random variable by exponential distribution in MATLAB and verification by  $X^2$  criterion and Kolmogorov criterion. With the help of the Matlab software package, random variables are modeled according to the exponential distribution law. The results are displayed as a histogram and checked by  $X^2$  and Kolmogorov criteria.*

### **Общая постановка проблемы**

Случайные процессы – современный и актуальный подход к изучению вероятностных наук. Параметры, которые входят в состав математической модели, часто имеют случайный характер и подчиняются различным законам распределения, таким как нормальное, равномерное, биномиальное, экспоненциальное или распределение Пуассона и еще много других.

В настоящее время случайные величины с заданным законом распределения находят применение в науке, технике и в любой другой области человеческой деятельности.

В компьютерных науках случайность нужна при тестировании программ, в разработке искусственного интеллекта, в играх, для сравнения алгоритмов.

Если генератор выдает числа, смещенные в какую-то часть интервала (одни числа выпадают чаще других), то результат решения задачи, решаемой статистическим методом, может оказаться неверным. Поэтому проблема использования хорошего генератора действительно случайных и действительно равномерно распределенных чисел стоит очень остро.

### **Исследование закона экспоненциального распределения**

Наибольшую популярность и широкое применение в теории и практике надежности нашло экспоненциальное распределение. Экспоненциальное распределение описывает интервалы времени между независимыми событиями, происходящими со средней интенсивностью.

Количество наступлений такого события за некоторый отрезок времени описывается дискретным распределением Пуассона. Экспоненциальное распределение вместе с распределением Пуассона составляют математическую основу теории надежности.

Кроме теории надежности, экспоненциальное распределение применяется в описании социальных явлений, в экономике, в теории массового обслуживания, в транспортной логистике – везде, где необходимо моделировать поток событий.

Случайная величина  $\varepsilon$  имеет экспоненциальное (показательное) распределение, если.

$$P(\varepsilon < x) = \begin{cases} 0 & \text{при } x \leq 0 \\ 1 - e^{-\lambda x} & \text{при } x > 0 \end{cases}. \quad (1)$$

Пусть  $\varepsilon$  – время ожидания события, тогда из формулы (1) следует, что вероятность того, что это событие наступит раньше  $x$ , равна  $1 - e^{-\lambda x}$ . Этот удобный формализм позволяет описывать моменты возникновения случайных событий.

Параметр  $\lambda$  оценивается на основе реальных данных. Плотность экспоненциального распределения имеет вид

$$P(x) = \begin{cases} 0 & \text{при } x \leq 0 \\ \lambda e^{-\lambda x} & \text{при } x > 0 \end{cases}. \quad (2)$$

Моделируемую выборку необходимо проверить согласно определенным критериям.

Один из самых известных статистических критериев – это «хи-квадрат» ( $\chi^2$ -критерий). Для данного случая проверка по критерию  $\chi^2$  позволит узнать, насколько полученная выборка случайных величин близка к эталону, то есть удовлетворяет ли она вышеуказанным критериям. Частотная диаграмма эталонной выборки случайных величин представлена на рисунке 1. Так как закон распределения эталонной выборки равномерный, то вероятность  $p_i$  (теоретическая) попадания чисел в  $i$ -ый интервал (где  $i=1..k$ ) равна  $p_i = 1/k$ . И, таким образом, в каждый из  $k$  интервалов попадет ровно по  $N \cdot p_i$  чисел ( $N$  – общее количество сгенерированных чисел).

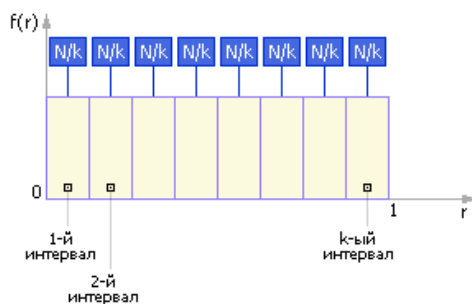


Рис. 1. Частотная диаграмма эталонного ГСЧ

Чтобы проверить ту или иную гипотезу ставят серию специальных опытов и полученные фактические данные сопоставляют с теоретически ожидаемыми согласно данной гипотезе. Если есть совпадение, то это может быть достаточным основанием для принятия гипотезы. Если же опытные данные плохо согласуются с теоретически ожидаемыми, возникает большое сомнение в правильности предложенной гипотезы.

Степень соответствия фактических данных ожидаемым (гипотетическим) измеряется критерием соответствия хи-квадрат:

$$\chi^2 = \sum_{i=1}^k \frac{(Z_{i-набл} - Z_{i-ожд})^2}{Z_{i-ожд}}. \quad (3)$$

Также для проверки опытных данных часто используется критерий Колмогорова, который позволяет с достаточно большой достоверностью проверить, принадлежат ли

статистические данные распределений вероятностей безотказной работы изделия к предполагаемому типу семейств законов распределения.

Достоинством критерия Колмогорова является то, что его можно использовать для малых  $n$  (порядка единиц и десятков). При проверке вычисляется опытное значение эмпирической функции последовательно во всех интервалах по формуле:

$$F_j = F_n(x_j) = \sum_{i=1}^j m_i/n. \quad (4)$$

### Моделирование случайных величин в MATLAB по экспоненциальному закону

Для моделирования можно воспользоваться формулой 2 и записать ее в следующем виде программного кода на языке MATLAB.

```
w(x) = L*exp(-L*x) % это плотность  
F(x) = 1-exp(-L*x) % это интегральная функция  
x = -log(1-F(x))/L; % это обратная функция
```

На диаграмме (рис. 2) отображена выборка по теоритическому и экспериментальному распределению, которая будет проверена по критерию хи-квадрат и критерию Колмогорова.

Ниже приведен код реализации.

```
clear, clc  
close all  
N = 10000; % объем выборки  
L = 2; % параметр распределения  
X = random('exp',1/L,1,N); % выборка случайных чисел  
[n x] = hist(X,30);  
yn = n/(N*(x(2)-x(1))); % считаем плотность вероятности по гистограмме  
y = pdf('exp',x,1/L); % теоретическая плотность вероятности  
bar(x,yn,1,'c'), hold on  
plot(x,y,'-r','linew',2)  
title(['Экспоненциальное распределение с \lambda=',num2str(L)])  
legend('эксперимент','теория')  
k = kstest2(yn,y); % проверяем по критерию Колмогорова  
if k==0  
    disp('Выборка соответствует теории по критерию Колмогорова')  
else  
    disp('Выборка не соответствует теории по критерию Колмогорова')  
end  
CDF = @(z) cdf('exp',z,1/L); % теоретическая интегральная функция  
h = chi2gof(X,'cdf',CDF); % проверяем по критерию хи квадрат  
if h==0  
    disp('Выборка соответствует теории по критерию хи квадрат')  
else  
    disp('Выборка не соответствует теории по критерию хи квадрат')  
end
```

### Проверка результатов

Проверка гипотез будет производиться в среде автоматизации технических расчетов MATLAB, т.к. данная система предоставляет удобный и гибкий функционал для решения задач подобного характера. Для проверки будет использован критерий согласия Пирсона («хи-квадрат») и критерий Колмогорова. Для этого используется функция chi2gof и kstest2 системы MATLAB соответственно, которые и дадут окончательный ответ. Функция chi2gof

и `kstest2` возвращают переменную `k` и `h`, которые принимают два значения 0 или 1. В случае 0 подтверждается гипотеза, иначе – не подтверждается.

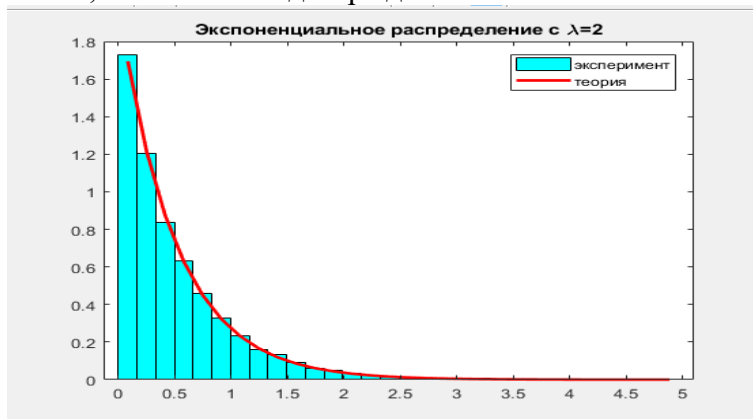


Рис. 2. Результат построения гистограммы

В нашем случае гипотеза на уровне значимости 5% подтвердилась (функции `chi2gof` и `kstest2` вернули результат 0). После выполнения данного кода, в окне выполнения (рис. 3) отображается результат работы команды `disp` по проверке сгенерированной выборки на соответствие заданным критериям.

```
22 - end
Command Window
New to MATLAB? See resources for Getting Started.
Выборка соответствует теории по критерию Колмогорова
Выборка соответствует теории по критерию хи квадрат
fx >>
```

Рис. 3. Результаты проверки выборки

### Выводы

На основании результатов выполнения программного кода, можно сделать вывод об успешной проверке моделируемой величины (т.к. функции `chi2gof` и `kstest2` вернули значение переменных `k` и `h` равных нулю). Таким образом, полученная экспериментальная выборка соответствует теории по критерию «хи-квадрат» и критерию Колмогорова.

### Литература

1. Венцель, Е.С. Теория вероятностей / Е.С. Венцель. – М. : Наука, 1969. – С. 170.
2. Калинина, В.Н. Математическая статистика / В.Н. Калинина, В.Ф. Панкин. – М. : Высшая школа, 1998. – С. 312.
3. Венцель, Е.С. Теория вероятностей / Е.С. Венцель, Л.А. Овчаров. – М. : Наука, 1973. – С. 322.
4. Венцель, Е.С. Теория вероятностей и ее инженерные приложения / Е.С. Венцель, Л.А. Овчаров. – М. : Наука, 1988. – С. 285.

УДК 004.8, 004.93

## РЕШЕНИЕ ПРОБЛЕМЫ РАСПОЗНАВАНИЯ ОБРАЗОВ ПРИ ПОМОЩИ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

**Колбасов С.Ю., Сорокин Р.А.**

Донецкий национальный технический университет, г. Донецк  
кафедра искусственного интеллекта и системного анализа

E-mail: [kolbasovsergey02@gmail.com](mailto:kolbasovsergey02@gmail.com), [eagle200877@gmail.com](mailto:eagle200877@gmail.com)

### *Аннотация*

*Колбасов С.Ю., Сорокин Р.А. Решение проблемы распознавания образов при помощи сверточных нейронных сетей. Работа посвящена обзору решения проблемы распознавания образов при помощи сверточных нейронных сетей. Приведены основные проблемы задачи распознавания образов, описаны основные шаги работы сверточных нейронных сетей.*

### *Abstract*

*Kolbasov S.Y., Sorokin R.A. Pattern recognition problem solving using convolutional neural networks. The work is dedicated to an overview of the solution to the problem of pattern recognition using convolutional neural networks. The main problems of the pattern recognition problem are given, the main steps of convolutional neural networks are described.*

**Введение.** В веке цифровых технологий с каждым годом совершаются значительные открытия и достижения, которые бесспорно улучшают, облегчают и продлевают человеческую жизнь. В попытке достичь поставленных задач иногда уходит несколько поколений научной деятельности ученых, связано это со многими причинами, такими как отсутствие различных технологий, способов наблюдения, ведения, которые позволят собирать огромные статистические данные при эмпирических наблюдениях. Одной из таких проблем является распознавание образов. Необходимость в таком распознавании возникает в самых разных областях – от военного дела и систем безопасности до оцифровки аналоговых сигналов [1]. Существует огромное множество задач, решение которых затруднительно, а порой и невозможно без участия специализированных программ для распознавания образов:

- оптическое распознавание символов;
- распознавание штрих-кодов;
- распознавание автомобильных номеров;
- распознавание лиц;
- распознавание речи;
- распознавание изображений;
- распознавание локальных участков земной коры, в которых находятся месторождения полезных ископаемых;
- классификация документов;

**Цель статьи.** Изучение особенностей решения проблемы распознавания образов при помощи сверточных нейронных сетей.

**Распознавание образов.** Создание устройств, выполняющих функции распознавания объектов в видеопотоке, в большинстве случаев позволит заменить человека специализированным автоматом. В настоящее время множество ученых работают в сфере исследований для решения проблемы автоматического распознавания объектов в



видеопотоке. Это стимулируется практическими требованиями, связанными с созданием систем военного и коммерческого назначения.

В задаче распознавания объектов существует несколько серьезных проблем:

- сильная зависимость от начальных параметров (необходимо знать достаточно много о тех объектах, которые собираемся искать в видеопотоке);
- идентификация объекта (различные похожие объекты могут быть похожи, например, мотоцикл и велосипед);
- внутриклассовая изменчивость.

Для достижения поставленной цели решены следующие задачи:

- исследование алгоритмов распознавания шаблонов;
- исследование влияния шума на качество распознавания;
- предложены методы, направленные на дальнейшее повышение эффективности и качества распознавания шаблонов.

Основной задачей распознавания является необходимость ответить на вопрос «Где на изображении (и есть ли) искомый объект?». Структурный подход предполагает ответ в виде ограничивающего прямоугольника вокруг объекта. Задача осложняется сложными фонами на изображениях и достаточно большой внутриклассовой изменчивостью объектов. Например, существуют сотни видов обычного кресла.

Формально задачу распознавания шаблонов можно описать двумя устройствами: датчик и классификатор. Датчик преобразует физические характеристики объекта, подлежащего распознаванию в набор признаков  $x = (x_1 \dots x_n)$ , которые характеризуют данный объект.

Пусть  $X$  – множество описаний объектов.  $W = (w_1 \dots w_m)$  – конечное множество номеров классов. Будем считать, что система распознавания допускает ошибку в том случае, если она относит к классу  $w_i$  объект, на самом деле принадлежащий классу отличному от  $w_i$ . Система распознавания  $R_1$  лучше системы  $R_2$ , в том случае, если вероятность совершения ошибки для системы  $R_1$  меньше, чем для системы  $R_2$ . Датчик выдает информацию в виде вектора  $x = (x_1, x_2, \dots, x_n)$ , где  $n$  – число измеренных характеристик каждого физического объекта. Предполагается, что вектор измерений (признаков)  $x$  принадлежит одному из  $M$  классов образов  $w_1 \dots w_m$ .

Решающая функция представляет собой функцию  $u(x)$ , относящую  $x$  точно к одному из  $M$  заданных классов. Оптимальной считается функция, которая дает наименьшую вероятность ошибки при всех допустимых значениях  $x$ .

Функция  $u$  относит набор  $x$  к классу  $w_i$  в том и только в том случае, если выполняется неравенство  $p(x|w_i) > p(x|w_j) \forall i \neq j$ .

**Сверточная нейронная сеть.** Наилучшие результаты в области распознавания лиц показала Convolutional Neural Network или сверточная нейронная сеть (далее – СНС), которая является логическим развитием идей таких архитектур НС как когнитрона и неокогнитрона. Успех обусловлен возможностью учета двумерной топологии изображения, в отличие от многослойного персептрона.

Сверточные нейронные сети обеспечивают частичную устойчивость к изменениям масштаба, смещениям, поворотам, смене ракурса и прочим искажениям. Сверточные нейронные сети объединяют три архитектурных идеи, для обеспечения инвариантности к изменению масштаба, повороту, сдвигу и пространственным искажениям:

- локальные рецепторные поля (обеспечивают локальную двумерную связность нейронов);
- общие синаптические коэффициенты (обеспечивают детектирование некоторых черт в любом месте изображения и уменьшают общее число весовых коэффициентов);
- иерархическая организация с пространственными подвыборками.

На данный момент сверточная нейронная сеть и ее модификации считаются лучшими по точности и скорости алгоритмами нахождения объектов на сцене. Начиная с 2012 года, нейросети занимают первые места на известном международном конкурсе по распознаванию образов ImageNet.

Сверточная нейронная сеть – это такой особый вид нейросетей прямого распространения, под прямым распространением понимается то, что переменные нейроны в этой сети разбиты на группы, называемые слоями [3]. И когда такая слоистая нейронная сеть применяется к данным, то активация слоев (значение этих переменных) подсчитывается последовательно: сначала значение активации первого слоя, потом значение активации второго слоя, и так до последнего слоя. Активация последнего слоя и служит выходами нейронной сети, и в этой нейросети есть много параметров, в каждом слое есть какие-то свои параметры, которые определяют, как активация следующего слоя зависит от активации предыдущего слоя. И что еще важно, активации внутри одного слоя могут подсчитываться параллельно, одновременно, они друг от друга не зависят, и это приводит к тому, что такие нейросети можно очень удобно и эффективно обчислывать на современных процессорах, в том числе на графических сопроцессорах.

Архитектура СНС может содержать большое количество различных слоев, однако основными слоями, которые присутствуют практически всегда, считаются слои свертки, пулинга (иначе подвыборки или субдискретизации) и полносвязные [2, 3].

Слой свертки назван так из-за операции свертки, которая является основным процессом данного слоя. Работа слоя заключается в использовании фильтра (обычно матрица размером  $3 \times 3$ ), значения которого умножаются на исходные значения пикселей изображения, после чего эти значения суммируются. Каждая уникальная позиция исходного изображения производит число, совокупность таких чисел называется картой признаков [2].

Слой пулинга (субдискретизации) выполняет операцию по понижающей дискретизации пространственных размеров (ширина и высота) исходного изображения, в результате чего объем сокращается. То есть, выполняется нелинейное уплотнение карты признаков. Обычно слой пулинга использует окно размером  $2 \times 2$  с шагом 2, то есть на каждом шаге в окно попадают только уникальные пиксели. Однако в различных архитектурах могут использоваться другие значения шага и размера окна. Уплотнение производится отбором пикселей, попавших на текущем шаге в окно, для отбора используются такие основные способы: выбор пикселя с максимальным значением или нахождение среднего значения среди всех пикселей, попавших в окно. Основная логика слоя субдискретизации заключается в том, что, если, на предыдущем слое свертки были выявлены некоторые признаки, то для дальнейшей обработки не требуется настолько подробное изображение, поэтому оно уплотняется до менее подробного.

Полносвязный слой выводит  $N$ -мерный вектор ( $N$  – число классов) для определения нужного класса, то есть выполняет классификацию. Работа данного слоя организуется путем обращения к выходу предыдущего слоя (карте признаков) и определения свойств, которые наиболее характерны для определенного класса. Полносвязный слой, по своей сути, является многослойным персептроном. СНС обычно содержат несколько таких слоев. Однако использование полносвязных слоев также накладывает ограничение на размер входных изображений – они все должны быть одинакового размера, поэтому некоторые современные СНС не используют данный слой и выполняют классификацию другими способами, что позволяет использовать изображения любого размера как во время обучения, так и во время классификации.

Обучение нейросети, то есть настройка ее параметров, происходит на большом количестве обучающих данных следующим образом: для каждого обучающего примера известно, что необходимо получить на выходе. Берется текущее состояние нейросети, применяется такая нейросеть к конкретному обучающему примеру, смотрится, что

получилось на выходе и как отличается от того, что ожидалось. Дальше путем концептуально несложных математических манипуляций, называемых методом обратного распространения ошибки, можно понять, как именно нужно модифицировать параметры каждого слоя, чтобы на выходе стало получаться что-то более похожее на то, что ожидается. И вот нейросеть поочередно смотрит на отдельные примеры или чаще на наборы примеров одновременно, чтобы увеличить правильность вычислений, и, глядя на эти отдельные примеры или пакеты примеров, она постепенно подстраивает свои параметры и добивается того, что на обучающих примерах она начинает предсказывать то, что нужно.

Достоинствами данного подхода является ряд признаков [3].

1. По сравнению с полносвязной нейронной сетью (типа перцептрона) – гораздо меньшее количество настраиваемых весов, так как одно ядро весов используется целиком для всего изображения, вместо того, чтобы делать для каждого пикселя входного изображения свои персональные весовые коэффициенты. Это подталкивает нейросеть при обучении к обобщению демонстрируемой информации, а не попиксельному запоминанию каждой показанной картинке в мириадах весовых коэффициентов, как это делает перцептрон.

2. Удобное распараллеливание вычислений, а следовательно, возможность реализации алгоритмов работы и обучения сети на графических процессорах.

3. Относительная устойчивость к повороту и сдвигу распознаваемого изображения.

4. Обучение при помощи классического метода обратного распространения ошибки.

Однако, при столь огромных преимуществах существует ряд весомых недостатков:

– плохая трансляционная инвариантность и отсутствие информации о ориентации (СНС имеют проблемы при повороте объектов или при изменении условий освещения – чтобы как-то решить эти проблемы – применяется аугментация данных);

– сильная потеря информации на пулинге.

**Выводы.** В ходе данного исследования была представлена проблема распознавания образов, как одного из методов классификации и индентификации предметов, важность решения данной проблемы для развития современной науки. В качестве решения данной проблемы были представлены сверточные нейронные сети, было приведено подробное описание алгоритма, достоинств и недостатков данного подхода к распознаванию образов.

### Литература

1. Барский А.Б. Нейронные сети: распознавание, управление, принятие решений / А.Б. Барский. – М.: Финансы и статистика, 2004. – 176 с.
2. Николенко С. Глубокое обучение. Погружение в мир нейронных сетей / С. Николенко, А. Кадури, Е. Архангельская. – СПб.: Питер, 2018. – 480 с.
3. Goodfellow I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. – The MIT Press, 2016. – 775 p.

УДК 004.85

## ИСПОЛЬЗОВАНИЕ МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ ДЛЯ ОПТИМИЗАЦИИ СОСТАВЛЕНИЯ РАССПИСАНИЯ В ОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЯХ

**Пашенко Д.А., Маслова Е.А.**

Донецкий национальный технический университет  
кафедра искусственного интеллекта и системного анализа

E-mail: [pashektor77@gmail.com](mailto:pashektor77@gmail.com)

### **Аннотация:**

*Пашенко Д.А., Маслова Е.А. Использование методов интеллектуального анализа данных для оптимизации составления расписания в образовательных организациях. Рассмотрен процесс анализа данных, определены его основные характеристики, а также выявлены возможности применения на практике для оптимизации составления и работы с расписанием.*

### **Annotation:**

*Pashchenko D.A., Maslova E.A. Using data mining techniques for optimizing scheduling in educational organizations. The process of data analysis was described, its main characteristics were determined, and the practical applications for optimizing the compilation and working with the schedule were identified.*

### **Общая постановка проблемы**

Одной из важнейших проблем качественной организации учебного процесса в образовательных организациях является задача формирования учебного расписания. Правильно и точно составленное расписание обеспечивает равномерную загрузку студенческих групп и профессорско-преподавательского состава.

Таким образом, необходимо найти способ для создания наиболее оптимального расписания и возможность его редактирования при необходимости.

### **Цель статьи**

Определить наиболее подходящие для составления и редактирования расписания методы интеллектуального анализа данных.

### **Актуальность данной статьи**

Проблема усовершенствования учебного процесса образовательной организации за счет планирования проведения занятий является известной, ее решением занимались многие ученые и практики по созданию автоматизированных систем. В настоящее время эта проблема остается открытой. Актуальность задачи определяется ростом требований к качеству обучения, планированию работы студентов в условиях дефицита аудиторного фонда и др. При создании плана расписания занятий необходимо учитывать множество противоречивых и нечетко определенных факторов: занятость преподавателей, аудиторного фонда, наличие основных и альтернативных занятий, а при назначении лабораторных занятий – занятость соответствующих лабораторий.

Составление расписания учебных занятий является одной из важнейших задач управления учебным процессом. В связи с этим проблема автоматизации составления расписаний учебных занятий в образовательных системах массового обучения по-прежнему остается одной из актуальных проблем организации учебного процесса.

### **Интеллектуальный анализ данных**

Интеллектуальный анализ данных — это обработка информации и выявление в ней моделей и тенденций, которые помогают принимать решения. Принципы интеллектуального анализа данных известны в течение многих лет, но с появлением больших данных они получили еще более широкое распространение.

Большие данные привели к росту популярности более широких методов интеллектуального анализа данных, отчасти потому, что информации стало гораздо больше, и она по самой своей природе и содержанию становится более разнообразной и обширной. При работе с большими наборами данных уже недостаточно относительно простой и прямолинейной статистики. Имея большое количество подробных записей о каждом проводимом занятии, недостаточно знать место и время их проведения. Чтобы лучше удовлетворить потребности преподавателей и студентов необходимо учитывать тип занятий, расстояние между корпусами и многие другие аспекты.

Эти требования привели от простого поиска и статистического анализа данных к более сложному интеллектуальному анализу данных. Для решения задач требуется такой анализ данных, который позволяет построить модель для описания информации и в конечном итоге приводит к созданию результирующего отчета. Этот процесс иллюстрирует рисунок 1.

Процесс анализа данных, поиска и построения модели часто является итеративным, так как в процессе работы возникает необходимость в выявлении дополнительных сведений, которые могут быть получены из дополнительных источников. Необходимо также понимать, как связать, преобразовать и объединить их с другими данными для получения результата. После обнаружения дополнительных элементов и характеристик, подход к выявлению источников и видов данных, с последующей их обработкой и получением требуемого результата, может измениться.

### **Ассоциация**

Ассоциация (или отношение), вероятно, наиболее известный, знакомый и простой метод интеллектуального анализа данных. Для выявления моделей делается простое сопоставление двух или более элементов, часто одного и того же типа. Например, отслеживая последовательность занятий, можно заметить, что сначала проводят лекционные занятия, а потом лабораторные или практические.

### **Классификация**

Классификацию можно использовать для получения представления о типе объектов, описывая несколько атрибутов для идентификации определенного класса. Например, студентов легко классифицировать по специальностям, определив читаемые для него дисциплины. Те же принципы можно применить и к преподавателям, например, классифицируя их по принадлежности к факультетам.

Кроме того, классификацию можно использовать в качестве входных данных для других методов. Например, для определения классификации можно применять деревья принятия решений. Кластеризация позволяет использовать общие атрибуты различных классификаций в целях выявления кластеров.



Рис. 1. Схема процесса анализа данных.

### **Кластеризация**

Исследуя один или более атрибутов, или классов, можно сгруппировать отдельные элементы данных вместе, получая структурированное заключение. На простом уровне при кластеризации используется один или несколько атрибутов в качестве основы для определения кластера сходных результатов. Кластеризация полезна при определении различной информации, потому что она коррелируется с другими примерами, так что можно увидеть, где подобию и диапазоны согласуются между собой. Метод кластеризации работает в обе стороны. Можно предположить, что в определенной точке имеется кластер, а затем использовать свои критерии идентификации, чтобы проверить это.

### **Прогнозирование**

Прогнозирование — это широкая тема, которая простирается от предсказания отказов компонентов оборудования до выявления мошенничества и даже прогнозирования прибыли компании. В сочетании с другими методами интеллектуального анализа данных прогнозирование предполагает анализ тенденций, классификацию, сопоставление с моделью и отношения. Анализируя прошлые события или экземпляры, можно предсказывать будущее.

Например, используя данные о количестве поступавших студентов за последние 5 лет можно сделать вывод о необходимости расширения или уменьшения аудиторного фонда.

### **Деревья решений**

Дерево решений, связанное с большинством других методов (главным образом, классификации и прогнозирования), можно использовать либо в рамках критериев отбора, либо для поддержки выбора определенных данных в рамках общей структуры. Дерево решений начинают с вопроса, который может иметь несколько ответов. Каждый ответ приводит к следующему вопросу, помогая классифицировать и идентифицировать данные или делать прогнозы.

При создании базы данных генерирующей расписание занятий образовательной организации использование методов интеллектуального анализа данных, обеспечивает удобный доступ к систематизированной и упорядоченной информации.

Учебный процесс, с точки зрения управления объектами в условиях ограниченных ресурсов, может быть разделен на три основных этапа.

Первый этап — этап планирования, этот этап может быть разделен на две составных части — планирование учебной нагрузки, т.е. закрепление учебных занятий в группах за конкретными преподавателями, и планирование поведения процесса, т. е. распределение этих занятий во времени (составление расписания). На втором этапе выполняется учебный процесс.

В период выполнения этого этапа возможны отклонения от начального значения состава преподавателей, аудиторного фонда, групп студентов. Управление на этом этапе сводится к минимизации потерь от этих отклонений без изменения общего расписания. Минимизация этих потерь может быть спланирована на этапе составления расписания при учете критериев устойчивости расписания к таким отклонениям.

На третьем этапе производится оценка результатов планирования и выполнения учебного процесса. Эта оценка имеет две составляющих: оценка достижения цели обучения и оценка качества расписания.

В первом случае оценивается учебный план и принимается решение о качестве обучения. Во втором — оценивается само расписание с точки зрения эргономических показателей, устойчивости и т. п. и могут быть выработаны критерии для построения расписания на следующий период работы.

В соответствии с предложенной схемой управления учебным процессом можно выделить две области, где возможна автоматизация управления, это этап составления расписания и расчет частных и обобщенной оценок результата учебного процесса. На рисунке 2 приведен процесс составления расписания.

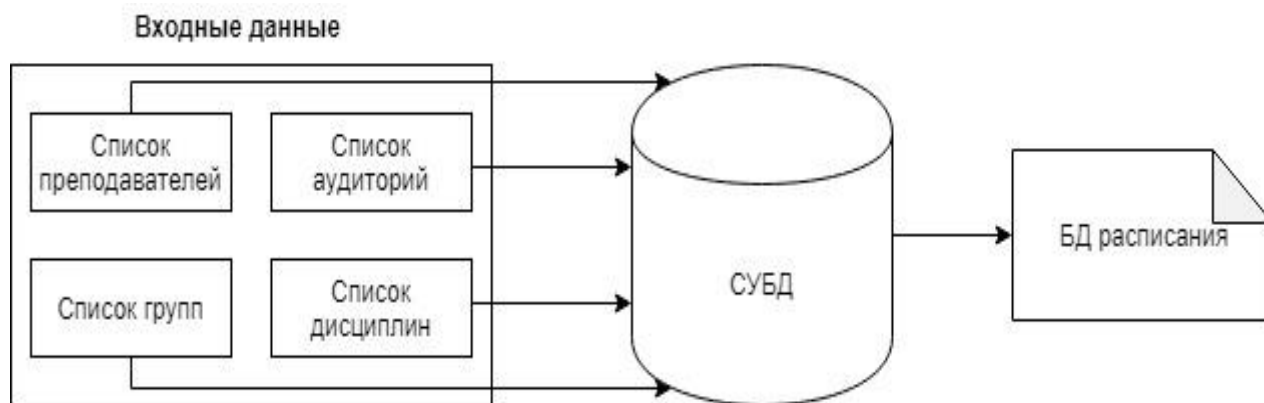


Рис. 2. Процесс составления расписания.

Для решения существующих проблем при составлении расписания для образовательных организация требуется построение гибкой и легко адаптируемой базы данных на основе новых принципов, с использованием современных компьютерных технологий. На рисунке 3 приведена схема потоков данных базы данных «Расписание занятий».

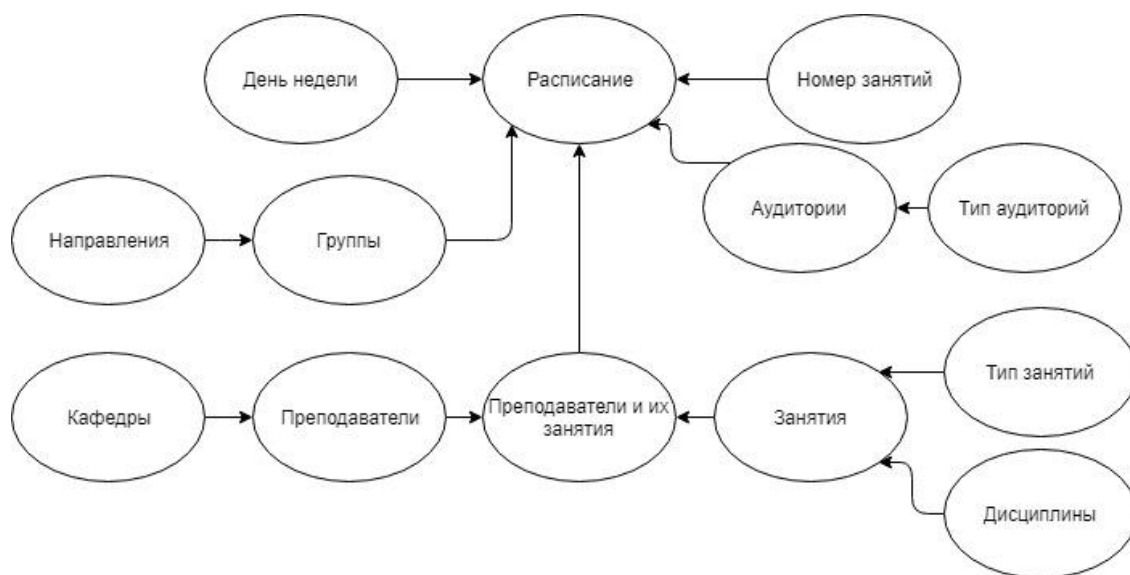


Рис. 3. Схема потоков данных БД «Расписание занятий».

### Выводы

Интеллектуальный анализ данных — это не только выполнение некоторых сложных запросов к данным, хранящимся в базе данных. Независимо от того какими средствами обрабатывается информация, необходимо работать с данными, форматировать или реструктурировать их. Требуется определить формат информации, на котором будет основываться последующий анализ и алгоритм реализации. Затем, когда информация будет сформирована в нужном формате, можно применять различные методы, не зависящие от требуемой базовой структуры данных или набора данных.

Данный анализ может успешно применяться в задачах составления расписания, учитывая большие объемы и разнообразную структуру обрабатываемой в этих задачах информации.

### Литература

1. Симбирская Л.М. Компьютерная система планирования учебной работы ВУЗа: Сборник научных трудов / Л.М. Симбирская, И.В. Клитная. – Харьков: Ранок, 2002. – 137 с.
2. Арвачева А.Э. Разработка информационной системы «Расписание занятий» для учреждений высшего образования / Молодой ученый, 2016. – 50-52 с.
3. Методы интеллектуального анализа данных [Электронный ресурс]. – Режим доступа: <https://www.ibm.com/developerworks/ru/library/ba-data-mining-techniques/index.html>. – Загл. с экрана.



УДК 004.6, 004.91

## РАЗРАБОТКА АНДРОИД-ПРИЛОЖЕНИЯ УЧЕТА И ВЕДЕНИЯ ЗАКАЗОВ ДОСТАВКИ ЕДЫ

**Радченко М.В., Сорокин Р.А.**

Донецкий национальный технический университет  
кафедра искусственного интеллекта и системного анализа

E-mail: [mikr.radchenko@gmail.com](mailto:mikr.radchenko@gmail.com), [eagle200877@gmail.com](mailto:eagle200877@gmail.com)

### *Аннотация*

*Радченко М.В., Сорокин Р.А. Разработка андроид-приложения учета и ведения заказов доставки еды. Работа посвящена разработке андроид-приложения учета и ведения заказов еды для облегчения работы сотрудников малого бизнеса. Приведены основные требования к функциональным возможностям системы, описаны составные части разрабатываемого приложения.*

### *Annotation*

*Radchenko M.V., Sorokin R.A. Development android application of accounting and ordering food delivery. The work is devoted to the development of an Android application for accounting and ordering food to facilitate the work of small business employees. The basic requirements for the functionality of the system are given, the components of the developed application are described.*

**Введение.** Сегодня активно развивается огромное количество малых бизнес идей, связанных с производством и распространением еды. Если доходы от бизнеса не велики, то возникает вопрос обеспечения качественных услуг доставки, которые, в то же время, не требуют больших затрат. Тут, на помощь предпринимателям, появляются не затратные мобильные приложения. Яркими представителями являются «Glovo» и «Uber Eats». Это сервисы доставки, в которых пользователи могут осуществлять заказ и следить за ходом его выполнения. Также необходимо отметить, что для работы курьером необходим минимальный набор инструментов, а именно: транспортное средство, телефон и желание работать. Данный способ подработки активно развивается в Америке, Европе, Японии и других странах. В данной работе описывается этапы построения приложения по осуществлению учета и ведения заказов доставки еды, его составные части.

**Цель статьи.** Разработка требований к андроид-приложению обеспечения учета и ведения заказов, а также статистический анализ данных, с помощью которого можно определять прибыльность того или иного товара.

**Постановка задачи.** В ходе реализации приложение должно выполнять следующие функции:

- взаимодействие с пользователем (еда и данные о доставке);
- отслеживание состояния и статуса товаров;
- хранение данных о реализованных товарах;
- интеллектуальный анализ данных по реализованным товарам (спрос на товар, прибыльность товара, расчет максимальной прибыли за отдельно взятый период).

Разрабатываемое приложение должно включать в себя базу данных, которая обеспечивает хранение поступающих, выполненных запросов, а также запросов, находящихся в обработке. База не должна иметь большой объем и должна быть хорошо приспособлена для работы с андроид-устройством.

Приложение должно иметь интуитивно понятный интерфейс, реализованный в нескольких рабочих экранах.

**Исследования.** Программное обеспечение разрабатывается с целью осуществления приема заказов, их оперативной обработки, ведения, сбор статистической информации и дальнейшей обработки собранной информации с целью проведения интеллектуального анализа.

В ходе разработки программного обеспечения были выполнены следующие шаги:

- анализ аналогичных приложений на рынке;
- архитектура типичного андроид-приложения;
- выполнение андроид-приложения[2];
- исследована работа с базами данных в андроид-приложениях;
- методы интеллектуального анализа.

В процессе исследования предметной области было выявлено, что на рынке программного обеспечения существует не так много небольших по размеру и затратам ресурсов программ, которые занимаются учетом и отслеживанием доставки различного рода продукции. Мобильность подобного приложения значительно упрощает ведение бизнеса и создает новые рабочие места. Как правило, такие приложения специализируются только на учете доставки товаров и в них отсутствует интеллектуальный анализ. Наличие же такого поможет предпринимателям выявлять наиболее прибыльные товары и ориентироваться на покупателя.

**Программная реализация.** На самом деле приложение «Android» – всего лишь набор файлов в заранее определенных каталогах. При построении приложения все эти файлы собираются воедино, и может быть получено приложение, которое можно запустить на устройстве. Платформа «Android» состоит из множества компонентов. В нее входят базовые приложения (например, Контакты), набор программных интерфейсов (API) для управления внешним видом и поведением приложения, а также множество вспомогательных файлов и библиотек поддержки[1].

Исполнительная среда «Android» включает в себя набор базовых библиотек, реализующих большую часть языка программирования Java. Каждое андроид-приложение выполняется в отдельном процессе. В самом основании системы лежит ядро «Linux». В «Android» оно обеспечивает работу драйверов, а также таких базовых сервисов как безопасность и управление памятью. Приложение состоит из активностей, макетов и других ресурсов. Одна из этих активностей является главной активностью приложения. У каждого приложения имеется главная активность, заданная в файле «AndroidManifest.xml»[3].

Для работы большинства систем управления базами данных необходим специальный процесс сервера базы данных. «SQLite» обходится без сервера; база данных «SQLite» представляет собой обычный файл. Когда база данных не используется, она не расходует процессорное время. Это особенно важно для мобильных устройств в целях экономии уровня заряда аккумулятора. С базой данных взаимодействует только приложение, поэтому можно обойтись без идентификации с именем пользователя и пароля. Базы данных «SQLite» невероятно надежны. Они поддерживают транзакции баз данных (другими словами, если при обновлении нескольких блоков данных что-то пойдет не так, «SQLite» сможет вернуться к исходному состоянию). Кроме того, операции чтения и записи данных реализуются на оптимизированном коде C. Этот код не только быстро работает, но и сокращает объем необходимых вычислительных ресурсов. Система «Android» включает набор классов для управления базой данных «SQLite». Основная часть этой работы выполняется тремя типами объектов. Помощник «SQLite» создается расширением класса «SQLiteOpenHelper». Он предоставляет средства для создания и управления базами данных. Класс «SQLiteDatabase» предоставляет доступ к базе данных. Класс «Cursor» предназначен для чтения и записи в базу данных. Класс «SQLiteOpenHelper» упрощает задачи создания и сопровождения баз

данных. Считается, что это своего рода личный ассистент, который берет на себя служебные операции по управлению базами данных.[4]

В разрабатываемом приложении будет проводиться интеллектуальный анализ данных выполненных заказов. Данный анализ предоставит предпринимателю полезную информацию по рентабельности своей продукции, и он сможет оптимизировать некоторые «тонкие участки» своей деятельности. Как пример: определение максимально прибыльного товара за определенный период, выявление самого неприбыльного, выявление различных закономерностей, связанных со временем заказа и т.д. Ниже приведены некоторые элементы интеллектуального анализа.

Методы ИАД позволяют выделять пять основных типов закономерностей.

1. Ассоциация – выявление связи (совместности) нескольких событий.
2. Последовательность – ассоциация с учетом временной составляющей.
3. Классификация – выделение признаков классов (сгруппированных данных).
4. Кластеризация – формирование классов.
5. Прогнозирование.

В таблице 1 приведены примеры использования интеллектуального анализа данных в различных прикладных областях.

Таблица 1 – Примеры использования ИАД

Задачи	Комментарии	Закономерности
Розничная торговля		
Анализ покупательской корзины (сходства)	Выявление товаров, которые стремятся покупать вместе. Применение - реклама, стратегия создания запасов и размещения товаров	Совместность
Исследование временных шаблонов активности покупателей	Применение - создание товарных запасов	Последовательность
Банковское дело		
Выявление мошенничества с кредитными карточками	Выявление стереотипов поведения мошенников в результате анализа исторических данных	Классификация
Сегментация клиентов	Выявление ориентированности различных групп клиентов на различные виды услуг	Кластеризация

**Выводы.** В ходе исследования и разработки программного продукта были определены цели и задачи, необходимые для оптимизации учета и ведения заказов еды на базе платформы «Android». Рассмотрена типичная структура андроид-приложения, активность устройства при выполнении программ, возможности работа с базами данных «SQLite», возможность анализа статистических данных.

#### Литература

1. Гриффитс, Д. Head First. Программирование под Android / Д. Гриффитс, Д. Гриффитс; [пер.с англ.]. – СПб.: Питер, 2016. – 704 с.

2. Гома, Х. UML. Проектирование систем реального времени, параллельных и распределенных приложений / Х. Гома; [пер.с англ.]. – М.: ДМК Пресс, 2012. – 704 с.
3. Медникс, З. Программирование под Android / З. Медникс, Л. Дорнин, Б. Мик, М. Накамура; [пер.с англ.] – 2-е изд. – СПб.: Питер, 2013. – 560 с.
4. Еранссон, А. Эффективное использование потоков в операционной системе Android / А. Еранссон; [пер.с англ.]. – М.: ДМК Пресс, 2015. – 304 с.

УДК 004.93

## АНАЛИЗ ЭМОЦИОНАЛЬНОГО СОСТОЯНИЯ ЧЕЛОВЕКА ПО ФОТОГРАФИЧЕСКИМ ИЗОБРАЖЕНИЯМ

А.С. Миненко, А.П. Семенова

Донецкий национальный технический университет, г. Донецк  
кафедра искусственного интеллекта и системного анализа

E-mail: [sam\\_dntu@mail.ru](mailto:sam_dntu@mail.ru), [nastena-semenova19@rambler.ru](mailto:nastena-semenova19@rambler.ru)

### *Аннотация*

*Миненко А.С., Семенова А.П. Анализ эмоционального состояния человека по фотографическим изображениям. В работе рассмотрены основные тенденции развития алгоритмов распознавания эмоций, базовые понятия теории распознавания эмоций, алгоритм распознавания эмоций по фотографическим изображениям лица.*

### *Abstract*

*Minenko A.S., Semenova A.P. Analysis of human emotional state by photographic images. The paper considers the main trends of the development of emotion recognition algorithms, the basic concepts of emotion recognition theory, algorithm of emotions recognition by photographic facial images.*

### **Введение**

В основе распознавания истинного лица человека лежит распознавание эмоций. Под эмоциями понимают либо внутренние чувства человека, либо проявления этих чувств. Эмоция – это психический процесс импульсивной регуляции поведения, основанный на чувственном отражении потребностной значимости внешних воздействий, их благоприятности или вредности для жизнедеятельности индивида. Таким образом, эмоцией называется особый вид психических процессов, которые отражают субъективное оценочное отношение к существующим или возможным ситуациям, к окружающему миру и самому себе в конкретный момент времени. Хотя эмоции могут быть выражены с помощью голоса, мимики, движения, позы и вегетативных реакций (частота сердечных сокращений и дыхания, артериальное давление), наибольшей значимостью обладает именно лицо человека [1-3].

Общение между людьми невозможно без проявления и анализа эмоций. Поэтому моделирование и распознавание эмоций является актуальным и важным направлением исследований при создании систем компьютерного распознавания и синтеза зрительных образов [4].

Невербальная, мимическая передача информации человеком стала предметом исследований в Массачусетском технологическом институте, Оксфордском, Санкт-Петербургском, Московском университетах, Институте кибернетики им. В.М. Глушкова,

Киевском национальном университете им. Тараса Шевченко, Донецком национальном техническом университете и др.

Задача распознавания эмоций может решаться в системах, применяемых в различных сферах человеческой деятельности. К наиболее значимым областям применения можно отнести: распознавание состояния водителя, криминалистику, маркетинговые исследования, человеко-машинное взаимодействие, системы безопасности и виртуальной реальности, online-обучение и др. [5].

#### **Формальная модель эмоций**

Рассмотрим базовые понятия, лежащие в основе подходов к распознаванию эмоций. Эмоции предназначены для решения определенных жизненных трудностей, сложных ситуаций. Каждая эмоция подготавливает человека к какому-то конкретному событию. Это событие может происходить как с внешним объектом, так и с самим человеком.

В работе [6] предложено описывать эмоции используя набор бинарных признаков. Для обозначения эмоции используется вектор со следующими признаками:

$$Em_i^n = (E_1, E_2, E_3), i = \overline{1,8}, \quad (1)$$

где  $E$  – бинарный признак, классифицирующий конкретные эмоции:

$E_1$  – признак, определяющий знак эмоции – позитивная (1) или негативная (0) эмоция;

$E_2$  – признак, определяющий время возникновения эмоции относительно действия (предвосхищающая (0) и констатирующая (1) эмоции);

$E_3$  – признак, определяющий направление конкретной эмоции.

Комбинируя три описанных бинарных признака, получаем 8 различных вариантов. Для классификации эмоций по источнику их происхождения введен 4-ый признак ( $\eta$ ) [7]:

–  $\eta=1$  – эмоции, связанные с удовлетворением (неудовлетворением) личных нужд человека;

–  $\eta=2$  – эмоции, которые возникают в результате сравнения некоторого объекта, самого себя или своих действий со своими же нормами, стандартами, правилами, убеждениями;

–  $\eta=3$  – эмоции, которые возникают в результате сравнения объекта с общественными правилами и нормами;

–  $\eta=4$  – эмоции, которые возникают в связи с потребностями других людей;

–  $\eta=5$  – эмоции, которые возникают в результате взаимоотношений с другим человеком;

–  $\eta=6$  – эмоции, которые возникают на основе презрения.

Путем соединения 4-х приведенных признаков можно описать 48 качественно различных эмоций.

В качестве базовых эмоций выделяют горе, радость, страх, надежду, гнев, удовлетворение, интерес, пренебрежение. Используя полученные вектора базовых эмоций ( $Em_i^1, i = 0 \dots 7$ ), можно построить векторную модель эмоций как выпуклую комбинацию двух эмоций по схеме, приведенной в работе [7].

#### **Информационные технологии при моделировании и распознавании лица человека по его мимическим изображениям**

В работах [8-10] рассмотрены некоторые положения процесса распознавания эмоций на мимических изображениях. В работе [8] предложено для выделения характерных признаков черт лица использовать модели, которые задаются с помощью неравномерных рациональных базисных сплайнов кривых. Подробно рассмотрен этап нахождения основных фрагментов лица (нос, брови, глаза, губы), процесс преобразования фотографических изображений с определенными эмоциональными состояниями в векторы значений,

содержащих координаты контуров основных фрагментов лица. Полученные контуры фрагментов лица служат основой для дальнейшего анализа и синтеза эмоций на изображении.

Для поиска пространства характеристических признаков, построения базиса этого пространства, воспроизведения производных эмоциональных состояний со следующим использованием выпуклой комбинации, предлагается [7,8]:

- создание множества фотографических изображений, на которых актерами воссоздаются ситуации ( $E_1, E_2, E_3$ ), в которых возникают базовые эмоции, и описание мимики, соответствующие этим эмоциям;
- анализ полученного множества с целью выявления областей, которые содержат характеристические признаки эмоций и их описание (используя анатомические признаки и методику Facial Action Coding System);
- создание в пространстве характеристических признаков базиса для последующего описания по нему произвольных векторов мимических проявлений эмоциональных состояний (как выпуклой комбинации базисных эмоциональных состояний);
- анализ характеристических признаков и ранжирование их по степени влияния в рамках предложенного базиса.

Для анализа полученного множества фотографических изображений с целью выявления областей, которые содержат характеристические признаки эмоций, используется подход, предложенный авторами методики FACS [11-13].

В ходе анализа мимических проявлений лица, было получено 21 характеристический признак [8,10], комбинация которых образует базис мимических проявлений эмоций. Т. о., мимические проявления эмоций ( $Em$ ) были представлены в виде вектора:

$$Em_i^n = (\mu_1, \dots, \mu_{21}), i = \overline{1,8}, \quad (2)$$

где  $\mu_j \in [0;1]$  – характеристический мимический признак (при  $\mu = 0$  – нет признака, а при  $\mu = 1$  влияние признака максимально).

Набор из 8-ми векторов, указанных таким образом, образует базис ( $B_{ij}$ , где  $i=1, \dots, 21$ ,  $j=1, \dots, 8$ ) – пространство мимических проявлений эмоциональных состояний.

Произвольный вектор  $b = (\mu_1, \dots, \mu_{21})$ , полученный путем анализа изображений с каким либо эмоциональным состоянием, можно разложить по базису  $B$  и получить описание эмоции, как выпуклой комбинации [10]:

$$x = (B^T B)^{-1} B^T b \quad (3)$$

где  $B$  – базисная матрица эмоциональных состояний;  $B^T$  – транспонирована матрица  $B$ ;  $b$  – вектор, описывающий мимическое проявление произвольного эмоционального состояния;  $x = (a_1, \dots, a_8)$ , где  $a_1$  – коэффициенты выпуклой комбинации ( $\sum a_1 = 1, a \in [0;1]$ ) для каждой из 8-ми базовых эмоций.

Так как мимически невозможно определить источник происхождения эмоций ( $\eta$ ), получаем множество из 21-й эмоции, которую можно определить предложенными способами.

### Вывод

Алгоритм для распознавания эмоций по фотографическим изображениям в общем виде можно разбить на несколько этапов: предварительная обработка фотоизображения лица (включает в себя геометрические и цветовые преобразования, удаление шумов), поиск лица

на изображении, выделение наиболее значимых частей лица (нос, брови, глаза, губы), построение контурной модели (построение сплайн кривых), классификация эмоций.

### **Литература**

1. Ильин Е.П. Эмоции и чувства / Е.П. Ильин – СПб: Питер, 2001. – 752 с.
2. Джеймс У. Психология / Джеймс У. – М.: Педагогика, 1991. – 368 с.
3. Ланге Г. Душевные движения / Г. Ланге – СПб.: Питер, 1996. – 180 с.
4. Визильтер Ю.В. Обработка и анализ изображений в задачах машинного зрения: Курс лекций и практических занятий. / Ю.В. Визильтер, С.Ю. Желтов, А.В. Бондаренко, М.В. Ососков, А.В. Моржин. – М.: Физматкнига, 2010. – 672 с.
5. Семенова А.П., Миненко А.С., Ванжа Т.В. Область применения алгоритма распознавания эмоций в информационных технологиях // Цифровой регион: опыт, компетенции, проекты: сборник статей Международной научно-практической конференции (г. Брянск, 30 ноября 2018 г.) [Электронный ресурс]. – Брянск: Брян. гос. инженерно-технол. ун-т., 2018. – С. 443-446.
6. Миненко А.С., Семенова А.П. Анализ эмоционального состояния человека на основе математического моделирования // Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ – 2018). Сборник материалов IX Международной научно-технической конференции в рамках IV Международного Научного форума Донецкой Народной Республики «Инновационные перспективы Донбасса». – Донецк: ДонНТУ, 2018. – С. 22-25.
7. Миненко А.С., Семенова А.П. Формальная модель эмоций / А.С. Миненко, А.П. Семенова // Проблемы искусственного интеллекта. – 2018. – №3(10). – С. 84-93.
8. Миненко А. С. Моделирование и информационные технологии при распознании лица человека по его мимическим изображениям / А.С. Миненко // Проблемы искусственного интеллекта. – 2016. – Донецк: ГУ ИПИИ. – № 2(3). – С. 48-54.
9. Миненко А.С. Вариационные задачи со свободной границей / А.С. Миненко. – Киев: Наукова думка, 2005. – 354 с.
10. Снетков В.А. Габитоскопия / В.А. Снетков – Волгоград: ВСИШ МВД СССР, 1979 – 182 с.
11. Friesen W. EMFACS-7: Emotional Facial Action Coding System. Unpublished manual / W. Frisen, P. Ekman. – California: University of California, 1983.
12. Ekman P. Facial Action Coding System: A Technique for the Measurement of Facial Movement [Text] / P. Ekman, W. Friesen. – Consulting Psychologists Press, Palo Alto, 1978.
13. DeBoor C. A. Practical Guide to Splines [Text] / DeBoor C. – New York: Springer-Verlag, 1978. – 392 p.

УДК 004.413

## МЕТОДОЛОГИИ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

**Грабельников В.А.**

Донбасская юридическая академия, ДНР, г. Донецк

E-mail: [eagle200877@gmail.com](mailto:eagle200877@gmail.com)

### **Аннотация:**

*Грабельников В.А. Методологии тестирования программного обеспечения. В работе проведен обзор методологий тестирования программного обеспечения и ведения проекта. Выделены направления развития современных методологий.*

### **Annotation:**

*Grabelnikov V.A. Software testing methodologies. The paper provides an overview of software testing and project management methodologies. The directions of development of modern methodologies are highlighted.*

### **Общая постановка проблемы**

Как и процесс разработки, процесс последующего тестирования программного обеспечения также следует определенной методологии [1]. Под методологией в данном случае понимаются разнообразные комбинации принципов, идей, методов и концептов, к которым прибегают во время работы над проектом.

В настоящее время существует довольно большое количество разнообразных подходов к тестированию, каждый со своими отправными точками, продолжительностью выполнения и методами, используемыми на каждом этапе. И выбор того или иного из них может быть довольно непростой задачей. В этой статье рассматриваются разные подходы к тестированию ПО и приводятся их основные особенности, чтобы помочь сориентироваться в существующем многообразии.

### **Каскадная модель (Линейная последовательная модель жизненного цикла ПО)**

Каскадная модель (Waterfall Model) является одной из наиболее старых моделей, которую можно применять не только для разработки или тестирования ПО, но также практически для любого другого проекта [2]. Его базовым принципом является последовательный порядок выполнения задач. Это значит, что мы можем переходить к следующему шагу разработки или тестирования только после того, как предыдущий был успешно завершен. Эта модель подходит для небольших проектов и применима только в том случае, если все требования точно определены. Главными достоинствами этой методологии являются экономическая эффективность, простота использования и управления документацией.

Процесс тестирования ПО начинается после завершения процесса разработки. На этой стадии все необходимые тесты переносятся с юнитов на системное тестирование для того, чтобы контролировать работу компонентов как по отдельности, так и в комплексе.

Помимо упомянутых выше достоинств, данный подход к тестированию также имеет и свои недостатки. Всегда существует вероятность обнаружения критических ошибок в процессе тестирования. Это может привести к необходимости полностью изменить один из компонентов системы или даже всю логику проекта. Но подобная задача невозможна в случае каскадной модели, поскольку возвращение на предыдущий шаг в этой методологии запрещено.

### **V-Model (Модель верификации и валидации)**

Как и каскадная модель, методика V-Model основана на прямой последовательности шагов [2]. Основным отличием между этими двумя методологиями является то, что



тестирование в данном случае планируется параллельно с соответствующей стадией разработки. Согласно этой методологии тестирования ПО, процесс начинается как только определены требования и становится возможным начать статическое тестирование, т.е. верификацию и обзор, что позволяет избежать возможных дефектов ПО на поздних стадиях. Соответствующий план тестирования создается для каждого уровня разработки ПО, что определяет ожидаемые результаты, а также критерии входа и выхода для данного продукта.

Основные этапы этой методологии могут изменяться, однако обычно они включают следующие:

- этап определения требований – приемочное тестирование относится к этому этапу, его основная задача состоит в оценке готовности системы к финальному использованию;
- этап, на котором происходит высокоуровневое проектирование, или High-Level Design (HDL) – этот этап относится к системному тестированию и включает оценку соблюдения требований к интегрированным системам;
- фаза детального дизайна (Detailed Design) параллельна фазе интеграционного тестирования, во время которой происходит проверка взаимодействий между различными компонентами системы;
- после этапа написания кода начинается другой важный шаг – юнит-тестирование – очень важно убедиться в том, что поведение отдельных частей и компонентов ПО корректно и соответствует требованиям.

Единственным недостатком рассмотренной методологии тестирования является отсутствие готовых решений, которые можно было бы применить, чтобы избавиться от дефектов ПО, обнаруженных на этапе тестирования.

#### **Инкрементная модель**

Данная методология может быть описана, как мультикаскадная модель тестирования ПО [2]. Рабочий процесс разделяется на некоторое количество циклов, каждый из которых также делится на модули. Каждая итерация добавляет определенный функционал к ПО. Инкремент состоит из трех циклов:

- дизайн и разработка;
- тестирование;
- реализация.

В этой модели возможна одновременная разработка разных версий продукта. Например, первая версия может проходить этап тестирования в то время, как вторая версия находится на стадии разработки. Третья версия в то же самое время может проходить этап дизайна. Этот процесс может продолжаться до самого завершения проекта.

Очевидно, что данная методология требует обнаружения максимально возможного количества ошибок в тестируемом ПО настолько быстро, насколько это возможно. Так же, как и фаза реализации, которая требует подтверждения готовности продукта к доставке конечному пользователю. Все эти факторы существенно увеличивают весомость требований к тестированию.

В сравнении с предыдущими методологиями, инкрементная модель имеет несколько важных преимуществ. Она более гибкая, изменение требований ведет к меньшим затратам, а процесс тестирования ПО является более эффективным, поскольку гораздо проще проводить тестирование и дебаггинг за счет использования небольших итераций. Тем не менее, стоит отметить, что общая стоимость все же выше, чем в случае каскадной модели.

#### **Спиральная модель**

Спиральная модель это методология тестирования ПО, которая основана на инкрементном подходе и прототипировании [3].

Она состоит из четырех этапов:

- планирование;

- анализ рисков;
- разработка;
- оценка.

Сразу после того, как первый цикл завершен, начинается второй. Тестирование ПО начинается еще на этапе планирования и длится до стадии оценки. Основным преимуществом спиральной модели является то, что первые результаты тестирования появляются незамедлительно после появления результатов тестов на третьем этапе каждого цикла, что помогает гарантировать корректную оценку качества. Тем не менее, важно помнить о том, что эта модель может быть довольно затратной и не подходит для маленьких проектов.

Несмотря на то, что эта модель является довольно старой, она остается полезной как для тестирования, так и для разработки. Более того, главная цель многих методологий тестирования ПО, включая спиральную модель, изменилась в последнее время. Мы используем их не только для поиска дефектов в приложениях, но также и для выяснения причин, их вызвавших. Такой подход помогает разработчикам работать более эффективно и быстро устранять ошибки.

### **Agile**

Гибкие методологии – это подходы к разработке программного обеспечения, ориентированные на использование итеративной разработки, динамическое формирование требований и обеспечение их реализации в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп, состоящих из специалистов различного профиля [4].

Методология гибкой (Agile) разработки и тестирования ПО может быть описана как набор подходов, ориентированных на использование интерактивной разработки, динамического формирования требований и обеспечения их осуществления как результата постоянного взаимодействия внутри самоорганизующейся рабочей группы. Большинство гибких методологий разработки ПО нацелены на минимизацию рисков посредством разработки в рамках коротких итераций. Одним из главных принципов этой гибкой стратегии является возможность быстрого реагирования на возможные изменения, нежели стремление положиться на долгосрочное планирование.

### **Экстремальное программирование (XP, Extreme Programming)**

XP (экстремальное программирование) – возможность вести разработку в условиях постоянно меняющихся требований. Признаки: игра в планирование, высокая частота релизов, контакт с клиентом, рефакторинг, стандарт выполнения кода, коллективная ответственность.

Экстремальное программирование является одним из примеров гибкой разработки ПО. Отличительной особенностью этой методологии является «парное программирование», ситуация, когда один разработчик работает над кодом, в то время как его коллега постоянно проводит обзор написанного кода. Процесс тестирования ПО является довольно важным, поскольку начинается даже раньше, чем написана первая строка кода. Каждый модуль приложения должен иметь юнит-тест, чтобы большинство ошибок могло быть исправлено на стадии написания кода. Другим отличительным свойством является то, что тест определяет код, а не наоборот. Это значит, что определенная часть кода может быть признана завершенной только в том случае, если все тесты пройдены успешно. В противном случае, код отклоняется.

Главными достоинствами такой методологии являются постоянное тестирование и короткие релизы, что помогает обеспечить высокое качество кода.

### **Scrum**

Scrum – часть методологии Agile, итеративный инкрементный фреймворк, созданный для управления процессом разработки ПО [5]. Согласно принципам Scrum, команда тестировщиков должна участвовать в следующих этапах:

- участие в Scrum планировании;
- поддержка в юнит-тестировании;
- тестирование пользовательских историй;
- сотрудничество с заказчиком и владельцем продукта для определения критериев приемлемости;
- предоставление автоматического тестирования.

Более того, участники QA-отдела должны присутствовать на всех ежедневных собраниях, как и другие члены команды, чтобы обсудить, что было протестировано и сделано вчера, что будет протестировано сегодня, а также общий прогресс тестирования программного обеспечения.

В то же время принципы Agile методологии в Scrum приводят к появлению специфических особенностей:

- оценка усилий, необходимых для каждой пользовательской истории является обязательной;
- тестировщик должен быть внимательным к требованиям, поскольку они могут постоянно изменяться;
- риск регрессии возрастает вместе с частыми изменениями в коде;
- одновременность планирования и выполнения тестов;
- недопонимание между членами команды в случае если требования заказчика не до конца ясны.

LD – бережливая разработка программного обеспечения – еще одно ответвление гибкой методологии, предполагающее сохранение высокого морально-функционального состояния разработчиков.

Гибкие технологии, в особенности XP, постепенно заняли главенствующее место в разработке программного обеспечения. Современные международные стандарты процессов жизненного цикла систем и программного обеспечения учитывают эти методы, и их можно применять для таких проектов. Соответственно инструменты ведения проектов, такие, например, как Rational Rose также приспособлены для них.

Гибкие технологии подходят для всех проектов, где нужна высокая гибкость и своевременный учет изменений. По крайней мере, их отдельные принципы позволяют существенно повысить успешность изменчивых проектов.

### **Выводы**

В заключение важно отметить, что сегодня практика использования той или иной методологии тестирования ПО подразумевает мультиверсальный подход. Иными словами, не стоит рассчитывать на то, что какая-то одна методология окажется подходящей для всех типов проектов. Выбор одной из них зависит от большого числа аспектов, таких как тип проекта, требования заказчика, поставленные сроки, а также многих других. С точки зрения тестирования ПО, для некоторых методологий характерно приступать к тестированию на ранних этапах разработки, в то время как при работе с другими принято ожидать до тех пор, пока система не готова полностью.

### **Литература**

1. Грабельников, В.А. Методологии разработки программного обеспечения // Цифровой регион: опыт, компетенции, проекты: сборник статей Международной научно-практической конференции. – Брянск: 2018. с.125-129.
2. Мирошниченко, Е. А. Технологии программирования: учебное пособие / Е. А. Мирошниченко. – 2-е изд., испр. и доп. – Томск: Изд-во Томского политехнического университета, 2008. – 128 с.

3. Евдокимов, И.В. Адаптация стандартов программных средств к проектам в области информационных технологий // Труды Братского государственного университета. Серия: Экономика и управление. 2010. Т. 2. с.97-101.
4. Вольфсон, Б. Гибкие методологии разработки. – М.: Эксмо, 2013. – 112 с.
5. Кон, М. Scrum: гибкая разработка ПО. – М.: «Вильямс», 2011. – 576 с.

УДК 681.518.5

## РАЗРАБОТКА СИСТЕМ ДИСПЕТЧЕРИЗАЦИИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКОЙ ЛИНИЕЙ

**А.В. Дробитько, Ю.К. Орлов**

Донецкий национальный технический университет  
кафедра автоматизированных систем управления  
E-mail: [alexandra.drobitko@gmail.com](mailto:alexandra.drobitko@gmail.com)

### **Аннотация:**

**Дробитько А.В., Орлов Ю.К. Разработка системы диспетчеризации автоматизированной системы управления технологической линией.** С учетом исходных условий, дефицита времени и возмущающих факторов на основе теоретико-множественных представлений выполнена декомпозиция функционирования, входных воздействий и процесса принятия решения оператором АСУ. Исследованы особенности выбора режима функционирования и формирования управляющих воздействий. Представлены модели человека-оператора и направления повышения надежности его функционирования в АСУ. Для математического описания состояния оператора в реальном масштабе времени применен векторно-множественный анализ с применением элементов нечеткой логики.

### **Annotation:**

**Drobitko A.V., Orlov J.C. Development of dispatching system of the automated control system of production line.** Decomposition of functioning, entrance influences, and process of decision-making by the operator of ACS is executed. Initial conditions, deficiency of time and revolting factors of steering are at the same time considered. Decomposition is executed with application of set-theoretic representations. Features of a choice of a mode of functioning and formation of steering influences are investigated. Models of the person operator are presented. The directions of increase of reliability of its functioning in ACS are considered. For the mathematical description of a status of the operator in real time the vectorial and multiple analysis using elements of fuzzy logic is applied.

### **Общая постановка проблемы**

В данное время все чаще поднимается проблема энергосбережения и надлежащего исполнения технологических процессов на линии. Комплексный подход в оценки ситуации требует тщательный процесс сбора и обработки данных [1].

Обзор литературы и интернет источников по данной тематике показал актуальность данной темы на сегодняшний день. Введение системы диспетчеризации позволяет обеспечить не только простейшие функции мониторинга, но и полноценное общее управление (получение реальных данных от датчиков, контроль работы всей линии и её отдельных секторов, построение графиков, отслеживание и архивация аварийных сигналов),

а также минимизировать «человеческий фактор» и влияние факторов и условий на оператора [2].

### Цель статьи

Выполнить системный анализ методов и инструментальных средств современных технологий контроля и прогностической оценки состояний системы контроллеры-оператор.

Для достижения поставленной цели необходимо осуществить анализ методов и инструментальных средств, декомпозицию функционирования оператора АСУ и входных воздействий. Так же построить модель датчики-оператор как динамического звена.

### Системный анализ системы диспетчеризации АСУ

В результате анализа было выявлено, что многими исследователями предложена модель «Пирамида АСУ промышленного предприятия» представленная на рисунке 1.



Рисунок 1 – «Пирамидальная» модель АСУ промышленного предприятия

В общем случае, основным назначением АСД является обеспечения высокой координации действий подразделений предприятия на оперативном уровне. Однако решение данной проблемы оказывается связанным с необходимостью интеграции всех функций оперативного управления, и, прежде всего, диспетчеризации. В данном случае оператор выступает как динамическое звено.

### Декомпозиция функционирования оператора АСУ

Оператор в АСУ выполняет множественные функции управления в нестационарных условиях (рис. 2).

Если оценивать оператора в виде некоторой эквивалентной информационной интеллектуально-физиологической системы, то он может быть представлен двумя уровнями: воспринимающей и исполнительской частями. На входе воспринимающей части этой системы воздействуют исходные условия, дефицит времени, внешние и внутренние возмущающие факторы, повышенная степень ответственности за результаты действий, самоконтроль и взаимоконтроль (при управлении в составе экипажа, команды) за решениями, действиями и результатами [4].

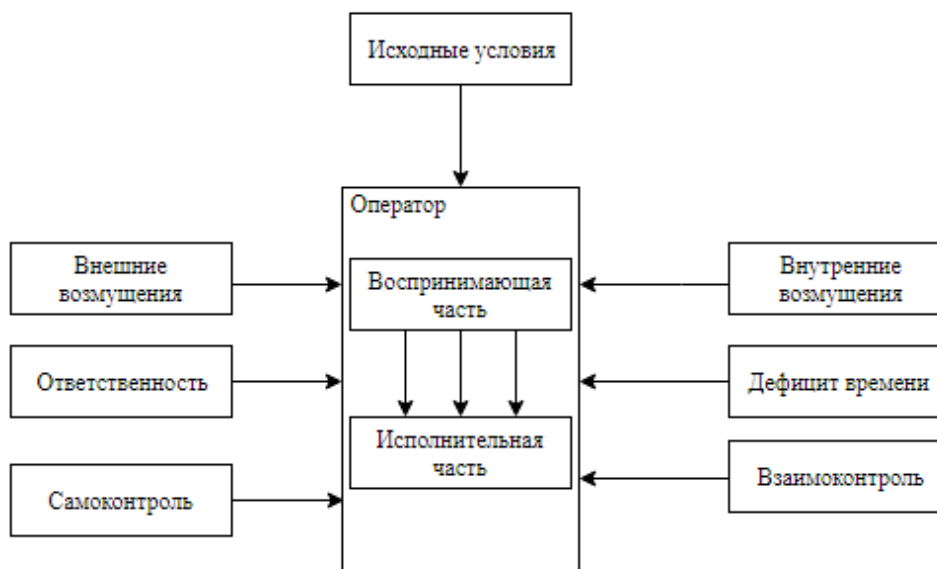


Рисунок 2 – Декомпозиция воздействий на оператора

На выходе формирующей части образуется ряд управленческих решений: по выбору режима функционирования АСУ, по формированию действий, направленных на управление объектом, выполнимости и исполнению отдельных разовых команд верхнего иерархического уровня при вмешательстве в оперативную деятельность оператора.

#### Декомпозиция входных воздействий

Исходные условия в работе оператора интерпретируются как множество требующих решения условий:

$$G(t) = G_{л}(t) \cup G_{ф}(t) \cup G_{к}(t),$$

где  $G_{л}(t)$  – множество условий (задач), требующих логического решения,  $G_{ф}(t)$  – множество формальных функциональных задач, требующих применения известного решения,  $G_{к}(t)$  – множество поступающих приоритетных и беспriorитетных разовых команд.

Чем больше функций управления в АСУ передается оператору, тем больше размерность логико-функционального множества  $B(t)$ :

$$B(t) = G_{л}(t) \cup G_{ф}(t); B(t) \subset G(t),$$

т. е. множества задач, которые требуют одновременно и логического, и формального решения.

Особую роль играют команды верхнего иерархического уровня управления, объединенные в множество  $G_{к}(t)$ . Если команды передаются оператору в реальном масштабе времени и с более глубоким анализом обстановки, чем это может выполнить оператор, их безусловное выполнение имеет позитивный результат. В противном случае может быть нанесен ущерб. Анализ команд для оператора во многих случаях сложнее, чем анализ

реальной обстановки по управлению объектом. Так что команды могут быть в некоторых случаях отнесены к возмущениям [7].

К исходным условиям принятия решения относится множество параметров  $Y_{oy}$ , характеризующих движение ОУ, поступающее на воспринимающую часть в форме векторов:

- $Y_{оп.ос}(Y_{oy})$ , образованного по результатам измерений датчиками обратной связи;
- $Y_{инт}(Y_{oy})$ , образованного по ощущениям и интуитивным заключениям самого оператора.

Дефицит времени обусловлен допустимыми временными интервалами, которые должны быть соблюдены при обработке условий управления, принятии и исполнении решений. Оценка дефицита времени – результат сравнения векторов:

$$\Delta D_{\tau}(t) = D_p(t) - D_{ж}(t),$$

где  $D_p(t)$  и  $D_{ж}(t)$  – соответственно векторы располагаемых и желаемых интервалов на обработку исходных условий, на принятие и исполнение управленческих решений.

Возмущающие факторы. Возможности человека по обработке информации и принятию решения не безграничны, для этого и существуют датчики. Например, он неспособен длительное время безошибочно выполнять монотонные действия. А в условиях утомления они не поддаются достоверному прогнозу. В большой степени поведение человека – следствие его субъективных психофизиологических особенностей. Возмущения в общем случае нестационарные и представляют собой объединение множеств:

$$F(t) = F_{оп\ вш}(t) \cup F_{оп\ вн}(t),$$

где  $F_{оп\ вш}(t)$  и  $F_{оп\ вн}(t)$  соответственно множество внешних и внутренних факторов.

#### **Декомпозиция принятия решения**

Цель функционирования оператора в АСУ – принятие управленческих решений. Соответственно, глубина декомпозиции ограничивается выходом за пределы цели исследования большой системы, а именно системы функционирования оператора в АСУ.

Выбор режима функционирования АСУ является следствием обработки оператором информации о состоянии объекта управления, системы автоматического управления и самоконтроля организма. Решению сопутствуют сведения о необходимости его принятия. Поэтому это тоже множество функций управления режимами ( $U_{реж}(t)$ ), которое включает всю эту информацию в виде его элементов.

Формирование действий, направленных на управление объектом, оценивается в зависимости от выбранного режима. Это соответственно множества управления техническими объектами ( $U_a(t)$ ,  $U_{дир}(t)$ ,  $U_{руч}(t)$ ) при автоматическом, директорном и ручном управлении [7].

Выполнимость и исполнение отдельных разовых команд верхнего иерархического уровня при вмешательстве в оперативную деятельность оператора оценивается множеством  $I(t)$ .

Рассмотренные особенности функционирования оператора АСУ показывают, что на функционирование оператора влияет множество дестабилизирующих случайных и детерминированных факторов [5].

#### **Модель контроллера-оператора как динамического звена.**

Существуют различные определения надёжности оператора. Наиболее приспособлен оператор к выполнению функций усилительного звена, полоса пропускания которого не превышает 0,5 Гц. Тренированный оператор может быть представлен стационарным динамическим звеном с передаточной функцией [1]:

$$W_o(p) = \frac{k_o(T_{o1}p+1)}{(T_{o2}p+1)(T_{o3}p+1)} e^{-T_{o4}p}, \quad (1)$$

где  $k_0 = 40 \dots 100$  – коэффициент передачи, описывающий мышечные усилия оператора во время управления подвижным объектом.

$$W_{01}(p) = T_{01}p + 1,$$

где  $T_{01} = 0.25-2.5$  с – передаточная функция форсирующего звена, характеризующего способность оператора предсказывать изменения сигналов управления.

$$W_{02}(p) = 1/(T_{02}p + 1),$$

где  $T_{02} \approx 0.1$  с, – передаточная функция аperiodического звена, описывающая инерционность мыслительных процессов оператора.

$$W_{03}(p) = 1/(T_{03}p + 1),$$

где  $T_{03} = 0.6-2$  с, – передаточная функция аperiodического звена, описывающая инерционность действий оператора при фильтрации событий.

$$W_{04}(p) = \exp(-T_{04}p),$$

где  $T_{04} = 0,13 \dots 0,2$  с, – передаточная функция звена с запаздыванием, описывающая запаздывание действий оператора, относительно полученных команд.

$p$  – комплексная переменная в преобразовании Лапласа.

Повышение надежности и продуктивности оператора автоматических систем розлива можно реализовать с помощью энергосберегающих преобразователей частоты, объединенных в общую систему на основе информационной шины CC-Link [4].

Высокоскоростной подачей бутылок в разливочную установку можно управлять с помощью контроллера System Q в сочетании с сетью CC-Link, обеспечивающей скорость передачи данных 10 Мбит/с и быстрые программные циклы в миллисекундном диапазоне. В номенклатуре периферийных модулей контроллеров System Q имеются также модули сети ETHERNET, позволяющие подключить контроллер к верхнему уровню управления и системам администрирования. Таким образом, производственные данные можно передавать в автоматизированные системы управления предприятием (ERP) и управления производством (MES) в реальном масштабе времени. Это позволяет оператору линии непосредственно оперировать реальными, мгновенными производственными данными, например, о производительности розлива, а не довольствоваться статистикой [1].

### Выводы

Существует объективная необходимость разработки методов контроля состояния оператора в реальном масштабе времени и технических средств реализации этих методов для достижения высокой надежности функционирования человеко-машинных систем.

Целесообразно для контроля состояния оператора АСУ в реальном масштабе времени вести обработку одного или нескольких сигналов организма, которые являются интегральным отражением функционирования всех критических систем организма. Для обработки сигналов организма оператора следует использовать цифровые методы обработки на основе теорий нечетких множеств и статистической обработки с реализацией на микропроцессоре.

### Литература

1. Балашов, В.Е. Техника и технология производства пива и безалкогольных напитков./ В.Е. Балашов, В.В Рудольф –М.: Легкая и пищевая промышленность, 1981. –248с.
2. Ермолаева, Г.А. Технология и оборудование производства пива и безалкогольных напитков: Учеб. для нач. проф. образования. / Г.А. Ермолаева, Р.А. Колчева – М.: ИРПО; Изд. центр «Академия», 2000. – 416 с.
3. Перегудов, Ф.И. Основы системного анализа. 3-е изд., исп. и доп. / Ф.И. Перегудов, Ф.П Тарасенко Томск: НТЛ, 2000. – 352 с.
4. Программируемые контроллеры [электронный ресурс] // Stela: [сайт]. [1992-2015]. URL:[http://www.ste.ru/siemens/pdf/rus/S7-1200\\_2014.pdf](http://www.ste.ru/siemens/pdf/rus/S7-1200_2014.pdf)



5. Федоров, Ю.Н. Справочник инженера по АСУТП. Проектирование и разработка. / Ю.Н. Федоров – М.: Инфра-Инженерия, 2008. – 928с.

6. Максимов, М.В. Радиоэлектронные системы самонаведения. / М.В. Максимов, Г.И. Горгонов – М. : Радио и связь, 1982. – 304с.

7. Конопкин, О.А. Инженерная психология и проблема надежности машиниста. / О.А. Конопкин, Л.С. Нерсенян – М.: Транспорт, 1976. – 239 с.

УДК 004.932.2

## РАЗРАБОТКА СИСТЕМЫ ОЦЕНКИ РЕЗУЛЬТАТОВ СТРЕЛКОВОЙ ТРЕНИРОВКИ

**Тарасов Н.Ю., Тарасова И.А.**

Донецкий национальный технический университет  
кафедра искусственного интеллекта и системного анализа

E-mail: [sonic1837@rambler.ru](mailto:sonic1837@rambler.ru)

### **Аннотация:**

**Тарасов Н.Ю., Тарасова И.А. Разработка системы оценки результатов стрелковой тренировки.** Проведён анализ существующих методов обработки изображений. Реализованы методы обработки и анализа изображения для системы оценки результатов стрелковой тренировки, а также пользовательский интерфейс для работы с данными.

### **Annotation:**

**Tarasov N.Y., Tarasova I.A. Development of a system for evaluating the results of shooting training.** The analysis of existing methods of image processing. Implemented methods of image processing and analysis for the system for evaluating the results of a rifle training, as well as a user interface for working with data.

### **Общая постановка проблемы**

Для хороших умений в стрельбе необходима качественная стрелковая тренировка, которая очень сильно влияет на развитие мастерства спортсмена-биатлониста или стрелка-любителя в этом деле. Стрелковые тренировки могут проводиться и стрельбой по доскам с нарисованной по ней мишенью, однако это негативно скажется на качестве тренировки. Для качественной стрелковой тренировки необходимо соответствующие хорошие методики тренировок и современное оборудование. Существует много методик и оборудования для стрелковых тренировок и поэтому важно также иметь программное обеспечение соответствующего уровня для повышения качества стрелковой тренировки. Таким программным обеспечением может стать система оценки результатов стрелковой тренировки, которая позволит повысить качество стрелковой тренировки спортсменов олимпийских дисциплин и стрелков-любителей.

**Целью статьи** является улучшение качества стрелковой тренировки за счёт моментального машинного исследования поражённых целей, благодаря подачи в систему изображения мишени, обнаружения попаданий, анализа и выдачи результата с рекомендациями и советами по улучшению стрельбы.

Для достижения поставленной цели необходимо выполнить ряд задач:

- проанализировать методы обработки и анализа изображения;
- реализовать методы обработки и анализа изображения для системы оценки результатов стрелковой тренировки;

- реализовать пользовательский интерфейс для работы с данными.

#### Анализ методов обработки и анализа изображения

Задачи обработки и анализа изображений основаны на корреляционной обработке, фильтрации изображений, на использовании методов нечёткой логики, искусственных нейронных сетей и так далее [2]. Однако существует проблема, которая состоит в том, что результаты решения задач этими методами не идентичны, что говорит о том, что методы имеют разную эффективность.

Для решения рассматриваемой задачи могут быть использованы следующие методы:

- метод корреляционной обработки;
- метод согласованной фильтрации;
- метод поиска центра фрагмента;
- метод моментов;
- метод детектора границ.

На рисунке 1 и 2 представлены результаты модельного эксперимента при разрешении исходного изображения 800x600 пикселей и отметки цели  $s_0(x,y)$  в виде окружности. В эксперименте формируется изображение, содержащее искомый фрагмент и аддитивный белый гауссов шум. Затем производится оценивание координат искомого фрагмента всеми рассматриваемыми методами. На рисунках приняты следующие обозначения: 1 – корреляционный приемник, 2 – процедура поиска центра фрагмента, 3 – фильтровый приемник, 4 – оценка координат центра тяжести, 5 – детектор границ. Погрешности оценивания выражены в миллиметрах.

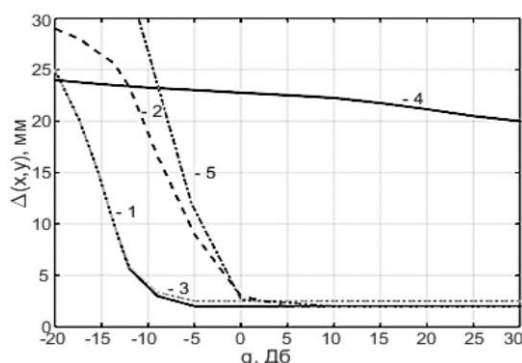


Рисунок 1 – Зависимости погрешностей измерения координат от отношения сигнал/шум

Также осуществлена оценка характеристик обнаружения искомого объекта на изображениях в виде зависимости вероятности правильного обнаружения от отношения сигнал/шум.

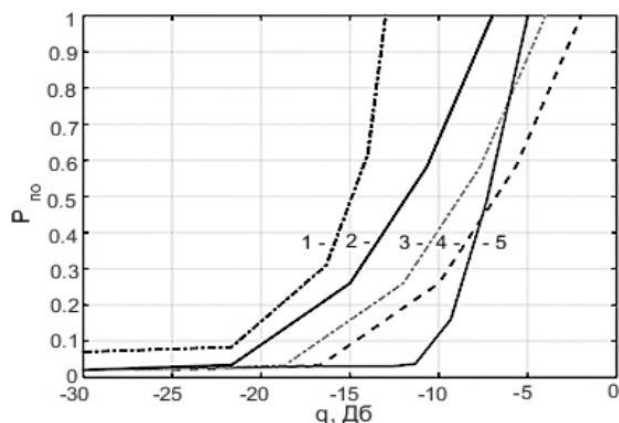


Рисунок 2 – Кривые вероятностей правильного обнаружения в зависимости от отношения сигнал/шум

Как видно, все методы плохо справляются с обнаружением искомого объекта при малом отношении  $q$  сигнал/шум. Это создаёт необходимость в предварительной фильтрации входящего видеопотока и создания условий, при которых выбранный метод обеспечит приемлемую вероятность обнаружения, исходя из требований к работе мультимедийных стрелковых тренажёров [1]. Оценки быстродействия рассмотренных методов приведены в таблице 1.

Таблица 1 – Оценки быстродействия рассмотренных методов

Разрешение	Время выполнения, с				
	Корреляционная обработка (1)	Поиск центра фрагмента (2)	Фильтровый приёмник (3)	Оценка центра тяжести (4)	Детекторы границ (5)
64x64	0.0014	0.0823	0.001	0.0038	0.0217
128x128	0.002	0.0124	0.0028	0.0164	0.0017
256x256	0.0048	0.0236	0.0114	0.0822	0.0026
512x512	0.0166	0.0922	0.0519	0.2746	0.008
768x768	0.0418	0.1532	0.1237	0.6401	0.0193
1024x1024	0.0608	0.4587	0.2219	1.0338	0.0296
1280x1280	0.1138	1.2347	0.3533	1.8869	0.0499
1600x1600	0.1656	1.5696	0.5581	3.0465	0.0848
2048x2048	0.2574	1.9477	0.9026	4.4834	0.1407

Результаты существующих экспериментов позволяют сделать заключение, что такие методы как: поиск центра фрагмента, фильтровый приёмник и оценка координат центра тяжести непригодны для работы в режиме реального времени. Для обработки видеопотока в реальном времени рекомендуется использовать методы корреляционной обработки или детектирования границ.

Так как для обнаружения секторов на мишени и попаданий на них будут искаться фрагменты изображений в форме круга, то в данном случае будет правильным использовать преобразование Хафа для нахождения окружностей на изображении.

### Реализация метода детектирования границ Канни и преобразования Хафа

Метод детектирования границ основывается на алгоритмах, которые выделяют точки цифрового изображения, где резко изменяется яркость или есть другие виды неоднородностей. Основной целью обнаружения резких изменений яркости изображения является фиксация важных событий и изменений. Они могут отражать различные предположения о модели формирования изображения. В данном методе более верным решением будет использовать алгоритм детектора границ Канни [6]. Алгоритм состоит из 4 шагов:

Шаг 1. Преобразование изображения в градации серого.

Шаг 2. Сглаживание. Размытие изображения для удаления шума.

Шаг 3. Подавление не-максимумов. Только локальные максимумы отмечаются как границы.

Шаг 4. Двойная пороговая фильтрация. Потенциальные границы определяются порогом.

Шаг 5. Получение границ изображения.

На вход алгоритма подаётся изображение. После выполнения шагов 1 и 2 изображение преобразовывается в градации серого и размывается для удаления шума. На выходе алгоритма получается изображение (см.рис.3), на котором присутствуют только границы изображения. Это изображение проходит преобразование Хафа, в котором находятся окружности на изображении. После преобразования Хафа на выходе получается изображение, представленное на рисунке 4. На рисунке 4 можно увидеть зелёные окружности, которые формируются за счёт преобразования Хафа для поиска окружностей на изображении.

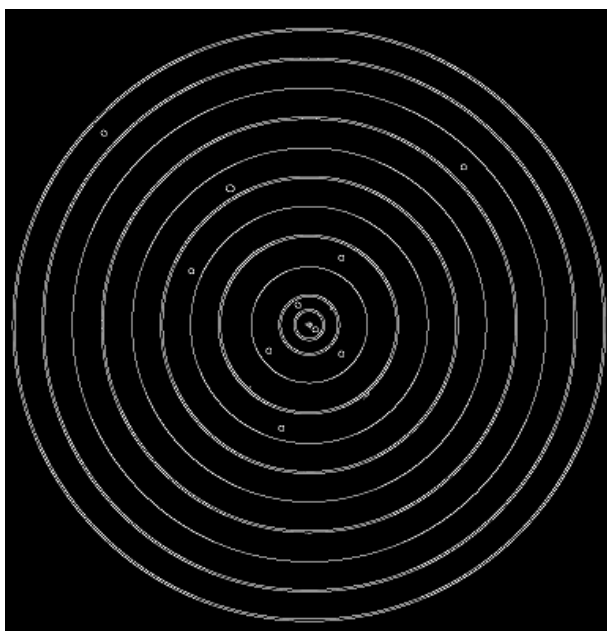


Рисунок 3 – Изображение, получаемое на выходе алгоритма

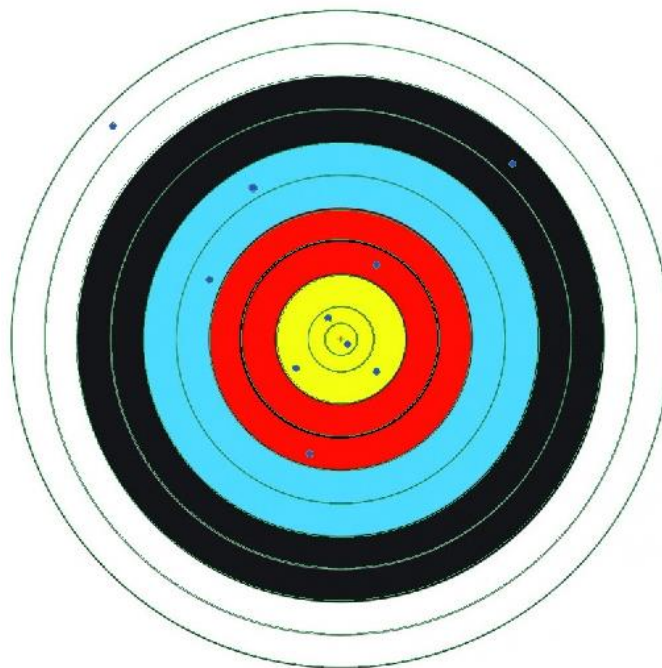


Рисунок 4 – Изображение с найденными окружностями

На рисунке 4 выделяется каждый сектор попадания и каждое попадание. Каждая выделенная окружность имеет три параметра, которые могут использоваться для вычислений и последующего анализа, а именно координата  $x$ , координата  $y$  и радиус окружности.

#### **Программная реализация системы оценки результатов стрелковой тренировки**

Результатом работы системы оценки результатов стрелковой тренировки, является оценка попадания в сектора мишени. На рисунке 5 представлен результат анализа изображения с границами и дана оценка стрелковой тренировки.

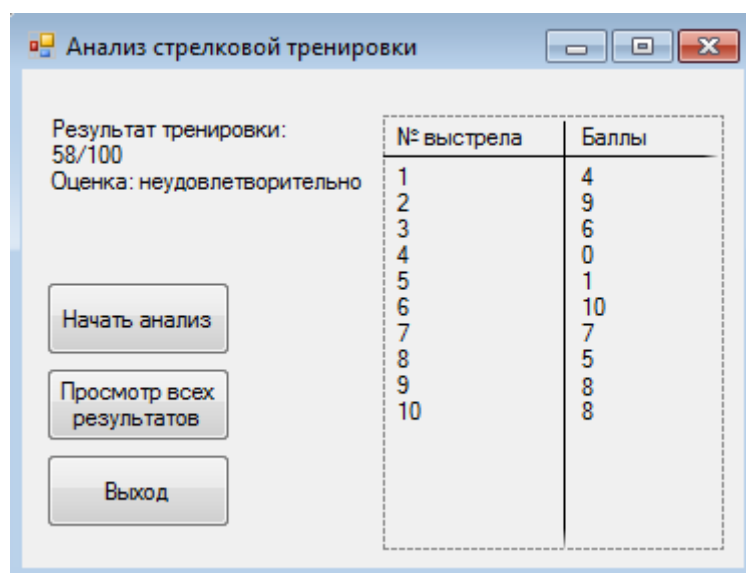


Рисунок 5 – Окно с результатом анализа стрелковой тренировки

После нажатия на кнопку «Начать анализ» высвечивается окно с изображением мишени, на которой присутствуют попадания и выделены окружности зоны попаданий,

после чего данные о попаданиях анализируются, и отображаются результатом анализа изображения, который представлен на рисунке 5.

### **Выводы**

Для разработки системы оценки результатов стрелковой тренировки потребовалось провести аналитический обзор, который показал широкий диапазон научных исследований, патентов и внедрённых комплексов. Данный анализ показал, что для улучшения качества стрелковой тренировки необходимо разработать систему, которая будет определять местоположение точки попадания в мишень. По результатам исследований, наиболее быстродействующими и эффективными являются методы корреляционной обработки и детектирования границ. Для обнаружения секторов и попаданий на мишени было применено преобразование Хафа, которая обнаруживает окружности на изображениях. Разработана система оценки результатов стрелковой тренировки, которая позволяет улучшить качество стрелковой тренировки.

### **Литература**

1. Колдунов С.А. Стрелковые тренажеры: вместо пули – лазерный луч / Колдунов С.А – М.: ООО «Синус», 1998. – 128 с.
2. Форсайт Д.А. Компьютерное зрение. Современный подход / Д.А. Форсайт, Ж. Понс. – М.: Вильямс, 2004. – 928 с.
3. Павлидис Т. Алгоритмы машинной графики и обработки изображений / Павлидис Т. – М.: Техносфера, 2003. – 395 с.
4. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М.: Техносфера, 2005. – 1072 с.
5. Ревенко, В.А. Экспериментальное обоснование использования технических средств в процессе обучения огневой подготовке личного состава учебных заведений МВД СССР: автореф. дис. ... канд. пед. наук / Ревенко В.А – Харьков, 1990. – 174 с.
6. Zhuowen Tu et. Al. Parsing: Unifying Segmentation, Detection and Recognition / Zhuowen Tu et. Al. // Int. Journal of Computer Vision. – 2005. – 140 с.

УДК 517.9

## МОДЕЛИРОВАНИЕ И ИССЛЕДОВАНИЕ ЗАДАЧИ СТЕФАНА

А.С. Миненко, А.В. Лёвкина, Е.В. Радевич

Донецкий национальный технический университет, г. Донецк  
кафедра искусственного интеллекта и системного анализа

E-mail: [sam\\_dntu@mail.ru](mailto:sam_dntu@mail.ru), [a.levkina13@gmail.com](mailto:a.levkina13@gmail.com),  
[radevich\\_katerina@mail.ru](mailto:radevich_katerina@mail.ru)

### Аннотация

*Миненко А.С., Лёвкина А.В., Радевич Е.В. Моделирование и исследование задачи Стефана. В работе рассмотрен процесс кристаллизации металла, который сопровождается конвективными перемешиваниями в жидкой фазе. Дана постановка задачи, в которой конвекция будет вызвана наличием заданного вихря.*

### Abstract

*Minenko A.S., Levkina A.V., Radevich E.V. Modeling and research of Stefan's problem. The paper considers the process of crystallization of the metal, which is accompanied by convective mixing in the liquid phase. Given the problem in which convection will be caused by the presence of a given vortex.*

### Введение

В работе, с помощью задачи Стефана, рассмотрен процесс кристаллизации. Рассмотрим задачу, в которой конвекция происходит из-за наличия заданного вихря. При выполнении данного процесса кристаллизующая система включает в себя две фазы: твердую и жидкую. Также необходимо отметить, что интенсивность конвекции кристаллизации металла главным образом зависит от объема пространства и температуры. Отметим, что основная цель исследования состоит в приближенном анализе свободной границы, которая зависит от интенсивности вихря.

### Постановка задачи

Рассмотрим задачу, в которой конвекция происходит из-за наличия заданного вихря. При выполнении данного процесса кристаллизующая система включает в себя две фазы: твердую и жидкую. Также необходимо отметить, что интенсивность конвекции кристаллизации металла главным образом зависит от объема пространства и температуры. Отметим, что основная цель исследования состоит в приближенном анализе свободной границы, которая зависит от интенсивности вихря [5].

Рассмотрим задачу, представленную в статье [4], с заданными граничными условиями для жидкой и твердой фаз. Добавим параметр температуры в жидкой фазе и получим условие конвективного теплопереноса:

$$\lambda_+ \left( \frac{\partial^2 u^+}{\partial x^2} + \frac{\partial^2 u^+}{\partial y^2} \right) - \frac{\partial \psi \partial u^+}{\partial y \partial x} + \frac{\partial \psi \partial u^+}{\partial x \partial y} = 0, (x, y) \in D_\gamma^+, \lambda_+ = \text{const} > 0. \quad (1)$$

Повторим аналогичные исследования с температурой  $u^-$  для твердой фазы. Следовательно, на свободной границе уравнение будет иметь следующий вид:

$$u^-(x, y) = 1, (x, y) \in \gamma. \quad (2)$$

Представленная выше задача была бы корректна, если кривая  $\gamma$  была бы заданна. В

силу того, что  $\gamma$  подлежит определению, на ней задается еще одно условие, а, именно, закон сохранения энергии:

$$|\nabla u^-|^2 - k^2 |\nabla u^+|^2 = 0, (x, y) \in \gamma, k = const, 0 \leq k \leq 1. \quad (3)$$

Данная задача нелинейная и «основное» неизвестное – это граница  $\gamma$ . Разрешимость подобного класса задач изложена в [1].

Предположим, что неизвестные рассматриваемой задачи можно представить в виде степенного ряда, тогда получим следующее выражение:

$$\psi(x, y; \mu) = \sum_{k=0}^{\infty} \mu^k \psi_k(x, y), \quad (4)$$

$$\mu^+(x, y; \mu) = \sum_{k=0}^{\infty} \mu^k \mu_k^+(x, y), \quad (5)$$

$$\mu^-(x, y; \mu) = \sum_{k=0}^{\infty} \mu^k \mu_k^-(x, y), \quad (6)$$

тогда, будем считать, что свободная граница  $\gamma$  допускает явное представление:

$$y = y(x; \mu), \quad (7)$$

причем

$$y(x, \mu) = \sum_{k=0}^{\infty} \mu^k y_k(x), -1 \leq x \leq 1. \quad (8)$$

Подставив эти разложения в задачу, рассмотренную выше, и приравняв их члены при степени  $\mu$ , мы получим бесконечное количество задач. Для начала, выпишем то нулевое приближение, которое соответствует значению  $\mu$  в степени 0. Тогда, функция  $\psi_0(x, y)$ , которая указанная в статье [6], является гармоничной. Так как она удовлетворяет нулевым граничным условиям Дирихле, то  $\psi_0(x, y) \equiv 0$  в  $\bar{D}_\gamma^+$ .

Выпишем условия, определяющие  $\mu_0^\pm$

$$\frac{\partial^2 u_0^\pm}{\partial x^2} + \frac{\partial^2 u_0^\pm}{\partial y^2} = 0, (x, y) \in D_{\gamma_0}^\pm. \quad (9)$$

$$\mu_0^+(x, 0) = v, -1 \leq x \leq 1, \quad (10)$$

$$u_{0x}^\pm(x, y) \pm \omega_0^\pm u_{0x}^\pm(x, y) = 0, x = \pm 1, (x, y) \in \partial D_{\gamma_0}^\pm, \quad (11)$$

$$\mu_0^\pm(x, y) = 1, (x, y) \in \gamma_0, \quad (12)$$

$$\mu_0^-(x, H) = 0, -1 \leq x \leq 1, \quad (13)$$

$$|\nabla u_0^-|^2 - k^2 |\nabla u_0^+|^2 = 0, (x, y) \in \gamma_0. \quad (14)$$

Так как задача (9) – (14) была рассмотрена в статьях [5,6], то мы можем сделать вывод, что задача имеет и притом единственное классическое решение в классе функций  $\mu_{0y}^\pm > 0$  и  $\mu_{0y}^- > 0$  в  $D_{\gamma_0}^+$  и  $D_{\gamma_0}^-$  соответственно. При этом граница  $\gamma_0$  является аналитической



кривой, монотонно возрастающей в правой половине, а функции  $\mu_0^\pm(x, y)$ ,  $\mu_0^-(x, y)$  непрерывны в  $\bar{D}_{\gamma_0}^+$  и  $\bar{D}_{\gamma_0}^-$  соответственно и непрерывно дифференцируемы всюду. Исключение составляют только угловые точки.

Итак, частный случай этой задачи выглядит следующим образом:

$$k = 1, \omega_0^+ = \omega_0^- = \omega_0. \quad (15)$$

При этом первое условие всегда выполнимо, если ввести

$$\tilde{u}^\pm = \begin{cases} ku^+(x, y), & (x, y) \in D_\gamma^+ \\ u^-(x, y) + k - 1, & (x, y) \in D_\gamma^-, \end{cases}$$

приводящую задачу (9) – (14) к случаю  $k = 1$ . Тогда на  $\gamma_0$  будут выполняться два условия  $u_0^+ = u_0^- = 1$ ,  $|\nabla u_0^+| = |\nabla u_0^-|$ .

А это значит, что задача (9) – (14) – это обычная задача о распределении температуры в области  $D$  без фазовых превращений вещества. Следовательно, мы можем построить функцию  $u_0(x, y)$  по формуле, указанной ниже

$$u_0(x, y) = \begin{cases} u_0^+(x, y), & (x, y) \in \bar{D}_{\gamma_0}^+ \\ u_0^-(x, y), & (x, y) \in \bar{D}_{\gamma_0}^-, \end{cases} \quad (16)$$

данная формула является решением задачи:

$$\begin{aligned} \Delta u_0 = 0, (x, y) \in D; u_0(x, y) = v, 0 \leq x \leq 1; u_{0,x}(0, y) = 0, \\ H \leq y \leq 0; \\ u_0(x, H) = 0, 0 \leq x \leq 1; u_{0,x}(1, y) + \omega_0 u_0(1, y) = 0, \\ H \leq y \leq 0. \end{aligned} \quad (17)$$

Следует отметить, что функция  $u_0(x, y)$  может быть эффективно найдена, к примеру, при помощи метода Фурье. Относительно функции  $u_0(x, y)$  можно заключить, что  $u_{0,y}(x, y) > 0$  в  $D$  [3]. Следовательно, уравнение  $u_0(x, y) - 1 = 0$ ,  $(x, y) \in D$  всегда разрешимо только в виде функции  $y = y_0(x)$ , которая задает кривую  $\gamma_0$ , т.е.  $\gamma_0 : y = y_0(x)$ ,  $-1 \leq x \leq 1$ .

### Первое приближение.

Выпишем сейчас ту краевую задачу, соответствующую множителю  $\mu$  в первой степени. Из условий и из разложений (4) – (8) для функций  $\psi_1(x, y)$  и  $u_1^\pm(x, y)$  вытекает задача

$$\psi_{1,xx} + \psi_{1,yy} = 1, (x, y) \in D_{\gamma_0}^+, \quad (18)$$

$$\psi_1(x, y) = 0, (x, y) \in \partial D_{\gamma_0}^+, \quad (19)$$

$$\lambda_+(u_{1,xx}^+ + u_{1,yy}^+) - \psi_{1,y} u_{0,x}^+ + \psi_{1,x} u_{0,y}^+ = 0, (x, y) \in D_{\gamma_0}^+, \quad (20)$$

$$u_1^+(x, 0) = 0, -1 \leq x \leq 1, \quad (21)$$

$$u_{1,x}^\pm \pm \omega_0^\pm u_1^\pm = 0, x = \pm 1, (x, y) \in \partial D_{\gamma_0}^\pm, \quad (22)$$

$$u_{0,y}^{\pm} y_1(x) + u_1^{\pm} \Big|_{\gamma_0} = 0, \quad (23)$$

$$u_{1,xx}^{-} + u_{1,yy}^{-} = 0, (x, y) \in D_{\gamma_0}^{-}, \quad (24)$$

$$u_1^{-}(x, H) = 0, -1 \leq x \leq 1. \quad (25)$$

Кроме того, на  $\gamma_0$  должно выполняться условие

$$y_1(x) \left[ (u_{0,x}^{-} u_{0,yy}^{-} + u_{0,y}^{-} u_{0,xy}^{-}) - k^2 (u_{0,x}^{+} u_{0,xy}^{+} + u_{0,y}^{+} u_{0,yy}^{+}) \right] + [u_{0,x}^{-} u_{1,x}^{-} + u_{0,y}^{-} u_{1,y}^{-}] - k^2 [u_{0,x}^{+} u_{1,x}^{+} + u_{0,y}^{+} u_{1,y}^{+}] = 0, \quad (26)$$
$$(x, y) \in \gamma_0.$$

Получившееся первое приближение имеет следующие характерные черты. Во-первых, эта задача линейна, во-вторых, ее нужно решать в известной области, соответствующей нулевому приближению. После того, когда функции  $u_0^{\pm}(x, y)$  и  $\psi_1(x, y)$  определены в областях  $D_{\gamma_0}^{\pm}$  и  $D_{\gamma_0}^{+}$ , соответственно из соотношений (20) – (26) находим функции  $u_1^{\pm}(x, y)$ , заданные в тех же областях  $D_{\gamma_0}^{\pm}$  и  $y_1(x)$ ,  $-1 \leq x \leq 1$ .

### Вывод

В качестве источника информации была исследована математическая модель, основанная на плоской задаче Стефана, с учетом конвективного движения и примесей в жидкой фазе. Из всего изложенного можно сделать вывод, что поставленная задача - нелинейная. Она имеет классическое решение, которое может быть построено с помощью нулевого приближения вариационным методом.

### Литература

1. Базалий, Б.В. Об одной стационарной задаче Стефана/Б.В. Базалий, В.Ю. Шелепов // Доклады АН УССР, № 1.–1974. –С. 5–8.
2. Базалий, Б.В. Об одном обобщении стационарной задачи Стефана /Б.В. Базалий, В.Ю. Шелепов // Мат. физика, №27.–1975.– С. 65–80.
3. Миненко, А.С. Приближенный анализ стационарной конвективной задачи Стефана /А.С. Миненко, А.И. Шевченко // Доповіди НАН України, № 5. –2010.— С. 36–40.
4. Миненко, А.С. Исследование конвективной задачи Стефана на плоскости/ А.С. Миненко, А.В. Лёвкина // Проблемы искусственного интеллекта, № 3.–2018.— С. 74-83.
5. Миненко, А. С. Приближенный анализ процесса кристаллизации металла при минимизации ступенчатой функции / А.С.Миненко, Е.В. Радевич// Проблемы искусственного интеллекта, № 2(5).– 2017.–С. 5–9.
6. Миненко, А.С. Задача приближенного анализа свободной границы в управлении информационно - коммуникационными технологиями /А. С.Миненко, А.В.Лёвкина, Е.В.Радевич// Цифровой регион: опыт, компетенции, проекты: сборник статей Международной научно-практической конференции. – Брянск: Брян. гос. инженерно-технолог. ун-т., 2018. –С. 341-345.
7. Миненко, А. С. Проблема минимума со свободной границей /А. С.Миненко // Искусственный интеллект, № 2.–1998.— С. 101–109.

Научное издание

**ИНФОРМАТИКА, УПРАВЛЯЮЩИЕ СИСТЕМЫ,  
МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ  
МОДЕЛИРОВАНИЕ  
(ИУСМКМ-2019)**

Материалы X Международной научно-технической  
конференции в рамках  
V Международного Научного форума  
Донецкой Народной Республики

22-24 мая 2019 г.

**Редактор: Р. В. Мальчева**

**Дизайн и верстка: М. П. Руденко**

Web-сайт конференции: <http://www.iuskm.donntu.org>