

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ**



ИНФОРМАТИКА И КИБЕРНЕТИКА

3 (13)

Донецк – 2018

УДК 004.3+004.9+004.2+51.7+519.6+519.7

ИНФОРМАТИКА И КИБЕРНЕТИКА, № 3 (13), 2018,
Донецк, ДонНТУ.

Представлены материалы по вопросам приоритетных направлений научно-технического обеспечения в области информатики, кибернетики и вычислительной техники.

Материалы предназначены для специалистов народного хозяйства, ученых, преподавателей, аспирантов и студентов высших учебных заведений.

Редакционная коллегия

Главный редактор: Павлыш В. Н., д.т.н., проф.

Зам. глав. ред.: Мальчева Р. В., к.т.н., доц.

Ответственный секретарь: Воронова А. И.

Члены редакционной коллегии: Аверин Г. В., д.т.н., проф.; Аноприенко А. Я., к.т.н., доц.;

Звягинцева А. ., к.т.н., доц.; Зинченко Ю. Е., к.т.н., доц.; Зори С. А., д.т.н., доц.;

Карабчевский В. В., к.т.н., доц.; Миненко А. С., д.ф-м.н., проф.; Привалов М. В., к.т.н., доц.;

Скобцов Ю. А., д.т.н., проф.; Федяев О. И., к.т.н., доц.; Шелепов В. Ю., д.ф-м.н., проф.

Рекомендовано к печати ученым советом ГОУ ВПО «Донецкий национальный технический университет» Министерства образования и науки ДНР. Протокол № 7 от 23 октября 2018 г.

Электронное периодическое издание «Научный журнал «Информатика и кибернетика» зарегистрирован в Министерстве информации ДНР.

Свидетельство о регистрации: серия ААА № 000145 от 20.06.2017.

Контактный адрес редакции

ДНР, 83001, г. Донецк, ул. Артема, 58, ГОУ ВПО «ДонНТУ»,

4-й учебный корпус, к. 36., ул. Кобозева, 17.

Тел.: +38 (062) 301-07-35, +38 (071) 334-89-11

Эл. почта: infcyb.donntu@yandex.ru

Интернет: <http://infcyb.donntu.org>

© Донецкий национальный технический университет
Министерство образования и науки ДНР, 2018

СОДЕРЖАНИЕ

Компьютерные и информационные науки

Анализ эффективности самотестирования цифровых схем на основе неприводимых полиномов	
<i>Дяченко В. О., Дяченко О. Н., Зинченко Ю. Е.</i>	5

Численное интегрирование быстро меняющихся функций вложенным методом Рунге-Кутты пятого порядка с адаптивным контролем величины шага	
<i>Маглеванный И. И., Астахова Н. А., Карякина Т. И.</i>	14

Информатика и вычислительная техника

Вероятностный метод оценки риска финансовой устойчивости предприятия	
<i>Ванжа Т. В., Орлов Ю. К.</i>	20

Исследование и разработка метода оптимизации роя частиц для распознавания динамических жестов	
<i>Потопахин А. А., Ручкин К. А.</i>	28

Математическое моделирование процесса кристаллизации металла при электрошлаковом переплаве	
<i>Миненко А. С., Лёвкина А. В.</i>	33

Обзор алгоритмов аутентификации в распределенных программных системах	
<i>Грищенко О. С., Чернышова А. В.</i>	37

О проектировании виртуальных ресурсов облачной образовательной инфраструктуры	
<i>Жданович Д. П., Жданович П. Б.</i>	42

Обучающая система построения дерева Штейнера в ортогональной метрике	
<i>Ольшевский А. И.</i>	50

Применение теории игр для оптимальной настройки ПИД-регулятора по выбранным критериям	
<i>Гуляев В. А., Орлов Ю. К.</i>	56

Разработка системы автоматического распознавания слитной русскоязычной речи на основе дискриминативного обучения	
<i>Пикалёв Я. С.</i>	61

Разработка системы нечеткого вывода для оценки компетенций претендентов на должности IT-отдела	
<i>Выключок А. В., Землянская С. Ю.</i>	68

Сервисы мониторинга социальных сетей	
<i>Анохина И. Ю., Черепов В. Г.</i>	73

Инженерное образование

Критерии проектирования моделей обучения основам алгоритмизации и программирования	
<i>Курилов М. А.</i>	81

Электронная обучающая система повышения квалификации педагогов: моделирование и внедрение	
<i>Волобуева Т. Б.</i>	87

Компьютерные и информационные науки

Анализ эффективности самотестирования цифровых схем на основе неприводимых полиномов

В. О. Дяченко, О. Н. Дяченко, Ю. Е. Зинченко
Донецкий национальный технический университет
do@donntu.org

Дяченко В. О., Дяченко О. Н., Зинченко Ю. Е. Анализ эффективности самотестирования цифровых схем на основе неприводимых полиномов. Рассмотрены вопросы организации самотестирования цифровых схем. Выполнен анализ эффективности компактного тестирования комбинационных схем, учитывающего характер распределения ошибок в тестовой реакции и структуру генератора тестовых последовательностей и анализаторов тестовых реакций, построенных на основе неприводимых примитивных и непримитивных минимальных полиномов. Предложены и проверены с помощью моделирования в системе автоматизированного проектирования цифровых схем рекомендации по выбору порождающих полиномов регистров сдвига с линейными обратными связями для различных вариантов самотестирования.

Ключевые слова: неприводимые, примитивные, минимальные полиномы, генератор тестовых последовательностей, анализатор тестовых реакций, конфигурация Галуа, порождающий полином

Введение

Кто владеет информацией, тот владеет миром. Эта фраза Натана Ротшильда, ставшая крылатой, в настоящее время становится все более актуальной. Все чаще обсуждаются проблемы инфосферы, продолжающегося информационного взрыва, постиндустриализма, информизации, информационного стресса, искусственного интеллекта и т. д. На современного человека обрушивается огромное количество информации, а в обозримом будущем количество генерируемых данных будет несоизмеримо больше. Прогнозы роста полезной информации (по различным оценкам 3% от 23% потенциально полезной или до трети от общего объема данных) довольно противоречивы. Время показывает, что реальная ситуация в зеттабайтную эпоху развития человечества часто непредсказуема. Тем не менее, не вызывает сомнений лавинообразный рост информационных данных.

International Data Corporation (IDC) - международная исследовательская и консалтинговая компания, основанная в 1964 году, занимается изучением мирового рынка информационных технологий и телекоммуникаций. IDC провела исследование, которое показало, что распространение технологий и доступа к Интернету привели к удвоению объема информации за последние 2 года. Исследование оценило объем сгенерированных данных в 2012 г. в 2,8 зеттабайта и прогнозирует к 2020 г. увеличение объема до 40 зеттабайт, что превосходит прежние прогнозы на 14%. Большие объемы

полезных данных теряются и большая часть информации плохо защищена.

Увеличение количества информации, которая передается, хранится и обрабатывается, приводит к требованиям обеспечения ее достоверности и надежности используемых аппаратных и программных средств [1].

Компьютеры время от времени ошибаются. Поезда сбиваются с расписания. Ракеты падают не туда, куда было задумано. Связь с космическим аппаратом «Фобос-1» была потеряна на трассе перелёта к Марсу. Связь с космическим аппаратом «Фобос-2» была потеряна после 57 суток полёта на орбите искусственного спутника Марса, за 10-11 суток до завершения программы исследований.

Широко известны случаи, когда ракеты попадают не в цель, а в гражданский самолет. По словам ректора Московского государственного технического университета гражданской авиации профессора Бориса Петровича Елисеева [2] – "при расследовании авиационных катастроф, не всегда известно главное - где ошибки летчика, а где "ошибки" бортового компьютера". Например, 3 июля 1988 года американский крейсер «Винсеннес» сбил над Персидским заливом иранский аэробус А-300. По одной из версий коренная причина катастрофы заключалась в ошибке компьютера на борту крейсера. Это может означать, что пассажиры и экипаж иранского авиалайнера стали одними из первых жертв «искусственного интеллекта», то есть ситуации, когда компьютеры не ограничиваются наблюдением за обстановкой и делают выводы, а также дают людям рекомендации.

Современный самолет - "летающий компьютер", не всегда безопасный. Стоит лишь посмотреть передачи «Расследование авиакатастроф». Его вычислительная система включает не меньше 80 бортовых мини-ЭВМ, которые обеспечивают полет по трассе, с изменениями курса, эшелона, высоты, скорости и т.д. То есть штурвал самолета на самом деле уже не "руль" - джойстик бортового компьютера воздушного судна. И вычислительная система самолета, контролируя пилотов, принимает решение на окончательное управление рулевыми поверхностями самолета [2].

Для устранения возможных ошибок из-за естественных природных явлений, либо искусственных причин, или дефектов аппаратных информационных средств, для защиты от разрушений, возникающих под действием жесткого космического излучения, используются современные технологии помехоустойчивого кодирования при проектировании микросхем памяти, весь спектр методов и средств встроенного самотестирования цифровых систем [1-18].

Одним из способов повышения надежности и тестопригодности СБИС микропроцессоров, устройств на ПЛИС является применение встроенных средств контроля, реализующих методы компактного тестирования. Метод сквозного сдвигового регистра за счет декомпозиции схемы на последовательностную и комбинационную части позволяет свести задачу тестирования сложного цифрового устройства к проверке сдвиговых регистров и комбинационных подсхем. С помощью незначительных аппаратных затрат сдвиговые регистры легко преобразуются в линейные переключаемые схемы. В результате получаются самотестирующиеся схемы, в которых реализуется компактное тестирование комбинационных схем (КС). При этом линейные переключаемые схемы исполняют роль генераторов тестовой последовательности (ГТП) и анализаторов тестовой реакции (АТР).

Применение компактного тестирования ставит задачу определения достоверности результатов тестового эксперимента.

В одной из первых работ, посвященных вопросу зависимости тестируемости схемы от вида ГТП и АТР, предлагается синдромное тестирование [5]. В этом случае в качестве ГТП используется любой счетчик, АТР – двоичный счетчик. В работах [7-9] рассматривается сигнатурно-синдромное исчерпывающее тестирование, при котором ГТП - двоичный счетчик, АТР – регистр сдвига с линейными обратными связями (РСЛОС).

В работах [1, 4, 6, 16, 17] рассматриваются вопросы комплексной оценки

достоверности тестирования КС при применении ГТП и АТР в виде РСЛОС. Такая оценка учитывает не только обнаруживающие способности АТР, но также структуру ГТП и характер тестовых реакций объекта диагностики. В частности, получен вывод о значительной зависимости эффективности сигнатурного анализа от выбора того или иного сочетания порождающих полиномов РСЛОС ГТП и АТР. Данная работа представляет собой продолжение исследований в этом направлении для случая тестирования на основе минимальных полиномов.

Цель статьи

Целью статьи является анализ эффективности компактного тестирования цифровых схем при предположении, что ГТП и АТР - РСЛОС с неприводимыми примитивными и непримитивными минимальными порождающими полиномами. На основе такого анализа определить комплексную оценку наиболее оптимального сочетания порождающих полиномов для исчерпывающего тестирования комбинационных схем и проверить полученные результаты с помощью моделирования.

Самотестирование в СБИС

Практически все ведущие зарубежные фирмы, связанные с разработкой микропроцессорных СБИС, уделяют значительное внимание средствам самотестирования: IBM (S/390, метод LSSD); Hewlett Packard (сигнатурный анализ); альянс компаний Apple, IBM и Motorola (Power PC); Motorola (MC 202-206); Intel Corporation (микропроцессоры 80386, Pentium Pro); Advanced Micro Devices (AMD-K6). При этом 5-8 % площади кристалла СБИС занимают встроенные схемы тестирования, которые позволяют обнаружить практически 100% дефектов. Например, диагностическое обеспечение микропроцессора S/390 включает: ОЗУ, кэш, память, схемы их управления со встроенными схемами самотестирования; триггеры, регистровые сети, образующие в режиме тестирования сканируемый путь по методу LSSD; встроенные ГТП; встроенный АТР - многоканальный сигнатурный анализатор; порт JTAG в соответствии со стандартом IEEE 1149.1. Методы исчерпывающего тестирования КС и сканирования позволяют вместе обнаруживать 95% неисправностей.

Применение разных псевдослучайных последовательностей, обеспечивает 99,9 % покрытия всех неисправностей СБИС.

Неприводимые, примитивные и минимальные полиномы

Прежде всего, рассмотрим основные понятия, связанные с характеристиками порождающих полиномов ГТП и АТР.

Полиномом над полем GF(q) называется математическое выражение

$$f(X) = f_{n-1}X^{n-1} + f_{n-2}X^{n-2} + \dots + f_1X + f_0$$

где символ X называют неопределенной (фиктивной) переменной, коэффициенты f_{n-1}, \dots, f_0 принадлежат полю GF(q), а индексы и показатели степеней являются целыми числами.

Нулевым полиномом называется полином $f(X) = 0$.

Приведенным полиномом называется полином, старший коэффициент f_{n-1} которого равен 1.

Степень ненулевого полинома $f(X)$ называется индекс старшего коэффициента f_{n-1} ; степень полинома $f(X)$ обозначается через $\deg f(X)$.

Неприводимыми называются полиномы, которые не могут быть представлены в виде произведения полиномов низших степеней с

коэффициентами из того же поля.

Приведенные неприводимые полиномы называются простыми.

Неприводимым называется полином, который не может быть представлен в виде произведения полиномов низших степеней. Аналогия - простые числа.

Примитивные и минимальные полиномы являются неприводимыми, но не наоборот.

Элемент β называется корнем полинома $p(X)$ или корнем уравнения $p(X) = 0$, если $p(\beta) = 0$.

Примитивным элементом поля GF(q) называется такой элемент α , что все элементы поля, за исключением нуля, могут быть представлены в виде степени элемента α .

Примитивным полиномом $p(X)$ над полем GF(q) называется простой полином над GF(q), такой, что в расширении поля, построенном по модулю $p(X)$, соответствующий полиному X элемент является примитивным.

В таблице 1 элемент X поля GF(2⁵) является примитивным. Поэтому $p(X) = X^5 + X^2 + 1$ и $pd(X) = X^5 + X^3 + 1$ являются примитивными, а символ α можно заменить символом X.

Таблица 1. Элементы полей Галуа GF(25) с двойственными порождающими полиномами

$p(X) = X^5 + X^2 + 1$				$pd(X) = X^5 + X^3 + 1$			
В виде степени	В 2-м виде	В виде степени	В 2-м виде	В виде степени	В 2-м виде	В виде степени	В 2-м виде
0	00000	α^{15}	11111	0	00000	$\alpha^{15} = \alpha^{-16}$	00110
α^0	00001	α^{16}	11011	$\alpha^0 = \alpha^{-31}$	00001	$\alpha^{16} = \alpha^{-15}$	01100
α^1	00010	α^{17}	10011	$\alpha^1 = \alpha^{-30}$	00010	$\alpha^{17} = \alpha^{-14}$	11000
α^2	00100	α^{18}	00011	$\alpha^2 = \alpha^{-29}$	00100	$\alpha^{18} = \alpha^{-13}$	11001
α^3	01000	α^{19}	00110	$\alpha^3 = \alpha^{-28}$	01000	$\alpha^{19} = \alpha^{-12}$	11011
α^4	10000	α^{20}	01100	$\alpha^4 = \alpha^{-27}$	10000	$\alpha^{20} = \alpha^{-11}$	11111
α^5	00101	α^{21}	11000	$\alpha^5 = \alpha^{-26}$	01001	$\alpha^{21} = \alpha^{-10}$	10111
α^6	01010	α^{22}	10101	$\alpha^6 = \alpha^{-25}$	10010	$\alpha^{22} = \alpha^{-9}$	00111
α^7	10100	α^{23}	01111	$\alpha^7 = \alpha^{-24}$	01101	$\alpha^{23} = \alpha^{-8}$	01110
α^8	01101	α^{24}	11110	$\alpha^8 = \alpha^{-23}$	11010	$\alpha^{24} = \alpha^{-7}$	11100
α^9	11010	α^{25}	11001	$\alpha^9 = \alpha^{-22}$	11101	$\alpha^{25} = \alpha^{-6}$	10001
α^{10}	10001	α^{26}	10111	$\alpha^{10} = \alpha^{-21}$	10011	$\alpha^{26} = \alpha^{-5}$	01011
α^{11}	00111	α^{27}	01011	$\alpha^{11} = \alpha^{-20}$	01111	$\alpha^{27} = \alpha^{-4}$	10110
α^{12}	01110	α^{28}	10110	$\alpha^{12} = \alpha^{-19}$	11110	$\alpha^{28} = \alpha^{-3}$	00101
α^{13}	11100	α^{29}	01001	$\alpha^{13} = \alpha^{-18}$	10101	$\alpha^{29} = \alpha^{-2}$	01010
α^{14}	11101	α^{30}	10010	$\alpha^{14} = \alpha^{-17}$	00011	$\alpha^{30} = \alpha^{-1}$	10100

Пусть GF(q) - некоторое поле, GF(Q) - расширение поля GF(q), α - элемент GF(Q). Простой многочлен $f(X)$ наименьшей степени над GF(q), для которого $f(\alpha) = 0$, называется минимальным полиномом элемента α над GF(q).

Алгоритм определения неприводимых,

примитивных и минимальных полиномов приведен в [3], там же приведена таблица неприводимых полиномов. Примитивные и минимальные полиномы обычно определяют с помощью таблиц.

Полином $K^*(X) = X^{\deg K(X)} K(X^{-1})$ - двойственный (обратный) полином по

отношению к полиному $K(X)$. Например, двойственным полиномом по отношению к полиному $K(X)=X^4+X+1$ будет полином $K^*(X) = X^4(X^{-4}+X^{-1}+1) = X^4+X^3+1$.

Обратный полином неприводимого полинома также неприводим.

Для сокращения размера таблицы в ней приведены неприводимые полиномы для элементов нечетных степеней, причем в таблицу включается только один из двух двойственных или самодвойственный полином.

Все полиномы таблицы являются минимальными. Для нулевого элемента минимального элемента нет, для элемента α^0 любого поля Галуа минимальный полином - $X+1$.

По выбранным полиномам из таблицы, могут быть определены минимальные полиномы для всех элементов поля построенного по модулю первого полинома для каждой степени – полином для элемента α^1 .

Все такие полиномы – примитивные с наименьшим количеством ненулевых коэффициентов.

При этом используются следующие свойства минимальных полиномов: минимальные полиномы одинаковы для элементов α^1 и α^2 и двойственны для элементов α^1 и α^{-1} .

Так, например, для степени 3 из таблицы неприводимых полиномов выбираем единственный полином 1 13F.

Эта запись означает, что это минимальный полином для элемента α^1 (а значит, и для α^2 , и для α^4). 13 – это ненулевые коэффициенты в восьмеричной системе счисления.

Символ F определяет свойства полинома, в частности, символы E, F, G, H означают, что полином является примитивным. $13_8=001\ 011_2=X^3+X+1$ (степени коэффициентов нумеруются слева направо, начиная с нуля, в данном случае, 543 210).

Поскольку число ненулевых коэффициентов поля $2^3-1=7$, то $\alpha^7=\alpha^0$, $\alpha^6=\alpha^{-1}$, $\alpha^5=\alpha^{-2}$, $\alpha^3=\alpha^{-4}$. Используя свойства минимальных полиномов, получаем левую часть таблицы 2.

Покажем, что полином X^3+X^2+1 является минимальным полиномом элемента α^3 . Подставив вместо переменной X z^3 (или $z+1$), получим $(X^3+X^2+1)\text{mod}(p(z)=X^3+X+1)=0$, то есть α^3 является корнем полинома X^3+X^2+1 .

Поле Галуа $GF(2^3)$ может быть построено и по другому примитивному полиному – двойственному 1 13F - $p^*(z)=z^3+z^2+1$ (правая часть таблицы 2).

Поле Галуа $GF(2^4)$ для примитивных полиномов (а их также, только два 1 23F и двойственный ему) может быть построено аналогично полям Галуа $GF(2^3)$.

Но поля Галуа $GF(2^4)$ существуют и для других порождающих полиномов.

Таблица 2. Поля Галуа $GF(23)$ с двойственными порождающими полиномами и минимальные полиномы

$p(X)=X^3+X+1$				$p(X)=X^3+X^2+1$			
В виде степени	В виде полинома	В 2-м виде	Минимальный полином	В виде степени	В виде полинома	В 2-м виде	Минимальный полином
0	0	000	-	0	0	000	-
α^0	1	001	$X+1$	α^0	1	001	$X+1$
α^1	z	010	X^3+X+1	α^1	z	010	X^3+z^2+1
α^2	z^2	100	X^3+X+1	α^2	z^2	100	X^3+z^2+1
α^3	$z+1$	011	X^3+X^2+1	α^3	z^2+1	101	X^3+X+1
α^4	z^2+z	110	X^3+X+1	α^4	z^2+z+1	111	X^3+z^2+1
α^5	z^2+z+1	111	X^3+X^2+1	α^5	z^2+z	110	X^3+X+1
α^6	z^2+1	101	X^3+X^2+1	α^6	$z+1$	011	X^3+X+1

Полином $g(X)=X^4+X^3+X^2+X+1$ (это полином 3 37D) является неприводимым, $\text{deg}(X)=4$, поэтому можно построить поле $GF(2^4)$ над полиномом $g(X)$ путем возведения в степень примитивного элемента.

Примитивным элементом такого поля является элемент $(\alpha+1)$ (этот элемент не равен X , поэтому полином – непримитивный), а

минимальным полиномом, соответствующим элементу $(\alpha+1)$, является примитивный полином: $[(X+1)^4+(X+1)^3+1] \text{mod} g(X)=0$.

В таблице 3 представлены элементы поля $GF(2^4)$ над полиномом $g(X)$ и элементы поля $GF(2^4)$ над полиномом $h(X)=X^4+X^3+1$ в степенном и двоичном обозначениях.

Таблица 3. Элементы поля GF(2⁴)

GF(2 ⁴) над g(X)		GF(2 ⁴) над p(X)		GF(2 ⁴) над g(X)		GF(2 ⁴) над p(X)	
0	0000	0	0000	(α+1) ⁸	1001	α ⁸	1110
(α+1) ⁰	0001	α ⁰	0001	(α+1) ⁹	0100	α ⁹	0101
(α+1) ¹	0011	α ¹	0010	(α+1) ¹⁰	1100	α ¹⁰	1010
(α+1) ²	0101	α ²	0100	(α+1) ¹¹	1011	α ¹¹	1101
(α+1) ³	1111	α ³	1000	(α+1) ¹²	0010	α ¹²	0011
(α+1) ⁴	1110	α ⁴	1001	(α+1) ¹³	0110	α ¹³	0110
(α+1) ⁵	1101	α ⁵	1011	(α+1) ¹⁴	1010	α ¹⁴	1100
(α+1) ⁶	1000	α ⁶	1111	(α+1) ¹³	0110	α ¹³	0110
(α+1) ⁷	0111	α ⁷	0111	(α+1) ¹⁴	1010	α ¹⁴	1100

Аналитический метод вычисления сигнатур

Предположим, что ГТП и АТР реализованы в виде РСЛОС конфигурации Галуа (с внутренними сумматорами в цепях обратной связи) с порождающими полиномами соответственно h(X) и g(X), причем оба полинома примитивные, а их корни связаны равенством β=αγ, m=deg h(X)=deg g(X).

Тестовые наборы, которые поступают на входы исследуемой КС, представляют собой ненулевые элементы поля GF(2m), являющегося расширением поля GF(2) над полиномом h(X).

Эти элементы поля могут быть представлены в двоичном, полиномиальном и степенном обозначениях. Каждому ненулевому элементу αγ поля GF(2m) соответствует минимальный полином, причем, если минимальный полином примитивный, то его степень равна m.

Если в качестве порождающего полинома РСЛОС АТР выбрать минимальный полином, соответствующий элементу αγ, то между корнями полиномов h(X) и g(X) будет выполнено равенство β=αγ.

Таблица 4. Элементы поля Галуа GF(24) с двойственными порождающими полиномами

p(z)=z ⁴ +z+1			p*(z)=z ⁴ +z ³ +1		
В виде степени	В виде полинома	В двоичном виде	В виде степени	В виде полинома	В двоичном виде
0	0	0000	0	0	0000
α ⁰	1	0001	α ⁰ =α ⁻¹⁵	1	0001
α ¹	z	0010	α ¹ =α ⁻¹⁴	z	<u>0010</u>
α ²	z ²	0100	α ² =α ⁻¹³	z ²	<u>0100</u>
α ³	z ³	1000	α ³ =α ⁻¹²	z ³	1000
α ⁴	z + 1	0011	α ⁴ =α ⁻¹¹	z ³ + 1	<u>1001</u>
α ⁵	z ² + z	0110	α ⁵ =α ⁻¹⁰	z ³ + z + 1	<u>1011</u>
α ⁶	z ³ + z ²	1100	α ⁶ =α ⁻⁹	z ³ + z ² + z + 1	1111
α ⁷	z ³ + z + 1	1011	α ⁷ =α ⁻⁸	z ² + z + 1	0111
α ⁸	z ² + 1	0101	α ⁸ =α ⁻⁷	z ³ + z ² + z	1110
α ⁹	z ³ + z	1010	α ⁹ =α ⁻⁶	z ² + 1	0101
α ¹⁰	z ² + z + 1	0111	α ¹⁰ =α ⁻⁵	z ³ + z	1010
α ¹¹	z ³ + z ² + z	1110	α ¹¹ =α ⁻⁴	z ³ + z ² + 1	1101
α ¹²	z ³ + z ² + z + 1	1111	α ¹² =α ⁻³	z + 1	0011
α ¹³	z ³ + z ² + 1	<u>1101</u>	α ¹³ =α ⁻²	z ² + z	0110
α ¹⁴	z ³ + z + 1	<u>1001</u>	α ¹⁴ =α ⁻¹	z ³ + z ²	1100
α ⁰	1	0001	α ¹⁵ =α ⁰	1	0001
α ¹	z	<u>0010</u>	α ¹	z	0010
α ²	z ²	<u>0100</u>	α ²	z ²	0100
α ³	z ³	1000	α ³	z ³	1000

Метод расчета сигнатур быть заключается в выборе степенного обозначения тестовых наборов. В этом случае значение сигнатуры для конъюнкции с рангом m может вычислено согласно следующему выражению: $S=M\gamma X-A\gamma$, где XA - степенное обозначение тестового набора, M - матрица для перехода от значений РСЛОС ГТП к значениям РСЛОС АТР.

Анализ таблицы минимальных полиномов [3] показывает, что для любой степени $m < 5$ существует только два примитивных полинома, причем $\beta = \alpha - 1$, т. е. эти полиномы являются двойственными (взаимобратными). Поэтому в дальнейшем для примеров будем рассматривать $h(X)$ степени $m=12$. А сначала рассмотрим взаимосвязь между элементами двойственных полиномов.

Несложно заметить, что между элементами двух полей присутствует

взаимосвязь. Во-первых, элементы следуют в противоположных направлениях, во-вторых элементы зеркально симметричны. Следующая таблица показывает такую же взаимосвязь и для ГТП и АТР с порождающими полиномами $p(z)=z^4+z+1$ и $p^*(z)=z^4+z^3+1$ соответственно.

Комплексная оценка наиболее оптимального сочетания порождающих полиномов.

В предшествующих работах степень полиномов для РСЛОС ГТП и АТР меньше пяти не рассматривалась, поскольку только начиная с пяти параметр γ может быть больше единицы. Однако, хотя для $m=\text{deg}h(X)=\text{deg}g(X)=5$ разных полиномов может быть шесть, все они примитивные. Для $m=\text{deg}h(X)=10$, $\text{deg}g(X)=5$ и $\text{deg}g(X)=2$ также все полиномы примитивные.

В работе [4] рассматривается $h(X)$ степени $m=12$ и доказано следующее утверждение.

Таблица 5. ГТП и АТР с порождающими полиномами $p(z)=z^4+z+1$ и $p^*(z)=z^4+z^3+1$ соответственно

ГТП $p(z)=z^4+z+1$			АТР $p^*(z)=z^4+z^3+1$	
В виде степени	В виде полинома	В двоичном виде	В виде степени	В двоичном виде
		0000		0000
α^0	1	0001	$\alpha^0=\alpha^{-15}$	1000
α^1	z	0010	$\alpha^1=\alpha^{-14}$	0100
α^2	z^2	0100	$\alpha^2=\alpha^{-13}$	0010
α^3	z^3	1000	$\alpha^3=\alpha^{-12}$	0001
α^4	$z+1$	0011	$\alpha^4=\alpha^{-11}$	1100
α^5	z^2+z	0110	$\alpha^5=\alpha^{-10}$	0110
α^6	z^3+z^2	1100	$\alpha^6=\alpha^{-9}$	0011
α^7	z^3+z+1	1011	$\alpha^7=\alpha^{-8}$	1101
α^8	z^2+1	0101	$\alpha^8=\alpha^{-7}$	1010
α^9	z^3+z	1010	$\alpha^9=\alpha^{-6}$	0101
α^{10}	z^2+z+1	0111	$\alpha^{10}=\alpha^{-5}$	1110
α^{11}	z^3+z^2+z	1110	$\alpha^{11}=\alpha^{-4}$	0111
α^{12}	z^3+z^2+z+1	1111	$\alpha^{12}=\alpha^{-3}$	1111
α^{13}	z^3+z^2+1	1101	$\alpha^{13}=\alpha^{-2}$	1011
α^{14}	z^3+z+1	1001	$\alpha^{14}=\alpha^{-1}$	1001

Утверждение. Пусть $h(X)$ – примитивный полином, $g(X)$ - неприводимый полином, $\text{deg}h(X)=m$, $\text{deg}g(X)=r$, причем $m/r=j$, j - натуральное число. Тогда вес $w(-\gamma)$ числа $-\gamma$ принимает максимальное значение, равное $m-j$ при $\gamma=(2^m-1)/(2^r-1) = 2^{(j-1)r} + 2^{(j-2)r} + \dots + 2^r + 1$; сигнатура конъюнкции с рангом $r < j$ равна нулю; $w(-\gamma)$ принимает минимальное значение, равное j при $\gamma=-(2^m-1)/(2^r-1)$, сигнатура конъюнкции с рангом $r < m-j$ равна нулю.

На основании приведенного утверждения можно выполнить простую сравнительную

оценку различных сочетаний порождающих полиномов РСЛОС ГТП и АТР.

Например, для $h(X)=X^{12}+X^6+X^4+X+1$ при $g(X)=X^{12}+X^6+X^4+X+1$ $z=1$, сигнатура конъюнкции с рангом $r < 1$ равна нулю;

при $g(X)=X^6+X^5+1$ $\gamma=2$, $k=(2^{12}-1)/(2^6-1)=65$, поэтому w принимает максимальное значение для $\text{deg}h(X)=12$ и $\text{deg}g(X)=6$, равное $12-2=10$, сигнатура конъюнкции с рангом $r < 2$ равна нулю.

Для различных сочетаний минимальных полиномов, выбранных из таблицы

неприводимых полиномов для степени 12 [13], приведены в таблице 6, а в таблице 7 аналогичные результаты для соответствующих двойственных полиномов.

Результаты моделирования с помощью САПР Active-HDL.

Результаты моделирования показали, что утверждение справедливо не только для примитивных, но также для непримитивных полиномов ГТП. С помощью САПР Active-HDL была проведена проверки результатов, полученных аналитически (табл. 6 и 7), для всех полиномов АТР степени 6, для некоторых основных характерных полиномов степени 4 и 2.

Проверка полностью подтвердила необходимое условие равенства нулю сигнатуры (основная характеристика неэффективности определенных сочетаний порождающих полиномов ГТП и АТР), как для примитивных, так и непримитивных полиномов АТР.

Кроме подтверждения табличных данных, полученных аналитически, по уже имеющимся результатам моделирования, можно сделать предположение, что при невыполнении условия, сформулированного в утверждении, сигнатура будет отлична от нуля.

Таблица 6. Результаты анализа для различных сочетаний минимальных полиномов

γ	$-\gamma, 2 \text{ c/c}$	$g(X), 8 \text{ c/c}$	$g(X)$	$m-w(-\gamma)=r<$
1	111111111110	10123	$X^{12}+X^6+X^4+X+1$	12-11=1
65	111110111110	00141	X^6+X^5+1	12-10=2
195	111100111100	00165	$X^6+X^5+X^4+X^2+1$	12-8=4
273	111011101110	00023	X^4+X+1	12-9=3
455	111000111000	00111	X^6+X^3+1	12-6=6
585	110110110110	00013	X^3+X+1	12-8=4
715	110100110100	00133	$X^6+X^4+X^3+X+1$	12-6=6
819	110011001100	00037	$X^4+X^3+X^2+X+1$	12-6=6
1365	101010101010	00007	X^2+X+1	12-6=6

Таблица 7. Результаты анализа для сочетаний двойственных минимальных полиномов

γ	$-\gamma 2 \text{ c/c}$	$g^*(X) 8 \text{ c/c}$	$g^*(X)$	$m-w(-\gamma)=r<$
-1	000000000001	14501	$X^{12}+X^{11}+X^8+X^6+1$	12-1=11
-65	000001000001	00103	X^6+X+1	12-2=10
-195	000011000011	00127	$X^6+X^4+X^2+X+1$	12-4=8
-273	000100010001	00031	X^4+X^3+1	12-3=9
-455	000111000111	00111	X^6+X^3+1	12-6=6
-585	001001001001	00013	X^3+X+1	12-4=8
-715	001011001011	00155	$X^6+X^5+X^3+X^2+1$	12-6=6
-819	001100110011	00037	$X^4+X^3+X^2+X+1$	12-6=6
-1365	010101010101	00007	X^2+X+1	12-6=6

Выводы

Рассмотрены вопросы организации самотестирования цифровых схем с применением компактного тестирования на основе методов циклического кодирования. Проиллюстрированы основные понятия порождающих полиномов, применяемых для построения ГТП и АТП – неприводимых, примитивных и непримитивных, минимальных полиномов. Для проверки комплексной оценки

эффективности самотестирования комбинационных схем, использована САПР Active-HDL (рис. 1-3).

Полученные результаты могут найти применение не только для организации самотестирования цифровых схем, но также для построения кодов БЧХ, кодов Рида-Соломона, для компактного анализа с локализацией ошибок.

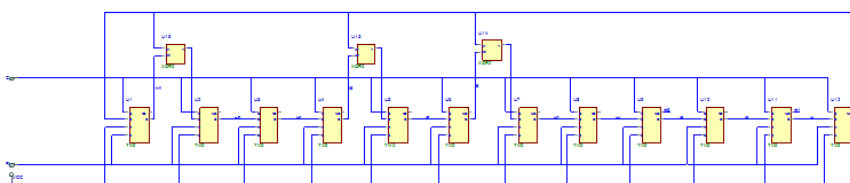


Рисунок 1 – ГТП с порождающими полиномами $h(X)=X^{12}+X^6+X^4+X+1$

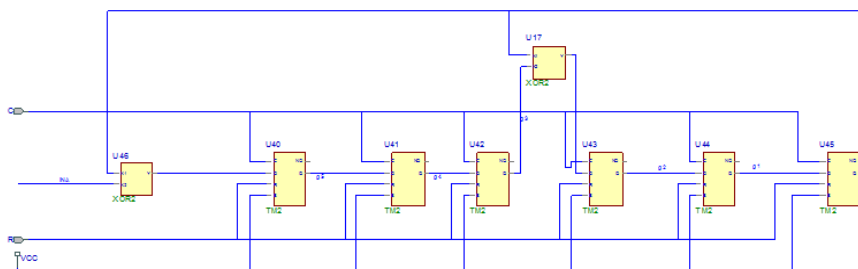


Рисунок 2 – АТП с порождающими полиномами $g(X)=X^6+X^3+1$

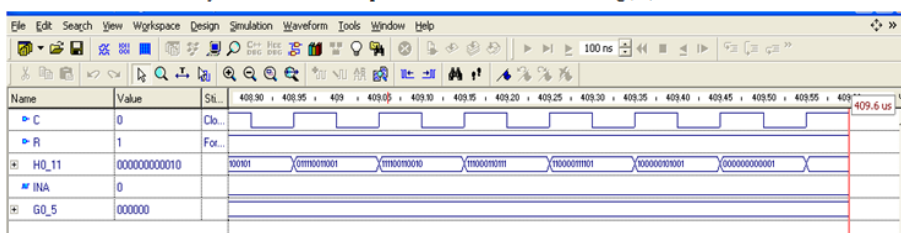


Рисунок 2 – Пример моделирования ГТП и АТП с порождающими полиномами $h(X)=X^{12}+X^6+X^4+X+1$ и $g(X)=X^6+X^3+1$ соответственно для конъюнкции с рангом 4

Литература

1. Дяченко В. О., Дяченко О. Н. Компактное тестирование на основе минимальных полиномов в цифровых схемах с самотестированием // *Материалы V Международной научно-технической конференции «Современные информационные технологии в образовании и научных исследованиях» (СИТОНИ-2017)*. – Донецк: ДонНТУ, 2018. – С. 367-371.

2. Ячменникова Н. Восстание машин. Может ли гаджет уронить самолет? / *Российская газета - Федеральный выпуск №7041 (173)*. Режим доступа - <https://rg.ru/gazeta/rg/2016/08/05.html>

3. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. – М.: Мир, 1976. – 594 с., ил.

4. Дяченко В. О., Дяченко О. Н. Комплексная оценка компактного тестирования цифровых схем на основе минимальных полиномов // *Информатика и кибернетика*. Донецк: ДонНТУ, 2018. № 1(11). – С. 36–41.

5. Savir J. Syndrome-testable design of combinational circuits // *IEEE Trans. Comput.* – 1980. – С-29. – Р. 442-451.

6. Ярмолик В. Н., Калоша Е. П. Эффективность сигнатурного анализа в самотестирующихся СБИС // *Электрон. моделирование*. - 1992.- 14,№3. - С.51-56.

7. Дяченко О.Н. Анализ сигнатурной тестируемости комбинационных схем // *Автоматика и вычислительная техника*. – 1990. – № 5. С.85-89.

8. Dyachenko O.N. Analysis of signature testability of combinational circuits// *Automatic Control and Computer Sciences*. - 05/1991; 24(5) – P. 77-81.

9. А.с. 1829035 СССР, МКИ5 G06F 11/00. Сигнатурно-синдромный анализатор / О.Н. Дяченко (СССР). - № 4864016/24; опубл. 23.07.93. Бюл. № 27. - 2 с.

10. Ершов А. Н., Петров С. В., Пятошин Ю. П., Коханько Д. В., Зяблов В. В. и др. Улучшение радиационной стойкости памяти с помощью помехоустойчивых кодов // Ракетно-космическое приборостроение и информационные системы. 2014, том 1, выпуск 4. - С.42-49.

11. Белл Д. Грядущее постиндустриальное общество. Опыт социального прогнозирования: Пер. с англ. 2-е изд., испр. и доп. - М.: Academia, 2004. - 788 с.

12. Гладких А. А., Климов Р. В., Чилихин Н.Ю. Методы эффективного декодирования избыточных кодов и их современные приложения. - Ульяновск: УлГТУ, 2016. - 258 с.

13. Richard E. Blahut. Algebraic Codes for Data Transmission / Cambridge University Press, 2012. - 498 p.

14. Дяченко О. Н., Дяченко В. О. Альтернативный метод укорачивания циклических кодов // Электронные информационные системы. 2017. № 1 (12). - С. 94-100.

15. Дяченко В. О., Дяченко О. Н. Альтернативный способ построения укороченных кодов Файра // Компьютерная и

программная инженерия. Сборник материалов международной научно-технической конференции студентов, аспирантов и молодых учёных 15-16 декабря 2015 года. - Донецк: ДонНТУ, 2015. - С. 86-89.

16. Дяченко О. Н., Зинченко Ю. Е., Дяченко В. О. Применение методов помехоустойчивого кодирования для компактного тестирования цифровых схем // Информатика и кибернетика. Донецк: ДонНТУ, 2017. № 3(9). - С. 55-59.

17. Дяченко В. О., Дяченко О. Н. Циклическое кодирование цифровой информации на основе двойственных полиномов // Современные тенденции развития и перспективы внедрения инновационных технологий в машиностроении, образовании и экономике: материалы II Международной научно-практической конференции (Азов, 19 мая 2015 г.) - Ростов н/Д, ДГТУ, 2015. - С. 71-76.

18. Дяченко О. Н., Дяченко В. О. Аппаратная реализация кодов BCH и кодов Рида-Соломона // Современные тенденции развития и перспективы внедрения инновационных технологий в машиностроении, образовании и экономике: материалы IV Международной научно-практической конференции (Азов, 25 мая 2017 г.). - Ростов н/Д, ДГТУ, 2017. - С.30-34

Дяченко В. О., Дяченко О. Н., Зинченко Ю. Е. Анализ эффективности самотестирования цифровых схем на основе неприводимых полиномов. Рассмотрены вопросы организации самотестирования цифровых схем. Выполнен анализ эффективности компактного тестирования комбинационных схем, учитывающего характер распределения ошибок в тестовой реакции и структуру генератора тестовых последовательностей и анализаторов тестовых реакций, построенных на основе неприводимых примитивных и непримитивных минимальных полиномов. Предложены и проверены с помощью моделирования в системе автоматизированного проектирования цифровых схем рекомендации по выбору порождающих полиномов регистров сдвига с линейными обратными связями для различных вариантов самотестирования.

Ключевые слова: неприводимые, примитивные, минимальные полиномы, генератор тестовых последовательностей, анализатор тестовых реакций, конфигурация Галуа, порождающий полином.

Dyachenko V. O., Dyachenko O. N., Zinchenko Y. E. Analysis of the self-testing efficiency for digital circuits based on irreducible polynomials. The questions of self-testing organization for digital circuits are considered. The analysis of the efficiency of compact testing for combinational schemes, taking into account the nature of the error distribution in the test reaction and the structure of the generator of test sequences and test reaction analyzers, built on the basis of irreducible primitive and non-primitive minimal polynomials, under study. Recommendations on the choice of shift registers generator polynomials with linear feedbacks for various self-testing variants are proposed and tested by means of simulation in the system of computer aided design.

Keywords: irreducible, primitive and minimal polynomials, generator of test sequence, analyzer of test reactions, Galois configuration, generator polynomial.

Статья поступила в редакцию 12.05.2018 г
Рекомендована к публикации профессором Аноприенко А. Я.

Численное интегрирование быстро меняющихся функций вложенным методом Рунге-Кутты пятого порядка с адаптивным контролем величины шага

И. И. Маглеванный, Н. А. Астахова, Т. И. Карякина
Волгоградский государственный социально-педагогический университет
sianko@list.ru

Маглеванный И. И., Астахова Н. А., Карякина Т. И. Численное интегрирование быстро меняющихся функций вложенным методом Рунге-Кутты пятого порядка с адаптивным контролем величины шага. 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.

Ключевые слова: метод Рунге-Кутты, численное интегрирование, уменьшение объема вычислений, оптимизация вычислительных методов.

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-Кутты

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]. 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 [5]. 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 Δ как разность между методами четвертого и пятого порядков

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

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 [5]). 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 Δ , и следует решить, как удержать ее в необходимых границах. Вычисляем

$$err_{\max} = \frac{|\Delta|}{scale},$$

$$scale = atol + rtol \cdot \max(u(t), u(t-h)) \quad (3)$$

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.

```
#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.;
```

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.

```

//////////
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
INTEGRATOR\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

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

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

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

Здесь параметр Γ - полуширина пика, а

высота пика равна $L_{\max} = 2/(\pi\Gamma)$. График типичной подинтегральной функции показан на рис.1.

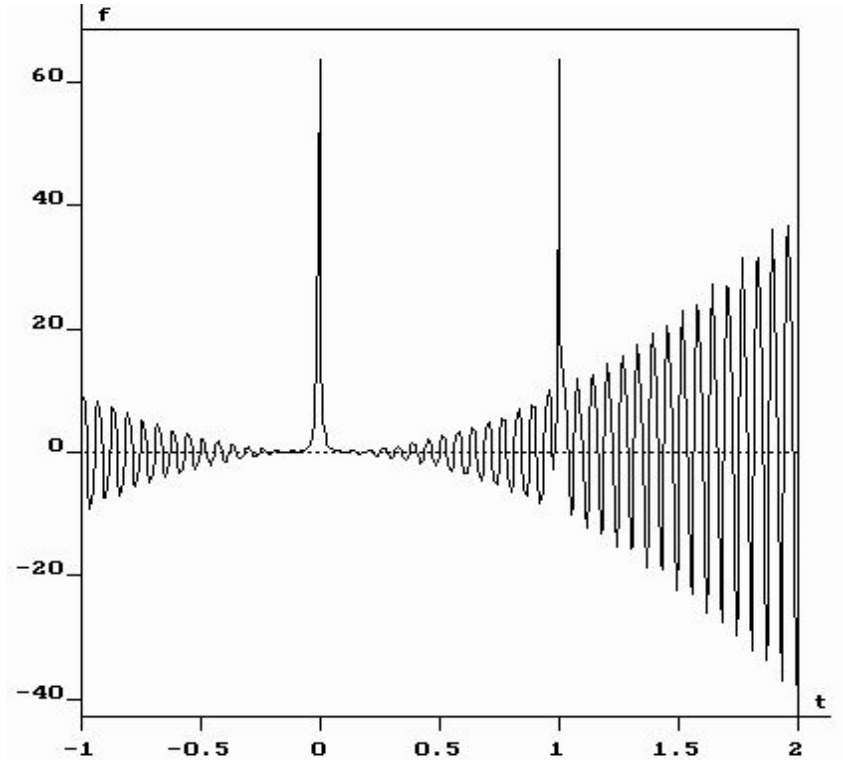


Рис. 1. График подинтегральной функции для интеграла (6) при $\Gamma = 0.01$, $\Omega = 100$.

В таблице 2 показаны численные оценки интеграла (5) для различных Ω и Γ , которые определяют функции с очень острыми пиками.

Таблица 2 – Численные результаты

Ω	Γ	L_{\max}	I_{exact}	I_{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_{exact} - точные значения интеграла, I_{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. I.I. Maglevanny, V.A. Smolar, Robust sampling-sourced numerical retrieval algorithm for optical energy loss function based on log–log mesh optimization and local monotonicity preserving Steffen spline, Nuclear Instruments and Methods in Physics Research B 367 (2016) 26–36

2. J.R. Cash, A.H. Karp, A Variable Order Runge-Kutta Method for Initial Value Problems with Rapidly Varying Right-Hand Sides, ACM Transactions on Mathematical Software, 16 (1990) 201-222.

3. E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I. Nonstiff Problems, 2nd ed., New York:Springer, 1993.

4. W. H. Press, S.A Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes: The Art of Scientific Computing (3rd ed.), New York: Cambridge University Press, (2007).

5. Fehlberg E. Klassische Runge-Kutta-Formeln funfter und siebenter Ordnung mit Schrittweitenkontrolle. // Computing, 1969, 4, S. 93-106.

Маглеванний И. И., Астахова Н. А., Карякина Т. И. Численное интегрирование быстро меняющихся функций вложенным методом Рунге-Кутты пятого порядка с адаптивным контролем величины шага. 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.

Ключевые слова: метод Рунге-Кутты, численное интегрирование, уменьшение объема вычислений, оптимизация вычислительных методов.

Maglevanny I. I., Astakhova N. A., Karyakina T. I. Numerical integration of rapidly varying functions by fifth-order embedded Runge-Kutta rule with adaptive step-size control. 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

Keywords: Runge-Kutta rule, numerical integration, reducing the amount of computations, optimization of computational methods.

Статья поступила в редакцию 22.06.2018 г
Рекомендована к публикации доктором технических наук Павлышом В.Н.

Информатика и вычислительная техника

УДК 65.011.56

Вероятностный метод оценки риска финансовой устойчивости предприятия

Т. В. Ванжа, Ю. К. Орлов
Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа
vanzha_tv7@mail.ru

Ванжа Т. В., Орлов Ю. К. Вероятностный метод оценки риска финансовой устойчивости предприятия. Исследован метод вероятностной оценки экономических рисков на примере финансовой отчетности предприятия горной промышленности. Разработаны алгоритмы работы системы управления экономическими рисками и их программная реализация.

Ключевые слова: экономический риск, вероятностная оценка, количественный анализ.

Введение

Оценка уровня риска является одним из важнейших этапов риск-менеджмента, так как для управления риском его необходимо, прежде всего, проанализировать и оценить.

Одним из самых актуальных подходов есть антикризисное управление горным предприятием с учетом экономического риска в силу его существенного влияния на технико-экономические показатели работы угледобывающего производства – экономические потери от этих рисков составляют около 80% от общего уровня непредвиденных потерь шахты.

Очевидным является то, что принятие оправданного риска в производственно-хозяйственной деятельности угольного предприятия, а не уход от него, и грамотное управление уровнем риска подталкивает к разработке и реализации управленческих решений, а в следствии приводит к повышению эффективности деятельности угледобывающего предприятия [1,2,3].

Целью данной работы является разработка автоматизированной системы управления финансовыми рисками, которая определяет, предотвращает и минимизирует опасности на предприятии горной промышленности.

Поставленная цель предопределила постановку и решение ряда взаимосвязанных задач:

- исследование технологии метода вероятностной оценки экономических рисков на примере финансовой отчетности предприятия горной промышленности;
- разработка алгоритма управления экономическими рисками;
- разработка программного обеспечения с использованием разработанного метода.

Объектом разработки является информационно-аналитическая система управления экономическими рисками предприятия угольной промышленности на примере отдела управления риском горного предприятия.

Предметом исследования есть структура системы управления.

Вероятностный метод оценки риска

Финансовые риски связывают с вероятностью финансовых потерь в виде снижения доходов, прибыли, потери капитала и прочего.

Существует три вида финансовых рисков [2,5,8].

К первому виду относятся риски, которые напрямую зависят от покупательной способности денег, а они делятся на подвиды: инфляционные, дефляционные, ликвидности и валютные риски.

Ко второму виду рисков относятся инвестиционные (это такие виды финансовых рисков, которые напрямую связаны с вложением капитала). Возникновение таких рисков обусловлено инвестиционной деятельностью предприятия.

Третий вид риска связан с формой организации хозяйственной деятельности предприятия. К ним относятся оборотные и авансовые риски.

К основным рискам, которые можно оценить на основе комплексной оценки финансового состояния предприятия относят:

- риск потери платежеспособности;
- риск потери финансовой стойкости и независимости;
- риски, которые обусловлены несбалансированностью структуры активов и пассивов.

Расчетные модели и методы, которые используются для определения уровня данных рисков, представлены в таблице 1.

Таблица 1. Методы оценки рисков финансового состояния организации

Виды рисков	Используемые расчетные модели и методы
Риски, обусловленные несбалансированностью структуры активов и пассивов	Метод анализа целесообразности затрат
Риск потери финансовой устойчивости и независимости	Абсолютные показатели структуры капитала
	Относительные показатели структуры капитала
Риск потери платежеспособности	Абсолютные показатели ликвидности баланса
	Относительные показатели ликвидности баланса

Метод анализа целесообразности затрат заключается в установлении «зон риска» путем анализа финансовой устойчивости организации (обеспеченность имеющимися средствами для текущей деятельности) на основе данных бухгалтерского баланса.

Финансовое состояние организации в данном случае делится на «зоны риска», исходя из структуры и размера элементов расходов организации, – в пределах которых они не превышают предельного значения установленного для каждой конкретной «зоны риска» [3,4,10].

Для выделения этих областей используют три показателя финансовой устойчивости предприятия:

– излишек или недостаток собственных средств;

– излишек или недостаток собственных среднесрочных и долгосрочных источников формирования запасов и расходов;

– излишек или недостаток общей величины основных источников для формирования запасов и расходов.

Эти показатели рассчитываются на основе балансовой модели устойчивости финансового состояния организации, которая имеет следующий вид (1):

$$BA+З+ДЗК+ВМП=СК+ЗП+ДП+КК+КЗС, \quad (1)$$

где BA – оборотные активы;

З – запасы и расходы;

ДЗК – денежные средства, краткосрочные финансовые вложения, дебиторская задолженность и другие активы;

ВМП – расходы будущих периодов;

СК – собственный капитал;

ЗП – обеспечение будущих выплат и платежей;

ДП – долгосрочные обязательства;

КК – краткосрочные кредиты;

КЗС – кредиторская задолженность и другие заемные средства.

Объем собственных оборотных средств определяется по формуле (2):

$$СОС = СК - BA, \quad (2)$$

где СОС – собственные оборотные средства.

Наличие собственных и долгосрочных заемных источников формирования запасов и затрат (СД) рассчитывается по формуле (3):

$$СД = СОС + ДП, \quad (3)$$

где ДП – долгосрочные пассивы (IV раздел пассива баланса).

Общая величина основных источников формирования запасов и затрат (ОИ) рассчитывается по формуле (4):

$$ОИ = СД + КЗС, \quad (4)$$

где КЗС – краткосрочные заемные средства.

При идентификации областей финансовой устойчивости используют трехкомпонентный показатель, представленный формулой (5):

$$S_x = S(\Delta СОС, \Delta СД, \Delta ОИ), \quad (5)$$

$$S_x(\Delta S_x) = 1, \text{ если } \Delta S_x \geq 0;$$

$$S_x(\Delta S_x) = 0, \text{ если } \Delta S_x < 0;$$

где $\Delta СОС$ – излишек или нехватка собственных средств, рассчитывается по формуле (6):

$$\Delta СОС = СОС - З, \quad (6)$$

где З – запасы.

$\Delta СД$ – излишек или нехватка собственных среднесрочных и долгосрочных источников формирования запасов и расходов, рассчитывается по формуле (7):

$$\Delta СД = СД - З. \quad (7)$$

$\Delta ОИ$ – излишек или нехватка общей величины основных источников для формирования запасов и расходов рассчитывается по формуле (8):

$$\Delta ОИ = ОИ - З. \quad (8)$$

Тогда, первая область финансового состояния характеризует условие (9):

$$S(\Delta\text{COC} \geq 0, \Delta\text{CD} > 0, \Delta\text{OI} > 0). \quad (9)$$

Вторая область нормального финансового состояния задается условием (10):

$$S(\Delta\text{COC} \approx 0; \Delta\text{CD} > 0; \Delta\text{OI} > 0). \quad (10)$$

Третья область неустойчивого финансового состояния, которое позволяет установить равновесие платежеспособности предприятия, задается условием (11):

$$S(\Delta\text{COC} < 0; \Delta\text{CD} \geq 0; \Delta\text{OI} \geq 0). \quad (11)$$

Четвертая область критического финансового состояния задается условием (12):

$$S(\Delta\text{COC} < 0; \Delta\text{CD} < 0; \Delta\text{OI} \geq 0). \quad (12)$$

Пятая область кризисного финансового состояния задается условием (13):

$$S(\Delta\text{COC} < 0; \Delta\text{CD} < 0; \Delta\text{OI} < 0). \quad (13)$$

Риск потери финансовой устойчивости и независимости можно оценить с помощью ряда показателей, которые представлены в таблице 2.

Для **коэффициента автономии формула** (14) выводится путем деления значения собственного капитала на сумму активов предприятия и имеет следующий вид:

$$\text{КФА} = \frac{\text{СК}}{\text{А}}, \quad (14)$$

где КФА – коэффициент автономии;
СК – собственный капитал;
А – активы.

Расчет коэффициента соотношения заемных и собственных средств достаточно прост.

Он определяется как отношение величины всех имеющихся долгов по заемным средствам (вне зависимости от срока их погашения) к сумме капитала, числящегося собственным, рассчитывается по формуле (15):

$$\text{К}_{\text{зис}} = \frac{\text{ЗаемКап}}{\text{СобКап}} = \frac{(\text{ДлЗаемОб} + \text{КрЗаемОб})}{\text{СобКап}}, \quad (15)$$

где $\text{К}_{\text{зис}}$ – коэффициент соотношения заемных и собственных средств;

ЗаемКап – имеющиеся долги по заемным средствам;

СобКап – величина капитала, числящегося собственным;

ДлЗаемОб – имеющиеся долгосрочные долги по заемным средствам;

КрЗаемОб – имеющиеся краткосрочные долги по заемным средствам.

Таблица 2. Показатели, оценивающие риск потери финансовой устойчивости и независимости

Название показателя	Рекомендованное значение	Характеристика показателя
Коэффициент автономии	Значение меньше 0,4 указывает на трудности, которые могут возникнуть в случае, если понадобится быстро покрывать непредусмотренные расходы	Характеризует степень независимости организации от внешних источников формирования активов
Коэффициент соотношения заемных и собственных средств	Значение больше 1,5 говорит об угрозе потери организацией своей финансовой независимости (автономии)	Указывает, какая сумма заемных средств находится в каждой денежной единице вложенных в активы собственных средств
Коэффициент обеспеченности собственными оборотными средствами	Значение меньше 0,1 указывает на неустойчивое финансовое состояние организации	Показывает наличие у организации необходимой суммы собственных оборотных активов для обеспечения ее финансовой устойчивости
Коэффициент финансовой устойчивости	Значение меньше 0,6 указывает на то, что организация испытывает финансовые трудности	Указывает, какая часть активов организации финансируется за счет устойчивых источников

Коэффициент обеспеченности собственными оборотными средствами определяется как отношение разности собственного капитала и необоротных активов предприятия к его оборотным активам, формула (16):

$$\text{К}_{\text{осос}} = \frac{(\text{СК} - \text{ВА})}{\text{АО}}, \quad (16)$$

Коэффициент финансовой устойчивости демонстрирует, насколько активы компании профинансированы за счет надежных и долгосрочных источников и рассчитывается по формуле (17):

$$\text{К}_{\text{фину}} = \frac{(\text{Ксоб} + \text{Обдс})}{\text{Побщ}}, \quad (17)$$

где $\text{К}_{\text{фину}}$ – коэффициент финансовой устойчивости;

Ксоб – собственный капитал, включающий имеющиеся резервы;

Обдс – долгосрочные займы и кредиты (обязательства), срок привлечения которых составляет более 1 года;

Побщ – итог по пассивам (иначе – валюта баланса).

Коэффициент, приближенный к 1, показывает, насколько мало привлечено компанией краткосрочных займов. Низкая доля краткосрочных заимствований как раз и называется финансовой устойчивостью [4,7,11].

Управление рисками – это комплекс знаний и навыков, позволяющих при условии выполнения бюджета и расписания проекта планировать и реализовывать действия по реагированию на негативные или позитивные события, которые с некоторой долей вероятности могут проявиться в ходе выполнения проекта.

Алгоритм работы программы

Алгоритм оценки финансовых рисков предприятия содержит такие этапы [5,9]:

- изучение всех существующих финансовых рисков и выбор из них самых значимых, тех, которые способны серьезно воздействовать на финансово-экономическую жизнь компании;
- определение приема расчета конкретного финансового риска, что дает возможность количественно (или качественно) обозначить опасность;
- прогнозирование колебания величины убытков (или их возможности) на разных путях развития компании, разработка управленческих решений для снижения неблагоприятных последствий.

Алгоритм работы программы поделен на два модуля, которые представлены на рисунках 1 и 2: оценка финансовой устойчивости предприятия и оценка риска потери финансовой устойчивости предприятия.

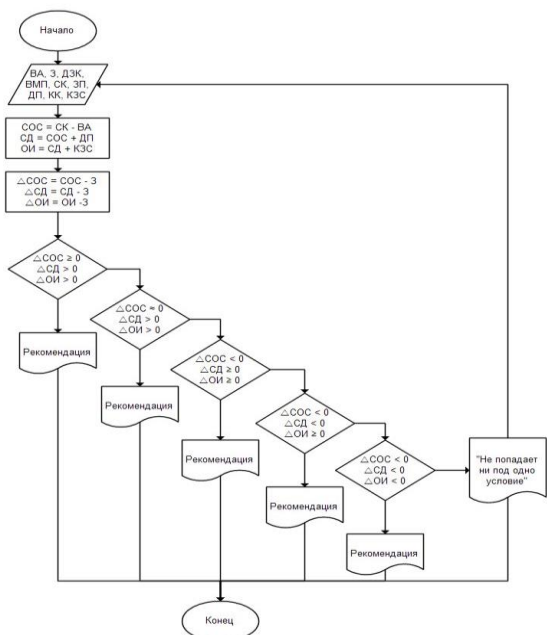


Рисунок 1 – Алгоритм оценки финансовой устойчивости предприятия

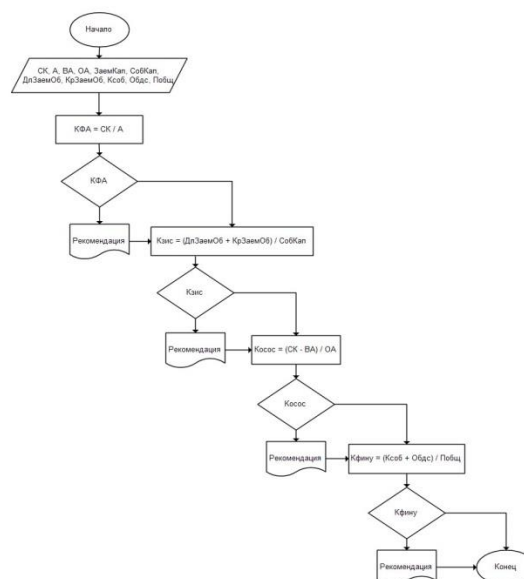


Рисунок 2 – Алгоритм оценки риска потери финансовой устойчивости

- Программа управления риском содержит:
- перечень всех расчетных коэффициентов, относительно которых будут проводиться меры по управлению, и методы управления, применяемые по отношению к ним;
 - план проведения мероприятий по снижению рисков;
 - анализ возможных мероприятий по группам рисков или по группам возможных убытков;
 - анализ остаточных рисков;
 - план мер по финансированию остаточных рисков;
 - анализ возможных методов финансирования рисков по группам возможных убытков;
 - систему мер по контролю над исполнением принятых решений.

Программная реализация продукта

Разработанная модель пользовательского интерфейса базируется на особенностях опыта конкретных пользователей, который характеризуется:

- уровнем подготовки в предметной области разрабатываемого программного обеспечения;
- интуитивными моделями выполнения операции в этой предметной области;
- уровнем подготовки в области владения компьютером;
- устоявшимися стереотипами работы с компьютером.

Модель изображена на рисунке 3 в виде диаграммы последовательности форм работы программного продукта.

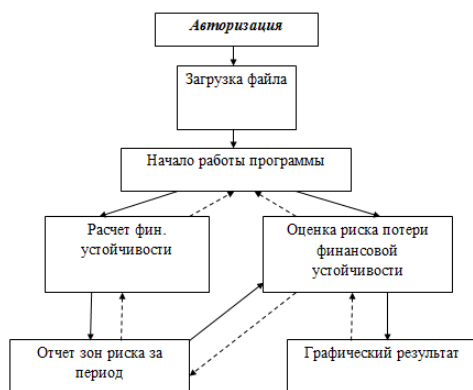


Рисунок 3 – Диаграмма последовательности форм

Окно проверки логина и пароля представлено на рисунке 4.

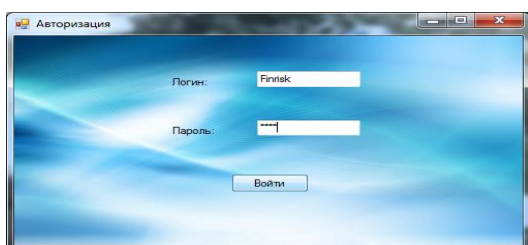


Рисунок 4 – Окно проверки логина и пароля

Окно проверки структуры файла представлено на рисунке 5.

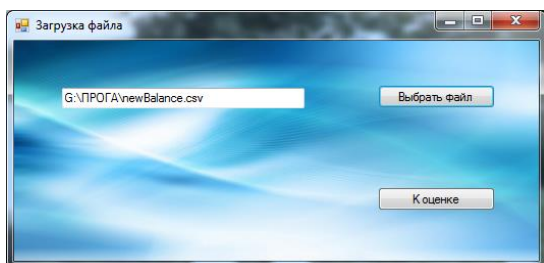


Рисунок 5 – Окно проверки структуры файла.

Форма начала работы в программе. После корректного ввода данных в поля «Логин» и «Пароль» появляется окно формы начала работы, представлено на рисунке 6.

При нажатии на кнопку «Финансовая устойчивость предприятия» идет переход на форму расчета финансовой устойчивости предприятия. При нажатии на кнопку «Риск потери» идет переход на форму оценки риска потери финансовой устойчивости

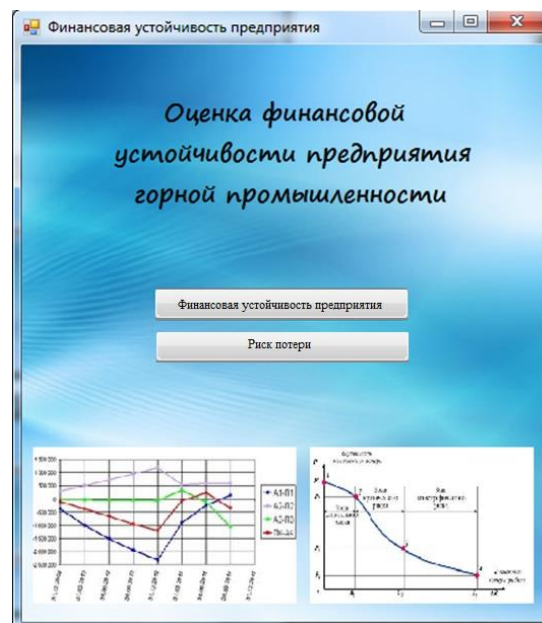


Рисунок 6 – Окно начала работы с программой

Окно формы финансовой устойчивости предприятия представлено на рисунке 7.

После нажатия на кнопку «Расчитать» идет расчет всех показателей, значение которых выводится на форме. Есть возможность перехода на форму выбора файла – кнопка «Назад», если были выбрана не та финансовая отчетность. При нажатии на кнопку «Определить зону риска» выполняется переход на следующую форму.

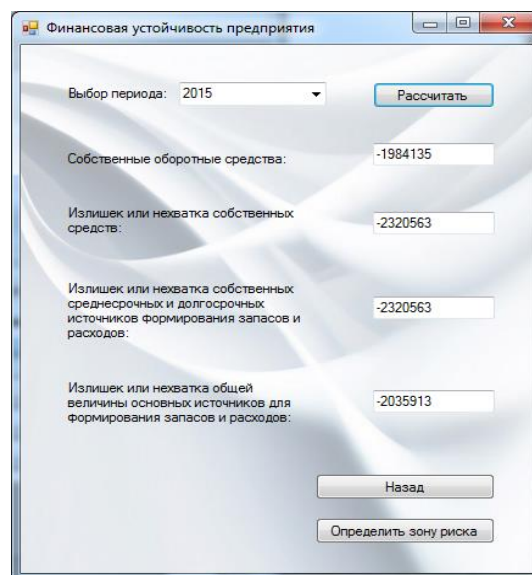


Рисунок 7 – Окно расчетов показателей финансовой устойчивости предприятия

Форма зоны риска. При нажатии на кнопку «Определить зону риска» идет переход в окно отчета по зонам риска, представленное на рисунке 8.

Отчёт ведется по каждому году финансовой отчетности. Есть возможность возврата на предыдущую форму. При нажатии на кнопку «Оценка риска потери» идет переход на следующую форму.

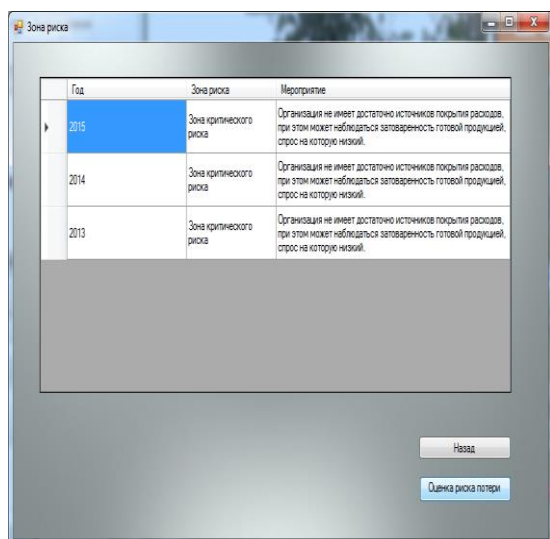


Рисунок 8 – Окно отчёта по зонам риска

Окно оценки риска потери финансовой устойчивости представлено на рисунке 9.

Форма графика финансовой устойчивости. Для графика используется встроенная библиотека Microsoft Chart.

Есть возможности перехода на предыдущие формы, выход из программы.

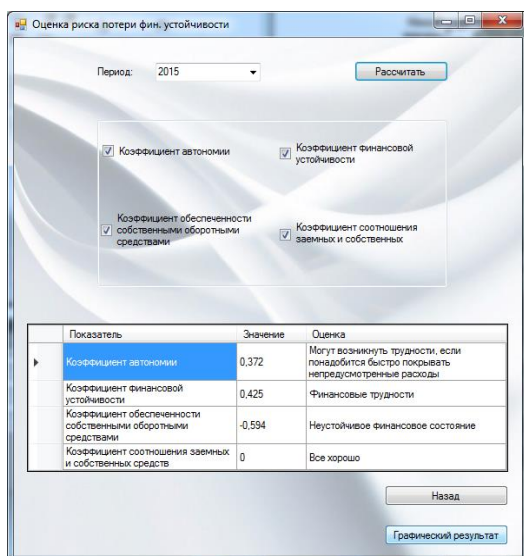


Рисунок 9 – Окно оценки риска потери финансовой устойчивости

Окно графика графического результата представлено на рисунке 10.

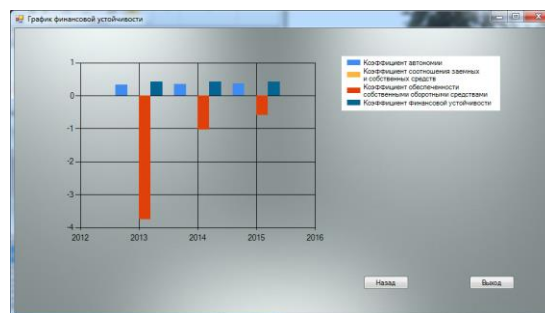


Рисунок 10 – Окно графического результата

Анализ финансовой устойчивости шахты В.М. Бажанова выявил значительные проблемы в финансовой стабильности предприятия. Рассчитанные показатели показали критическое состояние финансовой устойчивости, что несет за собой:

- трудности, которые могут возникнуть в случае, если понадобится быстро покрывать непредусмотренные расходы;
- угрозу потери организацией своей финансовой независимости;
- указывает на неустойчивое финансовое состояние организации;
- указывает на то, что организация испытывает финансовые трудности.

Сложность математического аппарата в системе расчетов затрудняет и сужает на практике применение в финансовой политике предприятия моделей управления рисками, что в свою очередь порождает недостаточную информированность в процессе принятия управленческих решений.

Таким образом, для уменьшения степени риска нужно использовать следующие мероприятия:

- контроль открытой валютной позиции;
- применение методов натурального хеджирования.

Выводы по работе

Предложена новая методика количественной оценки параметрических рисков горнодобывающей промышленности – вероятностный метод финансовой отчетности предприятия.

Он позволяет оценить финансовую устойчивость предприятия на основании коэффициентов устойчивости.

В качестве исходной информации для оценки рисков финансового состояния организации использовались следующие формы бухгалтерской отчетности:

- бухгалтерский баланс, который фиксирует имущественное и финансовое положения организации на отчетную дату;

– отчет о финансовых результатах, который показывает результаты деятельности организации за отчетный период.

По данным количественной оценки рисков рассчитаны финансовые риски предприятия горной промышленности. Перечислены методы управления финансовыми рисками предприятия горной промышленности и дана их характеристика.

Одно из наиболее важных требований системы контроля над финансовой деятельностью предприятий и учреждений состоит в том, чтобы размеры их собственного капитала соответствовали присущим им финансовым рискам.

В связи с этим перед финансовыми менеджерами всякий раз возникает задача определения размера оптимального резерва, необходимого для покрытия возможных будущих затрат.

Полученные зависимости показывают, что критерии оценки рисков, предложенные в настоящем исследовании, корректно отражают установленные причинно-следственные связи между параметрами и характеристиками финансовых рисков угледобывающего предприятия, позволяют переходить от интуитивных оценок рисков к объективным количественным показателям и, следовательно, повышать качество управления и, как следствие, заблаговременно вносить существенные и весьма важные корректировки в параметры производственной системы угледобывающего предприятия.

Литература

1. Ванжа Т. В., Орлов Ю.К. Анализ экономических рисков горного предприятия методом структурированного количественного анализа. ИНФОРМАТИКА И КИБЕРНЕТИКА. – Д.: ДонНТУ, – 2017. – № 2(8). – 43-49 с.

2. Т.В. Ванжа, Ю. К. Орлов. Разработка моделей и методов управления горным предприятием в условиях экономических рисков. Компьютерная и программная

инженерия. Сборник материалов международной научно-технической конференции студентов, аспирантов и молодых ученых 15-16 декабря 2015 года. – Донецк, ДонНТУ – 2015. – 314-317 с.

3. Т.В. Ванжа, Ю. К. Орлов. Разработка моделей и методов управления горным предприятием в условиях экономических рисков. Сборник статей студенческой научно-технической конференции, г.Севастополь, 04 – 08 апреля 2016 г/ М-во образования и науки РФ, Севастопольский государственный университет; науч. ред. Е.Н. Машенко – г. Севастополь: СевГУ, 2016. – 173-177 с.

4. Петросов А.А., Мангуш К.С. Экономические риски горного производства: Учебное пособие. М.: Издательство Московского государственного горного университета, 2002. – 142 с.

5. Лускатова О.В. Оценка риска и экономической устойчивости горных предприятий на основе нечеткой логики / Лускатова О.В. – Норильск: Норильский индустр. ин-т., 2004. – 221 с.

6. Балабанов И.Т. Риск-менеджмент / Балабанов И.Т. – М.: Юнити, 2002. – 192 с.

7. Райзберг Б.А. Предпринимательство и риск / Райзберг Б.А. – М.: Дело, 2001. – 61 с.

8. Чалый-Прилуцкий В.А. Рынок и риск. Методические материалы по анализу оценки и управления риском / Чалый-Прилуцкий В.А. – М.: НИУР, Центр СИНТЕК, 2001. – 80 с.

9. Мамаева Л.Н. Управление рисками: Учебное пособие / Мамаева Л.Н. – М.: Дашков и К, 2013. – 256 с.

10. Хенли Э.Дж. Надёжность технических систем и оценка риска / Э.Дж. Хенли, Х. Кумамото. – М.: Машиностроение, 1984. – 528 с.

11. Мартякова Е.В., Кочура И.В. Хозяйственные риски: оценка и прогнозирование : монография / Е. В. Мартякова, И. В. Кочура; ГБУЗ "Донец. нац. техн. ун-т, Ин-т экономики пром-сти НАН Украины. - Донецк, 2008. - 220 с.

Ванжа Т. В., Орлов Ю. К. Вероятностный метод оценки риска финансовой устойчивости предприятия. Предложена новая методика количественной оценки параметрических рисков горнодобывающей промышленности – вероятностный метод финансовой отчетности предприятия. Он позволяет оценить финансовую устойчивость предприятия на основании коэффициентов устойчивости. Объектом разработки является информационно-аналитическая система управления экономическими рисками предприятия угольной промышленности на примере отдела управления риском горного предприятия. Разработаны алгоритмы работы системы управления экономическими рисками и их программная реализация. По данным количественной оценки рисков рассчитаны финансовые риски предприятия горной промышленности. Проведен анализ финансовой устойчивости шахты В.М. Бажанова. Выявлены значительные проблемы в финансовой стабильности предприятия. Разработана модель пользовательского интерфейса, которая базируется на особенностях опыта конкретных пользователей.

Ключевые слова: экономический риск, вероятностная оценка, количественный анализ.

Vanzha T. V., Orlov J. K. Probabilistic method of assessing the risk of financial stability of an enterprise. A new method for quantitative assessment of parametric risks of the mining industry is proposed - a probabilistic method of financial reporting of an enterprise. It allows assessing the financial stability of an enterprise based on the coefficients of sustainability. The object of development is an information and analytical system for managing the economic risks of a coal industry enterprise using the example of the risk management department of a mining enterprise. The algorithms for the operation of the economic risk management system and their software implementation are developed. According to the quantitative risk assessment, the financial risks of a mining enterprise have been calculated. The analysis of the financial stability of the mine VM Bazhanov. Significant problems in the financial stability of the enterprise were revealed. A user interface model was developed, which is based on the peculiarities of the experience of specific users.

Keywords: economic risk, probabilistic estimation, quantitative analysis.

Статья поступила в редакцию 16.05.2018 г
Рекомендована к публикации доцентом Приваловым М. В.

УДК 004.021

Исследование и разработка метода оптимизации роя частиц для распознавания динамических жестов

А. А. Потопахин, К. А. Ручкин

Донецкий национальный технический университет
кафедра искусственного интеллекта и системного анализа
alexey_potopahin@hotmail.com

Потопахин А. А., Ручкин К. А. Исследование и разработка метода оптимизации роя частиц для распознавания динамических жестов. Проведено исследование эволюционного алгоритма оптимизации роя частиц (Particle Swarm Optimization) для решения задачи распознавания. Выполнена адаптация алгоритма PSO под задачу распознавания динамических жестов. Предложена модификация алгоритма PSO путем использования нормального распределения вероятностей в вычислении скорости движения частицы в пространстве поиска.

Ключевые слова: динамические жесты, распознавание, оптимизация роя частиц, адаптация, нормальное распределение.

Введение

Современные системы бесконтактного управления интерактивным контентом основаны на методах распознавания динамических и статических жестов человека. Для оптимизации задачи распознавания жестов используются эволюционные алгоритмы (метод движения роя частиц, метод колонии муравьев и т.д.), статистические модели (Скрытые Марковские Модели (СММ), метод Монте-Карло и т.д.) и др. Несмотря на отдельные успехи, качество разработанных алгоритмов распознавания жестов рук и пальцев, как динамических, так и статических, с использованием цветных видеокамер и трехмерных сенсоров все еще остается недостаточным для построения практических систем человеко-машинного взаимодействия.

Главными недостатками существующих методов являются чувствительность к изменениям освещения, потребность обучения системы для каждого оператора, невысокое качество распознавания жестов и небольшая скорость распознавания.

В данной работе проводится исследование и разработка метода оптимизации роя частиц (PSO), а также адаптация под задачу распознавания динамических жестов. В последнее время PSO успешно применяется во многих областях исследований и приложений. Оптимизация, роя частиц используется для подходов, которые могут применяться в широком спектре приложений, а также для специфических задач, которые ориентированы на конкретные требования.

Анализ современного состояния проблемы

Для захвата жестов человека могут использоваться различные устройства, например, ультразвуковые локаторы, кинематические датчики, системы структурированной подсветки и т.д. Но наиболее распространенным для получения данных о жестах пользователя является видеокамера и устройства подобные трехмерному сенсору Kinect от Microsoft, который помимо цветной видеокамеры, также состоит из двух сенсоров глубины. Применение двух и более видео-датчиков позволяет запечатлеть дополнительную информацию о трехмерном объекте (отверстия, невидимые с другой камеры, форма боковых проекций, и др.) [1].

Наиболее распространенным является подход с использованием Скрытых Моделей Маркова (СММ). Методы распознавания жестов на основе СММ представляют каждый жест набором состояний, которые связаны с тремя вероятностями (начальное, переходное, наблюдение) вычисляемые из заранее подготовленных эталонов. СММ выбирают модель с наилучшей вероятностью и классифицируют жест к соответствующей категории. Хотя системы распознавания на основе СММ выбирают модель с наилучшей вероятностью, но не гарантируется, что шаблон действительно похож на опорный жест. Для того, чтобы получить хорошие результаты, СММ должна быть хорошо обучена и настроена. Основными недостатками этого подхода является то, что для калибровки

моделей требуется большое количество образцов и длительное время обучения.

Распознавание жестов в режиме реального времени остается сложной задачей из-за неопределенности в методах определения границ жестов. Подход, основанный на движении роя частиц решает эту проблему, упрощая сегментацию жестов [5]. В предлагаемом подходе избегается преждевременная идентификация жеста, тем самым увеличивая точность результата распознавания. Данный распознаватель требует меньше вычислительных и временных ресурсов по сравнению с СММ и является хорошим кандидатом для реализации в приложениях, выполняющие распознавание в режиме реального времени.

Постановка задачи

В данной работе проводится исследование эволюционного алгоритма оптимизации роя частиц (PSO). Выполняется адаптация алгоритма PSO под задачу распознавания динамических жестов, а также модификация оригинального алгоритма путем использования нормального распределения вероятностей в вычислении скорости движения частицы в пространстве поиска.

Оптимизации с помощью роя частиц

Оптимизация роя частиц (PSO) – это метод стохастической оптимизации, основанный на популяции, разработанный доктором Эберхартом и доктором Кеннеди в 1995 году, которые вдохновились социальным поведением стаи птиц и рыб [2]. PSO имеет много общего с эволюционными методами вычисления, такими как генетические алгоритмы (ГА). Система инициализируется совокупностью случайных решений и ищет оптимальные параметры путем обновления эпох. Однако, в отличие от ГА, у PSO нет эволюционных операторов, таких как кроссовер и мутация. В PSO потенциальные решения, называемые частицами, двигаются через проблемное пространство, следуя текущим оптимальным частицам.

В PSO каждое отдельное решение может рассматриваться как частица в многомерном пространстве, которая движется с определенной скоростью. Скорость движения частицы постоянно изменяется в соответствии с параметрами частицы и соседними группами, которые перемещаются в проблемном пространстве [2].

Частица инициализируется следующими параметрами: $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ – позиция i -ой частицы; $V_i = (v_{i1}, v_{i2}, \dots, v_{in})$ – текущая скорость движения i -ой частицы; $P_i = (pBest_{i1}, pBest_{i2}, \dots,$

$pBest_{in})$ – лучшая позиция частицы, которая относится к месту, где достигается оптимальное значение фитнес-функции.

Пусть $f(x)$ – фитнес-функция, тогда лучшее значение i -ой частицы определено в формуле (1):

$$P_i(t+1) = \begin{cases} P_i(t) & f(X_i(t+1)) \geq f(P_i(t)) \\ X_i(t+1) & f(X_i(t+1)) < f(P_i(t)) \end{cases} \quad (1)$$

Лучшая позиция i -ой частицы обновляется в том случае, если новое значение фитнес-функции лучше, чем предыдущее. В противоположном случае остается без изменений.

Пусть M – группа всех частиц, а общая лучшая позиция среди всех частиц – $gBest(t)$ представлено в выражении (2):

$$f(gBest(t)) = \min(f(P_0(t)), f(P_1(t)), \dots, f(P_M(t))) \quad (2)$$

На основании вышеприведенных определений, эволюционные формулы алгоритма PSO описаны в (3) и (4):

$$v_i(t+1) = v_i(t) + (c_1 \cdot rand() \cdot (pBest_i - x_i(t))) \quad (3)$$

$$+ (c_2 \cdot rand() \cdot (gBest - x_i(t))),$$

$$x_i(t+1) = x_i(t) + v_i(t), \quad (4)$$

где $v_i(t+1)$ – новая скорость для i -ой частицы, c_1 и c_2 – весовые коэффициенты, $x_i(t)$ – это i -я позиция частицы во времени t , $pBest_i$ – это лучшая позиция i -ой частицы и $gBest$ – лучшая позиция среди всех частиц. Функция $rand()$ генерирует случайное число $[0, 1]$. В (4) частица получает новую скорость из (3) и передвигается на новую позицию. Блок-схема алгоритма оптимизации роя частиц представлен на рис. 1.

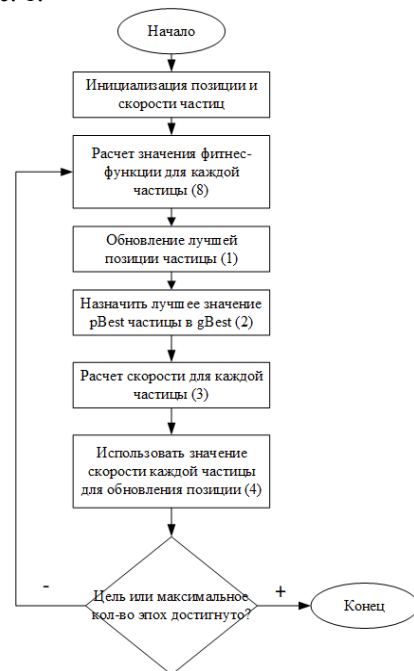


Рисунок 1 – Блок-схема алгоритма оптимизации роя частиц

Значение скорости частицы обычно ограничивается некоторым предопределенным значением. Максимальная скорость V_{max} в контексте оригинального алгоритма выбирается в интервале между 4 и 6.

Метод дискретной оптимизации роя частиц

Оригинальный метод оптимизации роя частиц разработан для работы в непрерывном пространстве. Для поставленной задачи необходимо адаптировать метод для работы в дискретном пространстве с двоичными переменными [3].

В двоичном пространстве позиция частицы $x_i(t)$ должна быть 0 или 1. Но скорость $v_i(t)$ по-прежнему может быть любым числом. Поэтому для того, чтобы новая позиция $x_i(t+1)$ была в дискретном пространстве, необходимо заменить формулу (4) на (5) и (6):

$$Sig(x) = \frac{1}{1 + \exp(-x)}, \quad (5)$$

$$x_i(t+1) = \begin{cases} 0 & rand() \geq Sig(v_i(t+1)) \\ 1 & rand() < Sig(v_i(t+1)) \end{cases} \quad (6)$$

В (6) $rand()$ – случайное число в диапазоне $[0, 1]$. Скорость $v_i(t)$ может воздействовать на текущую позицию частицы $x_i(t)$, а также направление, что позволяет алгоритму выполнять поиск в заданном пространстве. В дискретном PSO $v_i(t)$ может только показывать вероятность изменения $x_i(t)$. Если позиция частицы в одном измерении 0, то вероятность того, что позиция инвертируется в 1 является $Sig(v_i(t))$, а в противоположном случае вероятность будет составлять $1 - Sig(v_i(t))$.

Адаптация и модификация PSO для распознавания динамических жестов

Первым необходимым шагом для распознавания жестов и функционирования PSO (инициализация) необходимо представление жеста G в виде набора координат в двумерном пространстве (если запись жестов производится на 2D камеру). Каждая точка представлена набором координат $g(t) = [x(t), y(t)]$ в G во времени t [4].

Все жесты преобразуются в частицы и образуют пространство поиска SP в (7). Также каждая частица инициализируется набором случайных позиций $x_i(t)$ и скоростей $v_i(t)$.

$$SP = \{G_1 = [g_1, g_2, \dots, g_n], G_2, \dots, G_k\} \quad (7)$$

Фитнес-функция каждой частицы рассчитывается при помощи Эвклидова расстояния по формуле (8):

$$F(s, g) = \sqrt{(x_s - x_g)^2 + (y_s - y_g)^2}, \quad (8)$$

где s – текущая позиция частицы, а g – координата в жесте G .

Важным шагом в алгоритме PSO является определение скоростей частиц. Уравнение (3) содержит две случайные функции $rand_1()$ и $rand_2()$. Цель введения этих двух случайных функций заключается в том, чтобы имитировать непредсказуемое поведение природы. Как правило, эти две функции представляют собой два отдельных вызова, и в большинстве реализаций люди используют случайную функцию, равномерно распределенную между 0 и 1. Таким образом, тяговые силы $pBest$ и $gBest$ будут варьироваться от 0 до 1 с равномерной вероятностью в процедуре оптимизации.

Если расширить сферу охвата таких областей, как психология, астрономия и физика, то обнаружено, что многие наблюдаемые случайные переменные и данные следуют хорошо известному распределению Гаусса (нормальное). Несмотря на то, что некоторые из них точно не соответствуют распределенной кривой Гаусса, их можно довольно хорошо аппроксимировать при помощи центральной предельной теоремы [6]. Эта теорема утверждает, что сумма большого количества независимых и одинаково распределенных случайных величин будет приблизительно нормально распределена, если случайные величины имеют конечную дисперсию. Многие общие атрибуты, такие как тестовые оценки, высоты и т.д., следуют распределениям Гаусса, где небольшая вероятность встречается на высоком и нижнем концах, и большая вероятность встречается около середины. Поскольку алгоритм PSO основан на стохастическом свойстве, должно рассматриваться использование распределенных случайных величин Гаусса вместо равномерно распределенных переменных в (3). Также отмечается, что в алгоритме PSO весь рой может содержать 30 или более частиц и выполнять сотни или даже тысячи итераций на поиск оптимального значения. Поэтому случайность в движении частиц может быть более адекватно описана распределением Гаусса, чем путем равномерного распределения.

В данной работе предлагается модификация PSO алгоритма путем нормального распределения вероятностей в вычислении скорости $v_i(t)$ движения частицы в пространстве поиска. Использование нормального распределения вероятностей предложено в [7], где применяется в вычислении оператора кроссовера в простом генетическом алгоритме. Результаты работы модифицированных алгоритмов превосходят результаты без применения данного подхода.

Стандартным нормальным распределением принято считать нормальное

распределение с математическим ожиданием $a = 0$ и стандартным отклонением $\sigma = 1$, плотность которого равна (9):

$$\text{randN} = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (9)$$

Модифицированная формула вычисления скорости частицы представлена в (10):

$$v_i(t+1) = v_i(t) + (c_1 \cdot \text{randN}() \cdot (pBest_i - x_i(t))) + (c_2 \cdot \text{randN}() \cdot (gBest - x_i(t))), \quad (10)$$

где randN – нормально распределенное случайное число, которое ограничено действительным числом с математическим ожиданием, дисперсией и стандартным отклонением, описанное выше.

График зависимости скоростей частиц v_i от случайного числа в диапазоне $[0, 1]$ с использованием формулы расчета скорости (3) из оригинального алгоритма PSO представлен на рис. 2.

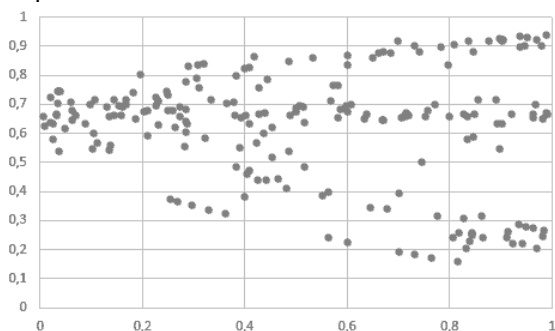


Рисунок 2 – Скорости частиц, полученные по формуле (3)

График зависимости скоростей частиц v_i с использованием нормального распределения (9) в модифицированной формуле вычисления скорости частицы (10) представлен на рис. 3.

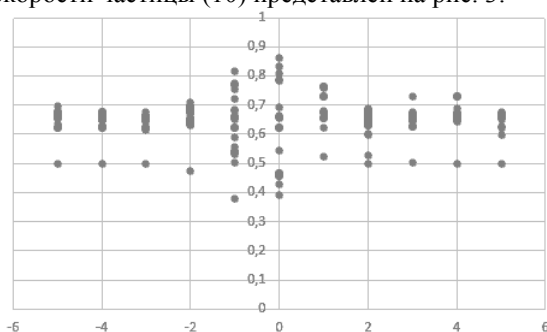


Рисунок 3 – Скорости частиц, полученные по формуле (10)

Таким образом нормальное распределение вероятностей в модифицированном способе вычисления скорости создает более натуральное распределение, тем самым увеличивает “генетическое разнообразие” популяции. Это

приемлемое решение для получения более успешных результатов.

Выводы

В работе проведено исследование эволюционного алгоритма оптимизации роя частиц (PSO), а также выполнена адаптация алгоритма под задачу распознавания динамических жестов.

Предложена модификация алгоритма PSO путем использования нормального распределения вероятностей в вычислении скорости движения частицы в пространстве поиска. Результаты работы модифицированного алгоритма превосходят результаты без применения данного подхода.

Главными недостатками существующих методов являются чувствительность к изменениям освещения, потребность обучения системы для каждого оператора, невысокое качество распознавания жестов и небольшая скорость распознавания.

Для распознавания сложных жестов рук необходимо использовать специализированные устройства ввода, а также проводить настройку и обучение программных средств.

Но для динамических жестов, где важно только знать траекторию движения руки (влево, вправо, вверх, вниз) достаточно использовать не сложные математические алгоритмы. Актуальной является задача создания новых моделей, методов и алгоритмов распознавания жестов руки, которые могут быть использованы для создания систем бесконтактного человеко-машинного взаимодействия.

Таким образом, метод оптимизации роя частиц может с правильно подобранной и настроенной модификацией может применяться для задачи распознавания динамических жестов, где необходима максимальная точность и эффективность распознавания.

По сравнению с ГА преимуществом PSO является то, что PSO легче реализовать, и требуется меньше параметров для настройки. PSO успешно применяется во многих областях: оптимизация функций, обучение искусственной нейронной сети и другие области, где может применяться ГА.

Литература

1. Терехин, А.В. Алгоритмы определения безразмерных признаков изображений проекций трехмерных объектов и их распознавание / Терехин А.В. // ВлГУ. – 2015. – 187 с.
2. Eberhart, R.C. A new optimizer using particle swarm theory / R.C. Eberhart, J. Kennedy // Proceedings of the Sixth International Symposium on Micromachine and Human Science. – 1995. – pp. 39-43.

3. Eberhart, R.C. A discrete binary version of the particle swarm algorithm / R.C. Eberhart, J. Kennedy // Proceedings of the IEEE International Conference on Systems. – 1997. – pp. 4104-4108.

4. Wei, Z. Network Coverage Optimization Strategy in Wireless Sensor Networks Based on Particle Swarm Optimization / Z. Wei, F. Zihao // Bachelor's Thesis in Electronics. – 2011. – P. 7.

5. Nyirarugira, C. Hand Gesture Recognition Using Particle Swarm Movement / C. Nyirarugira,

H. Choi, T. Kim // Mathematical Problems in Engineering. – 2016. – pp. 1-4.

6. Tijms, H., Understanding Probability: Chance Rules in Everyday Life / H. Tijms // Cambridge University Press, Cambridge. – 2007. – P. 442.

7. Ladhani G.S. A Genetic Algorithm with Weighted Average Normally-Distributed Arithmetic Crossover and Twinkling / G.S. Ladhani, T.B. Mohamed // Applied Mathematics. – 2012. – pp. 1220-1235.

Потопахин А. А., Ручкин К. А. Исследование и разработка метода оптимизации роя частиц для распознавания динамических жестов. Проведено исследование эволюционного алгоритма оптимизации роя частиц (Particle Swarm Optimization) для решения задачи распознавания. Выполнена адаптация алгоритма PSO под задачу распознавания динамических жестов. Предложена модификация алгоритма PSO путем использования нормального распределения вероятностей в вычислении скорости движения частицы в пространстве поиска.

Ключевые слова: динамические жесты, распознавание, оптимизация роя частиц, адаптация, нормальное распределение.

Potopakhin A. A., Ruchkin K. A. Research and development of a method for managing a three-dimensional scene using dynamic gestures. This research was conducted of the evolutionary particle swarm optimization algorithm (PSO) for solving the recognition problem. The adaptation of PSO algorithm was conducted to the dynamic gesture recognition task. A modification of the PSO algorithm was proposed by using the normal distribution probability in calculating velocity of a particle in the search space.

Keywords: dynamic gestures, recognition, particle swarm optimization, adaptation, normal distribution.

Статья поступила в редакцию 23.05.2018 г.
Рекомендована к публикации профессором Миненко А. С.

УДК 517.9

Математическое моделирование процесса кристаллизации металла при электрошлаковом переплаве

А. С. Миненко, А. В. Лёвкина

Донецкий национальный технический университет

a.levkina13@gmail.com

Миненко А. С., Лёвкина А. В. Математическое моделирование процесса кристаллизации металла при электрошлаковом переплаве. Исследуется задача Стефана с учетом конвекции в жидкой фазе. Составлено приближенное решение данной задачи с использованием малого параметра. Контроль над процессом выполняется с применением нечеткой логики.

Ключевые слова: электрошлаковый переплав, кристаллизация, тепловой поток, управление, моделирование.

Введение

Изучению тепла посвящено множество работ. Данный процесс имеет значительное влияние на ход протекания многих практически значимых процессов. Задачи класса по описанию распространения тепла, а именно фазовые превращения с выделением или поглощением тепла называются задачей Стефана. В большинстве случаев встречается при процессах плавления и затвердевания. Целью данной работы является математическое моделирование процесса кристаллизации металла при электрошлаковом переплаве (ЭШП) и изучение процесса получения слитка в кристаллизаторе путем его вытягивания.

Рассматривается рафинирование металлов, математическая модель теплопереноса при ЭШП и исследуемые теплофизические процессы с фазовыми переходами вещества. Так же рассмотрен квазистационарный режим переплава. Исследуется математическая модель, основанная на пространственной задаче Стефана, с учетом конвективного движения и примесей в жидкой фазе.

Постановка задачи

Электрошлаковый переплав (ЭШП) является вторичным процессом рафинирования металлов. Он применяется для дальнейшей очистки после завершения первичных операций по удалению примесей и рафинированию. В качестве сходного материала обычно применяется сплошной расходуемый электрод из первичного металла, который может быть литым, полученным обработкой давлением или состоящим из лома. Шлаковая ванна, содержащаяся в охлажденном кристаллизаторе, нагревается и расплавляется электрическим током (электросопротивлением), текущим

между электродом и охлажденным поддоном (см.рис.1) [1].

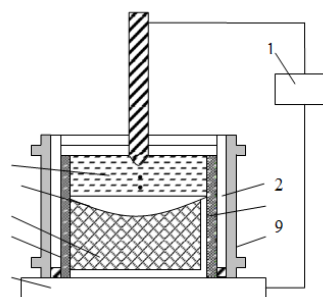


Рисунок 1 – Шлаковая ванна.

Здесь: 1 – трансформатор; 2 – медный водоохлаждаемый кристаллизатор; 3 – шлаковая ванна; 4 – жидкий металл; 5 – затвердевший металл; 6 – слой твердого шлака; 7 – водоохлаждаемый поддон; 8 – вода; 9 – корочка твердого шлака.

Когда температура шлаковой ванны превышает температуру плавления металла, электрод начинает оплавляться; капли, стекающие с конца электрода, падают в шлаковую ванну, образуя на поддоне затвердевающий металлический слой. Электрод попадает в шлаковую ванну, при этом слиток, служащий вторым электродом, постепенно растет. Жидкий шлак, таким образом, непрерывно перемещается вверх. Там, где поднимающийся шлак встречается со стенками охлаждающегося кристаллизатора, он затвердевает, что обеспечивает наличие сплошной корочки твердого шлака между кристаллизатором и затвердевающим слитком. При извлечении слитка она отшелушивается с поверхности, характеризующейся прекрасным качеством. Процесс аналогичен вакуумно-дуговому переплаву (ВДП): сплошной слиток образуется в результате затвердевания металла в вертикальном направлении. При соответствующем снижении силы тока к концу

операции обеспечивается полное отсутствие усадочной раковины и осевой пористости.

Рафинирование осуществляется вследствие реакции между металлом и шлаком, происходящей в 3-х случаях [2, 3]: при образовании капли через шлак; при прохождении отдельной капли металла в ванне, образующейся на поверхности части слитка; после накопления жидкого металла в ванне, образующейся на верхней части слитка.

При соответствующем выборе шлаков химическая реакция может быть усилена. Возможно снижение содержания серы до очень низкого уровня. Удалению неметаллических включений может способствовать их флотация и химическая реакция со шлаком. Шлаки могут быть подобраны таким образом, чтобы воспрепятствовать удалению ценных элементов. При рафинировании металла шлаком наиболее важными параметрами, которые определяют требования к питанию электроэнергией, являются форма и глубина ванны расплавленного металла. Как правило, наиболее предпочтительной является мелкая ванна с большим радиусом кривизны.

Радикальная кристаллизация протекает, при высокой силе тока и низком напряжении, высокой скорости переплава и мелкой шлаковой ванне. Доброкачественный слиток получается при условиях, обеспечивающих осевую кристаллизацию и отсутствие осевой пористости. Математическая модель теплопереноса при ЭШП расходуемых электродов в расходуемые кристаллизаторы учитывает особенности этого процесса, суть которого состоит в следующем (см.рис.2).

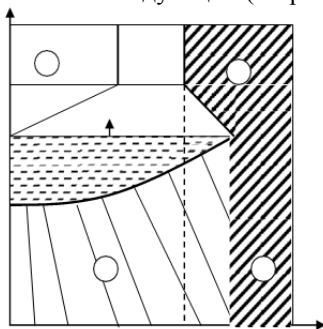


Рисунок 2 – Математическая модель теплопереноса при ЭШП.

Расходуемый электрод 1 плавится в расходуемом кристаллизаторе 2. Жидкий металл переплавляемого электрода и переплавившейся части расходуемого кристаллизатора образует электрошлаковый слиток 3, одетый в оболочку из не расплавившегося металла 4. Глубина проплавления стенки расходуемого кристаллизатора регулируется режимом ЭШП и режимом охлаждения наружной поверхности слитка (кристаллизатора).

Математическое моделирование этого процесса имеет свои особенности, поскольку решение уравнения теплопроводности ищется в сложной области $\{OABCDEF\}$. Основные принципы и подходы, используемые при моделировании теплопереноса обычного процесса ЭШП, здесь сохранены. Модель, приведенная в работе, учитывает сложную область, в которой исследуется теплоперенос, и специфику условий теплообмена на внешних границах области. Инженерные и технологические решения в спец металлургии обнаруживают тенденцию к увеличению размеров и усложнению геометрических форм как оборудования, так и готовых видов продукции, вследствие чего экспериментальные исследования призваны облегчить и ускорить изучение упомянутых процессов, выработать рекомендации по выбору рациональных режимов их протекания и привести к созданию теории оптимального управления, обеспечивающего максимально благоприятную микро- и макро- структуру металла.

Математическое моделирование теплофизических процессов в спец металлургии оперирует такими усредненными величинами: понятие сплошной среды, ее плотность, теплоемкость, температуропроводность, теплопроводность, вязкостью и др. Эти величины, составляющие основные дифференциальные уравнения для определения других усредненных характеристик: векторного поля скоростей, температурного поля и др. Все теплофизические параметры среды в той или иной степени зависят от температуры, а законы, управляющие этими зависимостями, носят экспериментально-эмпирический характер и до настоящего времени весьма неполны, особенно в области высоких температур, поэтому приходится рассматривать линейное приближение, применимое в случае малых перепадов температур, когда зависимостью упомянутых параметров от термодинамического состояния можно пренебречь. Адекватность математической модели физическому явлению возрастает вместе с уточнением упомянутых зависимостей. Линеаризованная теория задач массопереноса и теплопереноса оперирует также с известным количеством безразмерных параметров изучаемого явления, таких как число Рейнольдса, Пекле, Нуссельта, Прандтля и др. Значения этих параметров для возможно большего диапазона материалов и режимов, также приобретают повышенное значение.

Теплофизические процессы сопровождающиеся фазовыми переходами вещества, описываются как математическая модель, в которой температура каждой из фаз удовлетворяет уравнению переноса тепла со своими теплофизическими коэффициентами, на

границе раздела фаз, обе температуры постоянны и равны температуре фазового перехода (для химической однородной среды), а на заданных частях границы (стенках кристаллизатора, поддоне) поддерживается определенный режим (теплопровод, теплоизоляция и др.). Поверхность раздела фаз (фронт кристаллизации) является неизвестной или «свободной» границей, и для ее определения дополнительно задается так называемое «условие Стефана», означающее, что тепловой поток через фронт кристаллизации в сторону твердой фазы равен тепловому потоку со стороны жидкой фазы плюс скрытая теплота фазового перехода. Условие фазового перехода на неизвестной заранее поверхности раздела фаз превращает рассматриваемую математическую модель, особенно в общей пространственной постановке, в нелинейную проблему большой трудности, не нашедшую до сих пор полного решения. Кроме общих качественных исследований задачи Стефана (теорем существования и единственности, асимптотического поведения фронта кристаллизации для больших отрезков времени, свойства гладкости неизвестной границы и др.) большую практическую важность и теоретическую трудность представляют методы приближенного и численного анализа рассматриваемой математической модели.

С целью получения приближенных аналитических формул для описания изотерм был рассмотрен квазистационарный режим процесса переплава, при котором в равномерно движущейся вверх системе координат геометрия изотерм и формы ванночки не меняется с течением времени. Для упрощения кристаллизационной задачи была принята математическая модель при наличии только твердой фазы, тогда как тепловой поток со стороны жидкой фазы моделируется некоторой заданной функцией. Такая модель в случае кристаллизаторов произвольного поперечного сечения (круглых, прямоугольных и т.д.) имеет вариационную природу: отыскание ее решения редуцируется к проблеме минимума некоторого интегрального нелинейного функционала, область интегрирования которого и совпадает с твердой фазой слитка, следовательно, эта область заранее неизвестна. Процесс изучения нелинейных интегральных функционалов с переменной областью интегрирования, представляет весьма содержательный математический объект и стимулируется рядом важных не линейных проблем естествознания: равновесием фигур вращения, струйными, кавитационными и волновыми движениями жидкой среды, отмеченными выше теплофизическими процессами с фазовыми переходами и др. Создание основ

вариационного исчисления для функционалов указанного класса пересекается с рядом магистральных направлений современной математики.

Вариационный характер предложенной модели имеет ряд важных следствий. Первое из них состоит в том, что на этом пути в классе свободных границ, однозначно проектирующихся на горизонтальную поверхность, удается положительно решить проблему единственности, отражающую известную адекватность математической модели в теплофизической задаче с одной стороны, и делающую корректной задачу численного анализа этой модели с другой. Второе следствие состоит в том, что на этом пути прямыми методами вариационного исчисления можно дать полное решение проблемы существования решения.

Рассматриваемая модель характеризуется тремя безразмерными параметрами: числом Пекле ω , пропорциональным скорости протекания переплава и геометрическим размерам слитка; числом Нуссельта ω_0 , характеризующим теплопередачу через стенку кристаллизатора, геометрические размеры сечения слитка и его теплопроводностью; параметром Q , описывающим интенсивность теплового потока со стороны жидкой фазы.

Перейдем к описанию гидродинамических явлений в металлической ванночке. Жидкая фаза рассматриваемого процесса заслуживает специального исследования из-за априорной возможности существования поля скоростей, вызывающего интенсивную теплопередачу путем конвекции. Само же поле скоростей возникает под действием объемных сил земного притяжения, разности плотностей жидкого шлака и металла, архимедовых сил и объемного расширения при значительных температурных градиентах. Также, в случае электрошлакового переплава, сильных электромагнитных сил, способных играть доминирующую роль и сообщать конвекции, наряду с силами тяжести, вынужденный, а не свободный характер. Усиленная циркуляция в расплавленной шлаковой ванночке была обнаружена в исследованиях академиков Б.Е. Патона и Б.И. Медовара [1] и их сотрудников. А дальнейшее экспериментальное изучение этих явлений показало, что интенсивное перемешивание шлакового расплава имеет место по всему объему шлаковой ванночки и вызывается электромагнитными силами. Углубившись детальнее, видим, что под воздействием тока средняя скорость перемешивания частиц шлака достигает 7-10 см/сек, тогда как после отключения тока, несмотря на сохранение теплоотвода через стенки кристаллизатора,

скорости падают до 1 см/сек, т.е. уменьшаются на порядок. Отмеченные величины скорости вынужденной конвекции сравнимы с поперечными и продольными размерами шлаковой ванночки. Следовательно, конвективная теплоотдача в шлаковой ванночке значительно превосходит теплоотдачу через теплопроводность.

Основной темой дальнейшего изложения является гидродинамические и термодинамические явления в металлической, а не шлаковой ванночке, и поскольку их закономерности не менее сложны, а экспериментальные исследования отсутствуют, исходим из предположения, что существует по крайней мере три зоны со своими характеристиками кинематических и температурных полей: зона перегретого металла, поступающего в виде капель или сплошной струйки; пограничный слой вдоль фронта кристаллизации с заданной на нем постоянной температурой; пограничный слой вдоль стенки кристаллизатора при наличии теплоотвода через нее.

Последняя зона из-за теплоотвода через известную часть границы и наличия фазового перехода вдоль неизвестной части границы придает модели температурного пограничного слоя большую теоретическую сложность, поэтому остановимся на результатах исследования первых двух зон. Для описания поля скоростей в зоне поступления перегретого металла может использоваться математическая модель затопленной струи вязкой жидкости, основанная на известном в теоретической гидродинамике точном решении нелинейной системы дифференциальных уравнений Навье-Стокса.

Заключение

В заключении приведем обзор теории по данной тематике [4-8]. В данной статье

Миненко А. С., Лёвкина А. В. Математическое моделирование процесса кристаллизации металла при электрошлаковом переплаве. Исследуется задача Стефана с учетом конвекции в жидкой фазе. Составлено приближенное решение данной задачи с использованием малого параметра. Контроль над процессом выполняется с применением нечеткой логики.

Ключевые слова: электрошлаковый переплав, кристаллизация, тепловой поток, управление, моделирование.

Minenko A. S., Lovkina A. V. Mathematical modeling of process crystallization of the metal during electroslag remelting. The Stefan problem is investigated taking into account convection in the liquid phase. The approximate solution of the given problem with use of small parameter is made. Control over the process is performed using fuzzy logic.

Keywords: electroslag remelting, crystallization, heat flow, control, modeling.

смоделирован процесс кристаллизации металла при электрошлаковом переплаве. Это позволит в дальнейшем применить уравнение теплопроводности и условие Стефана для получения приближенного анализа процесса кристаллизации и приведет к получению необходимых значений теплового потока необходимого для создания слитка в кристаллизаторе электрошлакового переплава.

Литература

1. Меджибожский М. Я. Основы термодинамики и кинетики сталеплавильных процессов/Меджибожский М. Я. – Киев-Донецк: Вища школа, 1986. – 280 с.
2. Миненко А. С. Вариационные задачи со свободной границей/А. С. Миненко. – Киев : Наукова думка, 2005. – 354с.
3. Миненко А. С. Исследование осесимметричного течения со свободной границей/А.С. Миненко // Нелинейные граничные задачи. – 1993. – 65-71с.
4. Патон Б. Е. Избранные труды/ Б. Е. Патон – Киев : Институт электросварки им. Е.О. Патона НАН Украины, 2008. – 893 с.
5. Миненко А. С. Исследование одной конвективной задачи Стефана методом Ритца / А.С. Миненко // Укр. мат. журнал. – 2007. – №11. – С. 1546-1556.
6. Миненко А. С. Об одной проблеме Стефана / А.С. Миненко, А.И. Шевченко // Доповіді НАН України. – 2008. – № 1. – С. 26-30.
7. Миненко А. С. Проблема Минимума со свободной границей / А.С. Миненко // Искусственный интеллект. – 1998. – № 2. – С. 101-109.
8. Миненко А. С. Приближенный анализ стационарной конвективной задачи Стефана / А.С. Миненко, А.И. Шевченко // Доповіді НАН України. – 2010. – № 5. – С. 36-40.

Статья поступила в редакцию 15.05.2018 г

Рекомендована к публикации доктором технических наук Павлышом В.Н.

Обзор алгоритмов аутентификации в распределенных программных системах

О. С. Грищенко, А. В. Чернышова
Донецкий национальный технический университет
кафедра программной инженерии
alla@donntu.org, olleg1996@mail.ru

Грищенко О. С., Чернышова А. В. Обзор алгоритмов аутентификации в распределенных программных системах. Представлен обзор существующих алгоритмов аутентификации, используемых в современных распределенных системах. Описаны стандарты в данной области. Рассмотрен принцип работы распространенных протоколов аутентификации.

Ключевые слова: распределенная система, аутентификация, TLS, сертификат, однонаправленная аутентификация, RSA, Kerberos.

Постановка задачи

Аутентификация является важной частью любой распределенной программной системы. На сегодняшний день существует несколько основных подходов к аутентификации в программных системах и множество вариантов реализации этих подходов. Не все алгоритмы аутентификации обеспечивают одинаковый уровень защищенности и обычно имеют свою область применения. В связи с этим было решено рассмотреть алгоритмы и протоколы аутентификации, которые используются в современных распределенных программных системах, оценить их достоинства и недостатки.

Цель работы

Целью данной работы является анализ подходов к аутентификации в программных системах, рассмотрение принципов работы алгоритмов аутентификации, используемых в современных распределенных программных системах, исследование существующих стандартов в области аутентификации.

Основные способы аутентификации

Наиболее простым способом аутентификации является аутентификация по паролю. При регистрации на ресурсе пользователь задает свой логин (имя, телефон, email) и пароль. При аутентификации необходимо проверить соответствие логина и пароля. Такой способ аутентификации дает достаточно слабый уровень защиты. Прежде всего, пользователь сам задает и хранит пароль. При этом пароль может быть коротким (так что его можно будет легко подобрать простым перебором) или предсказуемым. Например, по состоянию на 2016 год в результате исследования 10000000 пользовательских

паролей было выяснено, что самыми популярными являются такие пароли как "123456", "123456789", "qwerty", "12345678". Причем пароль "123456" использовали 17% пользователей [1]. Взлом таких паролей с помощью современных инструментов подбора паролей займет не более нескольких секунд.

Можно добиться большей защищенности путем использования специальных генераторов паролей, с помощью регулярной смены пароля и за счет использования различных паролей для разных ресурсов. Можно выделить требования к системам, реализующим аутентификацию пользователя по паролю, которые помогут значительно повысить защищенность таких систем:

- приложение не должно позволять создавать простые пароли;
- приложение должно предусматривать защиту от перебора паролей;
- если в процессе работы приложения пароль передается по сети, то при передаче должно использоваться защищенное соединение;
- пароли не должны храниться в открытом виде;
- приложение должно предоставлять надежную процедуру изменения пароля;
- во время выполнения критических действий должна запрашиваться повторная аутентификация.

RFC 7235 [2] описывает протокол аутентификации для HTTP на основе логина и пароля. При обращении неавторизованного клиента к защищенному ресурсу, сервер отправляет ответ "401" и добавляет в заголовок описание схемы аутентификации целевого ресурса. Получив такой ответ, клиент запрашивает у пользователя логин и пароль и отправляет эти параметры серверу в поле Authorisation. Сервер проводит аутентификацию пользователя по полученным данным. Далее поле Authorisation добавляется ко всем запросам

на данный сервер. Описанный обмен данными должен проходить с использованием протокола TLS, так как эта схема не обеспечивает защиту пользовательских данных.

Часто при аутентификации по паролю используются токены. После аутентификации по паролю сервер генерирует некоторую информацию – токен (обычно на основе идентификатора сессии или данных пользователя) и отправляет клиенту. К последующим запросам пользователя автоматически добавляется токен, на основе которого сервер проводит авторизацию. Для усиления безопасности токен может быть подписан и зашифрован сервером и иметь срок действия. Данный подход позволяет отказаться от передачи пользовательских данных при каждом запросе и даже в случае перехвата токена злоумышленник сможет использовать его только в текущей сессии.

Более безопасным подходом к аутентификации является использование системы PKI (инфраструктура открытых ключей).

Основными компонентами PKI системы являются центр сертификации и пользователи. Пользователи взаимодействуют между собой, используя сертификаты, выданные центром сертификации. Понятие сертификатов определено в RFC5280 [3] (стандарт X.509). Для обеспечения аутентификации на основе сертификатов используются криптографические алгоритмы с открытым ключом. Сертификат представляет собой набор данных, идентифицирующий владельца, а также его открытый ключ.

Для использования на территории России RFC 4491 описывает сертификат, использующий российские криптографические алгоритмы GOST R 34.10-94, GOST R 34.10-2001, GOST R 34.11-94. [4]

Далее описан общий принцип аутентификации на основе сертификатов. Допустим есть участники информационного обмена А и В и центр сертификации ЦС. А генерирует открытый и закрытый ключи и получает сертификат у ЦС, содержащий открытый ключ А. А отправляет В сертификат. В проверяет подпись сертификата с помощью открытого ключа ЦС. После этого В может передавать данные, шифруя их с помощью открытого ключа А, содержащегося в сертификате. При этом центр сертификации должен пользоваться доверием у обоих участников информационного обмена.

На практике обычно используются иерархические PKI системы, состоящие из множества пользователей и центров сертификации. Центры сертификации создают иерархию, где нижестоящие центры доверяют вышестоящим. Вышестоящие ЦС выдают

сертификаты нижестоящим. Корневым в этой структуре является головной центр сертификации, который сам подписывает себе сертификат. Пользователи, желающие обменяться информацией, могут не принадлежать одному и тому же центру сертификации. В таком случае пользователь А не может доверять сертификату пользователя В, выданному неизвестным ЦС. Для решения этой проблемы необходимо построить путь сертификации от проверяемого сертификата до любого из доверенных центров. Например, пользователь А пытается проверить сертификат пользователя Е. А доверяет центру сертификации Б. Б подписал сертификат В, В подписал сертификат Г, а Г подписал сертификат Е. Так как существует цепь доверия от Б к Е (Б ->В->Г->Е) пользователь А может доверять сертификату пользователя Е.

Сертификат может быть отозван. В таком случае он помещается в CertificateRevocationList (список аннулированных сертификатов).

Государственные стандарты в области аутентификации

Стандарт аутентификации описывает ГОСТ Р ИСО/МЭК 9594-8-98 — Основы аутентификации [5]. Данный ГОСТ описывает 2 варианта аутентификации: простая на основе пароля и строгая, использующая асимметричные криптографические алгоритмы. Простая аутентификация предназначена в основном для локального использования.

Простая аутентификация может быть выполнена несколькими способами:

- передача имени пользователя и пароля в открытом виде получателю для проверки;
- передача хэша от имени пользователя, пароля и некоторой дополнительной информации;
- передача информации, описанной в предыдущем пункте, вместе со случайным числом и/или отметкой времени после применения однонаправленной функции.

Второй и третий способы представлены на рисунке 1. Эти способы простой аутентификации обеспечивают защиту передаваемых данных.

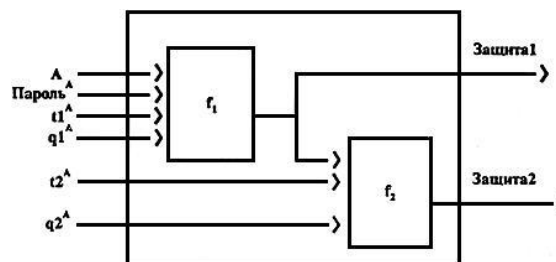


Рисунок 1 – Простая защищенная аутентификация

Здесь A - имя пользователя, t - отметка времени, q^{\wedge} - случайные числа, $f1$ и $f2$ однонаправленные функции.

Незащищенные пароли предполагают минимальную степень безопасности и не могут рассматриваться как основа услуг аутентификации.

Строгая аутентификация основана на использовании ассиметричных криптографических систем. Пользователи идентифицируются на основе владения личным ключом. Для того, чтобы проверить, обладает ли собеседник личным ключом другого пользователя, необходимо обладать ключом общего пользования этого пользователя. Для того, чтобы проверить принадлежность открытых ключей пользователей, должна существовать непрерывная цепочка доверительных точек. Для построения такой цепочки выполняется поиск общей точки доверия. Уполномоченный по сертификации (аналог центра сертификации РКІ системы), далее УС, создает сертификат пользователя, подписывая следующую информацию: имя пользователя, ключ общего пользования, и, опционально, уникальный идентификатор. Для того, чтобы любой пользователь имел доступ к ключу общего пользования УС, запись каждого уполномоченного по сертификации содержит сертификаты двух типов. Первый тип – это срочные сертификаты, созданные другими уполномоченными по сертификации. Второй тип – это реверсивные сертификаты. Сертификаты этого типа уполномоченный по сертификации выдает себе сам.

Список сертификатов, необходимый для того, чтобы пользователь мог получить общий ключ другого пользователя, называется "путь сертификации". Каждый элемент такого списка является сертификатом уполномоченного по сертификации следующего элемента в списке. В общем случае, перед выполнением аутентификации строится путь сертификации. Далее описаны процедуры строгой аутентификации.

Однонаправленная аутентификация использует одну передачу информации от пользователя A пользователю B и позволяет установить подлинность участников обмена, происхождение и получателей маркера аутентификации, целостность и новизну маркера аутентификации.

В двунаправленной аутентификации используется дополнительный ответ от B к A . Двунаправленная аутентификация дополнительно устанавливает, что маркер аутентификации ответа был сгенерирован B и предназначен для A , целостность и новизну маркера аутентификации пользователя B .

При трехнаправленной аутентификации добавляется дополнительная передача от A к B . Трехнаправленная аутентификация устанавливает те же свойства, что двунаправленная, но не нуждается в проверке меток времени.

Для каждой из описываемых ниже процедур аутентификации предполагается, что сторона A выполнила проверку всех сертификатов в пути аутентификации.

При однонаправленной аутентификации выполняются следующие шаги:

1) A создает rA (неповторяющийся номер), который используется для обнаружения повторных угроз и предотвращения подделок;

2) A посылает к B сообщение, содержащее следующие данные:

- путь сертификации от B к A ($B \rightarrow A$)
- отметка времени (tA), содержащая дату истечения срока действия маркера;
- rA ;
- имя или численный идентификатор B ;
- данные, аутентификация отправителя которых обеспечивается с помощью подписи;
- данные, в последствии используемые в качестве личного ключа, зашифрованные с помощью открытого ключа B .

На рисунке 2 представлена структура передаваемого сообщения.

$B \rightarrow A, A \{t^A, r^A, B, \text{sgnData}, B_o [\text{encData}]\}$.

Рисунок 2 – Структура сообщения при однонаправленной аутентификации

Здесь $A\{\text{данные}\}$ означает данные, подписанные A .

Пользователь B выполняет следующие действия:

- получает публичный ключ A , проверяет, что срок сертификата пользователя A не истек;
- проверяет целостность полученной информации;
- проверяет, что информация предназначена для передачи ему;
- проверяет, что отметка времени имеет значение "текущее";
- может проверить, что rA повторно не использован.

Двунаправленная аутентификация на первых этапах происходит так же, как и однонаправленная, но после получения маркера от A , B отправляет сообщение, содержащее метку времени tB , имя или численный идентификатор A , rA , данные, аутентификация отправителя которых обеспечивается с помощью подписи и данные, которые могут использоваться, в качестве личного ключа (зашифрованные открытым ключом B). На рисунке 3 представлена структура сообщения, передаваемого от B к A .

$B \{t^B, r^B, A, r^A, \text{sgnData}, \text{Ap}[\text{encData}]\}$.

Рисунок 3 - Структура сообщения,
передаваемого от В к А

Пользователь А проверяет целостность полученной информации, проверяет, что он является получателем, проверяет, что метка времени имеет значение "текущее". Также А может проверить, что r^A не используется повторно. При трехнаправленной аутентификации происходит передача сообщений от А к В и от В к А как в случае двунаправленной аутентификации, за исключением того, что метки времени t^A и t^B могут быть нулевыми и не должны проверяться. Затем пользователь А проверяет соответствие переданного r^A полученному и передает В маркер аутентификации, представленный на рисунке 4.

$A \{r^B, B\}$.

Рисунок 4 – Повторный маркер аутентификации,
передаваемый от А к В

В проверяет целостность полученной информации и сверяет полученный и отправленный r^B .

Ранее также использовались американские стандарты аутентификации FIPS 113 — COMPUTER DATA AUTHENTICATION и FIPS 196. Стандарт FIPS 113 описывал аутентификацию данных на основе алгоритма DES и был отозван в 2008 году как не удовлетворяющий текущим требованиям безопасности. Стандарт FIPS 196 описывал аутентификацию, основанную на открытом ключе, и был отозван в 2015 году по той же причине. Действующим стандартом остается алгоритм HMAC (FIPS PUB 198) [6]. Этот алгоритм описывает процедуру подписи данных на основе общего ключа. Алгоритм HMAC можно описать следующей формулой, представленной на рисунке 5.

$$\text{HMAC}_K(\text{text}) = \text{H} \left((K \oplus \text{opad}) \parallel \text{H} \left((K \oplus \text{ipad}) \parallel \text{text} \right) \right)$$

Рисунок 5 – Формирование подписи на основе
HMAC

Здесь $\text{H}()$ обозначает применение некоторой хэш-функции, text - подписываемые данные, K – ключ (дополненный или урезанный до размера блока), opad – блок вида (0x5c, 0x5c...0x5c), ipad – блок вида (0x36, 0x36 ... 0x36). Ключи, длина которых меньше половины длины хэша, возвращаемого H , считаются небезопасными. Реализация HMAC является обязательной для IPsec. Также алгоритм используется в протоколе TLS. Главный недостаток HMAC – требуются две доверяющие друг другу стороны (знающие общий ключ).

Описание современных протоколов аутентификации

Протокол TLS использует для аутентификации пользователей сертификаты, а для аутентификации и сохранения целостности данных HMAC.TLS ориентирован на обеспечение безопасности соединения в клиент – серверных приложениях. Процедура аутентификации и согласования ключей TLS состоит из следующих шагов:

– установление TCP соединения между клиентом и сервером;

– клиент отправляет на сервер спецификацию, содержащую версию протокола, которую он хочет использовать, поддерживаемые алгоритмы шифрования и хеш-функции;

– сервер утверждает версию используемого протокола, способ шифрования, прикрепляет свой сертификат и отправляет ответ клиенту, также сервер может запросить сертификат клиента;

– клиент проверяет сертификат сервера и инициирует обмен ключами с помощью алгоритма RSA или Диффи-Хеллмана (в зависимости от установленных параметров);

– сервер обрабатывает сообщение клиента, сверяет MAC, и отправляет клиенту сообщение ('Finished') в зашифрованном виде;

– клиент расшифровывает сообщение и сверяет MAC.

Предпочтительным считается согласование ключа по алгоритму Диффи-Хеллмана, так как в этом случае ключ не передается между клиентом и сервером. Протокол Kerberos использует описанный ниже алгоритм. Для получения доступа к какому-либо сервису клиент направляет запрос в службу KDC, которая хранит информацию об учетных записях всех клиентов сети и общие с этими клиентами долговременные ключи шифрования. Запрос содержит следующую информацию: идентификатор клиента, метку времени и идентификатор сервера. Служба KDC генерирует сеансовый ключ, действительный на протяжении некоторого времени. KDC отправляет клиенту ответ, зашифрованный с помощью долговременного ключа общего с клиентом. Ответ содержит сеансовый ключ, идентификатор сервиса, время жизни ключа и мандат. Мандат содержит данные о клиенте, метку времени и сеансовый ключ. Мандат шифруется долговременным ключом, общим с сервером. Получив ответ KDC, клиент извлекает из него мандат и свою копию сеансового ключа. После этого клиент посылает серверу сообщение, состоящее из мандата (который зашифрован долговременным ключом сервера), и своего аутентификатора, зашифрованного с помощью сеансового ключа. Сервер с помощью своего

долговременного секретного ключа расшифровывает сеансовый мандат и извлекает из него сеансовый ключ и данные о клиенте. С помощью полученного сеансового ключа сервер расшифровывает аутентификатор клиента и сверяет с информацией, содержащейся в мандате. Клиент может потребовать у сервера проведения взаимной аутентификации. В таком случае сервер шифрует с помощью сеансового ключа метку времени из аутентификатора клиента и пересылает её клиенту. Также для Kerberos существует расширение PKI NIT, позволяющее проводить описанную процедуру с использованием асимметричного шифрования. В этом случае появляются такие отличия:

1. При создании запроса к KDC указывается, что будет использоваться асимметричное шифрование. Сам запрос подписывается и кроме стандартной информации содержит сертификат пользователя.

2. Получив запрос, KDC проверяет достоверность сертификата и электронную подпись. После этого KDC проверяет локальное время, присланное в запросе (для защиты от повторов).

3. После проверки подлинности клиента, KDC формирует ответ, зашифрованный открытым ключом пользователя. Кроме того ответ подписывается и содержит сертификат KDC.

4. Получив ответ, пользователь проверяет подпись KDC и расшифровывает свой сеансовый ключ.

Выводы

Был проведен обзор существующих алгоритмов аутентификации, были рассмотрены стандарты в области аутентификации и принципы работы распространенных протоколов аутентификации. Можно сделать вывод, что наиболее используемыми способами

аутентификации являются аутентификация по паролю и аутентификация на основе сертификатов. Эти виды аутентификации используются современных протоколах аутентификации и описаны в государственных стандартах. При этом аутентификация на основе сертификатов является более защищенной, но при этом более сложна, чем аутентификация по паролю и требует наличия инфраструктуры.

Литература

1. Mostcommonpasswords [Электронный ресурс]. – Режим доступа: <https://keepersecurity.com/public/Most-Common-Passwords-of-2016-Keeper-Security-Study.pdf>
2. Hypertext Transfer Protocol (HTTP/1.1): Authentication passwords [Электронный ресурс]. – Режим доступа: <https://tools.ietf.org/html/rfc7235>
3. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile [Электронный ресурс]. – Режим доступа: <https://www.ietf.org/rfc/rfc5280.txt>
4. Using the GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms with the Internet X.509 Public Key Infrastructure [Электронный ресурс]. – Режим доступа: <https://tools.ietf.org/html/rfc4491>
5. ГОСТ Р ИСО/МЭК 9594-8-98 Информационная технология [Электронный ресурс]. – Режим доступа: <http://docs.cntd.ru/document/1200028710>
6. HMAC: Keyed-Hashing for Message Authentication [Электронный ресурс]. – Режим доступа: <https://tools.ietf.org/html/rfc2104.html>
7. The Transport Layer Security (TLS) Protocol [Электронный ресурс]. – Режим доступа: <https://tools.ietf.org/html/rfc5246>
8. The Kerberos Network Authentication Service [Электронный ресурс]. – Режим доступа: <https://tools.ietf.org/html/rfc1510>

Грищенко О. С., Чернышова А. В. Обзор алгоритмов аутентификации в распределенных программных системах. Представлен обзор существующих алгоритмов аутентификации, используемых в современных распределенных системах. Описаны стандарты в данной области. Рассмотрен принцип работы распространенных протоколов аутентификации.

Ключевые слова: *распределенная система, аутентификация, TLS, сертификат, однонаправленная аутентификация, RSA, Kerberos.*

Grichenko O. S., Chernyshova A. V. Overview of authentication algorithms in distributed software systems. An overview of the existing authentication algorithms used in modern distributed systems is presented. The standards of authentication are described. The principle of operation of authentication protocols is considered.

Keywords: *distributed system, authentication, TLS, certificate, unidirectional authentication, RSA, Kerberos.*

Статья поступила в редакцию 18.06.2018 г.
Рекомендована к публикации доцентом Федяевым О.И.

О проектировании виртуальных ресурсов облачной образовательной инфраструктуры

Д. П. Жданович¹, П. Б. Жданович²

Волгоградский государственный социально-педагогический университет¹

Волгоградский государственный медицинский университет²

dmzhd@yandex.ru¹, pavel.b.zhdanovich@gmail.com²

Жданович Д. П., Жданович П. Б. О проектировании виртуальных ресурсов облачной образовательной инфраструктуры. В настоящей статье мы предлагаем набор правил для проектировании виртуальных машин в образовательной облачной инфраструктуре. Мы также приводим пример классификации виртуальных машин, не зависящей от конфигурации, которую полезно использовать при планировании виртуальной инфраструктуры.

Ключевые слова: дистанционное обучение, IaaS, облачные технологии, лицензирование, свободное программное обеспечение, виртуальная машина.

Введение

Облачные технологии и виртуализация на протяжении нескольких последних лет назывались аналитиками в числе основных трендов развития мировых информационных технологий. В настоящее время они применяются уже вполне рутинно, но без них невозможно представить многое из того, что относят к основным направлениям развития ИТ сегодня [1].

Виртуальные ИТ-инфраструктуры создаются организациями на базе собственных ЦОДов (частное облако) или арендуются у провайдеров (публичное облако). В последнем случае речь идет об IaaS (infrastructure as a service, инфраструктура как услуга). IaaS относится к наиболее распространенным моделям предоставления облачных ИТ-услуг, наряду с PaaS (платформа как услуга) и SaaS (программное обеспечение как услуга). В IaaS заказчику предоставляется доступ к виртуальным машинам, сетевому оборудованию, устройствам хранения и др. Провайдер ИТ-инфраструктуры осуществляет развертывание в своем дата-центре необходимых виртуальных машин, выделяет физические ресурсы, обеспечивает отказоустойчивость, безопасность и требуемый уровень сервиса, предоставляет интерфейсы управления.

В сфере высшего профессионального образования виртуальные инфраструктуры весьма эффективно применяются при изучении информационных технологий. В виртуальной среде могут выполняться как отдельные лабораторные работы, так и комплексные ИТ-проекты. В совокупности с другими технологиями это позволяет реализовать полноценное дистанционное обучение на профильных специальностях. В некотором

смысле использование облачной инфраструктуры вообще стирает грань между «дистанционным» и «традиционным» обучением и попутно решает многие проблемы, с которыми сталкиваются образовательные организации при эксплуатации существующей ИТ-инфраструктуры [2]. Вот почему многие высшие учебные заведения по всему миру активно внедряют у себя облачные технологии, а некоторые из них сами вносят существенный вклад в их развитие [3].

Вопросам перехода к виртуальной ИТ-инфраструктуре посвящено большое количество публикаций и интернет-ресурсов. Однако в сфере образования такой переход имеет свои особенности, которые не позволяют напрямую применять многие общие подходы, выработанные для бизнеса. В настоящей работе предпринимается попытка выделить эти особенности, сформулировать правила, которыми целесообразно руководствоваться на этапе проектирования виртуальных ресурсов образовательного облака, и приводятся примеры их использования.

Некоторые особенности образовательной ИТ-инфраструктуры

Всюду далее будет рассматриваться ИТ-инфраструктура масштаба факультета высшего учебного заведения, на котором осуществляется подготовка ИТ-специалистов.

В традиционной образовательной ИТ-среде студент обычно выполняет задания на персональном компьютере, оснащенном всем необходимым прикладным программным обеспечением. Такой компьютер либо находится в лаборатории (компьютерном классе), либо принадлежит самому студенту, который в этом случае берет на себя обязательство соблюдать политику ограниченного использования коммерческого

ПО, если оно предоставлено ему вузом.

Недостатки использования традиционной ИТ-инфраструктуры хорошо известны. Одна из основных носит фольклорное название «компьютерный зоопарк» и заключается в том, что в организации используются компьютеры, приобретенные в разное время. Они отличаются друг от друга аппаратными ресурсами, версиями системного и прикладного ПО. Это сильно затрудняет централизованное управление, исключает унификацию ПО, создает проблемы совместимости и привязывает преподавание конкретной дисциплины к определенной совокупности рабочих станций. Каждый учебный компьютер используется ежедневно многими пользователями, и поэтому каждая рабочая

станция должна быть, по возможности, укомплектована всем программным обеспечением, которое может потребоваться для проведения занятия по любой теме, и обладать соответствующими аппаратными ресурсами. Таким образом, традиционная ИТ-инфраструктура образовательного учреждения тяжела в управлении, ее надежность снижается из-за проблем совместимости, а стоимость владения максимально высока. При более внимательном рассмотрении видно, что традиционная образовательная ИТ-инфраструктура эксплуатируется в более сложных условиях, чем «типовая» ИТ-инфраструктура в бизнесе. В таблице 1 приводятся наиболее значимые отличия.

Таблица 1. Отличия образовательной ИТ-инфраструктуры и ИТ-инфраструктуры в бизнесе

Образование	Бизнес
Каждая рабочая станция в течение дня может использоваться несколькими студентами	Обычно рабочая станция входит в состав рабочего места одного сотрудника
Каждый студент в течение дня может работать на нескольких рабочих станциях	Обычно пользователь работает постоянно на одной и той же рабочей станции или использует одно и то же мобильное устройство
Каждая рабочая станция должна быть оснащена всеми возможными программами, которые должны быть использованы в учебном процессе	Пользователю доступен ограниченный набор программ или даже единственная программа (kiosk mode)
Для выполнения учебных заданий студентам часто требуются права администратора	Права администратора даются пользователю лишь в исключительных случаях
При изучении ряда предметов (компьютерные сети, информационная безопасность) студенты должны изменять системные настройки, что может оказать влияние на работу других студентов, пришедших на следующее занятие	Пользователям, как правило, запрещено изменять конфигурацию их рабочей станции. Максимум настроек стараются задавать, используя централизованное управление (групповая политика и т.п.)
В ходе обучения студентам может потребоваться несколько операционных систем	На каждом рабочем месте используется, как правило, одна операционная система.

Полный или частичный перевод образовательной ИТ-инфраструктуры в облако позволяет преодолеть как перечисленные выше, так и многие другие проблемы. Однако попытка напрямую скопировать существующую инфраструктуру, перенеся в облако имеющиеся объекты путем прямой виртуализации, приведет к обратному эффекту: существующие проблемы сохранятся, инфраструктура усложнится, а производительность снизится

Частные или публичные облака

При организации обучения с применением облачных технологий всегда возникает проблема выбора: организовать собственный дата-центр, (частное облако) или воспользоваться услугами стороннего IaaS-провайдера. В работе [4] мы тезисно сформулировали аргументы в пользу частного облака, которое создается и управляется университетом, осуществляющим обучение. Необходимо рассмотреть этот вопрос более подробно, так как выбор между созданием

частного облака и арендой инфраструктуры является стратегическим, он определяет все дальнейшие технологические и организационные решения, связанные с применением облачной технологии в обучении.

Аренда инфраструктуры позволяет за минимальное время развернуть необходимые приложения для работы пользователей без создания собственного ЦОДа.

В то же время следует учитывать, что в большинстве случаев IaaS-услуги ориентированы на бизнес, который являлся и до сих пор является основным заказчиком в этой сфере. Со временем ведущие игроки на рынке облачных инфраструктур объявили о поддержке образовательной и научной деятельности, разработав специальные условия использования своих ресурсов студентами, преподавателями и научными работниками. Однако сами модели предоставления услуг не ориентированы на поддержку учебного процесса.

Одно из последствий перехода к облачным технологиям в образовании заключается в том, что, с одной стороны, количество виртуальных машин будет существенно превосходить количество обучающихся. С другой стороны, значительная, если не большая часть виртуальных машин будет использоваться довольно редко, как правило, не более двух академических часов в день, и очень ограниченное время, не более одного семестра. Впрочем, часть виртуальных ресурсов будет использоваться и постоянно, и интенсивно. Далеко не каждый IaaS-провайдер может предложить вузу услугу, обеспечивающую заказчику возможность:

- управлять большой библиотекой шаблонов виртуальных машин,
- быстро тиражировать шаблоны на всех студентов группы или курса,
- управлять резервным копированием в зависимости от назначения виртуальной машины,
- управлять лицензиями на ПО с учетом сложного расписания работы виртуальных машин.

Один из самых распространенных аргументов в пользу IaaS (в некоторых случаях он является единственным) заключается в отсутствии квалифицированного ИТ-персонала для обеспечения работы частного облака. Когда речь идет об учебном заведении, где студенты получают высшее образование в области ИТ, кадровой проблем, скорее всего, удастся избежать. Более того, управление собственным ЦОДом, пусть на первых порах небольшим, способствует подготовке квалифицированных специалистов по целому спектру современных информационных технологий.

Наконец, вуз, построивший собственное

частное облако с виртуальной инфраструктурой, может сам стать IaaS-провайдером для других учебных заведений и частных лиц.

Типы виртуальных машин и некоторые правила их проектирования

К физическим ресурсам дата-центра относят [5]:

- вычислительные узлы,
- устройства хранения,
- каналы связи.

Пополним этот перечень, считая физическими ресурсами коммерческие лицензии, позволяющие запустить ограниченное число копий определенного программного обеспечения.

Необходимо построить такую классификацию виртуальных машин для поддержки учебного процесса, которая помогала бы:

- описать жизненный цикл виртуальной машины,
- управлять доступом обучающихся к виртуальным ресурсам,
- осуществить планирование физических ресурсов.

Классифицировать можно по многим основаниям. Мы остановимся на следующих двух.

1. По длительности эксплуатации:
 - ✓ краткосрочного использования (несколько дней). Это виртуальные машины, которые используются при изучении конкретной темы для выполнения лабораторно-практических работ,
 - ✓ среднесрочного использования (1-2 семестра) для изучения курса, выполнения курсовых и семестровых работ,
 - ✓ долгосрочного использования (свыше 1 года).
2. По типу доступа:
 - ✓ Индивидуальная ВМ – предназначенная для использования одним обучающимся,
 - ✓ ВМ для рабочих групп – для использования в групповой работе с заранее известными участниками,
 - ✓ ВМ общего доступа – для использования не определенной заранее группой обучающихся.

Заметим, что отнесение ВМ к некоторому типу доступа в общем случае не определяет права доступа. Например, помимо самого студента естественно было бы обеспечить доступ к его ВМ и его преподавателю.

Рассмотрим пример двумерной классификации виртуальных машин при

использовании выбранных критериев (табл. 2).

Таблица 2. Классификация виртуальных машин

Тип доступа \ Длительность	Индивидуальная (И)	Для рабочих групп (Г)	Общего доступа (О)
Краткосрочного использования (К)	Рабочие станции для выполнения упражнений по изучаемой теме	ВМ для демонстрации работы ПО в рамках изучения одной темы	Обеспечение конференций и других краткосрочных мероприятий
Среднесрочного использования (С)	Рабочие станции для выполнения курсовой работы	СУБД, HTTP, DNS-серверы и другие общие ресурсы для поддержки соотв. учебного курса	ВМ для демонстрации работы ПО, для открытых проектов с определенными сроками
Долгосрочного использования (Д)	Личная ВМ студента	ВМ для поддержки групповых исследовательских проектов	Общедоступные сервисы: CIFS, HTTP, LDAP и т.д

Теперь мы можем сформулировать правила создания виртуальных машин, которые будут обсуждаться в последующих параграфах.

1. Если виртуализация возможна, то она необходима [6].
2. Правило минимальности.
3. Правило сегрегации.
4. Свободное программное обеспечение имеет приоритет.

Необходимость виртуализации

Виртуальная машина существует физически в виде одного или нескольких т. н. виртуальных носителей. Ими могут быть как обычные файлы, так и физические разделы жесткого диска гипервизора, неразмеченные жесткие диски, LVM-тома.

Считается, что при равных аппаратных ресурсах производительность виртуальной машины может быть в пределах нескольких процентов ниже, чем у системы без виртуализации [7,8], в зависимости от приложения, гипервизора (VMWare ESXi, Xen, KVM и т.п.) и типа виртуальных носителей. Такое падение производительности в большинстве случаев приемлемо. При этом виртуализация обладает следующими преимуществами.

Быстрота установки. Кардинальное сокращение периода развертывания достигается за счет клонирования виртуальной машины и ее конфигурации. Это позволяет создать набор шаблонов, а затем создавать из них виртуальные

машины и удалять их по требованию.

Управляемость. Динамическое управление аппаратными ресурсами заключается в изменении конфигурации машины путем присоединения к ней новых виртуальных носителей и других ресурсов, например, процессорных ядер, памяти, периферийных устройств.

В некоторых случаях для этого даже не требуется перезагрузка гостевой системы. Можно осуществлять «живую» миграцию виртуальной машины на другой гипервизор, которая происходит практически незаметно для пользователя, или просто приостановить машину, если она в данный момент не нужна.

Отказоустойчивость. Резервное копирование/восстановление виртуальной машины осуществляется гораздо легче, чем не виртуальной.

В качестве причины возможного отказа следует рассматривать теперь не только выход из строя виртуальной машины, но и возникновение исключительной ситуации на гипервизоре. В случае использования отказоустойчивого кластера гипервизоров срабатывает механизм fencing, и соседний узел берет управление виртуальной машиной на себя. Для последней данная ситуация означает, в худшем случае, незапланированную перезагрузку.

Если кластера нет, то восстановление производится в два этапа: переустановка программного обеспечения гипервизора (завершается относительно быстро) и

восстановление виртуальной машины из резервной копии или ее создание из шаблона. Даже в этом случае восстановление проходит в несколько раз быстрее, чем восстановление рабочей станции с полным набором программного обеспечения.

Доступность. Работа с виртуальной машиной осуществляется по протоколам удаленного доступа, поддержка которых может быть обеспечена как внутри гостевой системы, так и на гипервизоре. Используя удаленные подключения к сети университета, студенты могут получать удаленный доступ к своим виртуальным машинам в течение всего времени ее эксплуатации.

Безопасность. Виртуальная система защищена гораздо лучше, чем система без виртуализации, в силу следующих обстоятельств.

- Физическая безопасность обеспечивается лишь в границах дата-центра.
- Правила для сетевых протоколов можно (и предпочтительнее) задавать на межсетевом экране гипервизора, если функционал встроенного МСЭ виртуальной машины недостаточно.
- Подключение съемных носителей контролируется администратором гипервизора.

На протяжении ряда лет принцип «Virtualize everything» («виртуализировать всё») превратился в доминирующий, и случаи, когда виртуализация невозможна или нецелесообразна, рассматриваются отдельно как исключения [9].

Правило минимальности

Это и следующие правила состоят из нескольких пунктов, естественным образом согласованных друг с другом.

1. Виртуальная машина существует ровно столько времени, сколько она используется. После этого машина должна быть удалена или перенесена в оффлайн-архив.

2. Количество физических ресурсов (процессорное время, оперативная память, полоса пропускания канала), выделяемое виртуальной машине, должно быть не более рекомендуемого для запускаемых на ней задач.

3. Каждый виртуальный носитель нужно делать как можно меньше, оставляя на нем минимум свободного места.

Неоправданно большие виртуальные носители расходуют место на устройствах хранения и медленнее копируются во время операций создания и резервного копирования/восстановления.

Отметим, что в случае необходимости можно легко изменить размер виртуального носителя штатными средствами гипервизора.

Правило сегрегации

1. Одна виртуальная машина — одна задача. На виртуальной машине необходимо устанавливать только те прикладные программы, которые образуют набор взаимосвязанных приложений для решения одной учебной задачи. Например, если мы сегодня обучаем студентов экспортировать данные в документы LibreOffice из Java-приложения, нам требуется JRE, одна из IDE (например, Eclipse), собственно LibreOffice и средство для просмотра документов, чтобы читать справочные материалы. Однако в следующий раз, когда мы будем изучать компьютерные сети, нам понадобится уже другое программное обеспечение — и другая виртуальная машина.

2. Одна виртуальная машина — один студент или группа (в случае групповой работы). Не следует допускать совместного использования одной виртуальной машины двумя различными пользователями (по очереди или одновременно), если это не предусмотрено задачей, для которой она создана.

3. Данные, доступные виртуальным машинам, должны быть максимально сегрегированы [10]. Не следует создавать несколько разделов на одном виртуальном носителе, за исключением случаев, когда дополнительные разделы создаются автоматически программой-установщиком операционной системы, или когда студенты изучают инструменты для управления дисковыми разделами и файловыми системами.

Виртуальные носители не настолько жестко привязаны к гостевой системе, как жесткие диски к компьютеру. В процессе управления виртуальной инфраструктурой носители можно переназначать различным гостевым системам. По возможности следует хранить данные различных типов на различных виртуальных носителях, а для самих виртуальных носителей определять режим ввода/вывода и размещение на физических хранилищах в соответствии с частотой обращения, требуемой скоростью выполнения операций ввода/вывода, объемом и т. д.

4. Виртуальной машине должен быть разрешен только необходимый сетевой трафик. В зависимости от ситуации это может быть обеспечено на канальном уровне (VLANы) и на более высоких, когда задействуется МСЭ гипервизора.

Свободное программное обеспечение

Хотя обсуждение достоинств и недостатков свободного и проприетарного программного обеспечения выходит за рамки настоящей работы, следует отметить

следующее.

Как утверждается в [2], переход к виртуальной инфраструктуре позволяет эффективнее управлять лицензиями, поскольку одна и та же лицензия может использоваться различными ВМ при условии, что в каждый момент времени работает только одна из них. На практике, однако, так происходит далеко не всегда.

С началом массового использования виртуальных систем лицензирование коммерческого программного обеспечения усложнилось. Лицензионная политика вендоров в отношении виртуальных систем весьма разнообразна. Зачастую возможности использовать продукт в виртуальной среде намеренно ограничивается технически или юридически [11]. В ряде случаев условия лицензирования продукта заставляют заказчиков отказаться от некоторых продвинутых технологий, таких, как живая миграция, или от виртуализации вообще.

Многие крупные вендоры предоставляют образовательным учреждениям не только бесплатные и полнофункциональные версии своих программных продуктов, но и техническую поддержку. Однако условия предоставления такого ПО могут запрещать его использование в коммерческом обучении.

Следует также отметить, что некоторые вендоры начинают предоставлять свое ПО в собственной облачной инфраструктуре, ограничивая другие способы его получения.

В отдельных случаях существует риск потери доступа к коммерческому ПО из-за эмбарго, налагаемого в результате политических решений.

Всё это приводит к выводу, что использование свободного ПО в виртуальной образовательной среде обходится дешевле и несет меньше рисков. Оно может быть вполне оправданным, когда необходимо изучать не конкретный программный продукт, а технологию. Например, свободные программы Apache HTTP Server, Squid Proxy Server и Snort IDS/IPS можно изучать уже потому, что каждая из них определяет стандарт де-факто в своем классе.

Наконец, знакомство студентов с возможностями свободного ПО представляется авторам стратегически оправданным.

Резервное копирование

Проведя классификацию виртуальных машин по длительности жизненного цикла и типу доступа, мы можем переходить к решению двух важных задач:

- составление графика резервного копирования,
- планирование антивирусной защиты виртуальных машин.

Практика показывает, что резервное копирование в традиционной ИТ-инфраструктуре вызывает трудности. Это связано с тем, что полное резервное копирование полностью укомплектованной системы происходит долго, а для выборочного необходимо не только найти все источники копирования, но и периодически их уточнять. Если же виртуальная машина создана с учетом правил минимальности и сегрегации, то осуществить ее резервное копирование гораздо проще.

Создание резервных копий может отвечать двум целям: восстановление после сбоя и возврат системы к одному из предыдущих состояний. Некоторые важные вопросы организации резервного копирования обсуждаются в статье [12].

Резервное копирование средствами гостевой операционной системы считается малоэффективным, и его можно рассматривать лишь в отдельных случаях как учебную задачу.

Резервное копирование/восстановление виртуальной машины сводится к копированию/восстановлению ее виртуальных носителей, файла конфигурации и дампа оперативной памяти. Отметим также возможность создания мгновенных снимков (snapshots) системы.

Ниже приводится пример плана резервного копирования для различных типов ВМ (табл. 3).

Таблица 3. План резервного копирования для различных типов ВМ

Тип доступа \ Длительность	Индивидуальная (И)	Для рабочих групп (Г)	Общего доступа (О)
Краткосрочного использования (К)	Не выполняется	Не выполняется	Ежедневное, глубиной в 1 день

Тип доступа \ Длительность	Индивидуальная (И)	Для рабочих групп (Г)	Общего доступа (О)
Среднесрочного использования (С)	Ежедневное, глубиной в 1 неделю	Ежедневное, глубиной в 1 неделю	Ежедневное, глубиной в 1 неделю + в зависимости от частоты изменений
Долгосрочного использования (Д)	Еженедельное, глубиной в 1 мес.	Еженедельное, глубиной в 1 мес., + ежедневное, глубиной в 1 нед.	Ежедневное, глубиной в 1 нед. + в зависимости от частоты изменений

Подобный план позволяет уже на этапе проектирования рассчитать минимальный объем пространства хранения и более равномерно распределить по времени нагрузку, возникающую при выполнении операции.

Антивирусная защита

В традиционной, не облачной ИТ-инфраструктуре антивирусное программное обеспечение обычно устанавливается на все рабочие станции и серверы. Это стало настолько обычным, что антивирусное ПО считается неотъемлемой частью любого компьютера. С переходом к виртуальной инфраструктуре и IaaS подход к антивирусной защите нуждается в пересмотре. Как и любое средство защиты, антивирус предназначен для нейтрализации определенных угроз и должен применяться там, где вероятность реализации угрозы достаточно высока.

Следующие факторы повышают риск

заражения вирусом:

1. компьютер (рабочая станция или сервер) является длительно используемым объектом инфраструктуры,
2. на компьютере установлено всё программное обеспечение, которое когда-либо и кому-либо может понадобиться, в том числе и то, в котором часто обнаруживают уязвимости,
3. компьютер используется несколькими пользователями,
4. компьютер предоставляет общие ресурсы в локальной сети с большим количеством узлов,
5. пользователи осуществляют доступ к интернет-ресурсам,
6. на компьютере используются съемные носители.

При реализации вышеизложенных правил проектирования останутся актуальными лишь следующие факторы (табл. 4).

Таблица 4. Факторы, повышают риск заражения вирусом для различных типов ВМ

Тип доступа \ Длительность	Индивидуальная (И)	Для рабочих групп (Г)	Общего доступа (О)
Краткосрочного использования (К)	-	3	3
Среднесрочного использования (С)	1	1,3	1,3
Долгосрочного использования (Д)	1,5,6	1,3,5,6	1,3

Как видно из таблицы, антивирусная защита может с высокой вероятностью потребоваться для виртуальных машин типа (ДИ) и (ДГ) и, скорее всего, не требуется для (К*), (СИ). В остальных случаях в качестве компенсирующей меры может выступать резервное копирование.

Выводы

Нами предложены правила проектирования виртуальных машин, которые соответствуют особенностям виртуальных образовательных ИТ-инфраструктур и не зависят ни от конкретной программно-аппаратной реализации элементов физической инфраструктуры, ни от программной

конфигурации самих виртуальных машин, но в значительной степени определяют и то, и другое, позволяя минимизировать физические ресурсы, потребляемые виртуальной инфраструктурой, и избежать проблем, свойственных классической образовательной ИТ-инфраструктуре. Построенный пример классификации виртуальных машин, используемых в обучении, показывает, что уже на этапе проектирования виртуальной инфраструктуры можно в значительной мере ответить на некоторые вопросы, которые обычно возникают только на этапе эксплуатации.

Литература

1. Gartner, Inc. Gartner Top 10 Strategic Technology Trends for 2018. Contributor: Kasey Panetta. 03.10.2017. URL: <https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2018/> (дата обращения: 01.11.2017)
2. IBM Global Education. Virtualization in Education. White Paper. URL: <http://www-07.ibm.com/solutions/in/education/download/Virtualization%20in%20Education.pdf> (дата обращения: 01.11.2017)
3. North Carolina State University. VCL. URL: <https://vcl.ncsu.edu/> (дата обращения: 01.11.2017)
4. Жданович Д.П., Жданович П.Б. Построение прототипа IaaS-облака для дистанционного обучения // Ученые записки Института социальных и гуманитарных знаний, 2016 No2(14), ч. I. С.23-29
5. Ледовский М.В. Исследование различных способов построения алгоритмов мета-планирования ресурсов центра обработки данных // International Journal of Open Information Technologies, 2014. T.2. N 8. С. 6-10
6. Matt Murphy. 10 Trends Transforming Enterprise IT. URL: <https://techcrunch.com/2014/10/18/big-changes-big-money-10-trends-transforming-enterprise-it/> (дата обращения: 01.11.2017)
7. Major Hayden. Performance benchmarks: KVM vs. Xen. URL: <https://major.io/2014/06/22/performance-benchmarks-kvm-vs-xen/> (дата обращения: 01.11.2017)
8. How fast is KVM? Host vs virtual machine performance! URL: <https://forum.level1techs.com/t/how-fast-is-kvm-host-vs-virtual-machine-performance/110192> (дата обращения: 01.11.2017)
9. Rick Vanover. Virtualize everything (except in these four scenarios) URL: <https://www.techrepublic.com/blog/data-center/virtualize-everything-except-in-these-four-scenarios/> (дата обращения: 01.11.2017)
10. Younis A. Younis, Madjid Merabti, Kashif Kifayat. Secure Cloud Computing for Critical Infrastructure: A Survey. URL: https://www.researchgate.net/publication/262817790_Secure_Cloud_Computing_for_Critical_Infrastructure_A_Survey (дата обращения: 01.11.2017)
11. Microsoft Corp. Licensing Windows desktop operating system for use with virtual machines. URL: download.microsoft.com/download/9/8/D/98D6A56C-4D79-40F4-8462-DA3ECBA2DC2C/Licensing_Windows_Desktop_OS_for_Virtual_Machines.pdf (дата обращения: 01.11.2017)
12. Eric Siebert. Top five mistakes made when backing up VMs and how to prevent them. URL: <http://searchdatabackup.techtarget.com/tip/Top-five-mistakes-made-when-backing-up-VMs-and-how-to-prevent-them> (дата обращения: 01.11.2017)

Жданович Д. П., Жданович П. Б. О проектировании виртуальных ресурсов облачной образовательной инфраструктуры. В настоящей статье мы предлагаем набор правил для проектировании виртуальных машин в образовательной облачной инфраструктуре. Мы также приводим пример классификации виртуальных машин, не зависящей от конфигурации, которую полезно использовать при планировании виртуальной инфраструктуры.

Ключевые слова: дистанционное обучение, IaaS, облачные технологии, лицензирование, свободное программное обеспечение, виртуальная машина.

Zhdanovich D., Zhdanovich P. On Design of Virtual Resources for Education Cloud Infrastructure. In the present paper we suggest the set of rules that should be observed while designing virtual machines in education cloud infrastructure. We also give an example of a configuration-independent classification of VMs that can be useful while designing of virtual infrastructure.

Keywords: e-learning, IaaS, cloud, licensing, free software, virtual machine.

Статья поступила в редакцию 20.06.2018 г.
Рекомендована к публикации доктором технических наук Павлышом В.Н.

Обучающая система построения дерева Штейнера в ортогональной метрике

А. И. Ольшевский

Донецкий национальный технический университет

a_olshevskiy@mail.ru

Ольшевский А. И. Обучающая система построения дерева Штейнера в ортогональной метрике. Исследованы основные алгоритмы построения дерева Штейнера. Обоснован выбор языка программирования и использование OpenGL. Определен подход формирования объектов визуализации. Описаны основные структуры данных, схема классов, диаграммы взаимодействия и пакетов. Спроектирован интерфейс и описаны возможности программной реализации. Проанализирована эффективность разных методов.

Ключевые слова: структуры данных, дерево Штейнера, наборы операций, эффективность разных методов.

Введение

В настоящее время различные постановки задачи Штейнера (ЗШ) изучаются в таких областях как дискретная оптимизация, вычислительная геометрия, проблемы трассировки при проектировании СБИС, коммуникационных сетей, механических и электрических систем и т.д. При этом публикации в этой области можно разделить на две группы. В первом случае речь идет о классической ЗШ, ее частных случаях или обобщениях, и основное внимание уделяется изучению свойств этой задачи и построению алгоритмов ее решения. Вторая группа включает в себя публикации, посвященные либо задачам, постановка которых лишь «отдаленно» связана с классической и, по сути дела, отдается дань лишь термину «задача Штейнера», либо сугубо прикладным задачам, возникшим в связи с попыткой использования алгоритмов решения тех или иных постановок задачи Штейнера, в которых основное внимание уделяется не самой задаче, а особенностям реализации этих алгоритмов в конкретных прикладных аспектах [1].

Постановка задачи

Задача прямоугольного дерева Штейнера (ПДШ) в трёхмерном пространстве можно свести к задаче в двумерном пространстве, которая формулируется следующим образом: пусть дан граф $G = (V, E)$, где множество вершин V состоит из двух непересекающихся множеств, множества терминальных вершин A и множества точек Штейнера S .

Всюду в дальнейшем будем считать, что $|V| = p = n + s$, $|A| = n$, $|S| = s$, $|E| = m$. Ребрам графа приписаны неотрицательные веса, а под длиной дерева понимается сумма весов

входящих в это дерево ребер.

Задача состоит в нахождении такого подграфа G , который является деревом, покрывающим все терминальные вершины, и имеет минимальную длину среди всех подобных подграфов. Этот подграф называется минимальное дерево Штейнера (МДШ).

Исследования

Для решения задачи прямоугольного дерева Штейнера в трёхмерном пространстве рассмотрим, как эта задача решается в двумерном пространстве.

Существует множество алгоритмов построения минимальных связующих деревьев с дополнительными вершинами в ортогональной метрике [2-3]. Рассмотрим алгоритмы и выясним какой более применим к нашей задаче.

Известно, что для простого эвристического алгоритма построения ДШ в ортогональной метрике при соединении трех вершин единственная точка Штейнера g находится внутри прямоугольника, связывающего эти точки, причем координаты этой точки определяются как среднее арифметическое координат трех точек. Длина связывающих ребер определяется по формуле:

$$\sum_{i=1}^3 d(q_i, n_i) = \frac{1}{2} \prod (R(n_1, n_2, n_3)),$$

где $\prod(R)$ – длина периметра прямоугольника, построенного на трех вершинах.

Простой эвристический алгоритм построения ДШ, обладающий линейной сложностью $O(\alpha n)$, где n – число вершин [4].

Ранним решением проблемы ПДШ является теорема Ханана, которая обеспечивает сокращение от задачи ПДШ к задаче дерева Штейнера (МДШ). Ханан доказал, что для любого экземпляра существует оптимальное

ПДШ, в котором каждая точка Штейнера лежит на пересечении двух ортогональных линий, содержащих терминалы. Из теоремы Ханана следует, что граф G , называемый сетчатым графом Ханана, должен содержать оптимальное ПДШ [5], [6]. Граф G строится следующим образом: протягивайте горизонтальную и вертикальную линии через каждый терминал. Вершинам G соответствуют пересечения прямых. Между двумя вершинами существует грань, если они смежны вдоль линии. Вес ребра – это прямоугольное расстояние между конечными точками.

В дальнейшем пусть n обозначает число терминалов, а s обозначает число нетерминальных вершин в графе сетки Ханана. Заметим, что $s = O(n^2)$ в худшем случае.

Сокращение графа. Часто многие вершины в G могут быть удалены вместе с соседними ребрами, сохраняя при этом гарантию того, что G содержит оптимальное ПДШ. Было разработано множество методов сокращения для общих графиков и показано, что они достаточно эффективны для некоторых типов графиков. Однако, к сожалению, большинство этих сокращений неэффективны при применении к диаграмме решетки Ханана. Были разработаны другие сокращения, характерные для сетчатого графика Ханана. Одним из таких редукций является выпуклый корпус Янга и Винга [7]. Любая нетерминальная вершина, которая смежна с ровно двумя ортогональными краями e_1 и e_2 , может быть удалена, если присутствуют два других ребра, образующие прямоугольник с e_1 и e_2 . Вершины, оставшиеся после этого сокращения, были выполнены именно такими, которые лежат в прямолинейной выпуклой оболочке терминалов. В патологических случаях уменьшение выпуклой оболочки может не иметь никакого эффекта, но для небольших, случайно генерируемых множеств терминалов обычно достаточно эффективно. Сокращение выпуклого корпуса часто оставляет много терминалов степени 1. Такие терминалы могут быть удалены (вместе с их смежным краем), а их сосед сделал терминал, а соответствующий край добавлен обратно в окончательное решение. Самым ярким эффектом такого сокращения являются не терминалы, которые он удаляет, а скорее тот факт, что часто два или более терминалов сворачиваются в один новый терминал.

Решётка Ханана в трёхмерном пространстве строится по тому же принципу, что и в двумерном с учётом изменений мерности пространства. В трёхмерном пространстве также берётся пара точек, но добавляется 6 точек таким образом, чтобы они были углами параллелепипеда.

Проектирование программного обеспечения обучающей системы

Для реализации системы был выбран язык C++ с использованием технологии MFC. MFC – библиотека на языке C++, разработанная Microsoft и призванная облегчить разработку GUI-приложений для Microsoft Windows путем использования богатого набора библиотечных классов. Все функции реализованы в виде методов класса текущего окна.

Графическое отображение дерева Штейнера будет использовать графическую библиотеку OpenGL. Данная библиотека позволяет быстро и просто реализовать вывод точек в виде сфер в пространстве, а также связи между ними. Для разработки интерфейса будет использоваться библиотека MFC. При этом сцена OpenGL будет интегрироваться в рабочую область программы, сохраняя при этом привычный для пользователя вид окна.

Библиотека OpenGL представляет собой программный интерфейс для аппаратного обеспечения машинной графики. Этот интерфейс состоит приблизительно из 250 отдельных команд (почти 200 команд в ядре OpenGL и еще 50 команд в библиотеке утилит OpenGL), которые используются для того, чтобы определить объекты и операции, необходимые для создания интерактивных трехмерных прикладных программ.

Библиотека OpenGL разработана в качестве низкоуровневого, аппаратно-независимого интерфейса, допускающего реализацию на множестве различных аппаратных платформ. Для того чтобы достичь этих качеств, в состав библиотеки OpenGL не включены никакие команды для выполнения задач работы с окнами или для получения пользовательского ввода; вместо этого вы должны работать через любую систему управления окнами, которая работает с конкретными аппаратными средствами. Точно так же библиотека OpenGL не предоставляет команды высокого уровня для описания моделей трехмерных объектов. Такие команды могли бы позволить определять относительно сложные формы. При использовании библиотеки OpenGL вы должны создавать нужную модель из ограниченного набора геометрических примитивов – точек, линий и многоугольников.

Для хранения объектов сцены будет использоваться такая структура как стек. Параметрами объекта будут являться: тип объекта (идентификатор), его координаты или пара координат в случае прямой, соединяющей узлы графа, указатель на следующий элемент стека. Координаты объекта будут храниться в двумерном массиве с тремя столбцами, где каждый из столбцов хранит соответствующую

координату в пространстве. Количество строк в этом массиве будет равно либо одной либо двум в зависимости от идентификатора. Каждый объект стека обладает следующими параметрами:

- UID – уникальный идентификатор объек-та;
- Type – ключ, указывающий на тип объекта;
- Coord – двумерный массив, хранящий в себя координаты местоположения объекта;
- next – указатель на следующий элемент стека.

Для построения дерева Штейнера по алгоритму Крускала используется вектор объектов edges, который состоит из следующих элементов:

- x – начальная вершина;
- y – конечная вершина;
- w – вес ребра;
- edges {} – рёбра, вошедшие в дерево;
- edges(x, y, w): x(x), y(y), w(w) {}.

Рассмотрим функцию OnCreateStein, которая реализует основную часть алгоритма. В начале своей работы функция преобразует исходные данные из стека объектов obj в вектор объектов edges. После этого, согласно алгоритму, вектор сортируется с помощью встроенной функции sort, используя в качестве сравнения функцию cmp. Эта функция возвращает результат сравнения веса ребра w объекта edges. Далее мы проходимся по всем рёбрам edges и определяем, какому из подмножеств принадлежат вершины текущего ребра и вызывается функция unite. Если ребро добавить можно, то оно добавляется к списку рёбер дерева Штейнера. После этого мы проходим по всем рёбрам, содержащихся в списке рёбер дерева Штейнера и определяем их суммарный вес, а также выводим их на экран. Алгоритмы функций unite и OnCreateStein в виде блок-схем изображены на рисунке 1.

Диаграммы классов используются при моделировании программных систем наиболее часто. Они являются одной из форм статического описания системы с точки зрения ее проектирования, показывая ее структуру. Диаграмма классов не отображает динамическое поведение объектов изображенных на ней классов. На диаграммах классов показываются классы, интерфейсы и отношения между ними.

Обобщение на диаграммах классов используется, чтобы показать связь между классом-родителем и классом-потомком. Оно вводится на диаграмму, когда возникает разновидность какого-либо класса, а также в тех случаях, когда в системе обнаруживаются несколько классов, обладающих сходным поведением.

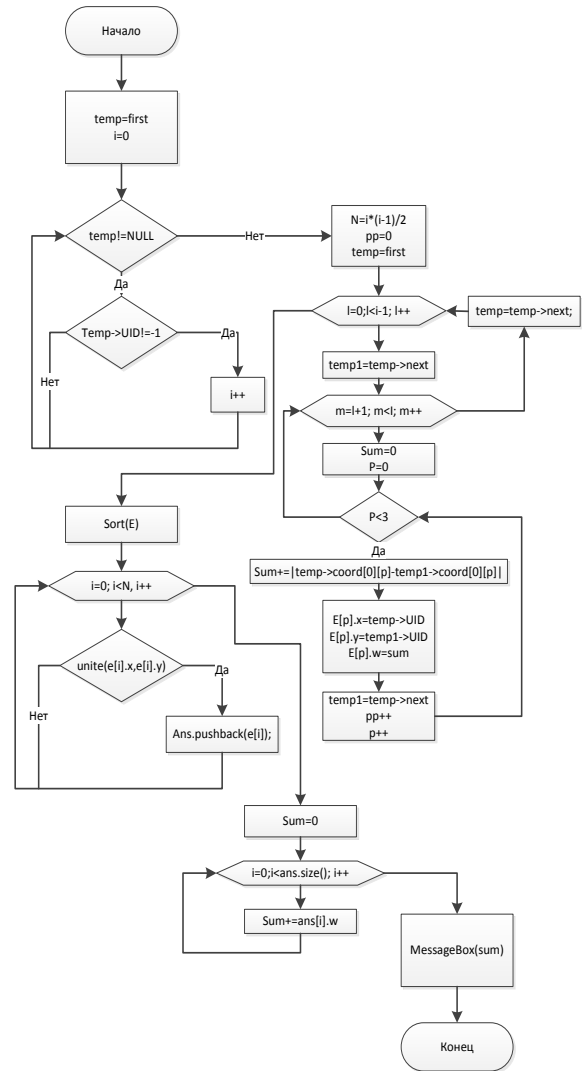


Рисунок 1 – Алгоритм функции OnCreateStein

Диаграмма классов для данного программного продукта изображена на рисунке 2.

Диаграммы последовательности предназначены для моделирования взаимодействия между несколькими объектами. Зачастую диаграммы последовательности создаются для моделирования взаимодействия в рамках одного прецедента.

На концептуальном уровне можно использовать диаграммы последовательности для моделирования взаимодействия между Бизнес-актерами, но зачастую подобные диаграммы обрастают лишними подробностями и плохо читаются. На данном уровне лучше подойдут диаграммы деятельности, исключение составляют случаи, когда необходимо смоделировать обмен сообщениями между двумя независимыми системами.

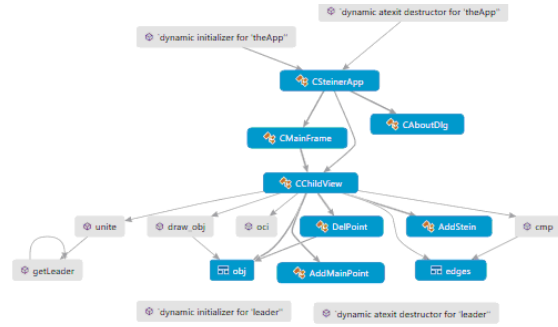


Рисунок 4 – Диаграмма пакетов

Рисунок 2 – Схема классов

Также диаграммы последовательности подойдут для моделирования взаимодействия пользователя и Системы в целом.

На уровне детальной спецификации требований диаграммы последовательности используются для моделирования взаимодействия компонентов Системы и пользовательских классов в рамках выбранного прецедента.

Диаграмма взаимодействий изображена на рисунке 3.

Реализация обучающей системы

В качестве рабочих алгоритмов для программного продукта были выбраны алгоритмы: на основе трех точек; Крускала; Прима, генетический и итерационный методы. Для них была разработана теоритическая часть с формальном описанием алгоритмов и определением сложности вычислений.

Система позволяет добавлять точки Штейнера или определяет их в процессе вычислений.

Программная реализация создана с учетом выбора алгоритма для построения минимального дерева Штейнера в трёхмерном пространстве (X,Y,Z) или двухмерном, если одна из плоскостей нулевая. Например: $(X,Y,Z=0)$.

Для визуализации решения данной задачи был выбран алгоритм Крускала, так как он не привязан к выбору первоначального остова, в отличие от алгоритма Прима.

На рисунке 5 пример формирования топологии в двухмерном пространстве $(X,Y,Z=0)$ для 5 терминальных вершин $T1(0,30)$, $T2(20,15)$, $T3(10,0)$, $T4(35,25)$, $T5(40,10)$ и двух точек Штейнера $Ш1(10,0)$, $Ш2(35,15)$. Суммарная длина ребер 85.

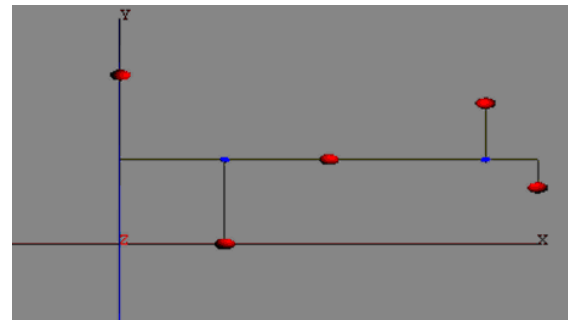


Рисунок 5 – Дерево Штейнера в двухмерном пространстве

Рисунок 3 – Диаграмма взаимодействий

Компоненты на диаграмме компонентов представляют собой физические модули программного кода. Зависимости между компонентами должны совпадать с зависимостями между пакетами. Эти зависимости показывают, каким образом одни компоненты взаимодействуют с другими. Направление данной зависимости показывает уровень осведомленности о коммуникации.

Диаграмма пакетов для данной системы изображена на рисунке 4

На рисунках 6-7 приведена исходная сцена программы и пример формирования топологии в трёхмерном пространстве для 5 терминальных вершин $T1(0,30,0)$, $T2(20,15,-20)$, $T3(10,0,-10)$, $T4(35,25,10)$, $T5(40,10,20)$. Суммарная длина ребер 160.

Система позволяет просматривать изображение графа с разных точек пространства, путем осевых вращений.

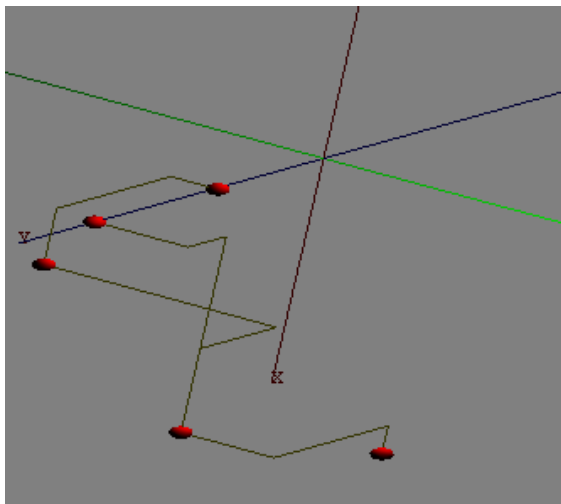


Рисунок 6 – Пример графа по оси Y

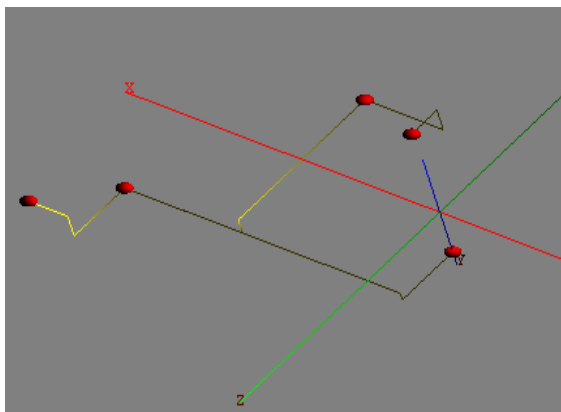


Рисунок 7 – Пример графа по оси Z

В качестве оценочного параметра используется соотношение времени построения дерева Штейнера к количеству заданных вершин графа.

Проанализировав полученные результаты можно сделать вывод, что с увеличением количества точек, время выполнения будет увеличиваться непропорционально, что является естественным для NP-полных задач. Так как программный продукт строит для каждой пары точек в трехмерном пространстве создаёт 6 потенциальных точек Штейнера время выполнения значительно увеличивается с ростом количества точек.

Выводы

Были исследованы основные алгоритмы построения дерева Штейнера. Для каждого из них было составлено формальное описание алгоритма выполнения и определена сложность.

При разработке программного продукта был спроектирован интерфейс, построены схема классов и диаграмма взаимодействия.

Был разработан модуль визуализации дерева Штейнера в ортогональной метрике и оптимизирована структура хранения данных. В качестве структуры для хранения объектов дерева Штейнера использовался один стек. В нём хранятся все объекты. Для их различия при чтении используются уникальные идентификаторы и идентификаторы типа. За счёт этого был уменьшен объём памяти и время её выполнения при построении дерева.

Достоинство данной разработки является её открытость, с возможностью добавления новых алгоритмов.

Визуализация графических объектов позволяет провести пошаговый анализ результатов работы системы для выбранных алгоритмов построения ПДШ, а также в интерактивном режиме оценить и выбрать лучший из них.

Разработанную систему можно использовать в ВУЗах для обучения студентов изучению различных алгоритмов построения прямоугольного дерева Штейнера, а так же при решении практических задач.

Литература

1. Agarwal P. K. / M. T. Shing // Algorithms for special cases of rectilinear steiner trees: Points on the boundary of a rectilinear rectangle. *Networks*, 20(4) – 1990 – p.85–453.
2. Alexander M. J. / G. Robins. // New performance-driven fpga routing algorithms. *IEEE Transactions Computer-Aided Design*, 15(12) – December 1996 – p.1505–1517.
3. Alpert C. J. / G. Gandham, M. Hrkic, J. Hu, A. B. Kahng, J. Lillis, B. Liu, S. T. Quay, S. S. Sapatnekar, and A. J. Sullivan // Buffered steiner trees for difficult instances. *IEEE Transactions Computer-Aided Design* 21(1) – January 2002 – p.3–14.
4. Ольшевский, А. И. Разработка программного модуля построения дерева Штейнера в ортогональной метрике / А. И. Ольшевский // *Инновационные перспективы Донбасса [Электронный ресурс] : материалы международной научно-практической конференции, г. Донецк, 20-22 мая 2015 г. Т. 5 : Компьютерные науки и технологии / М-во образования и науки ДНР и др. ; редкол.: Л.П. Полякова и др. - Донецк : ГВУЗ "ДонНТУ", 2015. - С. 110–115. - Библиогр.: с. 115.*
5. Нечепуренко М.С. Разработка программного модуля построения дерева Штейнера для случая ортогональной метрике в трехмерном пространстве / М.С. Нечепуренко, А. И. Ольшевский // *Сборник материалов международной научно-практической конференции студентов, аспирантов и молодых ученых «Компьютерная и программная*

инженерия – 2015». – Донецк, С 138

6. Нечепуренко М.С. Применение различных алгоритмов в решении задачи Штейнера / М.С. Нечепуренко, А. И. Ольшевский // Сборник материалов VII Международной научно-технической конференции в рамках II Международного Научного форума Донецкой Народной Республики 26 мая 2016г. «Информатика, управляющие системы, математическое и компьютерное моделирование» (ИУСМКМ -

2016). – Донецк, С 372-377

7. Нечепуренко М.С. Исследование и разработка алгоритма построения деревьев Штейнера для случая ортогональной метрики в трёхмерном пространстве / М.С. Нечепуренко, А.И. Ольшевский // «Информатика, управляющие системы, математическое и компьютерное моделирование» в рамках III форума «Инновационные перспективы Донбасса» (ИУСМКМ – 2017). – Донецк : ДонНТУ, 2017. – С. 464-469

Ольшевский А. И. Обучающая система построения дерева Штейнера в ортогональной метрике. Исследованы основные алгоритмы построения дерева Штейнера. Обоснован выбор языка программирования и использование OpenGL. Определен подход формирования объектов визуализации. Описаны основные структуры данных, схема классов, диаграммы взаимодействия и пакетов. Спроектирован интерфейс и описаны возможности программной реализации. Проанализирована эффективность разных методов.

Ключевые слова: структуры данных, дерево Штейнера, наборы операций, эффективность разных методов.

Olshevsky A. I. Teaching system for constructing a Steiner tree in an orthogonal metric. The basic algorithms for constructing a Steiner tree are investigated. The selection of the programming language and the use of OpenGL are justified. The approach of forming objects of visualization is determined. The main data structures, class diagram, interaction and packet diagrams are described. The interface is designed and the possibilities of software implementation are described. The effectiveness of different methods has been analyzed.

Keywords: data structures, Steiner tree, operation sets, efficiency of different methods.

Статья поступила в редакцию 22.06.2018 г
Рекомендована к публикации профессором Миненко А. С.

Применение теории игр для оптимальной настройки ПИД-регулятора по выбранным критериям

В. А. Гуляев, Ю. К. Орлов
Донецкий национальный технический университет
gvard12@gmail.com

Гуляев В. А., Орлов Ю. К. Применение теории игр для оптимальной настройки ПИД-регулятора по выбранному критерию. В данной работе предлагается алгоритм для точной настройки коэффициентов ПИД-регулятора на базе концепции теории игр. Приведены базовые теоретические сведения, необходимые для понимания работы алгоритма. Сформулирован алгоритм настройки регулятора, Построена модель в системе MATLAB, которая соответствует одному циклу алгоритма. Приведен ручной просчет одного шага алгоритма.

Ключевые слова: ПИД-регулятор, теория игр, цена игры, итерация, оптимальность настройки

Введение

Настройка ПИД-регуляторов является одной из наиболее распространенных операций при настройке автоматических систем управления в процессе наладки и эксплуатации [1]. Среди промышленных систем автоматического управления существуют такие, в которых по технологии функционирования необходимо обеспечивать аperiodический характер переходных процессов, например, регуляторы уровней, частотные преобразователи и т.д. В качестве регуляторов в указанных системах обычно используются промышленные ПИ- или ПИД-регуляторы. Главной их задачей является уменьшение колебательности и перерегулирования в системе, приводящее переходные процессы к аperiodическому виду. Такие регуляторы могут быть реализованы как непосредственно в регулируемых системах, так и отдельным регулятором на базе некоторого микроконтроллера или программно в системе управления технологическими процессами.

Существует комплекс методов для расчета параметров ПИД-регулятора, часть из которых проста и менее точна (например, метод Циглера-Никольса, метод СНР [2]), другая часть более точна, но более трудоемка (настройка с использованием генетических алгоритмов [3]). Также следует упомянуть настройки при помощи правил нечеткой логики и нейросетей как достаточно новые и развивающиеся направления подбора коэффициентов ПИД-регулятора.

Большинство достаточно простых методов не дают оптимальных результатов настройки ПИД-регулятора, поэтому встает вопрос тонкой настройки для достижения

наилучшего результата. Применение концепции теории игр в данном случае помогает и позволяет добиться наилучшего результата с заданной погрешностью.

В процессе создания статьи не было обнаружено исследований по данной тематике.

Теория ПИД-регуляторов

В общем виде система с ПИД-регулятором выглядит так, как показано на рисунке 1.

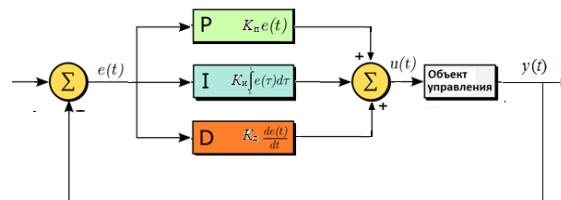


Рисунок 1 – Блок-схема модели управления с ПИД-регулятором

Основу ПИД-регулятора составляют три коэффициента:

K_P – пропорциональный коэффициент;

K_I – интегральный коэффициент;

K_D – дифференциальный коэффициент

[4].

При помощи варьирования этих параметров достигается оптимальное входное воздействие на объект управления, причем каждый из параметров по-своему влияет на то, какой будет выходное воздействие модели. Результатом ПИД-регулирования должна стать устойчивая система, т.е. с затухающими колебаниями, с установившемся значением, которое удовлетворяет условию задачи. Были выделены следующие критерии оптимальности настройки ПИД-регулятора, которые наиболее

часто встречаются при настройке, и значения которых должны стремиться к нулю:

- ошибка регулирования;
- перерегулирование;
- время регулирования [5].

В зависимости от условий задачи наладчик выбирает один из параметров и относительно его подбирает коэффициенты.

Собственно говоря, теория ПИД-регуляторов весьма широка, однако для данной работы достаточно использования вышеописанных терминов.

Теория игр

В [6] приведено исследование теории игр для принятия решений в условиях неопределенности и рисков. Согласно нему предполагается, что есть 2 стороны – игрок, заинтересованный в наиболее выгодном для него исходе игры, и выступающий против участника, совершенно безразличного к результату. Составляется матрица $N \times M$, где N – число вариантов доступных первому игроку, M – число вариантов доступных второму игроку, в каждую ячейку которой заносится значение, называемое ценой игры – некоторое выражение, необязательно численное, которое говорит о том к какому результату приведет выбор некоторых вариантов первым и вторым игроком. Применяв этот метод можно получить полный перебор возможных исходов при различных случаях выборов игроков, а используя такие критерии оптимальности можно получить численное представление эффективности выбора стратегии игры [7, 8]. К таким критериям относятся:

- критерий Байеса;
- критерий Лапласа;
- критерий Сэвиджа;
- критерий Гурвица;

Проведя расчёты по этим критериям лицо, принимающее решение, может получить полезную информацию для принятия решений.

В случае с простыми задачами теории игр достаточно ограничиться простым выбором наибольшей или наименьшей цены игры, т.е. воспользоваться принципом недостаточного обоснования Лапласа [9].

Концепция теории игр для настройки ПИД-регуляторов

Ключевая идея предлагаемого метода состоит в следующем: имея некоторые начальные значения регулятора, близкие к оптимальным, итеративно перебирать возможные варианты изменения параметров регулятора с заданным шагом до тех пор, пока не будет выполнено некоторое условие выхода.

Поскольку для настройки ПИД-регулятора используются 3 коэффициента, то

классическое представление теории игр об игре между двумя сторонами необходимо преобразовать в игру между тремя сторонами – таким образом, матрица игры превратится в трехмерную. Соответственно стратегиями игры будут выступать различные комбинации изменений параметров. Предлагается три варианта выбора для каждого из коэффициентов: оставить без изменений, уменьшить или увеличить на некоторое задаваемое наладчиком значение. Данный параметр может варьироваться, тем самым при уменьшении будет увеличиваться точность поиска, но также и увеличиваться время расчета при достаточно отдаленных от оптимума начальных значений.

Поскольку цена игры для сравнения в ходе итераций должна иметь численное значение, необходимо задать алгоритм, который бы определял какой её принимать. Предлагается следующая интерпретация: цена игры равна сумме 3 значений критериев оптимальности решения, умноженных на некоторые весовые коэффициенты, задаваемые пользователем на старте алгоритма. С помощью такого подхода можно обеспечить варьирование результатов в соответствии с предпочитаемым ожидаемым результатом.

На каждой итерации алгоритма после заполнения матрицы игры из всех элементов выбирается та цена игры, которая является наименьшей, в соответствии со стратегией по которой она сформирована обновляются коэффициенты ПИД-регулятора и алгоритм запускается заново.

Предполагается, что условием окончания работы алгоритма может служить одно из следующих условий:

- достигнут предел итераций алгоритма;
- достигнут предел времени работы алгоритма;
- решение на текущем шаге подразумевает сохранение предыдущих параметров ПИД-регулятора.

Реализация модели управления с возможностью варьирования коэффициентов регулятора

Для реализации предложенной концепции был выбран пакет прикладных программ MATLAB версии 2018a, поскольку он обладает возможностями по простой настройке ПИД-регуляторов и позволяет графически удобно представить модель и операции над ней.

В ходе разработки модели были использованы следующие модули программы: Simulink; DSP System Toolbox; Control System Toolbox.

На рисунке 2 показана разработанная модель управления

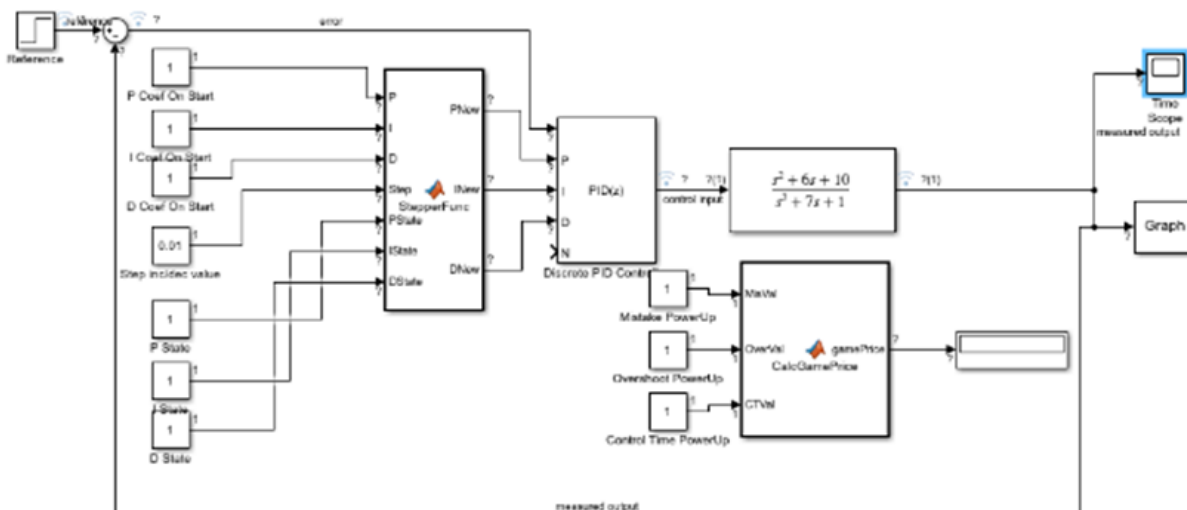


Рисунок 2 – Модель управления

Построенная модель является отображением одной итерации алгоритма. Пользователь задает следующие параметры:

- начальные коэффициенты ПИД-регулятора;
- шаг изменения коэффициентов;
- текущее состояние стратегии игры, выраженное в коэффициентах к каждому параметру (0 – уменьшение, 1 – без изменений, 2 – увеличение);
- весовые коэффициенты цены игры;
- собственно, передаточная функция, описывающая регулируемую систему.

На данной модели реализованы 2 функции – StepperFunc и CalcGamePrice. На вход функции StepperFunc приходят следующие параметры:

- начальные значения ПИД-регулятора;
- значение шага изменения параметров;
- коэффициенты состояния стратегии игры.

Сама функция представляет собой сравнение значений коэффициентов состояний и вывод скорректированных значений ПИД-регулятора непосредственно на блок регулятора. Далее реализуется базовая последовательность моделей регулирования, где после ПИД-контроллера идет непосредственно объект управления, описанный передаточной функцией. Полученные значения выводятся в блок TimeScope, который позволяет получить такие параметры системы, как перерегулирование и время регулирования, которые также записываются в переменную Graph.

Функция CalcGamePrice получает значения, записанные в переменную Graph, преобразует их, умножает на соответствующие весовые коэффициенты параметров и

складывает их. Полученное значение, являющееся ценой игры, выводится в блок, после чего наладчик может получить данные и использовать в дальнейшем.

Была разработана модель одной итерации предлагаемого алгоритма, поскольку это дает возможность достаточно просто модифицировать её и использовать, например, в связке с .NET приложением, что позволит использовать её при разработке промышленной системы типа SCADA.

Пример работы алгоритма

Для демонстрации эффективности работы алгоритма была взята модель системы управления как на рисунке 2. Путем подсистемы PID Tuner программы MATLAB были получены базовые коэффициенты регулятора (2,101 для пропорциональной, 1,274 для интегральной и 0,6729 для дифференциальной составляющих) [10]. На рисунке 3 показан график такой системы.

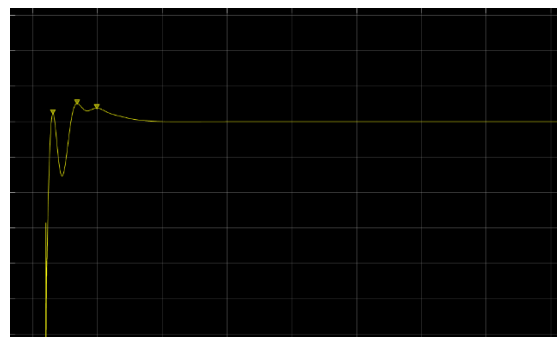


Рисунок 3 – График системы с изначальными параметрами регулятора

При этих значениях перерегулирование составляет 5,29%, время регулирования равно

0,293 с, ошибка регулирования равна 0 (система устойчива). Для данной задачи условлено, что необходимо добиться лучшего значения перерегулирования, поэтому весовой коэффициент этого параметра принято считать 3, в то время как остальных – 1.

Для оценки того насколько текущее регулирование эффективно в сравнении с предполагаемыми на шаге алгоритма необходимо рассчитать его цену игры, она же станет одним из значений в ячейках матрицы игры при условии не изменения параметров регулятора:

$$p_{base} = 3 \times 0,0529 + 1 \times 0,293 + 1 \times 0 = 0,446.$$

Далее для начала алгоритма необходимо задать последний параметр – шаг изменения коэффициентов. Пусть значение шага равняется 0,01. Теперь можно приступить к реализации алгоритма. В таблице 1 показана развернутая матрица игры 3x3.

Таблица 1 – Матрица игры

		Кд >	Кд =	Кд <
Кп >	Ки >	0,447	0,445	0,448
	Ки =	0,446	0,444	0,446
	Ки <	0,444	0,443	0,444
Кп =	Ки >	0,446	0,449	0,45
	Ки =	0,441	0,446	0,449
	Ки <	0,441	0,446	0,446
Кп <	Ки >	0,447	0,445	0,446
	Ки =	0,448	0,447	0,446
	Ки <	0,449	0,449	0,447

В соответствии с матрицей игры наилучшими стратегиями игры являются при неизменной пропорциональной, увеличенной дифференциальной, и или неизменной, или уменьшенной интегральной составляющих, но при детализации вычислений все же минимум достигается при уменьшении интегральной части. Таким образом, после этого шага основными параметрами ПИД-регулятора являются 2,101 для пропорциональной, 1,264 для интегральной и 0,6829 для дифференциальной составляющих. График модели системы будет едва заметно отличим от того, который изображен на рисунке 3, поэтому в работе не приведен. Затем алгоритм повторяется до тех пор, пока не выполнится некоторое выбранное условие окончания.

Выводы

Таким образом, в ходе исследования был получен алгоритм точной настройки ПИД-регулятора по выбранным критериям. Данный алгоритм позволяет путем прямого перебора с заданным шагом получить такие коэффициенты

регулятора, которые бы отвечали требованиям наладчика. В ходе исследования остались нерешенными следующие задачи:

- полноценная программная реализация алгоритма, которая бы давала готовый результат;
- оценка влияния на другие показатели регулирования и соответствие их требованиям наладчика.

На данном этапе результаты работы могут быть использованы на предприятиях промышленности, где повсеместно используются ПИД-регуляторы, специалистами АСУ для тонкой настройки параметров с целью повышения эффективности механизмов управления.

Литература

1. Тагиев, К.Ф. Технологические инновации на машиностроительном предприятии: риски и пути их снижения / К.Ф. Тагиев // Вестник экономики, права и социологии, 2010. – №4. – С.64–69.
2. Лазарева, Т. Я. Основы теории автоматического управления. Учебное пособие. 2-е изд., перераб. и доп. / Т. Я. Лазарева, Мартемьянов Ю. Ф. – Тамбов : Изд-во Тамб. гос. техн. ун-та, 2004. – 352 с.
3. Рубанов, В.Г. Теория автоматического управления (математические модели, анализ и синтез линейных систем): учебное пособие: часть 1. – Белгород: Изд-во БГТУ, 2005. – 199 с.
4. Батицкий, В.А. Автоматизация производственных процессов и АСУ ТП в горной промышленности / В.А. Батицкий, В.И. Куроедов, А.А. Рыжков. – М.: Недра, 1991. – 303 с.
5. Бесекерский, В.А. Теория систем автоматического управления / В.А. Бесекерский, Е.П. Попов.- 4-е изд., перераб. и доп.. – СПб.: Профессия, 2003. – 747 с.
6. Просветов, Г.И. Математические методы в экономике: учебно-методическое пособие / Г.И. Просветов. – М.: Изд-во РДЛ, 2004. – 364 с.
7. Саймон, Г.А. Теория принятия решений в экономической теории и в науке о поведении. Вехи экономической мысли: в 2 т. / Г.А. Саймон; под ред. В.М. Гальперина, С.М. Игнатьева, В.И. Моргунова. – СПб.: Экономическая школа, 2000. – 2 т.
8. Акимов, В.А. Основы анализа и управления риском в природной и техногенной сферах / В.А. Акимов, В.В. Лесных, Н.Н. Радаев. – М.: Деловой экспресс, 2004. – 352 с.
9. Шиянов, Б.А. Методы анализа рисков в системе регулирования неравновесными состояниями экономических систем, основанные на теории игр и нечеткой логики / Б.А. Шиянов, О.В. Силютин, В.С. Неженец /

Вестник Воронежского государственного технического университета. – Воронеж: ВГТУ, 2010. – 126 с.
Чертков, А.А. Параметрическая настройка ПИД-регуляторов динамических систем средствами

MATLAB / А.А. Чертков, Д.С. Тормашев, С.В. Сабуров // Вестник государственного университета морского и речного флота им. адмирала С.О. Макарова, 2014. – №5. – С.164-171.

Гуляев В. А., Орлов Ю. К. Применение теории игр для оптимальной настройки ПИД-регулятора по выбранному критерию. В данной работе предлагается алгоритм для точной настройки коэффициентов ПИД-регулятора на базе концепции теории игр. Приведены базовые теоретические сведения, необходимые для понимания работы алгоритма. Сформулирован алгоритм настройки регулятора, Построена модель в системе MATLAB, которая соответствует одному циклу алгоритма. Приведен ручной просчет одного шага алгоритма.

Ключевые слова: ПИД-регулятор, теория игр, цена игры, итерация, оптимальность настройки

Gulyaev V. A., Orlov Y. K. Application of game theory for optimum setup of the PID-regulator by the chosen criterion. In this work the algorithm for precise control of coefficients of the PID-regulator on the basis of the concept of game theory is offered. The basic theoretical data necessary for understanding of work of an algorithm is provided. The algorithm of the regulator setup is formulated. The model in the MATLAB system which corresponds to one cycle of an algorithm is constructed. The manual calculation of one step of an algorithm is given.

Keywords: PID-regulator, game theory, game price, iteration, control optimality

Статья поступила в редакцию 18.06.2018 г
Рекомендована к публикации профессором Миненко А. С.

УДК 004.522

Разработка системы автоматического распознавания слитной русскоязычной речи на основе дискриминативного обучения

Пикалёв Я. С.

Государственное учреждение «Институт проблем искусственного интеллекта»

pikaliov@gmail.com

Пикалёв Я. С. Разработка системы автоматического распознавания слитной русскоязычной речи на основе дискриминативного обучения. В данной работе рассмотрены методики распознавания речи на основе дискриминативного обучения. Рассмотрен процесс обучения системы автоматического распознавания русскоязычной слитной речи, используя small и medium языковую модель, а также различные вариации акустических моделей. Проведено сравнение методов, применяемых для извлечения акустических признаков. По результатам тестирования сделаны выводы об актуальности применения дискриминативного обучения для автоматического распознавания русскоязычной слитной речи.

Ключевые слова: автоматическое распознавание слитной речи, GMM-HMM, SGMM, дискриминативное обучение.

Введение

Автоматическое распознавание речи (ASR) является динамично развивающимся направлением в области искусственного интеллекта. Данная задача получила широкое распространение вследствие высокой применимости на практике: устройства управления; помощь операторам с занятыми руками; поиск ключевых слов или фрагментов текста; системы автоматического доступа; системы диктовки; системы речевой аналитики; системы биометрии. Сложность разработки ASR для слитной речи состоит в том, что в ней фраза обладает высокой вариативностью на всех уровнях обработки речи: отсутствует заранее подготовленная форма и содержание устного сообщения; разнообразие вариантов произнесения одних и тех же слов; присутствуют заполненные паузы, оговорки, повторы, смех и кашель. Помимо этого, слитная речь часто произносится в условиях наличия фонового шума, что дополнительно усложняет задачу её распознавания. Отдельная проблема состоит в распознавание речи «low resource languages» [10] (в их число входит русский язык), которым учёные и исследователи уделяют гораздо меньше внимания из-за недостаточного количества аннотированных речевых баз, а также фонетических проблем и недостаточного количество нормализованных текстов, которую учёные называют «data sparsity problem» (проблема отсутствия достаточного количества текстовых данных для статистического моделирования языка). Для русского языка в силу его флективности и свободного порядка слов в предложении data sparsity problem имеет большее влияние, чем для

остальных low-resource languages. В связи с этим на сегодняшний день в сфере распознавания русскоязычной слитной речи успехи достигнуты только в пределах ограниченного словарного запаса, т.е. распознавание слитной речи для русского языка до сих пор не имеет четкого решения. Учитывая вышесказанное, можно сделать вывод об актуальности исследования и разработки методов, алгоритмов и программных средств, обеспечивающих повышение точности распознавания слитной русскоязычной речи.

Цель данной работы заключается в разработке и тестировании системы автоматического распознавания русскоязычной слитной речи на основе дискриминативного обучения.

Структура системы автоматического распознавания речи

Современные ASR состоят из следующих частей (рис. 1).

1) Модуль обработки сигнала и извлечения признаков осуществляет шумоочистку аудиосигнала, который подаётся на вход распознавателя; преобразует сигнал из временной области в частотную; отделяет активную речь от фонового шума или тишины; извлекает из аудиосигнала векторы акустических признаков; адаптирует полученные признаки к диктору или окружению; определяется уровень помех и происходит шумоочистка и отделение полезного сигнала. Затем запись разделяется на маленькие фрагменты (фреймы), например, длиной 25 миллисекунд с шагом 10 миллисекунд, то есть внахлест. Таким образом из одной секунды речи получается сто фреймов.



Рисунок 1 - Структура ASR.

2) Акустическая модель (АМ) описывает плотность распределения вероятностей акустических классов (например, фонем) на заданном участке аудиосигнала. В системах для слитной речи применяется пофонемное распознавание: компьютер «учится» слышать отдельные звуки (фонемы) и соотносить их с возможным буквенным обозначением. Грубо говоря, система делает транскрипцию звуков и выдвигает предположение о том, какой символ этим звукам соответствует. Сначала каждый фрейм пропускается через акустическую модель. Система с машинным обучением, определяет варианты произнесенных слов и контекст. Точность результатов напрямую зависит от полноты фонетического алфавита системы. Для каждого звука изначально строится сложная статистическая модель, которая описывает произнесение этого звука в речи. Система распознавания сопоставляет входящий речевой сигнал с фонемами, а уже из них собирает слова. Каждый участок сопоставляется не с одной фонемой, а с несколькими, подходящими с разной степенью вероятности. Кроме того, система учитывает вероятности переходов, то есть определяет, какие фреймы могут идти следом за конкретной фонемой.

3) Языковая модель (ЯМ) описывает вероятность появления слова в контексте других слов. Для этого составляются лингвистические модели, подкрепляющие распознавание фонем правилами образования слов (морфология), словосочетаний и предложений (синтаксис), смысловых связей (семантика). Так, если акустическая модель не распознала, какова первая буква в слове «крюк», «к» или «т», то задача передается на языковой уровень. Система ищет соседние слова, и если перед проблемным словом стоит прилагательное «железный», то выбор делается в пользу «крюка». при помощи которой система определяет вероятный порядок слов и при необходимости восстанавливает не распознанные слова по смыслу, исходя из контекста и имеющейся статистики.

4) Словарь транскрипций устанавливает связь между последовательностями акустических классов, описываемых АМ, и словами, описываемыми ЯМ.

5) Декодер анализирует вероятности, генерируемые акустической и языковой моделями, и преобразует их в последовательность слов – результат распознавания. В результате полученная информация поступает в основной блок системы распознавания – декодер. Этот программный компонент совмещает данные от акустических и языковых моделей и на основании их объединения выдает конечный результат в виде наиболее вероятной последовательности слов.

О дискриминативном обучении

После стандартного обучения GMM-HMM (Gaussian Mixture Model - Hidden Markov Model; скрытые марковские модели, состояния которых описываются посредством смесей гауссовых распределений) [1] система, как правило, дополнительно обучается, чтобы увеличить правдоподобие (Maximum Likelihood, ML) истинной гипотезы относительно альтернативных гипотез. Оценка критерия ML оптимизирует распределение для порождающей модели, но для ASR нужна оптимизация для классификации. Непосредственная минимизация WER (Word Error Rate, процент ошибочно распознанных слов) трудна, потому что целевая функция не дифференцируема, что не позволяет применить методы на основе градиента. Выделяют ряд методик дискриминативного обучения.

1) Оценка дискриминативной модели по минимизации частоты ошибок классификации (Minimum Classification Error, MCE) [2], являющейся гладкой аппроксимацией коэффициента ошибок слова или предложения. Оценка MCE появилась из правила принятия

решений Байеса и значительно опережает оценку ML в распознавании речи.

2) Дискриминативные акустические системы могут быть обучены в соответствии с критерием максимизации взаимной информации который выражается как взаимная информация между данными наблюдения X и последовательностью эталонных слов. Оценка MMI параметров НММ обычно осуществляется с помощью расширенного алгоритма Баума-Уэлча, максимизирующего «слабый смысл» вспомогательной функции. Функция сглаживания задается суммой отрицательных расхождений Кульбака-Лейблера между устойчивыми распределениями. Обучение посредством MMI может интерпретироваться как оценка условий максимального правдоподобия (Conditional Maximum Likelihood, CML) [4] – максимизация \log posterior вероятности правильной последовательности слов.

3) Методы нормализации диктора, работающие в области характеристик, нацеленной на подготовку канонической характеристики пространства для устранения, на сколько возможно, изменчивости внедикторских вариаций: деформации оси частот, чтобы соответствовать длине вокального тракта контрольного громкоговорителя, как в длине тракта вокальной нормализации (Vocal преобразование функции максимизации вероятности в рамках нынешней модели как пространство характеристик максимального правдоподобия линейной регрессии (Feature space Maximum Likelihood Linear Regression, fMLLR) [6]. Применение этих методов позволяет сократить ошибку распознавания на

4) Методы пространственно-характеристического дискриминативного обучения. Область характеристик минимальных фоновых ошибок (fMPE) [7] – преобразование, обеспечивающее независимые от времени сдвиги регулярных характеристических векторов. Сдвиги получаются путем линейного проецирования из пространства высокой размерности гауссиан. Проекция обучается таким образом, чтобы повысить уровень распознавания между верными и некорректными последовательностями слов. В сочетании с моделированием области отличительных признаков данный метод обычно приводит к относительному улучшению производительности распознавания на 25% в некоторых задачах.

5) Ограниченное по области линейным преобразованием (Region Dependent Linear Transform, RDLT) [8], которое основывается на

методе fMPE. Оно включает в себя замену смещений на смесь аффинных преобразований.

6) Подход, основанный на методах с большим запасом классификации.

7) Для того чтобы сделать связь с большим запасом классификации более явной, применяется метод с модификацией критерия максимизации взаимной информации (Boosted Maximum Mutual Information, BMMI) [9], который рассматривается как критерий с большим запасом классификации принципа большой разницы (Principle of Large Margin, PLM) [10] и берёт свое начало из задачи ограничительной оптимизации для общей классификации с большим запасом. Применяя этот подход, увеличенная целевая функция получается за счёт введения параметра масштабирования K и увеличения коэффициента внутри целевой функции. Является лучшей дискриминативной обучающей схемой для распознавания слитной речи.

8) Использование метода для снижения степени вариативности речевого сигнала – нормализация среднего кепстра (Cepstral Mean Normalization, CMN) и дисперсии (Cepstral Mean

9) Включение временного контекста в кепстральные кадры для моделирования динамики речевого сигнала. Общая идея этого метода заключается в вычислении скорости и коэффициентов ускорения (дельта и дельта-дельта коэффициентов) для соседних кадров внутри окна, обычно, из 4 кадров. Эти коэффициенты добавляются к статической кепстре для формирования окончательного вектора признаков [12].

10) Применение метода линейной матрицы проекций, отображающей вектор, полученный путем объединения последовательных кадров в пространстве низкой размерности. Проекция разработана таким образом, чтобы максимально отделить фонетические классы в трансформированном пространстве. Разделители обычно вычисляются с помощью критерия линейного дискриминантного анализа (Linear Discriminant Analysis, LDA) [13]. Чтобы сделать гипотезу моделирования диагональной ковариации более допустимой, пространственные признаки LDA «переворачиваются» посредством преобразования полу-привязанной ковариации (Semi-Tied Covariance, STC) и линейные преобразования признаков, максимизирующие среднее правдоподобие (Maximum Likelihood Linear Transformation, MLLT) [14], целью которых является сведение к минимуму вероятности потери между полной и

диагональной ковариацией Гаусса. Использование каскада преобразований LDA и STC приводит к относительному уменьшению (на 10-15%) WER среди простых временных производных для распознавания слитной речи.

11) Стоит также выделить такой метод как подпространство моделей гауссовых смесей (Subspace Gaussian Model Mixtures, SGMM) [15], являющийся модификацией HMM-GMM. В SGMM все состояния HMM используют одну и ту же структуру GMM, с таким же количеством гауссиан в каждом положении. Однако значения и веса гауссовых смесей могут изменяться в подпространстве полного параметрического пространства, которое управляется глобальным преобразованием из векторного пространства в пространство параметров GMM. Т. е. модель определяется векторами, связанными с каждым состоянием N-размерности вместе с глобальным отображением из этого векторного пространства в пространство параметров GMM. Главным преимуществом SGMM является относительная компактность (число параметров, связанных с конкретными речевыми состояниями, довольно мало), что позволяет проводить обучение с меньшим количеством данных и позволяет использовать внеобластные и внеязыковые данные для обучения общих параметров. SGMM показывает больший прирост улучшения качества распознавания для небольшой обучающей выборки, чем для большой выборки. Главным минусом является то, что структура SGMM существенно сложнее GMM-HMM. Следовательно, требует гораздо больше времени для обучения.

Разработка ASR для распознавания слитной русской речи

В данной работе разработка ASR проводилась, применяя методы GMM-HMM с дискриминативным обучением, а также используя SGMM. Для этого использовался инструментарий Kaldi ASR [16], скомпилированный под ОС Ubuntu 16.4. Для обучения АМ использовались аудиофайлы формата wav (16-bit, 16 kHz). Обучающая выборка состоит из 500 дикторов (460 мужских, 40 женских; длительность около 18 ч), а для тестирования ASR использовалась выборка из 150 дикторов (148 мужских, 12 женских; длительность около 2 ч). Для создания словаря транскрипций использовалась нейросетевая модель seq2seq [17], обученная на словаре транскрипций формата ARPA: капельный – – транскрипций составляет 570 тыс. слов.

Разработка ASR состояла из 2 стадий. Для первой стадий разработки использовалась small ЯМ на основе 4-gram, состоящая из текста, взятого из обучающего и текстового наборов.

Для второй стадии разработки была использована middle ЯМ на основе 4-gram, обученная на относительно небольшой выборке (размер текстового корпуса составлял около 300 Мб).

Этапы обучения АМ на первой стадии разработки:

1) Обучение монофонной модели (mono) с 1000 связанных состояний. Используются 13-мерные акустические признаки MFCC (Мел-частотные кепстральные коэффициенты, Mel-Frequency Cepstral Coefficients) [11], дополненные первыми и вторыми производными (для дельта и дельта-дельта коэффициентов).

2) Обучение первой трифонной модели (tri1) с 11000 гауссиан и 2000 связанных состояний, на основе модели mono. Используются 13-мерные MFCC признаки с CMN, дополненные первыми и вторыми производными.

3) Обучение второй трифонной модели (tri2a) с 11000 гауссиан и 2000 связанных состояний, на основе модели tri1. Используются 13-мерные MFCC признаки с CMN, дополненные первыми и вторыми производными.

4) Обучение третьей трифонной модели (tri2b) с 11000 гауссиан и 2000 связанных состояний по всем данным, на основе модели tri2a. В качестве признаков используются взятые для 9 соседних кадров (центральный кадр и по 4 кадра слева и справа) 13-мерные MFCC признаки с CMN, к которым применено LDA-MLLT преобразование с понижением размерности векторов до 40.

5) Дискриминативное обучение третьей трифонной модели с использованием bMMI (tri2b_mmi) и MPE (характеристика минимальных фоновых ошибок, Minimum Phone

6) Обучение четвертой трифонной модели (tri3b) с 11000 гауссиан и 2000 связанных состояний по всем данным, на основе модели tri2b. В качестве признаков используются взятые для 9 соседних кадров (центральный кадр и по 4 кадра слева и справа) 13-мерные MFCC признаки с CMN, к которым применено LDA-MLLT преобразование с понижением размерности до 40, адаптированные к диктору с помощью метода

7) Дискриминативное обучение четвертой трифонной модели с использованием bMMI критерия (tri3b_mmi) по всем данным.

Основой декодера в Kaldi является подход к статическому декодированию, основанный на конечных автоматах (Finite State Automata, FSA) и «взвешенных преобразователях с конечным числом состояний» (Weighted Finite State

Transducers, WFSTs) [11]. А для процесса декодирования использовался модифицированный алгоритм Витерби (Витерби с обрезкой лучей) для оптимизации данных, т. к. генерируемый граф слишком велик при использовании обычного Витерби. Обрезка лучей означает, что мы обращаемся к векторам акустических признаков один за другим, отбрасывая на каждом участке состояния с наихудшей оценкой, причём используя данный подход точность распознавания практически не ухудшается.

В ходе обучения выяснилось, что в данном речевом корпусе содержится большое количество непригодных аудиофайлов для процедуры обучения (не содержащие речи; содержащие лишь шумы или отрывки иностранной речи; содержащие музыкальные вставки; продолжительность превышает 30 сек (рекомендуется использовать аудиофайлы продолжительностью до 30 сек для получения оптимальной АМ)).

Также были выявлены проблемы в аннотированных текстах для речевого материала: часть текстов отсутствовала; существовала проблема «е»-«ё», т. е. в слове, где должен быть символ «ё», присутствовал символ «е», что критично для создания АМ и ЯМ. Вышеуказанные проблемы были решены при помощи автоматизированных скриптов, предназначенных для выявления аудиофайлов без аннотированного текста, и скрипта, который токенизировал слова и проверял наличие такого слова в словаре. Но такой подход не позволил выявить все ошибки, поэтому была проведена дополнительная проверка вручную. Также проблема «е»-«ё» содержится и в файле транскрипции, которая на момент разработки не была устранена.

Одной из причин высокого WER являлось отсутствие некоторых слов в исходном словаре, а следовательно и отсутствие транскрипций и фонем. Для устранения этой проблемы использовалась модель seq2seq, которая автоматически генерировала транскрипцию для отсутствующего слова в формате ARPA, а также добавляла его в словарь транскрипций.

Результат тестирования ASR для первого этапа разработки отображён на табл.1, где для второго этапа разработки была изменена структура обучения: множество извлеченных акустических признаков делится на 3 части (2000 состояний, 5000 состояний и 8000 состояний).

Этапы обучения АМ на второй стадии разработки:

1) Обучение АМ посредством монофонов (mono) используется 2000 связанных состояний,

при её нормировании АМ дополняется 5000 связанными состояниями.

2) Обучение первой трифонной модели (tri1_5k) используется 2500 связанных состояний и 20000 гауссиан, а при её нормировании она дополняется 8000 связанными состояниями. Используются 13-мерные MFCC признаки с CMN, дополненные первыми и вторыми производными.

Таблица 1 – Результат тестирования ASR для первого этапа разработки.

model	WER _{before} , %	WER _{after} , %
mono	43,57	7,25
tri1	34.45	4.05
tri2a	30.09	3.49
tri2b	27.75	4.05
tri2b_mmi	26.33	3.37
tri2b_mpe	25.68	3.49
tri3b	24.03	3.93
tri3b_mmi	22.11	3.24

3) Обучение второй трифонной модели (tri2b_8k) на основе tri1_5k с 40000 гауссиан и 3000 связанных состояний по всем данным. В качестве признаков используются взятые для 9 соседних кадров (центральный кадр и по 4 кадра слева и справа) 13-мерные MFCC признаки с CMN, к которым применено LDA-MLLT преобразование с понижением размерности до 40.

4) Обучение третьей трифонной модели (tri3b_8k) на основе tri2b_8k с 60000 гауссиан и 4000 связанных состояний по всем данным. В качестве признаков используются взятые для 9 соседних кадров (центральный кадр и по 4 кадра слева и справа) 13-мерные MFCC признаки с CMN, к которым применено LDA-MLLT преобразование с понижением размерности до 40, адаптированные к диктору с помощью fMLLR преобразования.

5) Обучение четвёртой трифонной модели (tri4b) на основе tri3b_8k с 60000 гауссиан и 5000 связанных состояний по всем данным, к которым применено SAT (Speaker Adaptive Training, адаптация к диктору) [12]. При обучении АМ, основываясь на полученных значениях вероятностей произношений и тишины, изменяется ЯМ.

6) Обучение пятой трифонной модели (tri5b) на основе tri4b с 120000 гауссиан и 8000 связанных состояний по всем данным, к которым применено SAT. При обучении АМ, основываясь на полученных значениях вероятностей произношений и тишины, изменяется ЯМ.

7) Обучение SGGM модели (sgmm_tri5b) на основе модели tri5b, используя UBM-вектора признаков с 750 состояниями, а также применяя 8000 состояний для обучения AM и 120000 гауссиан.

8) Дискриминативное обучение SGGM модели (sgmm_tri5b_mmi_b0.1) на основе

Результаты тестирования ASR для второго этапа разработки отображены на табл.2.

Таблица 2 – Результаты тестирования ASR для второго этапа разработки.

model	WER,%
mono	43,57
tri1_5k	20.05
tri2b_8k	18.72
tri3b_8k	16.85
tri4b	17.4
tri5b	16.5
sgmm_tri5b	13.19
sgmm_tri5b_mmi_b0.1	12.45

Был проведен эксперимент с использованием метода PLP перцепционные коэффициенты линейного предсказания (Perceptual Linear Prediction, PLP) [19] для извлечения акустических признаков. Используя полученные данные было проведено сравнение двух методов извлечения признаков – MFCC и PLP. В результате был сделан вывод использовать в дальнейшем MFCC, т.к. разница в WER невелика, а используя PLP невозможно применить такие методы, как SGMM. Результат сравнения отображен на табл.3, где WER_{MFCC} – результат распознавания на основе MFCC; WER_{PLP} – на основе PLP.

Таблица 3 – Сравнение MFCC и PLP.

model	WER_{MFCC} , %	WER_{PLP} , %
tri1_5k	20.05	19.65
tri2b_8k	18.72	18.56
tri3b_8k	16.85	16.43

Заключение

Исходя из полученных результатов, можно сделать вывод об актуальности применения статистических методов применительно к задаче распознавания слитной

речи. В данной работе было показано преимущество применения дискриминативных методов для распознавания слитной речи. При разработке ASR с использованием small ЯМ наилучший показатель был получен, используя tri3b_mmi ($WER=3.24\%$); с использованием middle ЯМ наилучший показатель был получен, используя sgmm_tri5b_mmi_b0.1 ($WER=12.45\%$). Было проведено сравнение применения MFCC и PLP для извлечения акустических признаков. Главными проблемами на текущем этапе разработки являются: недостаточное количество аннотированных речевых данных, а также большой текстовый выборки для создания ЯМ. В будущем планируется использовать нейросетевой подход для разработки системы распознавания слитной речи, а также разработать методику для получения аннотированных русскоязычных речевых данных.

Литература

1. Jelinek, F. Statistical methods for speech recognition [Text] / F. Jelinek // MIT Press. – 1997. – p. 283.
2. Juang B.-H., Chou W., Lee C.-H. 1997. Minimum classification error methods for speech recognition. IEEE Trans. Speech Audio Processing. 5 (3): 257–265.
3. Derouault, A.-M. Improving speech recognition accuracy with contextual phonemes and MMI training [Текст] / A.-M. Derouault, B. Merialdo // ICASSP–89, vol.1 – 1989. – P. 116-119.
4. Layton M. Augmented statistical models for speech recognition [Текст] / M. Layton, M. Gales //, Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP). – 2006. – P. 129-132.
5. Hilger, F. Quantile based histogram equalization for noise robust large vocabulary speech recognition [Текст] / F Hilger, H Ney // IEEE Transactions on Audio, Speech, and Language Processing 14 (3) – 2006. – P. 845-854.
6. Wegmann S. Speaker normalization on conversational telephone speech [Текст] / S. Wegmann, D. McAllaster, J. Orloff, B. Peskin // ICASSP, no. 1, Atlanta. – 1996. – P. 339-341.
7. Huang, J. Rapid Feature Space Speaker Adaptation for Multi-Stream HMM-Based Audio-Visual Speech Recognition [Текст] / J. Huang; E Marcheret; K Visweswariah // IEEE International Conference on Multimedia and Expo (ICME). IEEE. – 2005. – P. 338-341.
8. Zhang, B. Discriminatively trained region dependent feature transforms for speech recognition / B. Zhang, S. Matsoukas, R. Schwartz // Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP). – 2006. – P. 313-316.
9. Saon, G. Large margin semi-tied covariance transforms for discriminative training

[Текст] / G. Saon, D. Povey, H. Soltau // Proc. Int. Conf. Acoustic, Speech, and Signal Processing (ICASSP). – 2009. – P. 3753-3756.

10. Yu, D. Large-margin discriminative training of hidden markov models for speech recognition [Текст] / D. Yu, L. Deng // ICSC. – 2007. – P.364-371.

11. Л. Рабинер, Р. Шафер, Цифровая обработка речевых сигналов. – М.: Радио и связь, 1981. – 496 с.

12. Chen, Y. The XMU phrase-based statistical machine translation system for IWSLT 2006 [Текст] / Y. Chen, X. Shi, C. Zhou // IWSLT-2006. – 2006. – P. 153-157.

13. Furui, S. Speaker-independent isolated word recognition using dynamic features of speech spectrum [Текст] / S. Furui // IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 34. – 1986. – P. 52-59.

14. Saon, G. Maximum likelihood discriminant feature spaces [Текст] / G Saon, M Padmanabhan, R Gopinath, S Chen // ICASSP'00, vol.2 – 2000. – P.1129-1132.

15. Povey, D. The Subspace Gaussian Mixture Model – a Structured Model for Speech Recognition [Текст] / D. Povey, L. Burget et. al. // Computer Speech and Language. – vol.25. – 2011. – P.409-439.

16. Povey D. et al. The Kaldi speech recognition toolkit //IEEE 2011 workshop on automatic speech recognition and understanding. – IEEE Signal Processing Society, 2011. – №. EPFL-CONF-192584.

17. Sutskever I., Vinyals O., Le Q. V. Sequence to sequence learning with neural networks //Advances in neural information processing systems. – 2014. – P. 3104-3112.

18. Povey D., Kuo H. K. J., Soltau H. Fast speaker adaptive training for speech recognition //Ninth Annual Conference of the International Speech Communication Association. – 2008.

19. Hermansky, H. Perceptual linear predictive (PLP) analysis of speech [Текст] / H. Hermansky // the Journal of the Acoustical Society of America. – 1990. – Vol. 87, no. 4. – P. 1738-1752.

Пикалёв Я. С. Разработка системы автоматического распознавания слитной русскоязычной речи на основе дискриминативного обучения. В данной работе рассмотрены методики распознавания речи на основе дискриминативного обучения. Рассмотрен процесс обучения системы автоматического распознавания русскоязычной слитной речи, используя *small* и *medium* языковую модель, а также различные вариации акустических моделей. Проведено сравнение методов, применяемых для извлечения акустических признаков. По результатам тестирования сделаны выводы об актуальности применения дискриминативного обучения для автоматического распознавания русскоязычной слитной речи.

Ключевые слова: автоматическое распознавание слитной речи, GMM-HMM, SGMM, дискриминативное обучение.

Pikaliov Y. S. Development of a system for automatic recognition of Russian-speaking speech, based on discriminative learning. In this paper, speech recognition techniques based on discriminative training are considered. The process of training the system of automatic recognition of Russian-speaking continue speech is considered, using a small and medium language model, as well as various variations of acoustic models. A comparison of the methods used to extract acoustic features is made. Based on the results of testing, conclusions were drawn about the relevance of the use of discriminative learning for automatic recognition of Russian-speaking continue speech.

Keywords: automatic continuous speech recognition, GMM-HMM, GMM, discriminative training.

Статья поступила в редакцию 19.05.2018 г.
Рекомендована к публикации доцентом Федяевым О.И.

УДК 004.048

Разработка системы нечеткого вывода для оценки компетенций претендентов на должности IT-отдела

А. В. Выклюк, С. Ю. Землянская
ГОУВПО «Донецкий национальный технический университет»
vyknastya95@gmail.com, zsaae07@gmail.com

Выклюк А. В., Землянская С. Ю. Разработка системы нечеткого вывода для оценки компетенций претендентов на должности IT-отдела. В статье описаны основные этапы оценки компетентности сотрудников, приведен алгоритм и рассмотрен поэтапный принцип работы систем нечёткого вывода для оценки компетентности персонала.

Ключевые слова: компетентность, оценка персонала, система нечёткого вывода, лингвистическая переменная, интеллектуальная система.

Введение

Одной из особенностей современного этапа развития общества является процесс его информатизации. Эффективная работа любой организации невозможна без автоматизации различных сфер её деятельности. Это существенно повышает востребованность специалистов в области информационных технологий в организациях практически любой сферы. Современный рынок труда предлагает большое количество IT-специалистов, однако в ситуации высокой конкуренции работодателям сложно реализовать выбор подходящего сотрудника с учетом требований вакантной должности.

Тестирование сотрудника производится с целью выявления его уровня компетенций, а именно, наличия определенных знаний и навыков, умений и личностных характеристик, а также способности ими пользоваться на должном уровне со стопроцентной отдачей. В каждой компании существуют свои корпоративные правила и негласные устои, которым нужно соответствовать для достижения максимального результата. Для адекватной оценки навыков сотрудников нужны кастомизированные методы.

Основной целью оценки сотрудников по компетенциям является организация эффективного использования имеющегося кадрового потенциала с учетом специфики предприятия и, как следствие, экономии фонда заработной платы, рабочего времени и затрат на вспомогательные ресурсы, требующиеся для выполнения определенных обязанностей.

В настоящее время большинство руководителей для оценки уровня подготовки кандидатов используют традиционные методы, без применения средств автоматизации. Как следствие, приходится затрачивать большое количество времени на проведение

тестирования, а также обработку результатов, причем результаты оценки могут оказаться недостаточно объективными.

Постановка проблемы

Для проведения аттестации IT-персонала следует применять сложные комбинированные методики для того, чтобы составить полное представление о том, насколько специалист соответствует должности и как правильно построить его дальнейшее развитие в структуре компании. Разработка экспертной системы, позволяющей автоматизировать работу специалиста по подбору персонала при проведении оценки сформированности необходимых профессионально-значимых и личностных качеств кандидатов и интерпретации полученных материалов позволит оказать интеллектуальную поддержку при принятии решения о соответствии кандидатов требованиям вакантной должности IT-отдела.

Процесс оценки компетентности персонала включает следующие этапы:

1. Формируется набор ключевых компетенций для данного отдела/предприятия.
2. Выбирается методика проведения тестового оценивания.
3. Составляется шкала оценивания.
4. Определяются критерии значимости каждого вида тестирования путём назначения весовых коэффициентов.
5. Проводится тестирование персонала.
6. Определяется степень соответствия сотрудника определённой должности на основании результатов оценивания.
7. Делаются соответствующие выводы и даются рекомендации.

Для эффективного анализа компетентности необходимо применять подход, допускающий нечёткости и частичные истины.

Таким требованиям соответствуют экспертные системы на основе нечёткого вывода (рис. 1).

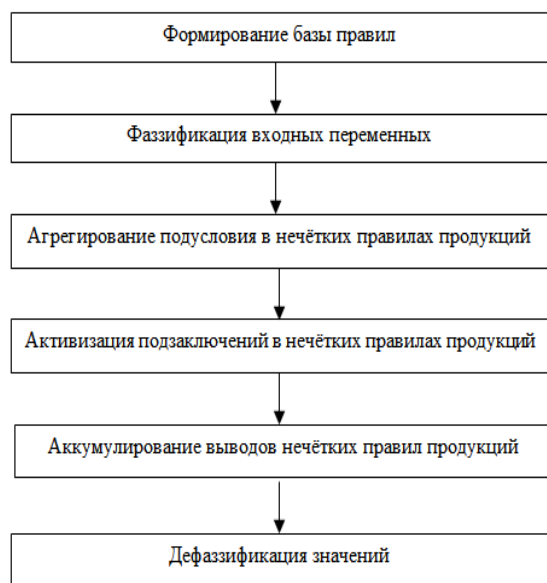


Рисунок 1 – Алгоритм работы системы нечёткого логического вывода

Этот подход имеет три отличительные черты:

1. В нём используются «лингвистические» переменные вместо числовых переменных или в дополнение к ним.
2. Простые отношения между переменными описываются с помощью нечетких высказываний.
3. Сложные отношения описываются нечёткими алгоритмами.

Лингвистической переменной называется набор $\langle \beta, T, X, G, M \rangle$, где:

β — имя лингвистической переменной;
 T — множество её значений (термножество), представляющее имена нечётких переменных;

X — универсум нечетких переменных, область определения;

G — синтаксическая процедура образования новых термов;

M — семантическая процедура, формирующая нечеткие множества для каждого терма данной лингвистической переменной.

Шкала оценивания каждой компетенции задается с помощью лингвистической переменной $K = [K_1, K_2, \dots, K_t]$, $t=5$.

β — уровень компетентности;
 $T = \{ \text{“Низкий”}, \text{“Ниже среднего”}, \text{“Средний”}, \text{“Выше среднего”}, \text{“Высокий”} \}$;

$X = [0, 1]$;
 Требования к компетентности сотрудников, а также сама компетентность являются непостоянными величинами в силу инновационного развития предприятия, научно-технического прогресса, морального

устаревания техники и технологий. Поэтому крайне важно своевременно формировать у персонала требуемую компетентность и поддерживать ее на соответствующем уровне.

Для решения этой задачи должны существовать инструменты и способы определения стандартов компетентности, инструменты и способы оценки компетентности сотрудников, а также методики и программы формирования компетентности согласно выработанным стандартам. Одним из таких способов является синтез модели компетенций. Термин «модель компетенций» подразумевает полный набор ключевых компетенций и индикаторов поведения, необходимых сотруднику.

Модель компетенций можно представить в виде иерархической структуры (рис. 1), где C_i — компетенция верхнего уровня; C_{ij} — компетенция 2-го уровня и т.д. Набор тесно связанных между собой компетенций образует кластер компетенций, например, компетенции C_{11}, C_{1m} образуют кластер C_1 .

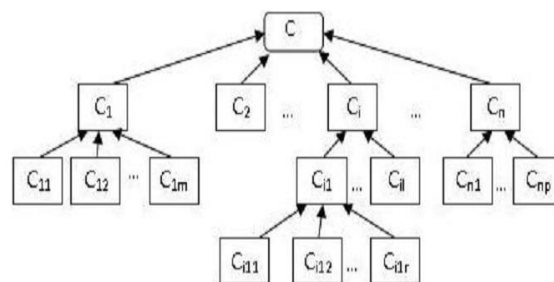


Рисунок 2 – Структура модели компетенций

Для каждой имеющейся или вновь создаваемой должности должна быть разработана модель компетенций, определены ее структура и глубина, разработаны индикаторы поведения (стандарты поведения в действиях человека, обладающего конкретной компетенцией). С помощью полученной модели проводится оценка компетентности персонала, и в зависимости от результата принимается решение относительно каждого конкретного сотрудника.

Исходной информацией для формирования модели компетенций являются названия компетенций, из которых формируется список имен лингвистических переменных, затем проводится их кластеризация, вводятся новые имена лингвистических переменных, соответствующих названиям полученных кластеров, определяются универсальные множества для каждой лингвистической переменной, множества названий лингвистических значений переменных (термножество), синтаксис и семантика лингвистических переменных; т.е. формируется

множество входных лингвистических переменных.

Таким же образом синтезируется выходная лингвистическая переменная, такая, что ее значение характеризовало бы степень соответствия компетентности сотрудника нечеткой модели компетенций. Далее на основании экспертных заключений происходит идентификация нелинейных зависимостей переменных нечеткими базами знаний (наиболее распространенные Мамдани и Сугено [3]). Полученная нечеткая модель обучается и тестируется, а при достижении удовлетворительных результатов запускается в процесс оценки компетентности персонала. Если результаты модели оказываются неудовлетворительными, вышеописанный процесс повторяется заново, либо вносятся коррективы. Если эти действия также не приводят к положительному результату, то следует изменить образ объекта (расширить, сузить, сдвинуть) и тем самым изменить список имен лингвистических переменных. Такой циклический процесс на определенной итерации приведет к построению адекватной нечеткой модели (формула 1)

$$FCM = f(KB, AL, PQ, T, Y), \quad (1)$$

где FCM – нечеткая модель компетенций, KB – база знаний, Y – множество выходных нечетких переменных, AL, PQ, T – множества входных нечетких переменных: AL – список должностей, PQ – множество персональных качеств, T – множество технических требований

Компетентность в области информационных технологий, характеризующая готовность будущих IT-специалистов к профессиональной деятельности, определяется уровнем сформированности профессионально-значимых качеств, с одной стороны и уровнем сформированности личностных качеств, с другой. Результаты мониторинга мнения руководителей по выбору наиболее важных профессионально-значимых и личностных качеств IT-специалистов [2] представлены в таблице 1.

Следует отметить, что если в качестве претендента на вакантную должность выступает выпускник вуза, то при оценке его профессиональной компетентности добавляются компетенции, указанные в ГОС ВПО для соответствующей специальности, при этом для формирования той или иной компетенции изучение различных дисциплин имеет неодинаковое значение, и сами компетенции, формируемые в процессе обучения, могут иметь различную значимость для разных работодателей.

Таблица 1 – Значимость профессиональных и личностных качеств IT-специалистов

№	Оцениваемые качества	Значимость, %	Весовые коэффициенты
Профессиональные качества			
1	Знания в области информационных технологий	12	0,9
2	Умение проектировать вычислительные сети	24	0,9
3	Навыки программирования	26	0,7
4	Опыт профессиональной деятельности	20	0,7
5	Мотивация к самореализации в проф. деятельности	18	0,7
Личностные качества			
6	Стремление к самообучению	26	0,5
7	Умение работать в команде	22	0,5
8	Ответственность	14	0,3
9	Стрессоустойчивость	10	0,3
10	Самостоятельность	8	0,3
11	Усидчивость	6	0,5
12	Коммуникабельность	6	0,9
13	Креативность	4	0,9
14	Аккуратность	2	0,7
15	Лидерство	2	0,7

Таким образом, можно говорить об относительном (зависящем от конкретного работодателя) уровне компетентности молодого специалиста (как следствие, один и тот же выпускник может представлять различный интерес для государственных организаций и для частных компаний IT-сферы), что еще более усложняет задачу эффективного подбора и оценки кадров.

При оценке компетентности будет использоваться итоговый рейтинг кандидатов по каждому типу компетенций: академических, социально-личностных и профессиональных.

Компетенции рассматриваются как критерии, и интегральная оценка профессиональной компетентности может проводиться с помощью одного из методов решения многокритериальных задач [7]. При использовании метода линейной свертки критериев оптимальной считается альтернатива (решение) с максимальным значением глобального или векторного критерия. Поскольку каждую из академических

компетенций можно оценивать по совокупности изученных дисциплин, причем роль конкретной дисциплины для одной компетенции может существенно отличаться от ее роли для формирования другой, то для оценки общего уровня профессиональной компетентности претендента на IT-должность будем использовать формулу:

$$K = \sum_{i=1}^{na} w_i A_i, \quad (2)$$

где K – оценка общего уровня компетентности претендента, na – количество компетенций, используемых для оценки профессионального академического уровня компетентности, w_i – нормированный коэффициент значимости i -й компетенции, A_i – уровень овладения i -й компетенцией:

$$A_i = \sum_{j=1}^{nd} z_{ij} d_j, \quad (3)$$

nd – количество дисциплин, участвующих в формировании i -й компетенции, z_{ij} – нормированный коэффициент значимости дисциплины j для формирования компетенции i , d_j – уровень усвоения дисциплины (по результатам итогового оценивания)

В предлагаемой экспертной системе при составлении списка должностей задается значение уровня компетентности, соответствующие каждой из них. Список должностей может редактироваться и дополняться. Уровень компетентности в области информационных технологий позволит судить о степени соответствия кандидата определённой должности. Шкала распределения уровней компетентности по должностям представлена в табл. 2.

Рассмотрим, как оценивается соответствие претендента определенной должности IT отдела с учетом профессиональных, личностных и академических компетенций.

Рассчитывается уровень компетентности претендента с учетом академических компетенций, соответствующих выбранной должности по формуле (2). Затем с учетом значимости профессиональных и личностных качеств, отобранных руководителем (табл.1) определяется вторая составляющая компетентности претендента следующим образом. Задаются уровни владения компетенциями для каждой из должностей L :

$$L = \{ \{ L_{min1}, \dots, L_{m1}, \dots, L_{max1} \}, \dots, \{ L_{min(nv)}, \dots, L_{m(nv)}, \dots, L_{max(nv)} \} \},$$

где nv – количество должностей

Таблица 2 – Соответствие уровней компетентности должностям

Уровень компетентности	Значение уровня компетентности	Соответствующая должность
Высокий	0,81—1,0	Руководитель отдела/департамента.
Выше среднего	0,61—0,8	Ведущий специалист (программист, веб-программист/дизайнер)
Средний	0,41—0,6	Администратор (системный, БД).
Ниже среднего	0,21—0,4	Инженер (службы технической поддержки, сетевой).
Низкий	0—0,2	Технический писатель (копирайтер, контент-менеджер)

На рисунке 3 представлен вид треугольной функции принадлежности для уровня компетентности.

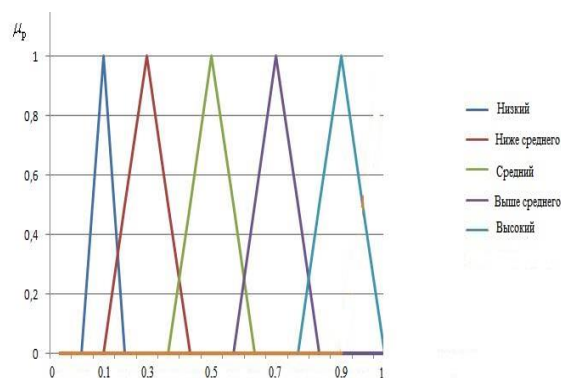


Рисунок 3 – График функции принадлежности для лингвистической переменной «Уровень компетентности»

По результатам тестирования, опросов определяются значения уровня владения каждой из компетенций для претендента на должность: $P = \{P_1, P_2, \dots, P_{np}\}$. Может быть составлена такая элементарная база правил для определения соответствия должности:

$d1$: “ЕСЛИ $P_1 = L_{max1}$ И $P_2 = L_{max1}$, И $P_3 = L_{max1}$, ТО $K = \text{Высокий}$ ”;

$d2$: “ЕСЛИ $P_1 = L_{m1}$ И $P_2 = L_{m1}$, И $P_3 = L_{m1}$, ТО $K = \text{Выше среднего}$ ”;

$d3$: “ЕСЛИ $P_1 = L_{m2}$ И $P_2 = L_{m2}$, И $P_3 = L_{m2}$, ТО $K = \text{Средний}$ ”;

$d4$: “ЕСЛИ $P_1 = L_{m3}$ И $P_2 = L_{m3}$, И $P_3 = L_{m3}$, ТО $K = \text{Ниже среднего}$ ”;

$d5$: “ЕСЛИ $P_1 = L_{min4}$ И $P_2 = L_{min4}$, И $P_3 = L_{min4}$, ТО $K = \text{Низкий}$ ”

Каждое правило имеет приоритет, который определяется в соответствии с важностью (значимостью) правила. Значимость назначается путем экспертной оценки W_i в виде числового значения от 0 до 1. Подобные правила составляются для всех компетенций и всех должностей, впоследствии база правил будет расширена и дополнена. Соответствие должности получаем как интегральную оценку академической и профессионально-личностной компетентности претендента.

Заключение

Произведен анализ распространенных методов и моделей для решения задач управления персоналом. Описан наиболее подходящий метод решения задачи оценки компетентности персонала на основе систем нечеткого вывода и многокритериального анализа. Приведено описание экспертной системы и процесса построения базы правил. Процесс оценки компетентности сотрудников и их соответствия занимаемым должностям целесообразно проводить с использованием интеллектуальной системы на основе нечетких методов, работа которой позволит получить объективную оценку соответствия уровня компетентности персонала требованиям руководства организации.

Литература

1. Насейкина Л. И., Соколова И. М. Автоматизация поддержки принятия решения подбора персонала IT-отдела на основе оценки

профессионально-значимых и личностных качеств кандидатов // Технические науки - от теории к практике: сб. ст. по матер. XXXIII междунар. науч.-практ. конф. № 4(29). – Новосибирск: СибАК, 2014.

2. Азарнова Т.В., Степин В.В., Щепина И.Н. Повышение эффективности методов управления развитием персонала на основе нейросетевых моделей и нечетких экспертных технологий // Вестник ВГУ. Серия: Экономика и Управление, 2014. – №3. – С. 121–130.

3. Штовба С.Д. Проектирование нечетких систем средствами MATLAB / С.Д. Штовба. – М.: Горячая линия-Телеком, 2007. – 256 с.

4. Звездова А.Б. Компетентностный подход в высшем профессиональном образовании. / А.Б. Звездова, В.Г. Орешкин. [Электронный ресурс] – Режим доступа: URL: http://www.miep.edu.ru/uploaded/zvezdova_oreshkin.pdf

5. Фот Ю.Д. Модели и алгоритмы системы отбора персонала на основе повышения достоверности данных при принятии решений. Автореф. дисс. на соиск. уч. степ. канд. тех. наук по спец. 05.13.10. Пенза, 2013. — 22 с.

6. Берестнева О.Г., Марухина О.В., Абунаваз Х.А. Алгоритмическое и программное обеспечение информационной системы оценки компетентности студентов технического вуза // Известия Томского политехнического университета. – 2006. – Т.309. – № 7. – С. 240–245.

7. Таха Х.А. Введение в исследование операций / Х.А. Таха. – М.: Вильямс, 2001. – 912 с.

Выкльок А. В., Землянская С. Ю. Разработка системы нечеткого вывода для оценки компетенций претендентов на должности IT-отдела. Описаны основные этапы оценки компетентности сотрудников. Сформулированы цели проведения такой оценки на предприятиях. Выделены основные профессиональные и личностные качества, необходимые IT-специалисту и их значимость. Разработан алгоритм и рассмотрен поэтапный принцип работы систем нечеткого вывода для оценки компетентности персонала.

Ключевые слова: Компетентность, оценка персонала, система нечеткого вывода, лингвистическая переменная, интеллектуальная система.

Vyklyk A. V., Zemlyansky S. Y. Development of fuzzy inference system of competencies of applicants for positions in the IT department evaluation. Main stages of assessing of employees competence are described. The purpose of such assessment in enterprises is formulated. The main professional and personal qualities necessary for the IT specialist and their significance are defined. The algorithm and the phased principle of fuzzy inference systems for evaluating staff competence are developed

Keywords: Competence, personnel assessment, fuzzy inference system, linguistic variable, intelligent system.

Статья поступила в редакцию 07.05.2018 г.
Рекомендована к публикации доцентом Приваловым М. В.

УДК 681.513.7

Сервисы мониторинга социальных сетей

И. Ю. Анохина, В. Г. Черепов

Донецкий национальный технический университет
Ingatula@mail.ru

Анохина И. Ю., Черепов В. Г. Сервисы мониторинга социальных сетей. Показана работа сервисов анализа социальных сетей. Рассматриваются характеристики оценки активности и степени вовлеченности аудитории. На примере страниц ДонНТУ в социальных сетях показано, как выбирается оптимальное время подачи новостей, как найти самых активных и авторитетных участников, как оценить уровень упоминаний в СМИ.

Ключевые слова: социальные сети, ВКонтакте, Твиттер, степень вовлеченности, активность аудитории, мониторинг сетей.

Введение

Социальные сети с каждым годом становятся все более важной и существенной частью социальной жизни. Появились отрасли науки, например, социодинамика, посвященные изучению влияния сетей на различные аспекты жизни.

Социальная сеть незаменима для бизнеса: продвижение товара, реклама, поиск новых клиентов и повышение лояльности уже существующих, создание своего имиджа и репутации на рынке. Социальные сети широко распространены в образовании, это страницы вузов, школ, техникумов. Существуют страницы политических групп и страницы, посвященные хобби, путешествиям; личные страницы.

О важности исследования социальных сетей еще в 2012г. писала газета Коммерсант в статье «Разведка ботом. СВР (служба внешней разведки) займется социальными сетями» [1]. В качестве основных задач ставились «исследование процессов формирования сообществ интернет - центров распространения информации в социальных сетях» и «определением факторов, влияющих на популярность и распространенность сведений».

Наш университет имеет свои страницы в нескольких социальных сетях: ВКонтакте, Фейсбуке, Твиттере, Одноклассниках.

Мы проанализировали указанные страницы. За неделю (с 3.11 по 10.11.17г.) на странице ДонНТУ в Контакте количество лайков и репостов составляло 28 и 6 соответственно. По данным сервиса <http://smmur.ru> количество реальных подписчиков 2240, активность подписчиков в сутки 0.35%, активность на один пост 0.088% от числа подписчиков. В среднем в сутки приходится семь лайков и один репост [2,3].

В твиттере за неделю понравилась информация только семи пользователям[4].

Аналогичные показатели и по страницам в Одноклассниках и Фейсбуке.

В связи с этим возникает вопрос, насколько целесообразно иметь одновременно несколько страниц в социальных сетях, может быть нужно иметь одну страничку в наиболее популярной сети для данного региона, города? Может быть, время подачи новостей следует ограничить определенным диапазоном. Влияет ли день недели на активность подписчиков.

Для того, чтобы дать ответы на эти вопросы, необходимо провести анализ групп в социальных сетях и на основании полученных данных разработать рекомендации.

Постановка задачи

Основными задачами информационно-аналитической работы с социальными сетями являются мониторинг и анализ социальных сетей, это необходимо для определения оптимальных путей достижения цели, будь то реклама предприятия, раскручивание бренда, агитация или попытка воздействовать на социум для продвижения политической идеи. Аналитика сетей позволит смоделировать происходящие в них процессы, а, следовательно, дает инструмент для управления и прогнозирования.

Многие промышленные гиганты сообщают о разработке специальных методов и моделей сбора первичных данных и анализа социальных сетей. Однако эти разработки не появляются в свободном доступе. Ясно, что небольшое предприятие/организация не будут заниматься глобальными разработками для анализа социальных сетей. А значит, требуется алгоритм, позволяющий на уровне существующих и по возможности бесплатных сервисов получать первичную информацию, оценивать тенденции сети, и, наконец, просто определять, в какой из сетей и в каком именно виде лучше предоставлять информацию.

На первом этапе нами была поставлена задача проанализировать существующие сервисы сбора первичной информации в социальных сетях, причем желательно найти те сервисы, которые могут одновременно работать с несколькими сетями.

Мы мониторили страницы нашего вуза с использованием нескольких сервисов, что позволило, во-первых, определить их доступность, работоспособность, а во-вторых, оценить сами анализируемые страницы.

Исследования

Сервис Livedune предоставляет статистику по блогерам из России, Казахстана, Беларуси, Украины и др. стран СНГ, пишущим в Instagram, ЖЖ, ВКонтакте и Твиттере, анализирует группы, работает с 2015г. [5].

Сервис предоставляет данные о количестве подписчиков, лайков, комментариев, тенденции изменения числа подписчиков и их активности, выкладывает материалы, которые пользовались максимальным спросом, так

называемые топы по лайкам и комментариям.

К сожалению, бесплатный период длится только три дня, дальше требуется оплата в размере приблизительно 1000р. в месяц. Но есть лазейка. При регистрации указывается адрес электронной почты, но не фиксируется IP – адрес компьютера, что позволяет при необходимости использовать сервис с разных электронных ящиков.

На рис. 1. показаны варианты работы с сервисом. Для примера нами были использованы страницы сайта Государственного музея изобразительных искусств имени А. С. Пушкина в Instagram (https://www.instagram.com/theartsmuseum/, цифра 1 рисунка); NASA в Твиттере (Национального управления по авиации и исследованию космического пространства, https://twitter.com/nasa, цифра 2 рисунка), Лаборатории Касперского в Фейсбуке (https://www.facebook.com/KasperskyLabRussia, цифра 3 рисунка).

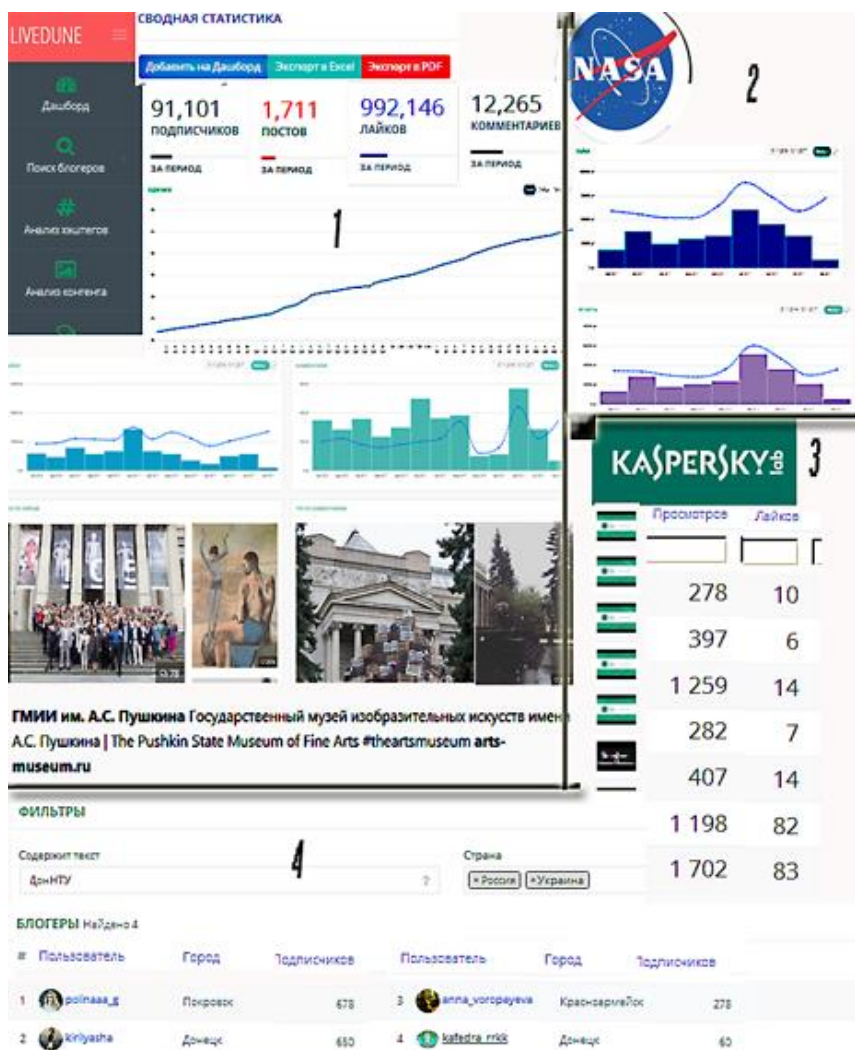


Рисунок 1 – Результаты работы сервиса Livedune

Как видно, у музея более 90 тысяч подписчиков, около миллиона лайков за месяц, при этом аудитория неуклонно растет. Сервис производит выборку наиболее интересных статей за анализируемый период. На основании этого можно делать вывод о том, что интересует участников в большей мере, а что не пользуется успехом.

У NASA около 27 млн. подписчиков, чаще всего сайт просматривают в выходные и понедельник, оптимальное время для выдачи новостей около 16 часов (цифра 2 рисунка).

Для Лаборатории Касперского мы ограничились просмотром соотношения количества просмотров и лайков.

Цифрой 4 отмечена еще одна интересная особенность сервиса. По запросу, в нашем случае «ДонНТУ», выдается список блогеров, пишущих о вузе. Найдено 4 блогера, правда, два из них описывают клон вуза в г. Красноармейске.

Таким образом, сервис позволяет получить достаточно полную информацию о группах в различных социальных сетях, а именно: количество подписчиков, характеристики их активности (просмотры, лайки, репосты, комментарии), тенденции изменения численности подписчиков в группе, распределение активности по дням недели и времени суток, осуществляет подборку топовых новостей, выдает список блогеров, пишущих по выбранной тематике.

Недостатком сервиса, как было сказано выше, является отсутствие бесплатного доступа.

Сервис Popsters.ru — онлайн-сервис статистики и аналитики контента. Поддерживает следующие социальные сети: Вконтакте, Facebook, Одноклассники, Instagram, Twitter, YouTube, Pinterest, Coub, Google+, Flickr.

К сожалению, бесплатным является лишь недельный период, затем ежемесячная стоимость составляет 399р. за одну сеть. Это означает, что если вы, допустим, хотите отследить группы Вконтакте, Facebook и Одноклассниках, то за каждую сеть придется платить указанную сумму. Кроме этого для отслеживания сети вы должны быть в ней зарегистрированы.

На рис.2 – исследование Твиттера.

Для поиска достаточно войти в нужную сеть и указать адрес или название страницы. Автоматически вы получите информацию за один месяц, но можно устанавливать произвольные строки и даты. Вы получаете различные типы графиков вовлеченности пользователей по дням недели и времени суток. Например, сайт NASA чаще всего посещают по понедельникам около 16 часов, что

подтверждается данными рассмотренного выше сервиса **Livedune**. Для ДонНТУ пиковым днем оказалось воскресенье, в остальные дни активность сохраняется практически постоянной. Максимальное число посетителей страницы – 8:00. Практически нет посетителей в период между 17 и 20, после 20:00 на странице опять появляются посетители. Это можно объяснить, в 8 утра многие открывают страницу, чтобы узнать новости, время 17-20 – время возвращения домой.

Таким образом, важные новости должны появляться на наших страницах к утру, как в старое время утренние газеты или после 20:00, как вечерняя пресса.

Сервис **Popsters.ru** позволяет оценить уровень вовлеченности читающих страницы.

Показателей, характеризующих уровень активности аудитории, довольно много. Наиболее применимые:

Вовлечение (Engagement) - единица реакции аудитории: лайк [like, нравится], комментарий, шер [поделиться, share, ретвиты, retweet], клики (переходы по ссылке), дизлайки [не нравится, dislikes] для YouTube.

Объем вовлечения (Engagement Volume) - общее количество вовлечений (сумма всех действий).

Уровень привлекательности (Love Rate) - количество лайков, деленное на количество участников в группе, сообществе, в %.

Уровень общительности (Talk Rate) - количество комментариев, деленное на количество участников в группе, сообществе, в %.

Охват (Reach) - количество людей, имевших хотя бы один контакт с публикацией.

Просмотры (Views) - количество просмотров публикации.

Сервис **Popsters** рассчитывает параметр ER-показатель уровня вовлеченности аудитории, вычисляется по формуле:

$$ER = \frac{\text{сумма всех интеракций в месяце}}{\text{количество подписчиков сообщества}} \%$$

Под интеракцией понимается любое действие пользователя.

Отметим, что именно социальные сети предоставляют уникальный шанс для авторов прямого взаимодействия с аудиторией. Ставя лайк, пользователь не только соглашается с публикацией, сообщает, что новость ему нравится, но и ставит свою уникальную подпись. Поэтому вопрос оценки уровня вовлеченности аудитории в коммуникацию является крайне существенным.



Рисунок 2 – Результаты работы сервиса Popsters

Сервис Followerwonk. Одним из лучших инструментов для аналитики Твиттера, к сожалению, платным, считается сервис Followerwonk (<https://moz.com/followerwonk/analyze/>) [6,7].

Известно, что твит «живет» не более 20 минут. Только в течение этого времени велика вероятность, что новость прочтут и ретвитнут, увеличив тем самым охват. Редко, когда читают ленту твитов каждый день от начала до конца или просматривают новости за предыдущий день. Исходя из этого, самым существенным является определение оптимального времени подачи важной информации и поиск влиятельных читателей, т. е. тех, кто ретвитнет информацию и у кого будут в свою очередь читать новость его подписчики.

На рис. 3 показана незначительная часть вариантов работы сервиса. Цифрой 1 отмечено месторасположение читателей твиттера ДонНТУ. Цифрой 2 – временная диаграмма их активности. Отметим, что результаты сервиса не совсем совпадают с данными проведенного ранее анализа. Пики активности приходятся на 21:00 и 14:00. Это лишь подтверждает наше мнение, что анализировать сети желательно с использованием различных сервисов, сравнивая их результаты.

Ниже на рисунке показана диаграмма оценки социального авторитета читателей. Авторитет изменяется от 1 до 100, чем выше значение, тем больше авторитет. Диаграмма отображает процент читателей с разными уровнями авторитета.

Например, уровень авторитета от 41 до 50 имеют 11.7% читателей. К ним принадлежат такие блоги, как «Новости науки и нанотехнологий», «Микроблог Северо-Восточного федерального университета имени Максима

Амосова», «Журнал интерьеров»

На сервисе дается статистика по возрастному составу читателей, оценивается процент женщин и мужчин, «возраст» присутствия в Твиттере вообще. И что особенно важно, по любому запросу (возраст, пол, социальный авторитет, активность...) можно получить список читателей и географию их размещения.

Сервис Babkee. Если ставится задача определить, как часто в Сети упоминается кампания/организация, оценить, в каком именно контексте происходит упоминание, можно использовать сервис Babkee (работает с 2011г.). Как пишут создатели сервиса, «основная цель Babkee - качественный мониторинг упоминаний в социальных медиа и СМИ».

Сервис платный, однако, на постоянной основе предоставляется возможность мониторить бесплатно два любых объекта, при этом сервис будет выдавать не более 3000 сообщений об объекте в месяц. Если мониторятся конкретные предприятия или организации, то такое количество вполне достаточно, а вот если требуется анализировать еще и сайты конкурентов, то предоставляемых бесплатно возможностей будет недостаточно.

На рис. 4 показаны варианты работы с сервисом. Нами были созданы два объекта мониторинга, цифрой 1 отмечены статьи, в которых упоминался ДонНТУ, каждую из статей можно открыть, что показано на примере упоминания ДонНТУ на страницах Голоса ДНР (цифра 2 рисунка).

Цифра 3 – список страниц, на которых упоминался вуз, например, на страницах Министерства образования, в lenta.ru, в новостях Горловки.

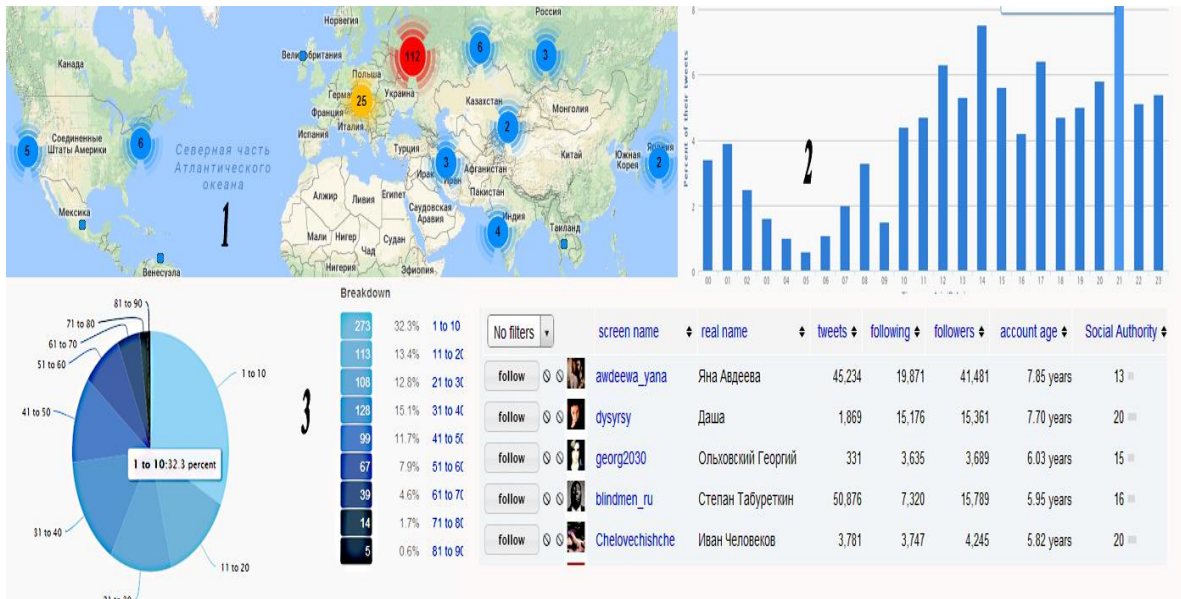


Рисунок 3 – Результаты работы сервиса Followerwonk

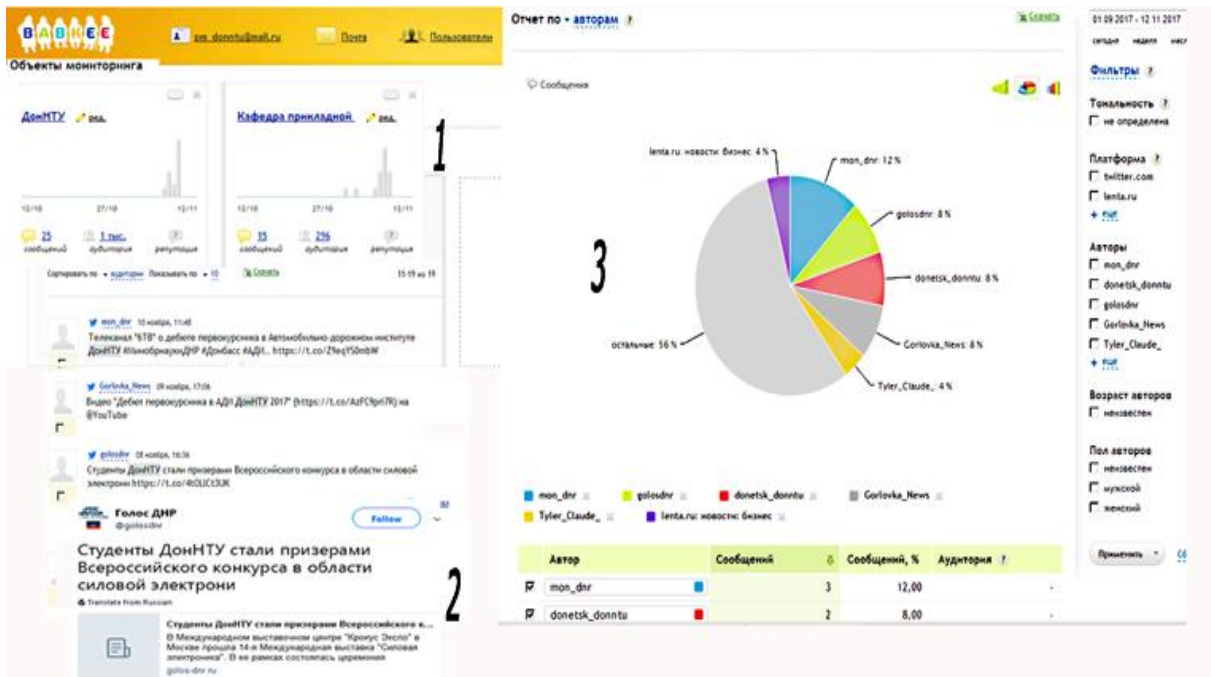


Рисунок 4 – Результаты работы сервиса Babkee

Говорят, что в Интернете ничего нельзя спрятать и потерять, или полностью удалить из Сети непонравившуюся страницу. Ярким подтверждением этому является Архив Интернета (англ. Internet Archive). Эта некоммерческая организация основана в 1996 году в Сан-Франциско Брюстером Кейлом. Архив собирает копии веб-страниц, графические материалы, видео-, аудиозаписи, программное обеспечение. Архив обеспечивает бесплатный доступ к базам данных любому пользователю. По мнению создателей Архива это необходимо для сохранения культурно-исторических ценностей в эпоху интернет-

технологий, создания и поддержки электронной библиотеки.

Web.archive.org — глобальный проект, целью которого является создание архива сайтов, размещенных в интернете. Сайты сохраняются не в виде скриншотов, а в виде полноценно работающих веб-страниц со всеми ссылками, картинками, комментариями. Для каждого сайта может накопиться по несколько копий, датированных разными этапами жизни ресурса [9]. Особенную ценность сервису придает то, что его можно использовать как практически единственный шанс восстановления.

На рис. 5 показаны результаты работы сервиса. Протестированы страницы ДонНТУ в Твиттере и Одноклассниках. Цифрой 1 отмечена временная шкала, показывающая за какие даты представлена в архиве информация и копия страницы твиттера ДонНТУ за 26 декабря 2016г. Наш вуз вошел в архив сервиса с 2016г. Цифрой 2 обозначена страница, на которой предлагается выбрать архивы страницы вуза в

Контакте.

Наличие такого архива позволяет не только восстановить сайт, как было сказано выше, но и свидетельствует о рейтинге страниц. Цифра 4 рис. 5 – копии страниц Первого канала России. Было снято 4525 копий за период с 1 октября 2002г по ноябрь 2017г.

Цифра 5 – копии страниц Министерства образования и науки ДНР.

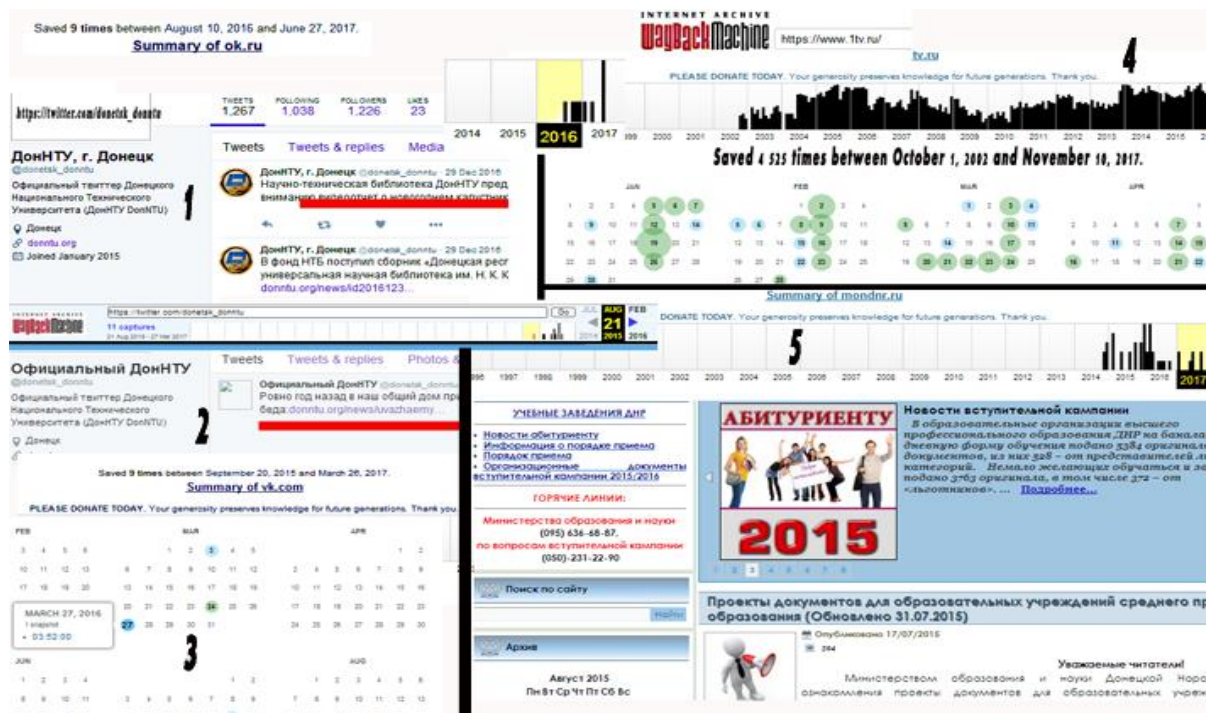


Рисунок 5 – Результаты работы сервиса Babkee

Наиболее оптимальным и доступным на наш взгляд, является сбор статистики в Контакте. Для примера приводим работу только одного сервиса Smmup.ru. Мы поставили целью провести сравнительный анализ страниц в Контакте некоторых вузов Донецка.

Наибольшее число подписчиков у

Донецкого национального медицинского университета (ДонНМУ). По активности лидирует ДонНУ (67 лайков, репостов и комментариев в день), меньше всего подписчиков на странице музыкальной академии, (см. рис.6).

Сообщество	"Живые" подписчики	Процент ботов	Л+Р+К (день)	Л+Р+К (пост)
ДонНМУ им. Горького	10450	7.5	21	8
Донецкий национальный университет. ДНР	9330	6.6	67	11
ГОУ ВПО «ДонаУиГС» (Бывший ДонГУУ)	3236	6	18	9
Донецкая музыкальная академия им.С.С.Прокофьева	543	4.9	7	4
Профком студентов ДонНУЭТ	8447	13.3	37	19
ДонНТУ, г. Донецк	2274	5.9	8	2

Рисунок 6 – Результаты работы сервиса Smmup

Выводы

Аналитика социальных сетей необходима, т.к. позволяет сравнить свои страницы с сайтами конкурентов, как это показано при описании сервиса Smmup, сделать выводы и исправить ошибки. Статистика позволяет оценить аудиторию паблика, оптимальное время подачи новостей, наилучшую форму контента. Несмотря на преобладание платных сервисов, бесплатно можно произвести достаточный по глубине анализ. И, наконец, статистика позволяет ответить на вопрос, нужно ли предприятию иметь несколько страниц в разных сетях или вполне достаточно одной.

Литература

1. Газета Коммерсантъ. Разведка ботом. СВР займется социальными сетями. Режим доступа: <https://www.kommersant.ru/doc/2009256>
2. ДонНТУ, официальная группа. Режим доступа: <https://vk.com/donetsk.donntu>.

3. Анализ сообществ. On-line сервис Smmup.ru. Режим доступа: <http://smmup.ru/activity.php>
4. ДонНТУ. Режим доступа: https://twitter.com/donetsk_donntu.
5. Actualtraffic. Обзор сервиса Livedune. Режим доступа: <http://actualtraffic.ru>
6. Acrit. Сервис Followerwonk. Режим доступа: https://www.acrit-studio.ru/it_is_important_to_know/twitter/followerwonkib1/
7. Эффективная работа с микроблогом Twitter: планирование твитов по времени. Режим доступа: <https://netpeak.net/ru/blog/effektivnaya-rabota-s-mikroblogom-twitter-planirovanie-tvitov-po-vremeni/>
8. Сервис babkee . Режим доступа: <http://www.babkee.ru>
9. Webarchive — веб-архив всего интернета и сайтов или машина времени на archive.org. KtoNaNovenkogo.ru. Все о создании сайтов, блогов, форумов, интернет-магазинов, их продвижении в поисковых системах и заработке на сайте. Режим доступа: <http://ktonanovenkogo.ru/web-obzory/webarchive-archive-org-veb-arxiv-interneta-sajtov.html>

Анохина И. Ю., Черепов В. Г. Сервисы мониторинга социальных сетей. Показана работа сервисов анализа социальных сетей. Рассматриваются характеристики оценки активности и степени вовлеченности аудитории. На примере страниц ДонНТУ в социальных сетях показано, как выбирается оптимальное время подачи новостей, как найти самых активных и авторитетных участников, как оценить уровень упоминаний в СМИ.

Ключевые слова: социальные сети, ВКонтакте, Твиттер, степень вовлеченности, активность аудитории, мониторинг сетей.

I. Yu. Anokhina, V. G. Cherepov Social network monitoring services. The work of social network analysis services is shown. The characteristics of the assessment of activity and the degree of involvement of the audience are considered. On the example of DonNTU pages in social networks it is shown how the optimal time of news delivery is chosen, how to find the most active and authoritative participants, how to assess the level of mentions in the media.

Keywords: social networks, VKontakte, Twitter, level of involvement, audience activity, network monitoring.

Статья поступила в редакцию 22.06.2018 г
Рекомендована к публикации доктором технических наук Павлышом В.Н.

Инженерное образование

Критерии проектирования моделей обучения основам алгоритмизации и программирования

М. А. Курилов

Донецкий национальный технический университет
кафедра прикладной математики
kurilov_ukr@mail.ru

Курилов М. А. Критерии проектирования моделей обучения основам алгоритмизации и программирования. В статье рассматриваются методологические аспекты создания систем обучения основам алгоритмизации и программирования, критерии проектирования таких систем и предпосылки решения актуальных вопросов обучения с учетом нынешних геополитических условий.

Ключевые слова: информатика, алгоритмизация, программирование, мотивация обучения, дистанционное обучение.

Постановка проблемы в общем виде и ее связь с важными научными или практическими заданиями

Настоящий момент в развитии Донецкой Народной Республики характерен чрезвычайной потребностью в высококвалифицированных специалистах практически во всех областях знаний. Особое место здесь занимают специалисты в области IT – Информационных Компьютерных Технологий.

Выбор критериев проектирования систем обучения с учетом современных геополитических условий, пожалуй, является актуальным в настоящее время. Основные пути качественной подготовки высококвалифицированных специалистов в этой области и объективные причины их выбора изложены в работе [1]. Автор предлагаемой работы ставит своей целью не только реанимацию, но и дальнейшее развитие основных идей основоположника мировой информатики В. М. Глушкова, автора ОГАС [2], системно – кибернетического подхода [3], отвергнутому более 40 лет назад [4-6].

Анализ последних исследований и публикаций, в которых начато решение данной проблемы и на которые опираются авторы

Системно – кибернетический подход к проблеме обучения, а также выбор критериев проектирования систем обучения предварим вначале описанием того, что из себя представляет такое понятие как система. Обратимся к [7], где даны варианты основных дефиниций рассматриваемого понятия:

- Система — сложное единство, сформулированное многими, как правило, различными факторами и имеющее (это

единство) общий план или служащее для достижения общей цели;

- Система — собрание или соединение объектов, объединенных регулярным взаимодействием или взаимозаменяемостью;

- Система — размещение множества или собрание вещей, связанных между собой таким образом, что вместе они образуют некоторое единство — целостность;

- Система — собрание сущностей или вещей, одушевленных или неодушевленных, которое воспринимает некоторые входы и действует согласно им для производства некоторых выходов, преследуя при этом цель максимизации определенных функций входов и выходов;

Академик В. М. Глушков осуществил дальнейшее “кибернетическое” развитие понятия системы, которое воплотилось в виде организационной системы управления (ОСУ) [3].

Этапы проектирования всякой организационной системы управления заключаются прежде всего в последовательном разукрупнении полученных описаний на отдельные (хотя обычно и взаимосвязанные) блоки. С каждым блоком (как и со всей системой в целом) связываются и описываются потоки входной (выходной) информации.

На определенных стадиях разукрупнения уточняется, какие блоки алгоритма управления будут выполняться людьми, а какие — ЭВМ. Соответствующим образом последовательно уточняется и структура базы данных. На каждом очередном этапе процесса разукрупнения блоков в процесс проектирования вовлекаются, вообще говоря, все новые и новые исполнители.

Конечная цель этого процесса - довести описание блоков, исполняемых ЭВМ, до машинных программ, а блоков, исполняемых людьми, до должностных инструкций, написанных столь подробно, чтобы люди с

соответствующим (указанным в проекте) образовательным цензом, но, быть может, не имеющие специального опыта работы, могли самостоятельно разобраться в инструкциях, однозначно интерпретировать их и фактически исполнять описываемые в них управленческие действия (сначала, возможно, в замедленном ритме).

В рамках практической реализации функционирования ОСУ В.М. Глушков рассматривал в качестве одной из ее составляющих – автоматизированную систему контроля исполнительской дисциплины.

Нерешенные ранее части общей проблемы, которым посвящается обозначенная статья

Во времена постсоветского периода доминирующую роль играл следующий принцип: *Бытие определяет сознание*. Нынешний период развития человечества, особенно в рамках глобальных информационных войн [1], характерен тем, что приведенный принцип сменился на свою противоположность: Именно сознание, читай разум, определяет благосостояние отдельного индивидуума, общественной группы и всего человечества.

На этом неоднократно акцентировал внимание В. М. Глушков [1-6] более 40 лет назад, хотя в те времена высказывания такого рода шли в разрез “политике партии и правительства”.

Научно – технический прогресс и особенно ускоренные темпы развития информационных технологий выдвинули на повестку дня подготовку высококвалифицированных специалистов с использованием современных методов обучения. Эти методы непосредственно связаны с основными задачами и проблемами, стоящими перед таким направлением науки, как искусственный интеллект.

“Мы знаем что-то, если можем это запрограммировать” – это высказывание А. П. Ершова, впервые официально появилось в кибернетическом научном издании, руководимым В. М. Глушковым [8] и, естественно, разделяемое последним. Всякое обучение должно проводиться с позиции интеллектуальной (профессиональной) силы того, кто обучает. Это полностью согласуется с принципом первого лица и инстинктом самосохранения, приоритетную роль которых В. М. Глушков продвигал всю свою сознательную жизнь [1].

С учетом этого, вопрос подготовки специалистов в области информационных технологий и особенно вопрос выбора тех лиц, которые отвечают за такую подготовку, приобретает приоритетное значение.

Формулировка целей статьи

Всякая статья, претендующая не только на статус научности, но и на статус практической ценности, должна предложить пути решения актуальных на текущий момент задач. Авторы данной статьи ставили перед собой целью предложить такие способы обучения специалистов, которые бы могли в кратчайшие сроки и наиболее эффективно решать поставленные задачи. При этом будем четко придерживаться той позиции, что все то, что обсуждается в нашей работе, имеет самое непосредственное отношение к искусственному интеллекту (ИИ). Многозначная интерпретация этого термина до сих пор вызывает различные споры между специалистами. Вспомним высказывание одного из основоположников искусственного интеллекта Г. С. Поспелова [10]. “...Этот термин гипнотизирует многих, особенно людей, далеких от кибернетики, заставляет их настроиться на определенную волну небывалых доселе возможностей и достижений, которые станут посильны столь могучему помощнику человека. Поэтому следует сразу очертить границы распространения термина, выделить главное: искусственный интеллект не есть синоним искусственного разума, а исследовательское направление в науке, к которому относятся проблемы и задачи не вычислительного характера, требующие переработки смысловой информации...”

В рамках подтверждения важности основных идей В.М. Глушкова в области информатизации общества, высказанных им более сорока лет назад, вспомним аналогичное высказывание основоположника информатики одного из крупнейших государств западной Европы [9]. “...Я полагаю, что в будущем кибернетика станет универсальной наукой. Кибернетик станет посредником между специалистами различных областей. Тот, кто хочет занять ведущую роль в государстве или экономике, должен изучать кибернетику. Еще в 1834 г. Ампер видел в кибернетике науку об управлении государством. Уже сейчас мне представляется сомнительным, чтобы человек, получивший только юридическое образование, мог быть способен оптимально управлять обществом с высоким уровнем развития техники.

Организация современного индустриального общества требует гораздо больше того, что дает юридическая подготовка. Кибернетические методы станут важнейшим инструментом не только в физиологии, но и в психологии. Я еще раз напомним о возможности такого моделирования поведения человека на электронных вычислительных машинах, при котором становится возможным предсказание его поведения в тех или иных условиях...” Это высказывание увидело свет в начале эпохи

первых компьютеров и для автора статьи эта цитата носит аксиоматический характер. Здесь на лицо полное подтверждение необходимости выбора системного подхода, о котором говорил В. М. Глушков. И, поэтому, вызывает удивление политика некоторых чиновников в системе образования, которые значительно сокращают, а порой и просто исключают учебные часы, отведенные для изучения основ компьютерных технологий, мотивируя это недостаточным финансированием изучения соответствующей отрасли знаний.

Следует отметить еще один фактор, о котором говорил академик [1-6]. Ускоренные темпы развития IT непременно приведут к тому, что компьютер будет самым мощным инструментом духовного, идейного и политического воздействия на общество. Речь идет о софизме, под которым понимают умышленный обман, основанный на нарушении правил языка или логики. Но обман тонкий и завуалированный, так что его не сразу и не каждому удастся раскрыть. Цель его - выдать ложь за истину. Достоверным подтверждением этого является соглашение между правительством Украины и правительством Соединенных Штатов Америки о гуманитарном и технико-экономическом сотрудничестве от 07.05.1992 г. Всемирная паутина более тридцати лет хранит и другие данные, которые подтверждают априорность аксиом программирования [1-6] и исключительную необходимость реанимации научного наследия В.М. Глушкова: <https://youtu.be/9TfpawfbVdca>, http://zakon2.rada.gov.ua/laws/show/840_295, <https://youtu.be/TqzkVDad9LM>

Изложение основного материала исследования с полным обоснованием полученных научных результатов.

Перейдем к основным вопросам статьи – критерии проектирования систем обучения, подготовка специалистов IT через современные методы обучения и формы такого обучения.

Начнем с критериев проектирования и выделим среди них следующие:

- Массовость;
- Доступность;
- Открытость;
- Современность;
- Побудительные мотивы;
- Развитие собственного “Я”;
- Профессионализм;
- Преемственность.

Рассмотрим подробнее предлагаемые критерии.

Массовость. Всякое обучение, особенно на уровне его самостоятельного проведения, предполагает достаточно большой объем знаний. Естественно, вся эта составляющая должна быть

представлена в электронном виде. Представлена она должна быть как авторскими разработками, так и уже существующими. Применительно к условиям высшей школы это может быть архив ранее выполненных и естественно тщательно проверенных лабораторных, курсовых и дипломных работ обучаемых.

Доступность. Доступ к информации может быть, например, обеспечен на индивидуальных интернет-ресурсах преподавателей или ресурсах образовательного учреждения.

Открытость. Предполагается непрерывное сопровождение соответствующих интернет ресурсов и постоянное их пополнение и обновление.

Современность. Содержательная часть должна отвечать текущему состоянию знаний, а также включать в себя компоненты воспитательного и патриотического аспектов с учетом своих исторических фактов. Для автора проведение “дебилизации” подрастающего поколения нашей страны с целью изменения существующего статуса очевидно.

Побудительные мотивы. У всех обучаемых необходимо развивать настоящее желание глубоко познать сущность рассматриваемого вопроса. “... Любите книгу – источник знаний...” – это высказывание известного пролетарского писателя еще никто не отменял.

Развитие собственного “Я”. Каждый обучаемый должен быть мотивированным принципом не быть хуже и глупее других. Вспомним высказывание экс госсекретаря США Дж.Керри о том, что американцы могут быть глупыми и это право дано им конституцией. Интерпретация такого высказывания для автора статьи очевидна.

Профессионализм. Семантическое наполнение данного критерия предполагает высокий профессионализм самого преподавателя и подготавливаемой им знаниевой компоненты.

Преемственность. Предполагается фактор непрерывности самого процесса обучения и органическая связь ранее полученных знаний с вновь получаемыми.

Пришло время, чтобы начать уважать свое прошлое. Использование системного подхода к проблеме обучения особенно актуально в эпоху высокоразвитой информатизации общества. Если общеобразовательный уровень такого общества и каждого отдельного его члена не соответствует уровню развития информационных технологий, то такое общество (читай “система”) обречено на исчезновение. Модель обучения всегда должна соответствовать тем социальным условиям, в которых эта модель функционирует. Очевидно, что приходится сталкиваться с рядом трудностей при создании таких моделей.

Большинство этих трудностей предвидел В. М. Глушков. Он, например, был одним из тех, кто предлагал проводить всякие занятия в форме семинаров и чтобы все присутствующие принимали в них самое активное участие. При этом иногда лектор (преподаватель) задает аудитории вопросы и он всегда разрешает самой аудитории задавать вопросы ему, если что-нибудь окажется неясным. Если лектор разрешает прерывать себя вопросами, то это значит, что он вполне владеет предметом и не боится услышать вопрос, на который не сможет дать ответ.

Вряд ли нужно говорить о том, что для лектора такого рода боязнь по меньшей мере неосновательна: никто не требует от преподавателя «всезнания». Преподавателя, действительно знающего свой предмет, невозможно поставить в тупик. Вполне очевидно, что такой метод ведения занятий куда более продуктивен, чем торжественное вещание с кафедры перед безмолвно внимающей аудиторией.

Настоящий момент характерен высоким уровнем развития информационных технологий. Ряд ведущих государств мира уже вступил в информационную эпоху. Экстраполируя идеи В. М. Глушкова относительно моделей обучения при помощи компьютера, отметим концепцию авторских электронных учебных модулей. Здесь главное - модульный принцип формирования собственного банка цифровых образовательных ресурсов. Один из основных элементов такого банка по курсу «Математика» для средних классов общеобразовательных учебных заведений подробно описан в [12]. Традиционно каждый из учебных модулей состоит из трех составляющих:

- определённого фрагмента (куска) учебной информации по данной теме;
- практической части (примеры и задачи);
- тестирующей части (контроль): проверка факта усвоения учебной информации.

Модуль большой темы состоит из ряда подчинённых модулей. В заключение изложения стратегического направления в обучении остановимся на описании одного из авторских вариантов модуля по дисциплине Информатика для школьников младших классов.

Представим саму задачу следующим образом. Необходимо описать алгоритм и составить программу для вывода двух или более графических изображений на экран дисплея. Автоматически обеспечить независимое горизонтальное их перемещение в пределах экрана или реализовать их перемещение при помощи клавиш управления курсором.

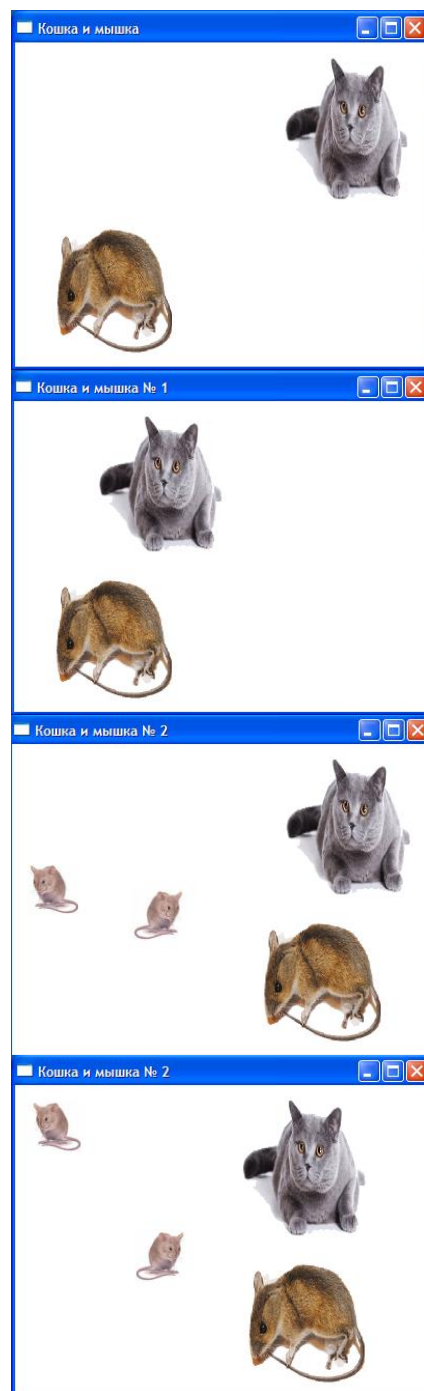


Рисунок 1 – Графические изображения

В качестве языка описания, при помощи которого будем составлять программы, реализующие поставленные ранее задачи, будем использовать уникальный язык, разработанный непосредственно автором статьи.

Здесь мы рассмотрим лишь те команды, которые используются для машинной реализации первой задачи (окно с именем «Кошка и мышка»).

Описание оформим в виде таблицы 1, где во втором ее столбце в скобках указано число самих параметров.

Таблица 1. Описание команд

Код команды	Параметры программы	Назначение команды
ГРАФИКА	(4)ДЛИНА, ШИРИНА, ЦВЕТА, ВИД ОКНА	Настраивает графический вывод на экране монитора. Задаёт в пикселях ширину(1), высоту(2) окна, режим цветности(3), вид окна (масштабируемое, развернутое и свернуто в иконку).
Н= ЗАГРУЗИТЬ_ ИЗОБРАЖЕНИЕ	(1) ИМЯ ФАЙЛА	Н – код картинки, который присваивается автоматически. Указывается имя графического файла изображения.
ОТРИСОВАТЬ_ ИЗОБРАЖЕНИЕ Н	(3) код картинки, X,Y	Н - Код картинки, полученный при ее загрузке из файла. Координаты левого верхнего угла изображения.
НАЧАЛО ... КОНЕЦ	Без параметров	Командный блок, в пределах которого помещаются команды управления изображениям и.

Практика общения автора статьи со школьниками начальных классов показала, что уже через несколько часов обучаемые вполне самостоятельно могут реализовывать простые программные модели игрового характера.

Такой подход в обучении обеспечивает достаточно быстрый переход школьников к самостоятельной постановке (вот она мотивация) более сложных задач на алгоритмизацию таких

сущностей, как сложные траектории движения, обработка их столкновений, преодоление препятствий и т.д. Проблемная направленность такой модели здесь для автора очевидна – Искусственный Интеллект.

В качестве заключительной, и пожалуй, основной части наших авторских изысканий, отметим следующее. В Российской Федерации в рамках своего ускоренного информационного развития создан ряд образовательных ресурсов.

Одним из них является Федеральный Центр информационно-образовательных ресурсов (<http://fcior.edu.ru/>), авторы которого подчеркивают «неограниченный жизненный цикл системы: поскольку каждый учебный модуль автономен, а система открыта. Открытая Модульная Система является динамически расширяемым образовательным ресурсом, не требующим сколько-нибудь существенной переработки в целом при изменении содержательных или технических внешних условий».

Об этом В. М. Глушков говорил более 40 лет назад и для всех здравомыслящих людей реанимация и практическое использование идей основоположника информатики гарантирует создание полноценного индивидуума, который успешно будет существовать в условиях всеобщей глобализации.

Выводы по исследованию и перспективы дальнейших поисков в данном направлении

В статье предложены критерии систем обучения, рассмотрены общие принципы системно – кибернетического подхода при подготовке специалистов широкого круга профессиональной ориентации основам информационных технологий (алгоритмизация, программирование, базы данных и т.д.).

Предложены адаптированные методы реализации обучения, широко используемые в странах Запада.

Реализация таких дидактических приемов требует серьезных изменений в среде нормативно – правовых отношений в сфере образования, так как революционные новшества в области образования (Болонская конвенция, кредитная система и т.д.), предложенные Западом, полностью подтвердили свою несостоятельность.

Принцип мотивации в обучении, повышение собственного “Я”, полностью лежащие в сфере проблем искусственного интеллекта – вот что может и должно вывести задачу подготовки высококвалифицированных специалистов на принципиально новый уровень.

Литература

1. Информатика, Алгоритмизация и программирование: Специальные формы обучения по методу академика В.М.Глушкова [Текст] / Меживой В.П., Курилов М.А. // Право Донецкой Народной Республики. – 2016. – № 1. – С. 92-100.
2. Глушков В.М. Что такое ОГАС? / Глушков В.М., Валах В.Я. – М.: Наука, 1981. – 160 с.
3. Глушков В.М. Основы безбумажной информатики / В.М. Глушков. – М.: Наука, 1987. (гл. ред. физ.-мат. лит.). – 552 с.
4. Малиновский Б.Н. Академик В.М. Глушков / Б.Н. Малиновский. – К.: Наукова думка, 1993. – 140 с.
5. Курилов М.А. Еще раз об аксиомах программирования и об обучении ему / Курилов М.А., Иванова С.Б., Пигуз В.Н. // Искусственный интеллект. – 2014. – № 3. – С. 4-11.
6. Аксиомы программирования и некоторые вопросы Дистанционного Обучения / А.И. Шевченко, М.А. Курилов, Л.П. Сыпченко, А.С. Барашко // Искусственный интеллект. – 2012. – № 1. – С. 244-251.
7. Уемов А.И. Системный подход и общая теория систем. М., / А.И.Уемов, “Мысль”, 1978 г., – 272 с.
8. Ершов А.П. О человеческом и эстетическом факторах в программировании / А.П. Ершов // Кибернетика. – 1972. – № 5. – С. 95-99.
9. Штейнбух К. Автомат и человек / К.Штейнбух, “СОВЕТСКОЕ РАДИО”, М., 1967. – 405с.
10. Интервью Г.С.Поспелова / Техника – молодежи. – 1981. – №7. – С.215 – 216.
11. Вопросы реализации дистанционного обучения в высших учебных заведениях юридического профиля [Текст] / Меживой В.П., Курилов М.А. // Право Донецкой Народной Республики. – 2016. – № 2. – С. 78-84.
12. Курилов М.А. Об одной модели дистанционного обучения программированию. [Текст] / М.А. Курилов // Научные труды Донецкого национального технического университета. Серия «Проблемы моделирования и автоматизации проектирования динамических систем». – Донецк, 2011. – Вып. 10(197). – С.51-57.

Курилов М. А. Критерии проектирования моделей обучения основам алгоритмизации и программирования. В статье рассматриваются методологические аспекты создания систем обучения основам алгоритмизации и программирования, критерии проектирования таких систем и предпосылки решения актуальных вопросов обучения с учетом нынешних геополитических условий.

Ключевые слова: информатика, алгоритмизация, программирование, мотивация обучения, дистанционное обучение.

Kurilov M. A. The Criteria of the designing the models of the education base algorithmization and programming. In article are considered methodological aspects of the making the systems of the education base algorithmization and programming, criteria of the designing of such systems and premiseses of the decision of the actual questions of the education with provision for present geopolitic conditions.

Keywords: informatics, алгоритмизация, programming, motivation of the education, remote education.

Статья поступила в редакцию 19.06.2018 г
Рекомендована к публикации доцентом Федяевым О.И.

УДК 371.315.7.014.3

Электронная обучающая система повышения квалификации педагогов: моделирование и внедрение

Т. Б. Волобуева

Донецкий институт последипломного педагогического образования

Волобуева Т. Б. Электронная обучающая система повышения квалификации педагогов: моделирование и внедрение. В статье рассматривается моделирование адаптивной электронной обучающей системы для повышения квалификации педагогических кадров/ Автор приводит требования к цифровой среде, модель управления качеством электронной обучающей системы, коэффициенты ее эффективности. Реализация данной модели позволяет обеспечить: четкую структуру системы и регламент функционирования; возможность персонификации обучения педагогов, удовлетворение индивидуальных образовательных запросов; многофункциональность механизмов обучения.

Ключевые слова: электронная обучающая система, педагог, повышение квалификации, качество, курсы.

Введение

Показателем современности образования сегодня является использование электронного обучения. Поэтому все более востребованными становятся электронные обучающие системы (ЭОС), платформы дистанционного обучения, другие программно-инструментальные среды. ЭОС предлагает обучающемуся учебный контент, контрольно-оценочный блок, справочную информацию. Однако, системы обучения нового поколения имеют более расширенные опции. Главное их нововведение – адаптивность. Они позволяют проектировать индивидуальные программы обучения, персонально отбирать учебные материалы.

Обучающая система

Для моделирования адаптивной электронной обучающей системы в процессе повышения квалификации педагогических кадров предлагаем пять ключевых шагов.

Шаг первый. Выбор технологии электронного обучения. Высокая технология отличается комфортностью пользования, легкостью освоения, большими возможностями коммуникаций. Программное обеспечение, требующее специальной подготовки пользователей, снижает мотивацию обучения и познавательную активность обучающихся.

Во всем многообразии средств организации электронного обучения [1] можно выделить *следующие группы:*

авторские программы (Authoring Packages);

системы менеджмента содержания учебных курсов (контентов) (Content Management Systems – CMS);

системы менеджмента учебного контента (Learning Content Management Systems – LCMS);

системы управления обучением (Learning Management Systems – LMS),

Авторские программы. Этот программный продукт, как правило, носит локальный характер. Разрабатывается для изучения определенных дисциплин или разделов предмета с использованием удобных для него технологий, например, PowerPoint, Lectura, HTML. Такие программные продукты имеют ряд недостатков: нет возможности контролировать процесс обучения и промежуточные результаты; трудоемкость коммуникаций большого количества обучающихся. Авторские программы обычно создаются для разработки занятий с мгновенной обратной связью. Они эффективны для самостоятельной работы, для усиления наглядности, мотивации обучения на аудиторных занятиях. Их мультимедийность повышает качество восприятия учебного материала. Однако низкий интерактивный потенциал делает их малоценными для ЭОС. Разработка подобных курсов в основном ведется преподавателями-энтузиастами и носит не систематический характер.

Системы менеджмента содержания учебных курсов содержат опции, позволяющие не только размещать учебные материалы в разных форматах, но варьировать его согласно целям или запросам обучающихся. Такие системы включают базы данных с возможностью сортировки, быстрого поиска, фильтрации. Они удобны, когда академический курс создает группа преподавателей, которым нужно вариативно использовать один и тот же учебный блок.

Системы менеджмента учебного контента соединяют возможности управления обучением и содержанием учебного материала. Организационно привлекательны для больших образовательных учреждений или их крупных

структур, так как реализуют менеджмент больших потоков обучающихся, позволяют быстро создавать курсы, предоставляют дополнительные модули.

Системы менеджмента обучения эффективны для обучения большого количества пользователей. Их разрабатывают как для учебных заведений, так и для корпоративного обучения. Они предоставляют возможность отслеживать процесс обучения, накапливать данные обучающихся, включая число их заходов на определенные вкладыши учебных материалов, затраченное время прохождения учебного блока и т.п. Регистрация обучающихся фиксирует целевую аудиторию, позволяет создавать учебные группы или динамические команды, использовать автоматическую рассылку, как учебных материалов, так и информацию о текущих событиях, отчеты, рекомендации. Кроме того, такие системы позволяют организовать проверку знаний и он-лайн общение.

Для моделирования адаптивной электронной обучающей системы в процессе повышения квалификации педагогических кадров нами выбраны системы менеджмента обучением (LMS).

Шаг второй. Выбор электронной обучающей среды. Сегодня на рынке программного обеспечения позиционируется два вида сред электронного обучения: коммерческие и бесплатные (свободного доступа).

Коммерческие системы разрабатываются под дистанционное образование или электронное обучение в пределах учебного заведения. Определенную популярность завоевали у нас среды “Amigo.CMS”, “inDynamic 2.3”, “Прометей”, “Битрикс”, “NetCat и др. Их недостатками являются: большая стоимость, жесткие аппаратные требования, ограниченное число образовательных лицензий.

К **бесплатным средам** относятся свободно распространяемые LMS/LCMS: Moodle, ATutor, Dokeos, Claroline, OLAT, LAMS, , Sakai, OpenACS и др.

Используя классификацию требований по К. Вигерсу (рис. 1), мы выделяем три уровня требований к программной среде для электронного обучения педагогов на курсах повышения квалификации: педагогические, пользовательские и функциональные [2].



Рисунок 1 – Классификация требований по К. Вигерсу

Педагогические требования очерчивают глобальные цели и миссию образовательной организации. Пользовательские требования описывают возможности, предоставляемые системой, для решения задач пользователей.

Функциональные требования отображают пакет функций программного обеспечения, заложенных в техническое задание разработчиком для реализации своих задач

пользователями.

Нефункциональные требования описывают эксплуатационные характеристики и признаки качества (например, целостность, устойчивость, эффективность, практичность). Здесь же перечисляются ограничения, т.е. указывается лимит выбора возможных решений.

Нормативные правила – включают законодательные акты, образовательные

стандарты, корпоративную политику, учетные таблицы и алгоритмы вычислений.

Мы выделили следующие основные критерии выбора электронной обучающей среды для повышения квалификации педагогических кадров (рис. 2):

- *Функциональность.* В системе должны быть заложены интерактивные функции (форумы, чаты), менеджерские опции (анализ активности педагогов, управление ресурсами и т.п.)

- *Надежность.* В среде должны присутствовать подсистемы защиты информации, шаблоны формирования контента, удобные в пользовании управляющие модули. Важным показателем привлекательности системы является простота администрирования.

- *Стабильность.* Среда должна обеспечивать надежность работы в разных режимах, при большом количестве одновременных вхождений. Она должна быть устойчивой к сбоям.

- *Привлекательность цены.* Стоимость системы суммируется из затрат: на приобретение (аренду), внедрение, создание курсов, сопровождение обучения. Она должна быть приемлемой для института.

- *Лимит лицензий.* Для построения электронной обучающей системы повышения квалификации педагогических кадров ценно отсутствие ограничений на количество лицензий для обучающихся. За год в Донецком ИППО проходят курсы повышения квалификации около девяти тысяч педагогов, еще столько же принимают участие в постоянно действующих межкурсовых мероприятиях, поэтому неограниченное право доступа к электронному обучению большого числа пользователей является весомым требованием к электронной среде.

- *Инструментальность.* Для работы в системе необходимы средства разработки учебного контента. Встроенные текстовые, табличные и другие редакторы, шаблоны,

библиотеки и приложения не только облегчают создание обучающих и проверочных материалов, но и обеспечивают единое представление различной информации.

- *Совместимость.* Система должна поддерживаться SCORM – международными стандартами дистанционного обучения. Это важно для интеграции, переносимости учебных материалов, мобильности обучения.

- *Контрольно-оценочная система.* Для проверки знаний обучающихся нужны модули, обеспечивающие входное, выходное тестирование, промежуточный контроль в виде контрольных работ, оценивания активности в дискуссиях, форумах, выполнения практических заданий и т.д.

- *Комфортность эксплуатации.* Работа в системе не должна требовать специальной подготовки педагогов. Трудности при использовании среды резко снижают мотивацию к обучению. Комфортность пользователя должна обеспечивать удобный интерфейс, быстрый поиск, интуитивно понятные опции, доступность коммуникаций, легкость навигации.

- *Модульность.* Модульная структура контента позволяет манипулировать учебными блоками для решения различных педагогических задач. Такая организация учебных материалов обеспечивает большую вариативность их использования, интеграцию и дифференциацию информации.

- *Доступность.* Привлекательность электронного обучения состоит еще и в индивидуальном режиме работы:

- учиться педагог может в любое удобное время и в любом месте (обязателен беспрепятственный доступ в систему);

- проблемы здоровья не будут мешать обучению, так как предполагаются настройки среды под особые потребности (слабое зрение, другие ограниченные возможности);

- постоянная связь с тьютором.



Рисунок 2 – Критерии выбора электронной обучающей среды

Мы выбрали самую популярную систему Moodle. С ее помощью можно организовать on-line курсы повышения квалификации, целевые сайты, Web-факультативы, виртуальные педагогические мастерские и другие обучающие мероприятия. Мы используем преимущества социально-педагогического конструирования, заложенного в среде:

- обучение проводится с актуализацией жизненного опыта педагогов и имеющихся знаний;
- взаимообучение – обязательная составляющая повышения квалификации.

Электронное обучение строится на принципах андрагогики с использованием интерактивных методов, ситуативного моделирования и педагогического творчества. Наши преподаватели освоили специфику ролей тьютора, модератора и фасилитатора, используя

их в соответствии с целями и форматами курсов повышения квалификации. Moodle также дает нам возможность применять гейм-технологии, гибридное обучение, большие данные, адаптивное обучение.

Еще одно важное преимущество этой системы – мультязычность. Бесплатная поддержка работы в многоязыковой среде позволяет создавать модули для учителей иностранных языков, использовать зарубежные ресурсы и развивать поликультурную компетентность педагогов.

Шаг третий. Проектирование архитектуры электронной обучающей системы.

На основе всего вышесказанного в данной работе предлагается следующая модель (рис. 3) построения электронных курсов повышения квалификации:



Рисунок 3 – Архитектура электронной обучающей системы

Шаг четвертый. Формирование баз данных. Основные требования к организации баз данных (БД): установление многосторонних связей; производительность; минимальные затраты; возможность поиска; целостность (восстановление данных); безопасность.

Шаг пятый. Формирование электронных курсов повышения квалификации. Мы используем единый структурный шаблон для составления электронных курсов повышения квалификации: входной и выходной контроль, учебные модули, контрольные работы, дискуссии,

дополнительные материалы, справочные материалы, методические рекомендации, практические работы, курсовая работа, консультации. Входной и выходной контроль представляют собой on-line тесты по всему курсу. Первый тест позволяет очертить проблемное поле и акцентны обучения, последний – оценить когнитивный результат курсов. Учебные модули предоставляют возможность выстроить индивидуальную траекторию обучения, создавать творческие мини группы. Каждый модуль, как правило, содержит одну проблемную лекцию, практикум

и вебинар. Особой популярностью у педагогов пользуются видео мастер-классов, web-тренингов, on-line супервизий. Требования к учебным материалам:

- четкая структурированность текста;
- разбиение на блоки;
- подача текстового материала в таблицах, схемах, картах;
- гиперссылки, подстрочные подсказки;
- мультимедийность – наличие презентаций, видеоматериалов, анимации.
- Содержание учебных материалов тоже имеет свою специфику:
 - в них отражены самые последние педагогические новации и достижения педагогической науки и практики;
 - они носят опережающий характер и направлены на обеспечение перспективных видов учебной деятельности;
 - имеют практико-ориентированную направленность: разрабатываются под контингент.

Контрольные работы содержат три вида заданий: репродуктивный, частично-поисковый, творческий и имеют разный максимальный оценочный балл.

Инструктивно-методические материалы содержат конкретные рекомендации, четкие алгоритмы работы и советы по повышению эффективности обучения.

Справочные материалы не перегружены

лишними словарями. В них находятся тезаурусы по теме, методические карты, компиляция нормативных документов и т.п.

Дополнительные материалы предоставляются текстовыми и видеофайлами, аудиокнигами и Е-хрестоматиями, профильными on-line журналами и ссылками на сайты по теме.

Тьюторинг включает постоянное консультирование педагогов, форумы, чаты и предметные конференции, мотивационную рассылку, организацию дискуссий и презентаций творческих работ.

Курсовая работа представляет собой творческий продукт педагога, например, авторский проект, пакет мультимедийных дидактических материалов, сборник ситуативных задач по конкретным темам предмета, набор эвристических заданий, мини видеотека предметного содержания, анимация опытов по физике или химии и т.п.

При этом в работе должны быть: отражены новые навыки и компетенции, полученные на курсах; обоснована целесообразность ее применения в педагогическом процессе.

Электронные курсы направлены на развитие профессиональной компетентности педагогов. Результат обучения в электронной системе повышения квалификации можно представить пятиуровневой спиралью (рис. 4).



Рисунок 4 – Результаты обучения на курсах повышения квалификации

Первый уровень – базовые компетентности, которые опираются на деятельностные процессы и оказываются в контексте структуры педагогической деятельности;

Второй уровень – общепрофессиональные компетентности, которые относятся к определенной категории специалистов, объединенных одним предметом

деятельности (например, категории учителей, управленцев, методистов и др.);

Третий уровень – специально-профессиональные компетентности – это частные компетентности, которые приобретаются во время работы по определенным специализациям: дидактика, воспитание, психология, логопедия, методика, язык, математика, физика, химия, история и т.п.

Четвертый уровень – моделирование, педагогическое творчество.

Пятый уровень – личностные изменения, готовность к саморазвитию не только профессиональному, но и личностному.

Качество обучения отслеживается и через анализ педагогической практики в послекурсовой период, полноту использования приобретенных на курсах знаний и навыков, уровень их адаптации к конкретным условиям учебного заведения.

Шаг шестой. Оценивание качества электронной обучающей системы Требования

к электронным обучающим системам каждый год повышаются и дополняются. Поэтому важной составляющей моделирования ЭОС является оценивание их эффективности и управление их качеством. Под качеством ЭОС мы понимаем пакет индикаторов успешного усвоения материала педагогами. Воздействовать на качество можно через коррекции контента и процессов, супервизии тьюторов. Процесс управления качеством электронных обучающих систем показан на рисунке 5.

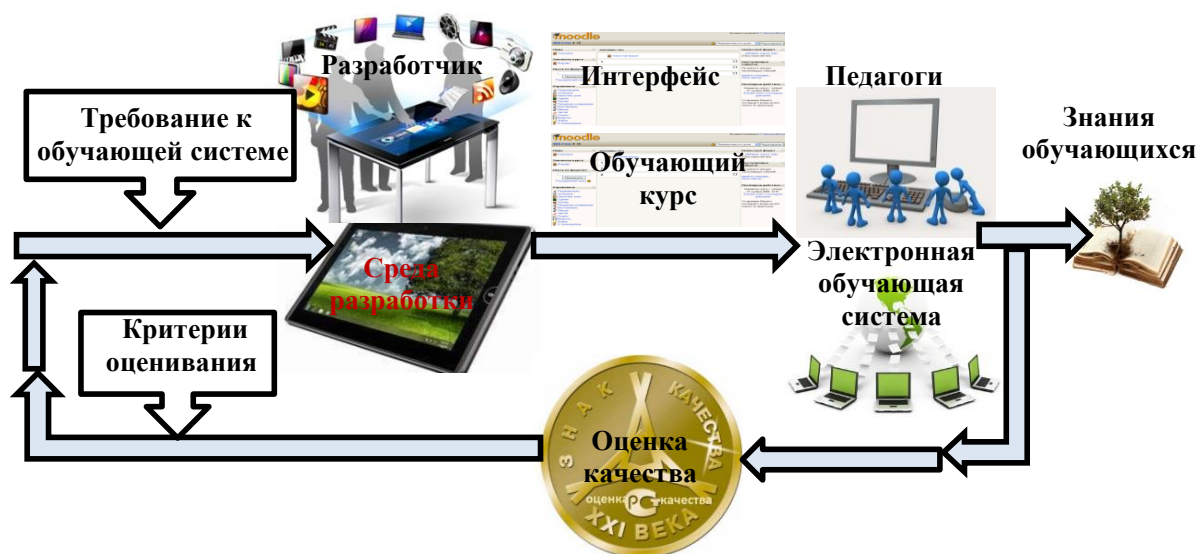


Рисунок 5 – Процесс управления качеством электронной обучающей системы

В настоящее время для оценивания качества ЭОС используют два подхода: целостный и многослойный. Оценивание «в целом» использует: несколько общих критериев (например, безопасность, продуктивность); проверку соответствия стандартам; отзывы пользователей. Как показывает практика, такой подход недостаточен для проведения серьезных коррекций и модернизации системы.

Для адаптивных электронных обучающих систем чаще используют второй подход. В них на конечный результат влияют непосредственно используемые механизмы адаптации. Сама ЭОС состоит из различных подсистем, которые

взаимодействуют друг с другом. Для оценивания качества задействуют различные наборы критериев для каждой подсистемы. Таким образом, диагностируется весь процесс обучения, более полно выявляются недостатки и риски, разрабатывается целевая коррекция.

На рисунке 6. проанализированы примеры автоматизированных систем оценивания качества электронных обучающих систем. Их общий недостаток – узкая специализация, привязка к определенной обучающей системе.



Рисунок 6 – Анализ автоматизированных систем оценки качества процесса обучения и обучающих материалов

Мы взяли главным критерием качества электронной обучающей системы повышения квалификации педагогов – эффективность обучения.

Чтобы оценить эффективность нашей модели мы рассмотрели четыре уровня оценки обучения, предложенных Дональдом Киркпатриком (рис. 7). [5].

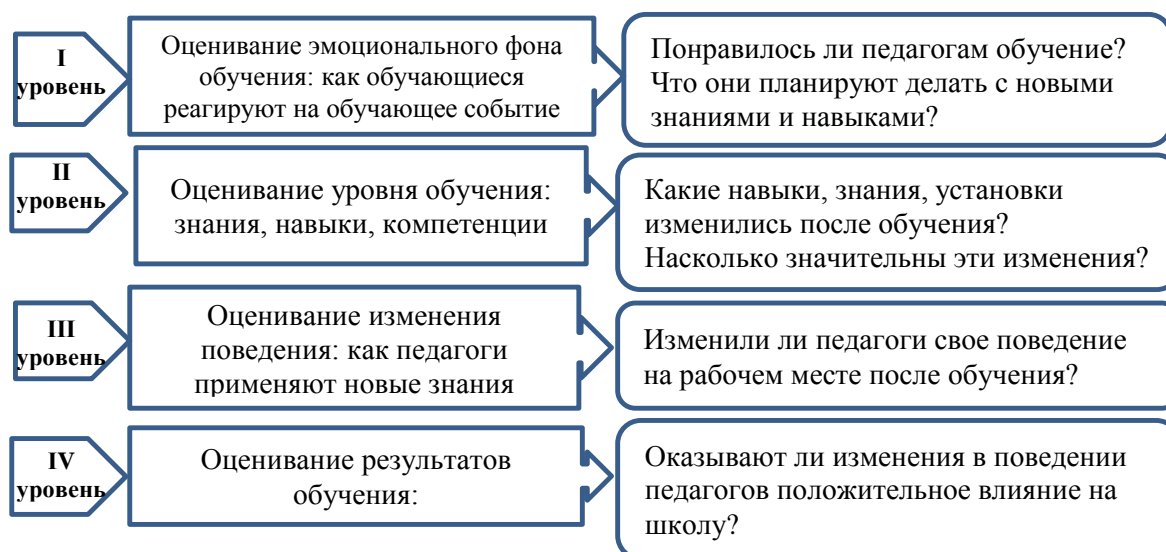


Рисунок 7 – Уровни оценки Дональда Киркпатрика

Не смотря на простоту такого подхода, учтены не все факторы, влияющие на качество обучения. Поэтому мы адаптировали набор

коэффициентов, предложенных исследователями [3] к педагогической системе повышения квалификации (рис. 8).

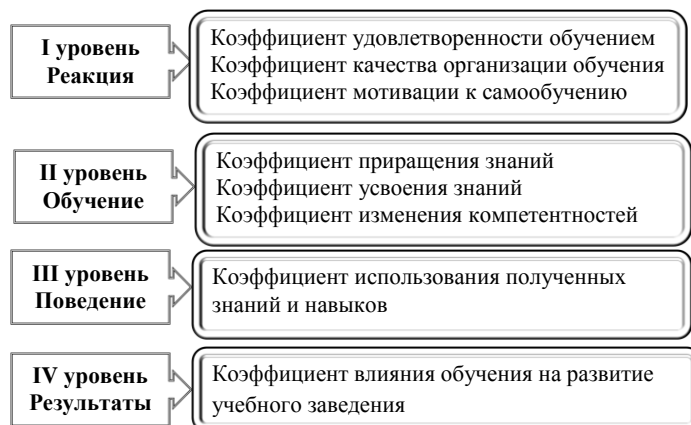


Рисунок 8 – Показатели оценки эффективности ЭОС

Статистические расчеты позволяют точнее и глубже провести диагностические процедуры, получить более полную картину качества, определить преимущества и недостатки отдельных составляющих системы и оптимальнее провести коррекцию.

Оценивание качества электронной обучающей системы является важным элементом повышения квалификации педагогов. Новые компетенции педагогов, полученные при работе в ЭОС, применяются в педагогической практике. Это влияет на качество обучения школьников, результативность работы образовательных организаций, а значит и региональной системы образования в целом.

Выводы

Предложенная адаптивная электронная обучающая система повышения квалификации педагогических кадров моделируется по следующему алгоритму:

- 1) выбор технологии электронного обучения;
- 2) выбор электронной обучающей среды;
- 3) проектирование архитектуры;
- 4) формирование баз данных;
- 5) формирование электронных курсов повышения квалификации;
- 6) оценивание качества электронной обучающей системы.

Реализация данной модели позволяет обеспечить:

- четкую структуру системы и регламент функционирования;
- возможность персонификации обучения педагогов, удовлетворение индивидуальных образовательных запросов;
- многофункциональность механизмов обучения;

- поэтапный самоконтроль обучения педагога и контроль со стороны руководителя курса;
- повышение мотивации непрерывного обучения;
- постоянное обновление содержания, инновационность;
- повышение информационной культуры педагогов;
- пространство для профессионального самосовершенствования.

Литература

1. Богомолов В.А. Обзор бесплатных систем управления обучением. [Электронный ресурс] – <http://cyberleninka.ru/article/n/obzor-besplatnyh-sistem-upravleniya-obucheniem>
2. Вигерс Карл. Разработка требований к программному обеспечению. Пер, с англ. – М.: Издательско-торговый дом "Русская Редакция", 2004. – 576с.
3. Гусева А.И., Дроздова А.А. Анализ применения электронных обучающих курсов в системе корпоративного обучения банковских сотрудников. [Электронный ресурс] – http://www.rae.ru/fs/?section=content&op=show_article&article_id=10004912
4. Дмитриев Л.В. Об электронном обучении на базе свободного ПО // Третья конференция «Свободное программное обеспечение в высшей школе». – Переславль, 2 – 3 февраля 2008 года. – Тезисы докладов. – М.: ALT Linux, 2008. – С. 33-34.
5. Сидоров В.А. Образование и подготовка кадров в условиях новой технической реконструкции. – М.: Высш. шк., 2004. – 271 с.

Волобуева Т. Б. Электронная обучающая система повышения квалификации педагогов: моделирование и внедрение. В статье рассматривается моделирование адаптивной электронной обучающей системы для повышения квалификации педагогических кадров по следующему алгоритму: 1) выбор технологии электронного обучения; 2) выбор электронной обучающей среды; 3) проектирование архитектуры; 4) формирование баз данных; 5) формирование электронных курсов повышения квалификации; 6) оценивание качества электронной обучающей системы. Автор приводит требования к цифровой среде, модель управления качеством электронной обучающей системы, коэффициенты ее эффективности. Реализация данной модели позволяет обеспечить: четкую структуру системы и регламент функционирования; возможность персонализации обучения педагогов, удовлетворение индивидуальных образовательных запросов; многофункциональность механизмов обучения; поэтапный самоконтроль обучения педагога и контроль со стороны руководителя курса; повышение мотивации непрерывного обучения; постоянное обновление содержания; повышение информационной культуры педагогов; пространство для профессионального самосовершенствования.

Ключевые слова: электронная обучающая система, педагог, повышение квалификации, качество, курсы.

Volobueva T. B. Electronic education system of the in-plant training of teachers: design and introduction. The author examines in the article the simulation of adaptive e-learning systems for professional development. It is modeled by the following algorithm: 1) selection of e-learning technology; 2) selecting an e-learning environment; 3) architecture design; 4) databases; 5) formation of electronic training courses; 6) assessment of the quality of e-learning systems. The author cites requirements for the digital environment, the model of e-learning systems, the coefficients of its effectiveness. This model provides: a clear structure and order; the organization of individual approach to the training specialist, the flexibility of information presentation, versatility training systems; staged self-control training of the teacher and supervision by the course Director; a permanent interest in learning; the relevance of the training course; the opportunity for professional self-improvement.

Keywords: electronic learning system, teacher, training, quality, courses.

Статья поступила в редакцию 19.06.2018 г.
Рекомендована к публикации доцентом Приваловым М. В.

CONTENT

Computer and information science

Analysis of the self-testing efficiency for digital circuits based on irreducible polynomials <i>Dyachenko V. O., Dyachenko O. N., Zinchenko Y.E.</i>	5
Numerical integration of rapidly varying functions by fifth-order embedded Runge-Kutta rule with adaptive step-size control <i>Maglevanny I.I., Karyakina T.I.</i>	14

Informatics and computer engineering

Probabilistic method of assessing the risk of financial stability of an enterprise <i>Vanzha T.V., Orlov J.K.</i>	20
Research and development of a method for managing a three-dimensional scene using dynamic gestures <i>Potopakhin A.A., Ruchkin K.A.</i>	28
Mathematical modeling of process crystallization of the metal during electroslag remelting <i>Minenko A. S., Lovkina A.V.</i>	33
Overview of authentication algorithms in distributed software systems <i>Grichenko O. S., Chernyshova A.V.</i>	37
On Design of Virtual Resources for Education Cloud Infrastructure <i>Zhdanovich D., Zhdanovich P.</i>	42
Teaching system for constructing a Steiner tree in an orthogonal metric <i>Olshevsky A.I.</i>	50
Application of game theory for optimum setup of the PID-regulator by the chosen criterion <i>Gulyaev V.A., Orlov Y.K.</i>	56
Development of a system for automatic recognition of Russian-speaking speech, based on discriminative learning <i>Pikaliov Y.S.</i>	61
Development of fuzzy inference system of competencies of applicants for positions in the IT department evaluation <i>Vyklyk A. V., Zemlyansky S. Y.</i>	68
Tools for analyzing social networks <i>Anokhina I. Yu., Cherepov V. G.</i>	73

Engineering education

The Criteria of the designing the models of the education base algorithmization and programming <i>Kurilov M.A.</i>	81
Electronic education system of the in-plant training of teachers: design and introduction <i>Volobueva T. B.</i>	87

Электронное периодическое издание

Научный журнал

ИНФОРМАТИКА И КИБЕРНЕТИКА

(на русском, английском языках)

№ 3 (13) - 2018

Ответственный за выпуск Р. В. Мальчева

Технический редактор Р. В. Мальчева

Компьютерная верстка А. И. Воронова

Подписано к выпуску 20.10.2018. Усл. печ. лист. 11,2. Уч.- изд. лист. 7,2.
Адрес редакции: ДНР, 83001, г. Донецк, ул. Артема, 58, ГОУ ВПО «ДонНТУ»,
4-й учебный корпус, к. 36., ул. Кобозева, 17.
Тел.: +38 (062) 301-07-35, +38 (071) 334-89-11
E-mail: infcyb.donntu@yandex.ru, URL: <http://infcyb.donntu.org>