

ISSN 1998-0663

№4(18)–2011

<http://bijournal.hse.ru>

БИЗНЕС- ИНФОРМАТИКА

МЕЖДИСЦИПЛИНАРНЫЙ НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ НИУ ВШЭ

BUSINESS INFORMATICS

Учредитель:

Национальный
исследовательский университет
«Высшая школа экономики»

Редакционная коллегия

Абдульраб А. (Франция)
Авдошин С.М.
Алескеров Ф.Т.
Белов В.В.
Грибов А.Ю.
Громов А.И.
Гюнтер Х. (Германия)
Ильин Н.И.
Калягин В.А.
Каменнова М.С.
Козырев О.Р.
Кузнецов С.О.
Мальцева С.В.
Миркин Б.Г. (Великобритания)
Моттль В.В.
Мулазани М. (Италия)
Пальчунов Д.Е.
Силантьев А.Ю.
Таратухин В.В.
Терзани С. (Италия)
Ульянов М.В.

В ЭТОМ НОМЕРЕ:

**ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ
В АЗЕРБАЙДЖАНЕ**

**ПАРАМЕТРИЧЕСКИЕ
АЛГОРИТМИЧЕСКИЕ
СТРУКТУРЫ**

**РАСПОЗНАВАНИЕ
ЧЕЛОВЕЧЕСКИХ ЛИЦ**

**РАЗМЕЩЕНИЕ
СЕТЕВЫХ ХРАНИЛИЩ**

*В соответствии с решением
президиума ВАК РФ
журнал «Бизнес-информатика»
с 19.02.2010 включён в Перечень
ведущих рецензируемых
научных журналов и изданий,
в которых должны быть опубликованы
основные научные результаты
диссертаций на соискание
ученых степеней кандидата
и доктора наук.*

БИЗНЕС- ИНФОРМАТИКА

№4(18)–2011

СОДЕРЖАНИЕ

Принятие решений и бизнес-интеллект

Р.А. Караев, Р.Г. Гюльмамедов, Н.Ю. Садыхова, М.А. Нагиев
Индикаторы состояния и факторы развития
ИКТ в республике Азербайджан 3

С.М. Авдошин, М.С. Горбатовский, А.В. Чернов
Концепция создания интеллектуальной системы
ситуационного реагирования и обеспечения
безопасности железных дорог современной России 8

Т.К. Кравченко
Оценка эффективности стратегических
решений службы информационных технологии 16

Программная инженерия

К.В. Герценбергер, Е.В. Чепин
Аналитическая модель оценки производительности
многопроцессорной обработки данных для набора
параллельных алгоритмических структур 24

Н.Л. Щеголева, Г.А. Кухарев
Применение алгоритмов двумерного анализа
главных компонент для задач распознавания
изображений лиц 31

П.Ю. Кудинов, В.А. Полежаев
Композиция случайных инкрементных
деревьев и восстановление структуры таблиц 39

Математические методы и алгоритмы решения задач бизнес-информатики

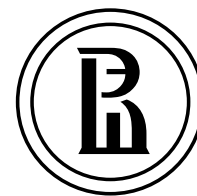
А.Г. Исавнин, Д.Р. Галиев
Использование библиотеки SEDUMI для робастной
оптимизации инвестиционного портфеля 47

Информационные системы и технологии в бизнесе

И.В. Пырлина
Классификация операционных рисков
при сервисно-ориентированном подходе
к созданию информационной системы 54

О.В. Есиков, Д.В. Изотов
Задача выбора числа и мест размещения
центров хранения и обработки информации
в компьютерной сети 62

Annotations 68



БИЗНЕС-ИНФОРМАТИКА

№4(18)–2011

Междисциплинарный
научно-практический журнал
НИУ ВШЭ

Журнал рекомендован ВАК
для научных публикаций

Подписной индекс издания
в каталоге агентства
«Роспечать» –72315

Учредитель:
Национальный
исследовательский университет
«Высшая школа экономики».
Выходит 4 раза в год.

Главный редактор
Голосов А.О.

Заместители главного редактора
Горбунов А.Р., Исаев Д.В.

Научный редактор
Лычкина Н.Н.

Технический редактор
Осипов В.И.

Корректор
Витюк Е.Б.

Дизайн обложки
Борисова С.Н.

Компьютерная вёрстка
Богданович О.А.

Администратор веб-сайта
Проценко Д.С.

Адрес редакции:
105187, г. Москва,
ул. Кирпичная, д. 33/5.
Тел.+7 (495) 771-32-38,
e-mail: bijournal@hse.ru

За точность приведённых сведений
и содержание данных,
не подлежащих открытой публикации,
несут ответственность авторы

При перепечатке ссылка на журнал
«Бизнес-информатика» обязательна

Тираж 500 экз.

Отпечатано в типографии НИУ ВШЭ
г. Москва, Кочновский проезд, 3.

© Национальный
исследовательский университет
«Высшая школа экономики»

ИНДИКАТОРЫ СОСТОЯНИЯ И ФАКТОРЫ РАЗВИТИЯ ИКТ В РЕСПУБЛИКЕ АЗЕРБАЙДЖАН

Р.А. Караев,

профессор, руководитель лаборатории Института кибернетики НАН Азербайджана

Р.Г. Гюльмамедов,

доцент кафедры информационной экономики и технологии Азербайджанского государственного экономического университета

Н.Ю. Садыхова,

научный сотрудник Института кибернетики НАН Азербайджана

М.А. Нагиев,

генеральный директор консалтинговой компании USTAD LLC

Адрес: Азербайджан, г. Баку, ул. Ф. Агаева, д. 9

E-mail: karayevr@rambler.ru; ici@minco.gov.az; natella5@rambler.ru; miragabey@yahoo.com

Отмечается важность своевременного и точного представления данных о состоянии ИКТ для принятия директивных и инвестиционных решений в современной информационной экономике. Приводятся сведения, отражающие индикаторы состояния и факторы развития ИКТ в Республике Азербайджан.

Ключевые слова: информационно-коммуникационные технологии, индикаторы состояния, факторы развития, Азербайджан.

1. Введение

Высокий потенциал информационных и коммуникационных технологий (ИКТ) как катализатора социально-экономического развития сегодня неоспорим. Международная практика [1, 2, 3, 4, 5] убедительно свидетельствует о том, что ИКТ способствуют повышению производитель-

ности и экономическому росту, однако, как именно и в какой степени, — остается, по-прежнему, предметом дискуссий. В решении этого актуального вопроса наметились несколько подходов, но первоочередными на данном этапе являются два: разработка системы индикаторов состояния и факторов развития ИКТ и анализ качества динамики этих показателей [3, 4].

Своевременные и точные данные о состоянии ИКТ необходимы для принятия взвешенных решений в приоритетных областях экономики и соответствующего распределения ресурсов. В условиях, когда важность включения проблематики ИКТ в национальные стратегии социально-экономического развития получила широкое признание, дефицит данных об ИКТ является серьезной преградой для эффективного планирования. Информация о распространении и использовании ИКТ помогает в принятии продуманных деловых и инвестиционных решений. В докладе ЮНКТАД [4] отмечается, что на национальном уровне важно поддерживать проекты по сбору данных о ИКТ с тем, чтобы обеспечить долгосрочное международное сотрудничество в деле количественного анализа ИКТ для целей соответствующей директивной деятельности.

В настоящей статье представлены некоторые результаты такого проекта, выполненного в Республике Азербайджан.

2. Состояние ИКТ-сектора

Цель проекта состояла в исследовании ключевых индикаторов, отражающих общее состояние ИКТ сектора, и выявлении основных факторов, определяющих возможности развития ИКТ в различных отраслях.

При проведении исследований использовались методы экспертного опроса (интервьюирование, анкетирование, метод круглого стола), методы стратегического анализа – SWOT-анализ и PEST-анализ. В качестве источников наряду с авторскими данными [6] использовались также отчеты Госкомстата Азербайджана (Статистические ежегодники) [7, 8], пресс-релизы и доклады Министерства связи и информационных технологий Азербайджана, аналитические обзоры Всемирного Банка [5], Комитета ОСЭР по ИКТ [3] и группы Economist Intelligence Unit (EIU) (<http://www.eiu.com>).

В процессе проведения исследований наряду с количественными и качественными показателями, дающими общую характеристику ИКТ сектора [1], нас интересовали три ключевых вопроса, являющихся сегодня приоритетными для большинства развивающихся стран:

1. мнение менеджмента предприятий (в первую очередь, малых и средних предприятий) относительно роли и возможностей ИКТ и степень ис-

пользования ИКТ в текущей деятельности предприятий;

2. основные внутренние и внешние факторы, препятствующие достижению максимальных результатов от внедрения ИКТ на предприятиях;

3. субрегиональные показатели ИКТ, отражающие степень «цифровой асимметрии» экономических районов Республики.

Как показал опрос, проведенный среди 72 руководителей и сотрудников финансовых подразделений предприятий из различных отраслей экономики (табл. 1), только 24 % (А) опрошенных (из 72) уверены, что затраты на ИКТ положительно влияют на производительность, и это влияние может быть измерено; 42 % (В) признают положительное влияние, но считают, что это влияние не может быть измерено; 10 % (С) не находят связи между затратами на ИКТ и производительностью; остальные 24 % (D) не уверены в ответе (рис. 1).

Опрос показал также, что взаимоотношения между финансовыми и ИКТ-подразделениями предприятия для достижения бизнес-целей «очень важны» с точки зрения 25 % (четверти) опрошенных; еще 70 % оценивают их как «важные»; 4 % частично признают эту точку зрения.

Таблица 1.

К какой отрасли экономики принадлежит ваше предприятие? (в процентах от общего числа респондентов)

Финансовые услуги (банки, страховые компании)	7
Услуги в сфере связи и телекоммуникаций	5
Услуги профессионального характера (врачи, адвокаты, брокеры по операциям с недвижимостью)	8
Производство стройматериалов	9
Здравоохранение	4
Торговля и общественное питание	11
Наука, образование и обучение	8
Транспорт и транспортные услуги	8
Индустрия спорта, туризма, отдыха, реклама, СМИ, издательская деятельность	6
Машиностроение	4
Химические и нефтехимические предприятия	7
Топливная промышленность (добыча нефти и газа, нефтепереработка, торговля нефтепродуктами, АЗС)	10
Электроэнергетика (ТЭЦ, ГЭС, торговля электроэнергией)	7
Прочее	9

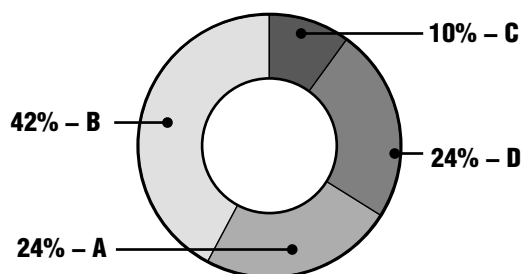


Рис. 1. Мнение менеджмента предприятий относительно роли ИКТ.

Оценка соответствия ИКТ-проектов бизнес-целям предприятий показала, что для большинства из них разрабатываемые и внедряемые ИКТ-проекты напрямую не отвечают коммерческим целям руководства и владельцев предприятий (рис. 2).

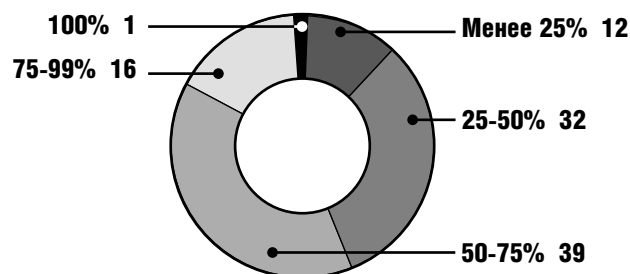


Рис. 2. Диаграмма соответствия ИКТ-проектов бизнес-целям предприятий (распределение ответов на вопрос: «Какая часть ваших ИКТ-проектов отвечает бизнес-целям предприятия?», в процентах от общего числа респондентов).

Все еще низкой остается степень использования ИКТ в торговых операциях, проводимых предприятиями (рис. 3).

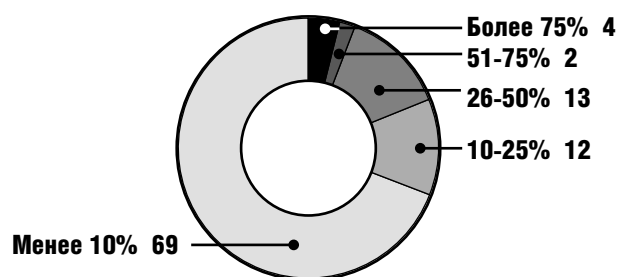


Рис. 3. Диаграмма использования Интернет в торговле (распределение ответов на вопрос: «Какая часть ваших товаров или услуг продается через Интернет?» в процентах от общего числа респондентов).

Результаты SWOT-анализа, направленного на выявление сильных и слабых сторон предприятий в их взаимодействии с угрозами и возможностями внешней среды (*Strengths* – сильные стороны, *Weaknesses* – слабые стороны, *Opportunities* – возможности, *Threats* – угрозы), позволили установить

внутренние факторы («слабые стороны» предприятий) и внешние факторы («угрозы» внешней среды), препятствующие достижению максимальных результатов от использования ИКТ. Эти результаты приведены в табл. 2 и 3.

Таблица 2.

Каковы основные внутренние факторы, препятствующие получению максимальных результатов от использования ИКТ? (в процентах от общего числа респондентов; респонденты могли указать не более двух ответов)

Недостаток ИКТ-навыков у менеджеров высшего звена	37
Отсутствие эффективного взаимодействия руководителей коммерческих и ИТ-подразделений	32
Недостаточная интеграция различных технологий в бизнесе	31
Ценовые ограничения	39
Неспособность эффективно работать с данными и использовать их в бизнес-проектах	23
Недостатки в планировании и/или реализации ИКТ-проектов	26
Сопrotивление сотрудников внедрению ИКТ	22
Прочее	7

Таблица 3.

Каковы основные внешние факторы, препятствующие получению максимальных результатов от использования ИКТ? (в процентах от общего числа респондентов)

ИКТ слишком часто не соответствуют потребностям бизнеса	36
Нехватка единых технических стандартов	33
Плохая ИКТ-инфраструктура	31
Недостаток прозрачности стоимости владения для информационных технологий	26
Недостаток профессиональной квалификации в сфере ИКТ	28
Высокая скорость устаревания ИКТ	19
Плохое послепродажное обслуживание со стороны поставщиков ИКТ	15
Недостаток стимулов, поощряющих инновации и инвестиции в ИКТ	24
Ограничивающие методы работы	14
Последствия автоматизации рабочих мест в глазах общественности	8
Прочее	3

Экономическая практика [1, отчеты группы EIU], а также исследования, проведенные Комитетом ОЭСР по ИКТ [3], показывают, что чрезвычайно важное значение для реализации потенциальных возможностей ИКТ на предприятиях имеют «условия ведения бизнеса» в регионе («качество бизнес-среды»). Сводные перечни таких условий представлены в указанных отчетах. Однако эти перечни носят открытый (справочный) характер. Естественно, что при обследовании конкретных регионов эти перечни пришлось корректировать с учетом особенностей регионов. Эти особенности мы выявляли с помощью стандартной четырехкомпонентной модели PEST-анализа, предназначенной для выявления ведущих политических, экономических, социальных и технологических факторов внешней среды предприятий (Policy – политика, Economy – экономика, Society – общество, Technology – технология). Результаты идентификации условий ведения бизнеса, выполненной с помощью PEST-анализа, и полученные оценки их значимости для предприятий-респондентов, приведены в табл. 4.

Таблица 4.

Насколько важны перечисленные «условия ведения бизнеса» для того, чтобы ваше предприятие могло получить пользу от ИКТ? (в процентах от общего числа респондентов)

Условия ведения бизнеса	1	2	3	4	5
	очень важно	важно	средне	неважно	очень неважно
1. Свободный от регулирования и конкурентный рынок ИКТ	40	30	19	8	3
2. Государственная политика, способствующая распространению ИКТ среди потребителей	27	35	25	11	2
3. Инвестиционное и налоговое стимулирование предприятий, активно использующих ИКТ	30	35	27	6	2
4. Действенная законодательная база и правоприменение, защищающие электронную торговлю	41	25	27	6	1
5. Эффективные законы, защищающие интеллектуальную собственность и сетевые ресурсы юридических и физических лиц	45	26	15	3	2

В заключительной части статьи мы приводим:

◆ ключевые индикаторы состояния ИКТ в Республике, широко используемые в международной практике, и динамику изменения этих индексов за

период 2005–2010 гг. (рис. 4);

◆ сводные индексы развития ИКТ по отдельным экономическим районам Республики (рис. 5).

Индексы развития ИКТ определялись по методике EIU и исчислялись как среднее арифметическое комплекса первичных показателей: (1) распространенность фиксированных телефонных линий (число линий на 100 человек); (2) распространенность мобильных телефонов (на 100 человек); (3) персональных компьютеров (на 100 человек); (4) пользователей Интернета (на 100 человек); (5) количество интернет-серверов на 1 млн человек; (6) распространенность широкополосного доступа (на 10 000 человек).

Каждый показатель преобразовывался в балльную оценку (по шкале 1 – 100 баллов, исходя из минимального и максимального значения этого показателя в выборке региона).



Рис. 4. Ключевые индикаторы состояния ИКТ.

Представленное на рис. 5 распределение свидетельствует о существенной «цифровой асимметрии» национального информационного пространства. Оно должно быть преодолено на второй фазе государственной программы «Е-Азербайджан» [6]. Программа осуществляется под пристальным вниманием и при всемерной поддержке со стороны высшего руководства страны и продиктована растущим спросом на ИКТ, динамичным развитием экономики страны и улучшением уровня жизни населения.

3. Заключение

Международный Союз Электросвязи оценил Азербайджан как одну из быстроразвивающихся в области ИКТ стран (<http://www.regnum.ru/news/1133149.html?forprint>). За последние 3 года темпы доходного роста в секторе ИКТ примерно в



Рис. 5. Распределение индексов развития ИКТ по отдельным экономическим районам (2010). Исходные данные взяты из источников [6, 7, 8].

2,5–3 раза опережают темпы всемирного развития и в среднем ежегодно составляют 30–35% (<http://www.bakutel.az/2011/?p=index>). В настоящее время по темпам роста сектор занимает в Республике второе место после нефтегазового сектора. Ожидается, что такая тенденция будет продолжена и в течение нескольких лет они приблизятся к доходам

от нефтегазового сектора, а в 2018–2020 годах даже превысят их (http://news.day.az/hitech/165073_print.html). Это обстоятельство делает проблему мониторинга ИКТ-сектора еще более актуальной и ставит вопрос детального изучения местных условий для правильной интерпретации рамочных документов МСЭ, Евростат и ОЭСР [9]. ■

Литература

1. Штрик А.А. Использование информационно-коммуникационных технологий для экономического развития и государственного управления в странах современного мира // Информационные технологии. № 6. 2009. Приложение. 32 с.
2. Международная конференция «ЮНЕСКО между двумя этапами Всемирного саммита по информационному обществу». 2005. URL: http://confifap.cpic.ru/conf2005/rus/info/progr_ru.htm (дата обращения 28.06.2011).
3. Организация Экономического Сотрудничества и Развития (ОЭСР). Наука, технологии и промышленность: Перспективы 2008. URL: <http://www.oecd.org/html> (дата обращения 17.06.2011)
4. ЮНКТАД. Конференция ООН по торговле и развитию. Доклад об информационной экономике. 2010. URL: <http://www.unctad.org/Templates/webflyer.asp?docid=13912&intItemID=3594&lang=1> (дата обращения 19.05.2011).
5. IC4D. Information and Communications for Development 2009: Extending Reach and Increasing Impact. (IC4D is a regular publication of the World Bank on the critical role of information and communication technology (ICT) in economic development) URL: <http://go.worldbank.org/DMY979SNP0> (дата обращения 12.05.2011).
6. Gulmamedov R.H. The National ICT Strategy and Key Indicators on Information Society of Azerbaijan Republic / Proceedings of International Conference on e-Government Sharing Experiences. Antalya. Turkey, 8–11 December 2009. P. 373–381.
7. Телекоммуникация и связь в Азербайджане. Статистический ежегодник. Госкомстат Республики Азербайджан. 2010. – 356 с.
8. Информационное общество в Азербайджане. Информационные и коммуникационные технологии. Статистический ежегодник. Госкомстат Республики Азербайджан. 2010. – 147 с.
9. ООН. Экономический и Социальный Совет. Статистическая комиссия. Доклад Партнерства для статистического измерения информационно-коммуникационных технологий в интересах развития: статистика информационно-коммуникационных технологий. 2009. URL: <http://www.itu.int/ITU-D/ict/partnership/material/2009-19-ICT-R.pdf> (дата обращения 01.07.2011).

КОНЦЕПЦИЯ СОЗДАНИЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ СИТУАЦИОННОГО РЕАГИРОВАНИЯ И ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ЖЕЛЕЗНЫХ ДОРОГ СОВРЕМЕННОЙ РОССИИ¹

С.М. Авдошин,

профессор, руководитель отделения программной инженерии, заведующий кафедрой управления разработкой программного обеспечения Национального исследовательского университета «Высшая школа экономики», академик Всемирной академии наук комплексной безопасности

М.С. Горбатовский,

директор по развитию бизнеса в финансовой отрасли ООО «ИБМ Восточная Европа/Азия»

А.В. Чернов,

директор по развитию бизнеса в транспортной отрасли ООО «ИБМ Восточная Европа / Азия»

Адрес: г. Москва, ул. Кирпичная, д. 33/5

E-mail: savdoshin@hse.ru, mikhail.gorbatovskiy@ru.ibm.com, alexander.chernov@ru.ibm.com

Предлагается архитектура и компоненты модели создания интеллектуальной системы ситуационного реагирования и безопасности на транспортной инфраструктуре на основе методов и алгоритмов обработки потоковых первичных данных и строительства распределённой системы сбора и анализа данных из разнородных источников. Предлагается архитектура прогнозирования и принятия решений в режиме реального времени на сверхбольших объемах данных, а также подход к динамической оптимизации расписаний на железнодорожной инфраструктуре.

Ключевые слова: обработка потоков данных, ситуационное реагирование, динамическая оптимизация, прогнозирование по поведенческим шаблонам, сети датчиков.

¹ Статья подготовлена в рамках государственного контракта № 07.514.11.4039 на выполнение научно-исследовательских работ по теме: «Исследование и разработка инновационных комплексных моделей интеллектуальной системы ситуационного реагирования и контроля безопасности железных дорог современной России», проводимых по федеральной целевой программе «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы».

1. Проблематика современных железных дорог

Современные мировые тенденции свидетельствуют о глобализации и интеграции ключевых областей человеческой деятельности. Сегодня мы живем на разумной планете – развивающейся комплексной системе, которая делает окружающий мир все более доступным, интегрированным и управляемым. Научно-технический прогресс способствует росту технологической оснащенности человеческой деятельности, а тенденции к глобализации и интеграции приводят к появлению взаимосвязей и постоянных коммуникаций между всеми отраслями экономики.

Транспорт является основной коммуникационной инфраструктурой, обеспечивающей человеческую деятельность. Рост населения, повышение его мобильности и глобализация экономики повышает зависимость от транспорта. Железные дороги занимали и продолжают занимать наибольшую долю растущего рынка перевозок и поэтому должны быть наиболее тесно интегрированы в мировую экономику и социальную сферу.

О значимости железных дорог свидетельствуют следующие цифры:

- ◆ По железным дорогам ежегодно перевозится около 10 миллиардов тонн грузов и 21 миллиард пассажиров.

- ◆ В предкризисные годы мировой рынок железнодорожных перевозок рос в среднем на 9%.

- ◆ Железные дороги от 2 до 5 раз эффективнее автомобильного и авиационного транспорта (один литр условного топлива может переместить тонну груза на 190 километров). При этом пассажирские путешествия по железным дорогам приводят к выбросам CO₂ в 3-10 раз меньше по сравнению с автомобильными или авиационными путешествиями.

Сегодня железные дороги стоят перед глобальной проблемой, которая заключается в росте потребностей в перевозках при высокой загруженности существующей инфраструктуры, недостаточной гибкости и операционной эффективности, а также структурных ограничениях. В нынешних условиях достигнут предел возможностей экстенсивного развития, что становится узким местом в развитии экономики. Наступает время изменений подходов к развитию, в которых на первое место выходят оптимизация, рационализация и интеллектуализация.

Как применить понятия интеллекта и интеллектуализации к информационным системам и поддерживаемой ими системе управления компанией?

Интеллектуальные железные дороги представляют собой технологически оснащенную систему, компоненты которой тесно взаимодействуют между собой в едином информационном пространстве, что позволяет улучшать операции и осуществлять проактивное управление деятельностью за счет того, что вся необходимая и свежая информация для принятия решений находится «под рукой». Проактивность (предупреждение, прогнозирование, своевременность) – это основная черта интеллектуальной деятельности [4]. Такие железные дороги могут претендовать на место центрального, интегрирующего звена экосистемы, в которую входят различные транспортные операторы и их логистическая инфраструктура, интермодальные перевозчики, клиенты и представительства власти.

В современных условиях интеллектуализация железных дорог становится насущной задачей не только благодаря появлению технических возможностей для улучшения эффективности деятельности, но во многом благодаря ожиданиям общества от транспортной отрасли, которые сегодня реализуются лишь с помощью инновационных подходов.

2. Какими будут железные дороги в недалеком будущем?

Как будут выглядеть железные дороги будущего? «Интеллект» железных дорог будет проявляться в следующем:

- ◆ Маршруты, расписание и графики движения динамически регулируются за счет анализа факторов спроса, погодных, сезонных и инфраструктурных условий.

- ◆ Задержки поездов снижаются, а безопасность услуг возрастает благодаря системам, своевременно идентифицирующим проблемы и поломки.

- ◆ Возрастает средняя скорость движения и объем перевозок без существенных капитальных вложений за счет анализа операций и результатов деятельности, а также эффективного управления активами.

- ◆ Возрастает прозрачность и эффективность интермодальных перевозок за счет модернизации, стандартизации и интеграции систем различных операторов.

✧ Пассажиры и клиенты грузовых перевозок могут гибко управлять своими действиями на протяжении всего цикла транспортировки, пользуясь доступными и стандартными каналами получения интегрированного набора услуг.

Ряд решений, способствующих повышению «интеллекта» системы управления, успешно используются в железнодорожных компаниях ряда стран. Развиваются беспроводные системы мониторинга, позволяющие отслеживать положения подвижного состава и грузов, корректность эксплуатационной работы, а также насыщенность пассажиропотоков в транспортных сетях. Внедряются системы оплаты проезда в городском и пригородном железнодорожном транспорте с использованием единых транспортных карт, при этом собираемые данные о поездках помогают пассажирам выбирать оптимальные маршруты движения. Применяются аналитические средства для динамического управления расписанием и движением, что позволяет сокращать объемы условного топлива и организовывать высокоскоростные пассажирские маршруты. Разрабатываются решения по предоставлению интегрированных сервисов для пассажиров и посетителей вокзальных комплексов, что способствует развитию основного и смежного бизнесов, а также росту лояльности пассажиров. Устанавливаются средства интеллектуального видеонаблюдения на вокзалах и крупных станциях, которые способствуют улучшению операционной деятельности и повышению безопасности.

Основными факторами интеллектуализации являются технологическая оснащенность, тесное взаимодействие компонент и вытекающая из этого проактивность управления [4]:

- Повышение технологической оснащенности железных дорог поможет собирать необходимую информацию для взаимодействия компонент системы железных дорог в едином информационном пространстве:

- Сенсоры и средства анализа событий, связанных с подвижным составом и инфраструктурой.

- Интеллектуальные средства видеонаблюдения на вокзалах, станциях и логистических терминалах.

- Мобильные средства отслеживания состояния и нахождения грузов.

- Взаимодействие компонент системы железных дорог, основанное на интенсивном и управляемом

обмене информацией, позволит осуществлять более взвешенное и оптимальное управление:

- Решения для оперативного управления (ситуационные центры и риск-анализ), анализа пассажиропотоков, динамического управления расписанием и оптимизации использования подвижного состава.

- Решения для совместной деятельности различных операторов на высокоскоростных пассажирских линиях.

- Интеграция информационных сервисов для клиентов грузовых перевозок, пассажиров и посетителей вокзальных комплексов.

- Проактивность управления, базирующаяся на своевременном получении и анализе необходимой информации, технологическое оснащение и организация взаимодействия компонент создадут все предпосылки для повышения эффективности управления [5]:

- Средства многофакторного анализа исполнения графиков движения.

- Решения для управления жизненным циклом производственных активов.

- Решения для анализа грузовых и пассажирских потоков с оптимизацией маршрутов движения.

Перечисленные направления качественного преобразования железнодорожной отрасли – это путь к интеграции железных дорог в глобальную экономику и приобретению ими долгосрочного конкурентного преимущества. Интеллектуализация железнодорожной отрасли – это реальность сегодняшнего дня. Во многих странах – США, Китай, Япония, Франция – развивается высокоскоростное движение, для организации которого необходима не только хорошая железнодорожная инфраструктура, но и механизмы управления информацией на качественно новом уровне – чем выше скорость, тем больше факторов, которые необходимо принимать в расчет при управлении, тем быстрее надо принимать решения и тем серьезнее ответственность за возможные ошибки. Также применяются средства управления производственными активами, позволяющие оптимизировать использование вагонного парка, и системы динамического управления движением и расписанием – например, на железных дорогах Нидерландов применяется система, позволяющая ежедневно управлять расписанием 5000 пассажирских поездов, перевозящих около 1 000 000 пассажиров по расписанию, которое динамически изменяется при учете более чем 50 000

факторов. Наконец, в ряде развитых европейских стран и в Великобритании успешно зарекомендовали себя решения для обслуживания пассажиров на всем цикле поездки.

Процесс интеллектуализации железных дорог – это пошаговое внедрение и интеграция технологических инноваций в различных областях деятельности, от управления движением и управления активами до обслуживания клиентов и пассажиров с последующим выстраиванием проактивной системы управления. Интеллектуализация является долгосрочной инициативой и как ключевой ориентир должна поддерживаться стратегическими решениями высшего руководства железнодорожной компании. Принятие таких решений должно основываться на понимании следующих вопросов:

- ◆ Можем ли мы сегодня измерять и анализировать качество и результаты операций, чтобы использовать эту информацию для принятия своевременных и эффективных решений?
- ◆ Достаточно ли гибкой является наша деятельность, чтобы соответствовать меняющимся требованиям клиентов и пассажиров?
- ◆ Можем ли мы планировать использование и ремонт наших активов, исходя из реальной потребности в этих активах и их реального состояния?
- ◆ Можем ли мы в реальном времени отслеживать состояние и местонахождение вагонов и грузов?
- ◆ Получают ли наши пассажиры интегрированный и удобный в управлении набор услуг, помогающий им на протяжении всего цикла поездки?
- ◆ Являются ли наши информационные системы технологически развитыми и интегрированными в той мере, чтобы обеспечить внутренние коммуникации, а также внешнее взаимодействие с другими участниками отрасли?
- ◆ Насколько справедливы наши предположения о конкурентоспособности компании в различных сегментах рынка?

3. Примеры интеллектуальных подходов к информатизации железных дорог

Интеллектуальный подход к информатизации на железных дорогах востребован сегодня для поддержки деятельности ситуационных центров, которые должны своевременно (в режиме, приближенном к реальному времени) получать необходимые разнородные первичные данные по операционной деятельности, сопоставлять эти данные, оценивать

уровни рисков тех или иных наступивших или ожидаемых событий и своевременно осуществлять необходимые действия. В масштабах страны речь идет о сверхбольших объемах данных, поступающих в распределенную систему в режиме реального времени [2]. Как правило, для этого нужно знать:

- ◆ Актуальные координаты поездов, находящихся в пути, и маршруты их следования.
- ◆ Техническое состояние локомотивов и вагонов, сведения о владельцах вагонов.
- ◆ Ситуацию на перегонах, по которым движутся поезда.
- ◆ Ситуацию на станциях по ходу следования и их технических возможностях.
- ◆ Необходимые сведения о перевозимых грузах и их владельцах.

Постоянно собирая множество таких первичных данных и анализируя их «на лету», информационная система ситуационного центра идентифицирует повышенный риск в следующем гипотетическом случае:

- Грузовой состав перевозит цистерны с опасным грузом и движется со скоростью 80 км/ч.
- На одной из платформ обнаружена неисправность, которая при таком режиме движения может привести к аварии (например, нарушена геометрия размещения цистерны на платформе или зафиксирован чрезмерный нагрев букса колесной пары).
- На ближайшей станции находится вагонное депо, которое может выполнить необходимые ремонтные работы.

Анализ этих данных «на лету» и применение к этим данным определенных правил, настроенных в информационной системе, позволяет инициировать в автоматическом режиме следующие действия:

- Отправить локомотивной бригаде сообщение о неисправности и необходимости снизить скорость до 40 км/ч.
- Отправить сообщение диспетчеру станции по ходу движения поезда о неисправности, указав номер вагона.
- Отправить сообщение владельцу вагона и владельцу груза о возникшей ситуации.

Другой областью, где востребован интеллектуальный подход к информатизации на железных дорогах, является деятельность грузового перевозчика, который получает заявки на грузовые перевозки

и стремится оптимизировать свою эксплуатационную работу по этим заявкам, снижая порожний пробег вагонов. В самом общем случае для этого надо принимать в расчет:

■ Содержание заявок на грузовые перевозки (какой тип груза, когда, откуда и куда следует перевезти).

■ Актуальное и прогнозируемое на некоторое время вперед состояние вагонного парка (какой вагон, в каком техническом состоянии, в каком статусе и где находится).

■ Правила эксплуатационной работы, тарифы и ограничения по перевозкам различных видов грузов.

Располагая информацией по поступающим заявкам на грузовые перевозки и сопоставляя эти заявки с актуальными сведениями о вагонах, информационная система оценивает различные варианты исполнения заявок и предлагает оптимальное решение. Предположим, в упрощенном случае, что:

◆ Поступила заявка перевезти со станции А сыпучий груз, который может уместиться при соблюдении нормативов перевозки в двух полувагонах. Для соблюдения сроков доставки необходимо начать перевозку не позднее, чем через два дня.

◆ На станции А имеется один свободный полувагон, находящийся в пригодном техническом состоянии, несколько полувагонов, находящихся в вагонном депо станции на ремонте, который закончится через неделю, и несколько свободных полувагонов, находящихся на станции Б в двух днях пути.

◆ На станцию А через день прибудет состав, в котором имеется один порожний полувагон, который готов к погрузке данного сыпучего груза. Состав будет расформировываться и проследует дальше на следующий день.

Анализируя различные варианты исполнения заявки, информационная система предлагает разместить половину груза в свободном полувагоне, который уже находится на станции А, затем дожидаться прибытия на станцию второго полувагона и разместить в нем оставшуюся часть груза, после чего отправить груз в срок с составом, в котором пришел второй свободный полувагон.

Перечисленные усилия способствуют повышению средней скорости и объемов железнодорожных перевозок, их безопасности и качества обслуживания клиентов железных дорог.

4. Какие программные платформы необходимы для интеллектуальной информатизации?

Реализация описанного выше интеллектуального подхода к информатизации железных дорог как крупных хозяйствующих субъектов требует сбора и обработки сверхбольших объемов данных в потоковом режиме, приближенном к реальному времени [1, 3]. Крупные вендоры, понимая данную тенденцию, инвестируют в разработку методик и создание решений для интеллектуализации железных дорог, а также в разработку программно-аппаратных платформ для высокопроизводительных вычислений на сверхбольших объемах разнородных данных, помогая железнодорожным компаниям внедрять инновационные средства для качественного преобразования отрасли.

Соответствующие программные платформы классифицируются по трем уровням:

◆ *Уровень операционного управления* требует принятия правильных решений в режиме, приближенном к реальному времени. Этой задачей занимаются ситуационные центры, и на сегодняшний день во многих железнодорожных компаниях они используют устаревшие технологии — телефонную связь, факсограммы и периодические отчеты из различных информационных систем. При таком подходе основная нагрузка по обработке данных и принятию решений ложится на людей, что повышает риски операционной работы. Снизить эти риски позволяют программные платформы, позволяющие обрабатывать сверхбольшие объемы разнородных первичных данных в потоке (без использования технологии хранилищ), сопоставлять эти данные на основе определенных бизнес-правил, определять уровни рисков и должным образом реагировать на возникающие ситуации, вовлекая людей лишь в случае необходимости [7]. Наиболее зрелыми, по мнению авторов, в техническом и коммерческом плане являются платформы IBM Infosphere Streams [6], технология на основе открытого кода StreamBase и технология Sybase Aleri. При этом первичные данные могут поступать из самых различных источников: датчиков технического состояния вагонов, систем линейного уровня, систем управления коммерческой деятельностью и даже тех же факсограмм.

◆ *Уровень тактического управления* требует поиска оптимальных вариантов решения часто возникающих типовых задач. Этой задачей занимаются

специалисты по управлению вагонным хозяйством, когда требуется сформировать пассажирский состав или исполнить заявки на грузовые перевозки с минимальными издержками. На сегодняшний день такие задачи решаются, как правило, путем целевых приписок вагонов к регулярным составам и маршрутам, а при возникновении неплановых ситуаций к решению задачи подключаются люди: запрашиваются отчеты по актуальному расположению резервных вагонов, уточняются прогнозы по ремонтам вагонов, актуализируются статусы используемых вагонов и прогнозы по разгрузке требуемых вагонов, после чего принимается решение, не учитывая все возможные варианты. Повысить качество таких решений позволит программная платформа, позволяющая формировать математические модели определенных участков деятельности, выполнять имитационное моделирование при различных вариантах решения задачи и находить оптимальное решение. В промышленных системах, решающих задачи оптимизации, наиболее широко используется технология IBM iLog Optimization на базе математических библиотек CPLEX и методов программирования по ограничениям CP. Важно понимать, что имитационное моделирование выполняется на основе тех же первичных данных, которые используются на уровне операционного управления, поэтому комплекс тактического управления должен представлять собой «надстройку» поверх комплекса операционного управления, а соответствующие программные платформы должны интегрироваться друг с другом.

✧ **Уровень стратегического управления** требует выработки планов деятельности на основе качественных прогнозов и различных видов анализа. Этой задачей занимается руководство компании, и для ее решения требуется глубокое понимание грузовых и пассажирских потоков, их динамики и тенденций. Традиционно такие задачи решаются путем создания аналитического комплекса, работающего поверх хранилищ данных, но во многих практических случаях наблюдается проблема полноты, достоверности и своевременности первичной информации, что сказывается на качестве аналитических отчетов и прогнозов. Повысить качество аналитической работы и решения задач планирования деятельности позволят программные платформы бизнес-аналитики (BI – business intelligence), которые обладают широким спектром аналитических инструментов и хорошо интегрируются с решениями для операционного управления, что позволяет успешно решать проблемы с первичными данными [5].

Высокоуровневое представление интеллектуального подхода к построению архитектуры программного комплекса, решающего описанные задачи, авторы схематично представили на *рис. 1*.

Потоковая обработка сверхбольших объемов данных выполняется на вычислительном кластере на базе аппаратной платформы x86 серверов производства Intel, IBM, Hitachi и др. Обработка потоков данных с датчиков и систем линейного уровня производится в оперативной памяти путём выполнения различных операций над потоком – матема-

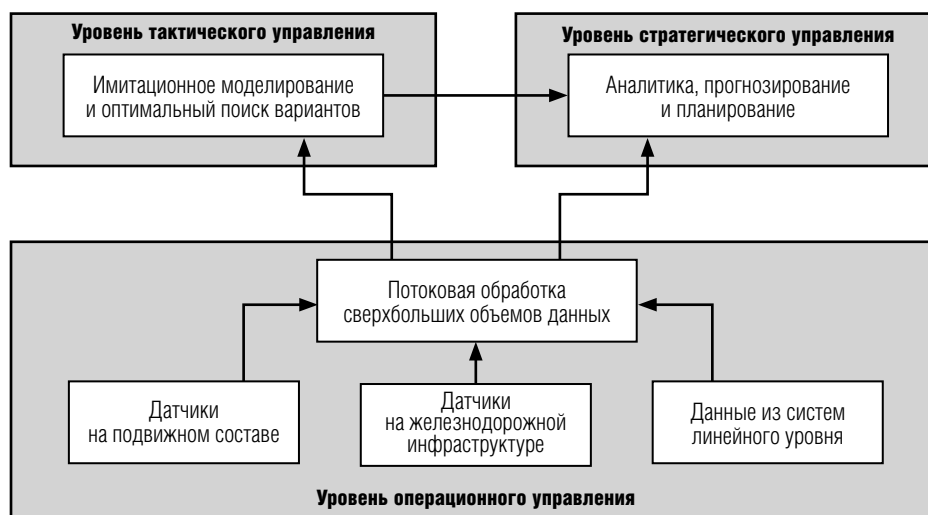


Рис. 1. Интеллектуальный подход к построению архитектуры программного комплекса.

тических, логических, операций типов join и split, синхронизации потоков во времени и др. Выполнение вычислений в оперативной памяти позволяет в десятки раз повысить их производительность и соответственно снизить стоимость транзакции.

Наиболее зрелой по функциональности, надёжной и высокопроизводительной является технология IBM Infosphere Streams, способная в отличие от большинства обработчиков дискретных событий (CEP-engines), таких как StreamBase, Oracle CEP, Progress Arama, WBE, обрабатывать на одной платформе как дискретные события, так и потоковое видео, аудиопоток, текст и пространственно-локационные данные. Поток результатов обработки на выходе из системы может быть помещён в файл, базу данных, интеграционную шину, отправлен по протоколу TCP/IP или UDP. Это позволяет построить территориально распределённую систему и организовать обработку больших объёмов первичных данных на линейном уровне станций, подъездных путей, отдельных объектов и предприятий, тем самым не загружать каналы связи несущественной информацией.

Компонент оптимизации и имитационного моделирования содержит в виде графа статическую модель сети железных дорог и объектов инфраструктуры, а также свойства этих объектов, различные параметры и атрибуты. Системы транспортной инфраструктуры, содержащие информацию о движении поездов, заявках на пассажирские и грузоперевозки, на ремонты и обслуживание и другие данные, наполняют систему имитационного моделирования динамической информацией о текущем и планируемом расписании движения, ремонтов, погрузок/разгрузок. Применяя набор математических библиотек и инструментов параллельных расчетов на вычислительном кластере система проводит оптимизацию по одному из выбранных критериев.

Одной из редких технологий, поддерживающей оптимизацию в промышленных масштабах для крупных предприятий и параллельные расчеты, является iLog OPL. ILOG OPL, и в частности о IBM ILOG CP Optimizer, который предназначен для эффективного моделирования и решения практических задач планирования. Данное программное обеспечение является новым шагом программирования в ограничениях, показывающее эффективную работу в задачах составления расписания, а также некоторых задач комбинаторной оптимизации, которые не могут быть легко линеаризованы и достаточно трудны в решении традиционными методами математического программирования.

Основным преимуществом подхода, используемого в CP Optimizer, является то, что для временных промежутков используются специальные переменные, что заметно сокращает количество неизвестных по сравнению с методами математического программирования. Поэтому такая временная детализация используется для описания ограничений модели, которая может быть сформулирована и решена более эффективно.

Роль системы потокового анализа первичных данных на уровне операционного управления, интегрированной с системой имитационного моделирования и оптимизации, заключается в данном случае в предоставлении команды о необходимости пересчёта расписания. Это требуется в случае возникновения аварии или её признаков на одном из узлов транспортной сети, изменения в приоритетах заказов, необходимых для динамической оптимизации расписания.

Аналитику, прогнозирование и планирование на уровне стратегического управления целесообразно выполнять с использованием специализированного ПО, позволяющего организовать цикл аналитического процесса: планирование исследования, сбор данных, всесторонний анализ с использованием математического аппарата ПО до отчетности и распространения результатов. Решение этих задач выполняется на базе промышленных технологий исследования данных (data mining), таких как IBM SPSS, SAS, R. Технологии исследования данных выполняют аналитические (читай OLAP) операции с хранимыми данными в пакетном режиме.

Для выполнения процедуры анализа данных системы потокового анализа должны сначала наполнить хранилище данных информацией, соответствующей требованиям системы прогнозирования. Далее в офф-лайн режиме выполняется изучение данных с применением как универсальных методов анализа – анализ связей между переменными, параметрические и непараметрические методы, регрессии, общие линейные модели и т.д., так и специальных – методов временных рядов, прогнозирования количественных и категориальных исходов, моделирование сложных взаимосвязей, нейронные сети. Получив результаты анализа данных, мы переводим их в модель исследования данных и прогнозирования, а также создаем в визуальном инструментарии автоматические задания, поддерживающие процесс принятия решения [3].

Запуск заданий по исследованию данных и принятию решений по подготовленным алгоритмам реализуются в двух вариантах:

◆ задания, сформированные в системе прогнозирования и анализа данных, выгружаются в формате языка разметки для прогнозного моделирования PMML, а затем запускаются в системе обработки потоковых данных. Это позволяет выполнять обнаружение аномалий, прогнозирование и прочую аналитику в режиме реального времени с низкими задержками;

◆ задания реализуются на хранимой информации в пакетном режиме (например, ночью или на выходных), когда требуется дополнительный и углублённый анализ исторической информации. Это используется, например, когда требуется повысить качество модели или автоматически обновить её с учётом новых факторов.

Заключение

Таким образом, при правильном выборе и развитии современных программных платформ появляется реальная возможность решения актуальных задач крупных железнодорожных компаний, ведущих многопрофильную и масштабную деятельность. Усложнение этих задач в условиях инфраструктурных ограничений и рост потребностей в перевозках требует новых, интеллектуальных подходов к информатизации, и применение таких подходов является на сегодняшний день не опцией, а неизбежностью, диктуемой сложившимися реалиями. Сегодня успешная информатизация становится ключевым фактором конкурентоспособности транспортных компаний. ■

Литература

1. Antonopoulos N. Handbook of Research on P2P and Grid Systems for Service-Oriented Computing: Models, Methodologies, and Applications. – IGI Global, 2010.
2. Bizarro P., Chandy M.K., Stojanovic N. Event processing grand challenges // DEBS 2011.
3. Yang L., Guo M. High-Performance Computing: Paradigm and Infrastructure. – John Wiley & Sons, 2006.
4. Koyuncugil A.S., Ozgulbas N. Surveillance Technologies and Early Warning Systems: Data Mining Applications for Risk Detection. – NY: Hershey, 2011.
5. Richardson J. Need for Speed Powers In-Memory Business Intelligence. – Gartner Research, 2011.
6. Foster K., Spicer M., Nathan S. IBM Infosphere Streams: Assembling Continuous Insight in the Information Revolution. – San Jose, California: International Technical Support Organization, 2011.
7. Buchmann A., Hinze A.M. Principles and Applications of Distributed Event-Based Systems. – NY: IGI Global, 2010.

ОЦЕНКА ЭФФЕКТИВНОСТИ СТРАТЕГИЧЕСКИХ РЕШЕНИЙ СЛУЖБЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

*Т.К. Кравченко,
доктор экономических наук, профессор,
заведующий кафедрой бизнес-аналитики
Национального исследовательского университета
«Высшая школа экономики»*

*Адрес: г. Москва, ул. Кирпичная, д. 33/5
E-mail: tkravchenko@hse.ru*

В статье предложен подход для выявления признаков оценки стратегических решений службы информационных технологий. Особенностью подхода является интеграция преимуществ системы сбалансированных показателей для ИТ-службы – IT Balanced Scorecard (IT BSC) и стандарта COBIT для получения иерархической структуры признаков (показателей) оценки эффективности решений в сфере информационных технологий.

Ключевые слова: служба информационных технологий, стратегические решения, сбалансированная система показателей, стандарт COBIT, перспективы, вклад в бизнес, пользователи, качество эксплуатации, долгосрочная ориентация, причинно-следственные связи, системы поддержки принятия решений.

1. Введение

Служба информационных технологий (далее ИТ-служба) занимает особое место в любой компании, поскольку предоставляет и поддерживает множество сервисов, являющихся критически важными для бизнеса. В компетенцию ИТ-службы входит обоснование многих стратегических решений, к числу которых относятся следующие:

♦ выбор стратегии развития информационных систем;

♦ выбор вариантов внедрения информационных систем;

♦ расстановка приоритетов при формировании ИТ-бюджета;

♦ выбор технологических платформ для корпоративных приложений;

♦ выбор поставщиков ИТ-ресурсов;

♦ принятие решений о внесении изменений в ИТ-инфраструктуру;

♦ выбор методики обучения пользователей;

♦ принятие решения об интеграции системы управления информационными технологиями с корпоративным управлением.

В большинстве случаев при оценке вариантов стратегических решений в недостаточной степени учитывается специфика деятельности ИТ-службы. В данной работе предложен подход к формированию признаков (показателей эффективности), с позиций которых рекомендуется оценивать варианты таких решений.

2. Методологические предпосылки исследования

В основе предлагаемого подхода лежит идея совместного использования методологии формирования сбалансированной системы показателей, адаптированной для ИТ-сферы (IT Balanced Scorecard) и стандарта COBIT, содержащего набор рекомендаций для аудита и управления в сфере ИТ.

Методология сбалансированной системы показателей (Balanced Scorecard, BSC), созданная Робертом Капланом и Дэвидом Нортоном, используется для формирования и реализации стратегии компании: от разработки стратегических целей до конкретных мероприятий по их реализации и мониторингу эффективности через систему соответствующих показателей [1, 2].

При группировке показателей изначально использовались четыре перспективы:

- ♦ финансы (традиционные финансовые цели, результаты работы и факторы, которые на них прямо или косвенно влияют);
- ♦ клиенты (показатели, отражающие качество работы с клиентами);
- ♦ внутренние бизнес-процессы (показатели, определяющие эффективность бизнес-процессов);
- ♦ обучение и развитие (факторы, отражающие компетентность персонала компании, возможности для инноваций, развития и расширения компании).

Позднее на основе традиционной сбалансированной системы показателей (BSC) была разработана аналогичная система показателей для ИТ-службы – IT Balanced Scorecard (IT BSC) [5, 6]. В этой системе вместо четырех классических перспектив были выделены следующие (табл. 1).

Таблица 1.

Перспективы IT BSC

Вклад в бизнес
Перспектива показывает, как деятельность ИТ-службы оценивается руководством компании. Нацелена на получение наибольшей ценности бизнеса за счет инвестиций в ИТ. Включает контроль расходов на ИТ, ценность ИТ-проектов для бизнеса, обеспечение новых возможностей для бизнеса.
Пользовательская перспектива
Показывает, как пользователи реагируют на функционирование ИТ. Нацелена на поддержку позиции ИТ-службы как наиболее предпочитаемого сервиса подобных услуг и удовлетворение потребностей пользователей. Включает предложение наиболее качественного с точки зрения поддержки пользователей программного обеспечения и функциональных решений.
Качество эксплуатации
Показывает, насколько эффективны и результативны процессы ИТ. Нацелена на предоставление отлаженных процессов при разработке, внедрении и поддержке программ. Включает поддержку качества операций и разработок в ИТ.
Долгосрочная ориентация
Показывает, насколько хорошо ИТ-служба организована для решения будущих проблем. Нацелена на развитие возможностей и предоставление ресурсов для решения новых задач. Включает обучение и проверку качества подготовки ИТ-персонала, исследование смежных технологий и обновление имеющегося программного обеспечения.

Control Objectives for Information and related Technology (COBIT) – методология, включающая набор стандартов и рекомендаций для аудита и управления ИТ, разработанная Ассоциацией аудита и контроля информационных систем (Information Systems Audit and Control Association, ISACA). Стандарт состоит из взаимосвязанных компонент, оказывающих поддержку в области управления, контроля и аудита как высшему руководству, совету директоров, руководителям бизнес-сегментов и ИТ-менеджменту, так и специалистам в различных областях [4].

Рассматривая COBIT со стороны стратегического управления, следует отметить, что стандарт ориентирован на бизнес. Для того чтобы организация обеспечивала себя необходимой для достижения целей информацией, она должна инвестировать и управлять ИТ-ресурсами посредством структурированного комплекса процессов, которые обеспечивают сервисы для предоставления информации. Методология COBIT нацелена на управление и контроль информации, что помогает соответствовать требованиям бизнеса.

При наличии четкого руководства со стороны бизнеса в части формулировки требований к ИТ-службе информационные технологии поддерживают выполнение корпоративной стратегии организации.

Процесс использования ИТ для поддержки деятельности организации выглядит следующим образом: корпоративная стратегия преобразуется в бизнес-цели, связанные с ИТ-инициативами. Далее формулируются непосредственно цели ИТ-службы.

СОВИТ представляет деятельность ИТ в виде модели процессов, состоящей из четырех доменов, которые отражают традиционные виды деятельности, риски и зоны ответственности в сфере ИТ, связанные с планированием, созданием, сопровождением и мониторингом [4].

1). Планирование и организация

Данная группа процессов охватывает стратегию и тактику, а также то направление деятельности ИТ, которое внесет больший вклад в достижение бизнес-целей. Отвечает на следующие вопросы управления:

- ◆ согласованы ли между собой ИТ-стратегия и корпоративная стратегия;
- ◆ оптимально ли используются ресурсы компании;
- ◆ понимают ли сотрудники цели, поставленные перед ними службой ИТ;
- ◆ осуществляется ли управление ИТ-рисками;
- ◆ соответствует ли качество систем ИТ-потребностям бизнеса;
- ◆ определение доли ИТ-бюджета в совокупном бюджете компании.

2). Приобретение и внедрение

Для реализации ИТ-стратегии должны быть выявлены, разработаны или приобретены ИТ-решения, которые также должны быть внедрены и интегрированы в бизнес-процесс. Кроме того, необходима уверенность в том, что ИТ-решения соответствуют бизнес-целям, поэтому к этому домену относится изменение и обслуживание существующих систем. Здесь рассматриваются следующие вопросы:

- ◆ предлагают ли новые проекты решения, отвечающие потребностям бизнеса;

- ◆ выполняются ли новые проекты в срок и в рамках бюджета;

- ◆ будут ли новые системы после внедрения работать должным образом;

- ◆ удастся ли внедрить изменения без сбоев в текущих бизнес-операциях.

3). Эксплуатация и сопровождение

Домен связан с предоставлением услуг, обеспечением безопасности и непрерывности сервисов, поддержкой конечных пользователей, управлением данными. Рассматриваются следующие вопросы:

- предоставляются ли ИТ-услуги в соответствии с приоритетами бизнеса;
- оптимизированы ли затраты на ИТ;
- способен ли ИТ-персонал использовать ИТ-системы эффективно и безопасно;
- обеспечивается ли должный уровень конфиденциальности, целостности и доступности.

4). Мониторинг и оценка

Домен связан с управлением эффективностью, мониторингом систем внутреннего контроля и проверкой соответствия требованиям регулирующих норм. Рассматриваются следующие вопросы управления:

- как выявить проблемы, связанные с эффективностью ИТ-процессов;
- существуют ли результативные и эффективные системы внутреннего контроля;
- как установить обратную связь между эффективностью ИТ и целями бизнеса;
- обеспечивается ли в рамках информационной безопасности должный уровень конфиденциальности, целостности и доступности.

Внутри каждого из доменов СОВИТ выделяет список используемых процессов. Они могут применяться для проверки полноты отдельных видов деятельности, но не обязательно всех без исключения: считается достаточным выделение ключевых видов деятельности и их комбинирование в соответствии с требованиями конкретной организации.

Каждый из процессов имеет свое описание, набор целей контроля и показателей эффективности.

Таким образом, процессы управляют ИТ-ресурсами для достижения целей, соответствующих

бизнес-целям корпоративного уровня, что обеспечивает поддержку реализации корпоративной стратегии.

3. Интеграции преимуществ IT BSC и стандарта COBIT для построения иерархической структуры признаков (показателей) оценки эффективности ИТ-решений

При выявлении признаков (показателей) оценки эффективности ИТ-решений внутри каждой из перспектив IT BSC каждый домен стандарта COBIT рассматривается отдельно (табл. 2).

Таблица 2.

Структура системы показателей для выявления показателей эффективности стратегических решений ИТ-службы

Планирование и организация (PO)				
	Вклад в бизнес	Пользовательская перспектива	Качество эксплуатации	Долгосрочная ориентация
PO 1.				
...				
PO 10.				
Приобретение и внедрение (AI)				
	Вклад в бизнес	Пользовательская перспектива	Качество эксплуатации	Долгосрочная ориентация
AI 1.				
...				
AI 7.				
Эксплуатация и сопровождение (DS)				
	Вклад в бизнес	Пользовательская перспектива	Качество эксплуатации	Долгосрочная ориентация
DS 1.				
...				
DS 13.				
Мониторинг и оценка (ME)				
	Вклад в бизнес	Пользовательская перспектива	Качество эксплуатации	Долгосрочная ориентация
ME 1.				
...				
ME 4.				

Ввиду большой размерности полностью заполненной табл. 2 приведем ее фрагмент (табл. 3). В качестве примера показатели табл. 3 приведены для домена «Планирование и организация» в разрезе перспектив «Пользовательская перспектива» и «Качество эксплуатации».

Таблица 3.

Показатели оценки эффективности стратегических решений ИТ-службы (домен планирование и организация)

	Пользовательская перспектива	Качество эксплуатации
PO 1. Разработка стратегического плана развития ИТ	Доля заседаний по обсуждению планов ИТ, в которых принимали активное участие представители бизнеса	Доля тактических планов ИТ, соответствующих ранее определенной структуре
	Уровень корпоративной удовлетворенности текущим состоянием проекта и портфеля приложений	Доля в портфеле ИТ-проектов, зафиксированных в тактических планах ИТ
	Степень удовлетворенности владельцев бизнеса стратегическими и тактическими планами развития ИТ	Степень соответствия нормативным и бизнес-требованиям
PO 2. Определение информационной архитектуры	Уровень участия пользователей	Доля элементов данных, не являющихся частью корпоративной модели данных
	Доля удовлетворенных пользователей моделью данных (дружествен ли по отношению к пользователям справочник данных)	Доля приложений, не соответствующих информационной архитектуре

При формировании признаков оценки вариантов решений каждой конкретной задачи следует учитывать, на каких этапах деятельности принимаемое решение будет считаться значимым, т.е. процессы каких доменов включает в себя данная задача и, соответственно, какие признаки могут быть использованы для оценки вариантов решения в разрезе четырех перспектив.

Таким образом, целью интеграции методологии IT BSC и стандарта COBIT является получение иерархической структуры признаков (показателей) оценки эффективности ИТ-решений. В качестве верхнего уровня иерархии признаков используются перспективы, определенные в IT BSC: вклад в бизнес, пользовательская перспектива, качество эксплуатации и долгосрочная ориентация. При выявлении признаков следующего уровня иерархии определяется, процессы каких доменов и соответствующие им показатели эффективности ИТ включает в себя задача в каждой из перспектив.

Выборка признаков оценки вариантов решения для различных задач осуществляется на основе заполненной таблицы 2.

4. Выявление признаков (показателей) оценки эффективности вариантов решения задачи выбора стратегии внедрения информационной системы

В качестве примера рассмотрим процедуру выявления признаков для оценки вариантов решения конкретной задачи — задачи выбора стратегии внедрения информационной системы (ИС).

Стратегия внедрения определяет общий характер организации проектов по автоматизации, нацеленной на достижение стратегических бизнес-целей компании.

Возможны следующие варианты стратегии:

- ◆ внедрение системы на основе существующих бизнес-процессов;
- ◆ оптимизация бизнес-процессов предприятия, внедрение адаптированной под них системы;
- ◆ адаптация процессов предприятия под стандартные процессы системы.

Первый вариант предполагает, что предприятие использует уже отлаженные бизнес-процессы и существующая практика просто переводится на платформу новой информационной системы. Поскольку существенного вмешательства в деятельность компании не происходит, риски такого проекта невысоки, но и эффективность от внедрения мала, улучшения незначительны. Такой вариант может быть использован в том случае, если в рамках данной компании существующая практика бизнеса вполне приемлема, а последующая постепенная оптимизация процессов предусматривается в рамках уже внедренной информационной системы.

В случае оптимизации бизнес-процессов и внедрения адаптированной под них системы существуют риски, связанные с изменением процессов предприятия. Эти риски, вызывающие необходимость доработки и изменения системы, как правило, проявляются на этапе эксплуатации. Также нередко появляется необходимость привлечения специалистов узкого профиля для доработки системы. Таким образом, повышаются требования к качеству сопровождения ИТ-сервисов компании, а эффективность внедрения зависит от качества проведенной оптимизации бизнес-процессов.

При адаптации процессов предприятия под стандартные процессы информационной системы деятельность компании изменяется в угоду шаблонам или стандартам внедряемого программного обеспечения, что, в свою очередь, снижает риски эксплуатации системы. Внедряемая система остается

стандартной, что упрощает обновления, требует от специалистов знаний только стандартной функциональности, способствует сокращению штата специалистов, поддерживающих систему, а также снижает затраты на поддержку. Компания, внедряющая ИС, как правило, предлагает свое видение лучших мировых практик, а заказчик, ориентируясь на них, организует свою деятельность в соответствии с предложенными моделями. Эффективность такого подхода зависит от того, насколько выбранная информационная система оказалась подходящей для предприятия.

Задача выбора стратегии внедрения ИС затрагивает процессы следующих трех доменов стандарта COBIT:

- ◆ «Планирование и организация»;
- ◆ «Приобретение и внедрение»;
- ◆ «Эксплуатация и сопровождение».

Для формирования признаков оценки вариантов стратегии внедрения информационной системы соответствующие показатели выбираются в рамках трех указанных доменов по каждой из перспектив ИТ BSC из заполненной *таблицы 2*.

Перспектива «Вклад в бизнес» оценивает затраты компании на ИТ, нацелена на получение наибольшей ценности для бизнеса за счет инвестиций в ИТ. Это точка зрения менеджмента, который берет на себя контроль расходов на ИТ для обеспечения новых возможностей для бизнеса.

В рамках данной перспективы лиц, принимающих решения (ЛПР), интересует, насколько результативны инвестиции для реализации определенной стратегии внедрения. Необходимо учесть как изначальную стоимость приобретения программного обеспечения, так и дальнейшие расходы на поддержку и приобретение сопутствующих услуг. В зависимости от выбранной стратегии возникают разного рода и уровня риски, на управление которыми необходимо предусмотреть некоторую часть бюджета. Компанию необходимо снабжать пользовательской и обучающей документацией, стоимость которой может возрасти при условии необходимости ее дополнительной разработки.

Соответственно, в рамках домена «Планирование и организация» могут быть выбраны следующие признаки:

- ◆ доля инвестиций в ИТ, приносящих заранее запланированные результаты (РО 5. Управление ИТ инвестициями, сокращенно — «Результативные ИТ-инвестиции»);
- ◆ доля бюджета на ИТ, затраченного на меро-

приятия по управлению рисками (РО 9. Оценка и управление ИТ рисками, сокращенно – «Управление рисками»);

В рамках домена «Приобретение и внедрение» могут быть выбраны следующие признаки:

◆ уменьшение стоимости разработки/поддержки пользовательской документации, операционных процедур и обучающих материалов (AI 4. Обеспечение выполнения операций, сокращенно – «Разработка документации»);

◆ снижение стоимости приобретаемых товаров и услуг в расчете на единицу продукции (AI 5. Поставки ИТ ресурсов, сокращенно – «Приобретение товаров и услуг»);

В рамках домена «Эксплуатация и сопровождение» могут быть выбраны следующие признаки:

● затраты на отдельную услугу в течение определенного периода времени (DS 6. Определение и распределение затрат, сокращенно – «Затраты на услуги»).

Пользовательская перспектива показывает, как пользователи реагируют на функционирование ИТ. Следует учитывать как удовлетворенность пользователей функциональностью и качеством, так и улучшения в работе сотрудников ИТ и наличие общего осязаемого эффекта от проведенного внедрения системы.

Признаки данной перспективы с учетом всех доменов:

■ доля удовлетворенных работой сотрудников ИТ («Удовлетворенность ИТ сотрудников»);

■ доля заинтересованных сторон, удовлетворенных качеством ИТ («Качество ИТ»);

■ доля пользователей, удовлетворенных реализованной функциональностью («Функциональность»);

■ доля увеличения удовлетворенности пользователей от внедрения услуг, систем и новых технологий («Результаты внедрения»).

Перспектива «Качество эксплуатации» отражает отлаженность информационных процессов, применяемых при разработке и внедрении программ, и включает в себя поддержку качества операций и ИТ-разработок. Здесь можно уделить внимание соответствию процессов нормативным и бизнес-требованиям, а также уровню интеграции с бизнес-процессами компании. Также имеет смысл оценить усилия по обеспечению и поддержке функциональности внедренной системы, учесть ошибки при

транзакциях и неполадки, выявленные в ходе оценки рисков.

Признаки данной перспективы с учетом всех доменов:

◆ степень соответствия нормативным и бизнес-требованиям («Соответствие требованиям»);

◆ доля выявленных ИТ-инцидентов в ходе оценки рисков («Результаты оценки рисков»);

◆ доля мероприятий, направленных на поддержку существующих приложений («Поддержка системы»);

◆ оценка усилий по программированию, направленных на обеспечение функциональности («Обеспечение функциональности»);

◆ число приложений, в которых ИТ-процедуры гармонично интегрированы с бизнес-процессами («Интеграция с бизнес-процессами»);

◆ доля ошибок при транзакциях («Ошибки при транзакциях»).

Перспектива «Долгосрочная ориентация» подразумевает учет ресурсов, необходимых ИТ для поддержки оказания услуг и решения проблем в будущем. Включает в себя обучение и проверку качества подготовки ИТ-персонала, исследование смежных технологий и обновление имеющегося программного обеспечения.

В данном случае используются показатели численности сертифицированного и компетентного персонала, текучки кадров в службе ИТ, а также функциональности, по которой проводится обучение, и количества времени, запланированного на проведение обучения.

Признаки данной перспективы с учетом всех доменов:

◆ доля сертифицированного ИТ-персонала («Сертифицированный ИТ-персонал»);

◆ доля сотрудников ИТ, компетентность которых соответствует должностным требованиям, определенной стратегией («Соответствие компетенций требованиям»);

◆ показатель текучести кадров ИТ («Текучесть кадров ИТ»);

◆ число приложений, подкрепленных адекватным обучением пользователей и обслуживающего персонала («Функциональность, подкрепленная обучением»);

◆ число часов, затраченных на обучение персонала в год («Часы на обучение»).

Иерархическая структура задачи выбора страте-

гии внедрения информационной системы представлена на *рисунке 1*.

5. Выбор эффективной стратегии внедрения информационной системы

Для получения решения описанной выше задачи выбора стратегии внедрения информационной системы может быть использована система поддержки принятия решений (СППР) Expert Choice (www.expertchoice.com). Параметры задачи принятия решений, реализованные в системе Expert Choice, приведены на *рис. 2*. В левом окне экранной формы указаны коэффициенты относительной значимости признаков оценки альтернатив, а в правом верхнем окне – коэффициенты относительной значимости вариантов решения, позволяющие сделать вывод, что наиболее предпочтительным является вариант «Адаптация процессов предприятия под стандартные процессы системы».

Данный результат получен при задании экспертом оценок предпочтительности для каждой пары альтернатив по каждому признаку. При этом предварительно оценивается относительная значимость каждой пары признаков. При изменении исходных оценок может измениться и результат решения задачи.

6. Заключение

В данной работе предложен подход для выявления признаков оценки стратегических решений службы информационных технологий, позволяющий сделать процесс принятия решений более качественным, надежным и учитывающим особенности предметной области.

В качестве методологических предпосылок исследования были рассмотрены стандарт COBIT для аудита и управления ИТ и система сбалансированных показателей для ИТ-службы – IT BSC.

Целью интеграции IT BSC и стандарта COBIT является получение иерархической структуры признаков (показателей) оценки эффективности ИТ-решений.

В качестве верхнего уровня иерархии признаков используются перспективы, определенные в IT BSC: вклад в бизнес, пользовательская перспектива, качество эксплуатации и долгосрочная ориентация.

При выявлении признаков нижестоящих уровней иерархии определяется, процессы каких доменов и соответствующие им показатели эффективности ИТ включает в себя задача в каждой из перспектив.

В целях демонстрации подхода по выявлению признаков оценки альтернатив решена задача выбора стратегии внедрения информационной системы. С использованием СППР Expert Choice была

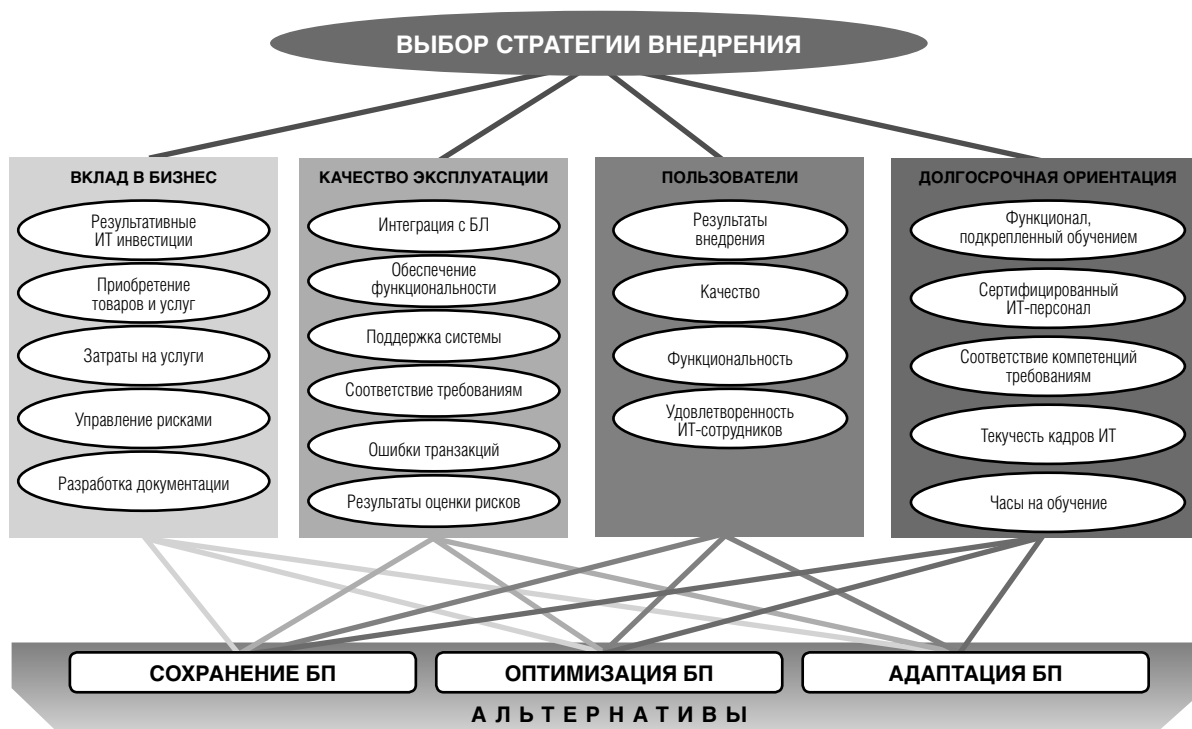


Рис. 1. Иерархическая структура задачи выбора стратегии внедрения информационной системы.

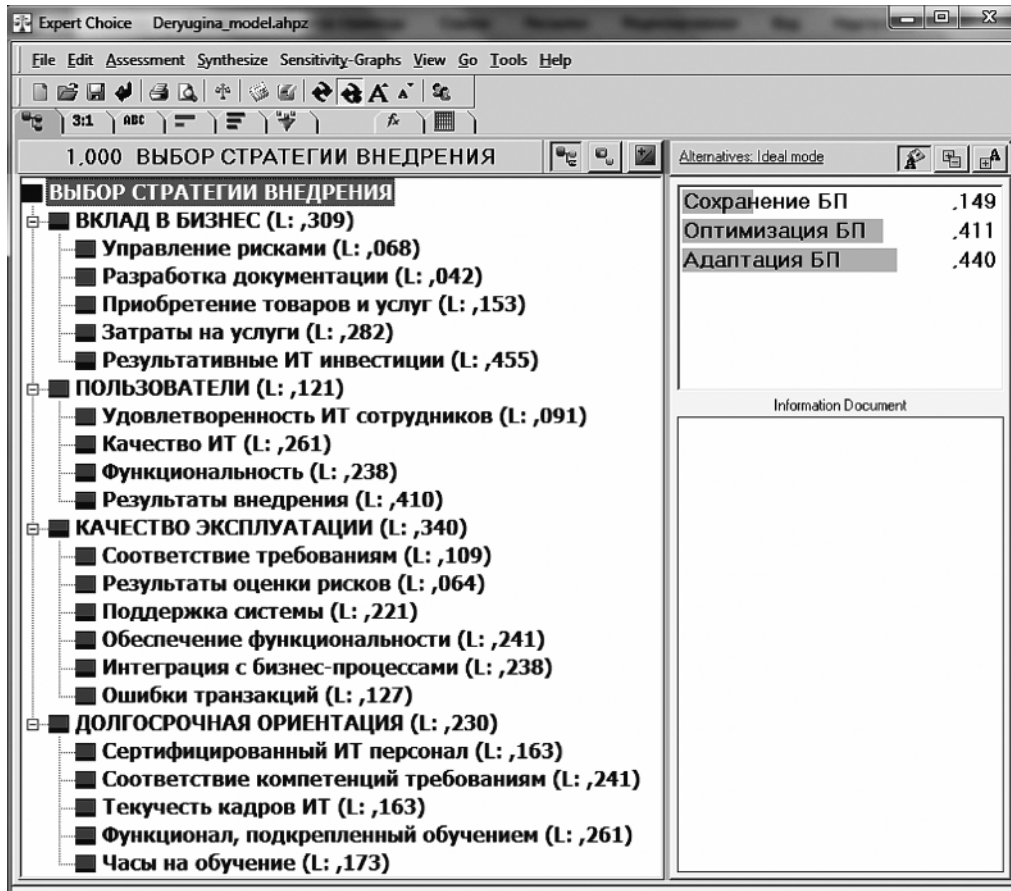


Рис. 2. Решение задачи в СППР Expert Choice.

произведена оценка относительной значимости системы признаков и выявлены предпочтения альтернативных стратегий, к числу которых относятся: внедрение системы на основе существующих бизнес-процессов, оптимизация бизнес-процессов предприятия и внедрение адаптированной под них системы, адаптация процессов предприятия под стандартные процессы системы.

По результатам оценки наиболее предпочтитель-

ным является вариант решения, предусматривающий адаптацию бизнес-процессов предприятия под процессы информационной системы, основанные на лучших практиках ее внедрения.

Предложенный подход может быть использован для решения других стратегических задач деятельности ИТ-сервиса компании, поскольку позволяет существенно повысить качество принимаемых решений с учетом особенностей предметной области. ■

Литература

1. Каплан Р., Нортон Д. Сбалансированная система показателей. От стратегии к действию. — М.: Олимп-Бизнес, 2006.
2. Каплан Р., Нортон Д. Стратегические карты. Трансформация нематериальных активов в материальные результаты. — М.: Олимп-Бизнес, 2007.
3. Кравченко Т.К. Разработка системы ключевых показателей для оценки эффективности процессов службы информационных технологий // Информационные технологии, 2010. № 8. С. 48-53.
4. COBIT 4.1.— IT Governance Institute. 2007.
5. Saull R. The IT Balanced Scorecard — A roadmap to effective governance of a shared services IT organization // Information Systems Control Journal (previously IS Audit and Control Journal), Volume 2, 2000, pp. 31-38.
6. Van Grembergen W., Van Bruggen R. Measuring and Improving Corporate Information Technology Through the Balanced Scorecard Technique // The Fourth European Conference on the Evaluation of Information Technology, Delft University Press, 1997.

АНАЛИТИЧЕСКАЯ МОДЕЛЬ ОЦЕНКИ ПРОИЗВОДИТЕЛЬНОСТИ МНОГОПРОЦЕССОРНОЙ ОБРАБОТКИ ДАННЫХ ДЛЯ НАБОРА ПАРАЛЛЕЛЬНЫХ АЛГОРИТМИЧЕСКИХ СТРУКТУР

К.В. Герценбергер,

младший научный сотрудник, Объединенный институт ядерных исследований,
г. Дубна

Е.В. Чепин,

кандидат технических наук, доцент Национального исследовательского
ядерного университета (МИФИ), Московский физико-технический институт
(государственный университет) (МФТИ)

Адрес: Московская область, г. Дубна, ул. Жолио-Кюри, д. 6

E-mail: k.gertsenberger@gmail.com, evchepin@gmail.com

Эффективность параллельной обработки данных напрямую зависит от алгоритма распараллеливания и типа аппаратной архитектуры. Важно, чтобы оценка прогнозируемой эффективности распараллеливания задачи для конкретной аппаратной платформы происходила на наиболее раннем этапе разработки. Для этой цели предлагается разработанная аналитическая модель предсказания производительности будущей программно-аппаратной системы для трех основных параллельных алгоритмических структур – алгоритмов распараллеливания.

Ключевые слова: аналитическая модель, предсказание производительности, распределенные вычисления, параллельная аппаратная архитектура, параллельная алгоритмическая структура, параллельное программное обеспечение.

1. Введение

Традиционные низкоуровневые методы и инструменты параллельного программирования зачастую приводят к сложному, непереносимому, немасштабируемому программному коду, делают задачу создания программ узкоспециализированной, нетривиальной, трудоемкой с

большими временными затратами, не защищают как от программных ошибок, так и низкой эффективности. В процессе развития программирования уровень абстракции неуклонно повышается. В настоящее время активно ведутся исследования в области создания высокоуровневых инструментальных средств параллельного программирования.

В контексте повсеместного распространения многоядерных процессоров и многопроцессорных систем, использование разработанной авторами статьи высокоуровневой методики, основанной на параллельных алгоритмических структурах, автоматизирующей основные этапы и позволяющей разработчику с невысокими знаниями параллельного программирования быстро реализовывать прикладные программы, успешно функционирующие в многопроцессорной среде, является крайне актуальным. Предложенный подход, представленный в работе [4], содержит следующие этапы разработки (курсивом выделены этапы, выполняющиеся полностью автоматически) параллельного программного обеспечения:

1. Совместное задание схемы многопроцессорной/многоядерной системы и цепочки функций обработки данных.

2. Выбор параллельных алгоритмических структур из набора для распараллеливаемых функций в соответствии с логикой их работы и зависимостью по данным.

3. *Проведение аналитического и/или имитационного моделирования планируемой системы.* В случае неудовлетворительных модельных характеристик или поиска оптимального решения корректировка схемы многопроцессорной системы, параллельного алгоритма и проведение нового моделирования.

4. Сопоставление исходного кода функций с блоками выбранных параллельных алгоритмов.

5. *Проведение автоматической генерации параллельного кода функций.*

6. *Запуск полученного программного обеспечения на реальной системе при помощи планировщика и оценка эффективности решения.*

В данном подходе разработчикам необходимо задать схему аппаратной и функциональной части, сопоставить параллельные алгоритмические структуры из заданного набора с распараллеливаемыми функциями, а остальные этапы автоматизированы. Такой подход освобождает программистов от разработки параллельных алгоритмов, выбора инструментария для используемой аппаратной платформы, написания параллельного исходного кода, отладки и ручного таймирования реализованной системы, сложного запуска на многопроцессорной системе.

Как и для других подходов разработки параллельного программного обеспечения, одним из

наиболее важных моментов является начальный этап предсказания эффективности планируемой программно-аппаратной связки. Точный метод предсказания дает возможность проектировщику оценить производительность параллельного алгоритма до реализации и исследовать эффективность при выборе альтернативного алгоритма или изменения параметров аппаратной части, например, количества процессоров и пропускной способности сети, может помочь в поиске точек, критичных для производительности.

На данный момент наиболее распространённым подходом при проектировании параллельных платформ является использование экспертных оценок. Этот подход, безусловно, позволяет в некоторой степени оптимизировать структуру проектируемых систем. Однако решения носят субъективный характер, поэтому для сложных распределённых систем полученные оценки и решения дают значительные отклонения от оптимальных значений. В случае ошибки выбора аппаратной платформы разработчикам приходится закупать другое оборудование, в случае неэффективного алгоритма распараллеливания – корректировать алгоритмическую часть и проводить разработку заново.

При проектировании и оптимизации вычислительных систем часто используют математическое моделирование. Моделирование планируемой системы дает возможность оценить максимальную производительность при имеющихся ресурсах, найти критичные точки аппаратной архитектуры, избежать ошибок проектирования, основанных на экспертной оценке. В ходе реализации методики, основанной на параллельных алгоритмических структурах, были разработаны аналитическая и имитационная модель параллельной обработки на многопроцессорной (многоядерной) системе.

Аналитическое моделирование предлагает более простой и формализованный подход для представления системы, более приспособлено к поиску оптимального решения. Имитационная модель, в свою очередь, не требует такого числа упрощений и допущений, позволяет получить более точную оценку работы сложной системы как единого целого. В данной статье представлена разработанная аналитическая модель. В качестве основного результата модели выбраны следующие характеристики: коэффициент ускорения, эффективность и масштабируемость. Так как представленные характеристики зависят от параллельного алгоритма, то

в статье проводится оценка эффективности параллельной обработки для каждого класса параллельных алгоритмов.

2. Основные типы параллельных алгоритмических структур

Разработчики в большинстве случаев используют существующие алгоритмы распараллеливания, которые давно известны, или их незначительные модификации. В связи с этим использование готового, но расширяемого набора параллельных алгоритмических структур для автоматизации является крайне актуальным и перспективным на сегодняшний день. В ходе анализа существующих алгоритмов распараллеливания был сформирован следующий набор параллельных алгоритмических структур для аналитической модели:

1. Точечный алгоритм реализует операцию над однородным множеством величин. Функция, применяемая к элементу множества, не зависит от остальных элементов. Неперекрывающиеся сегменты данных в этом случае распределяются по процессорам и независимо обрабатываются.

2. Локальный алгоритм (функция, применяемая к элементу множества, зависит от нескольких соседних элементов) аналогичен точечному алгоритму с тем отличием, что множество данных разбивается на перекрывающиеся фрагменты [2].

3. В алгоритме параллельной конвейеризации каждая ступень конвейера создается на отдельном процессоре, который производит обработку данных, полученных с предыдущей ступени конвейера. Конвейерная обработка сама по себе не обладает масштабируемостью, так как число ступеней фиксировано, однако в ряде случаев позволяет ускорить обработку для плохо распараллеливаемых задач.

4. Редукция использует ассоциативную операцию ко всем элементам множества, сводя множество к одному элементу. Алгоритм параллельной редукции заключается в распределении по процессорам неперекрывающихся сегментов данных, независимо выполнению редукции до одного элемента и сборке результатов для редукции на управляющем процессе (потоке).

5. Парадигма «разделяй-и-властвуй» осуществляет рекурсивную декомпозицию задачи на две или более подзадачи того же типа, но меньшего размера, с последующим комбинированием частичных

результатов в конечный результат.

6. Параллельный алгоритм волновой схемы заключается в вычислении по каждому направлению волны одновременно различными процессорами. Зачастую данный алгоритм также сводится к параллельной рекурсивной схеме.

Предложенный набор содержит подходы распараллеливания как по данным (например, локальный алгоритм, параллельная редукция), так и по задачам (например, конвейер). Очевидно, что список не является полным, в будущем планируется его расширение новыми элементами. Стоит отметить, что, например, в области обработки изображений и сигналов он охватывает больше 80 % операций. Однако, первые три алгоритма из данного списка наиболее часто, на наш взгляд, встречаются в практических задачах, связанных с моделированием по методу Монте-Карло, с оптимизацией на основе генетических алгоритмов, с задачами обучения и распознавания. Примеры задач из различных прикладных областей, для которых целесообразно использование таких моделей для распараллеливания ПО, можно найти в статьях [6-8].

3. Аналитическая модель предсказания производительности

Для построения моделей параллельной обработки на многопроцессорной системе используется тот факт, что любая аппаратная схема сводится к набору вычислительных узлов, выполняющих параллельную обработку данных, хранилища данных, а также коммутативной среды, связывающей вычислительные узлы между собой и хранилищем данных. В аналитической модели производится замена схемы гетерогенной сети аналогичной по структуре гомогенной сетью, у которой все вычислительные узлы и коммуникационные средства равны по производительности самому медленному узлу и самой медленной связи исходной схемы. В силу симметричности выбранных параллельных алгоритмических структур и закона Амдала в данном случае такие схемы эквивалентны для аналитического расчета. Исторически сложилось так, что время чтения/записи с жесткого диска заметно превышает время доступа к оперативной памяти, поэтому времена работы с оперативной памятью и кэшем не учитываются.

Выполнение обработки на многопроцессорной системе определяется параллельной алгоритми-

ческой структурой, поэтому анализируя все множество алгоритмов, можно построить модель для предсказания производительности, результатом работы которой будут выбранные характеристики. Используя формулу коэффициента ускорения можно спрогнозировать масштабируемость заданной системы следующим способом: для каждого числа процессоров от 1 до p рассчитывается соответствующий коэффициент ускорения. По полученной зависимости ускорения от числа процессоров аппаратной платформы можно оценить масштабируемость параллельной обработки на выбранной системе. Ниже подробно рассмотрены аналитические модели первых трех типов алгоритмов из приведенного выше списка.

3.1. Точечный алгоритм распараллеливания

а) Многопроцессорная система с распределенной памятью.

В структуре алгоритма можно выделить три этапа: чтение данных, независимая обработка сегментов на p -процессорах и запись полученного результата. Если обозначить количество узлов за P_{node} , среднюю скорость линейного чтения/записи носителя данных – W , пропускную способность сети – B , объем обрабатываемых данных – n , а T_1 – время обработки на одном процессоре, то можно математически представить коэффициент ускорения и эффективности для каждой параллельной алгоритмической структуры. Время последовательной обработки определяется следующим выражением:

$$T_{посл}(n) = T_{посл.чтение}(n) + T_1(n) + T_{посл.запись}(n), \quad (1)$$

$$\text{где } T_{посл.чтение}(n) = T_{посл.запись}(n) = n / W.$$

Таким образом, время последовательного выполнения функции можно представить следующим образом:

$$T_{посл}(n) = 2 \cdot n / W + T_1(n). \quad (2)$$

Согласно точечному алгоритму распараллеливания, время независимой обработки уменьшится в P_{node} раз и составит $T_1(n)/P_{node}$.

Скорость доступа к данным определяется соотношением пропускной способности сети и скоростью чтения/записи носителя информации. В случае, если пропускная способность сети выше скорости линейного чтения/записи, то производительность определяется медленным элементом:

временем линейного чтения/записи W . В противном случае, носитель информации успевает считывать и передавать в сеть данные, пока последняя передает их рабочему узлу, поэтому пропускная способность сети B является определяющей. Таким образом, на первом шаге анализируется, какой компонент чтения/записи более медленный: сеть или носитель информации. В качестве скорости чтения/записи данных S выбирается минимальная характеристика: либо пропускная способность сети, либо средняя скорость линейного чтения/записи.

Время чтения определяется по приходу всех байт на все процессоры и равно:

$$T_{пар.чтения}(n) = T_{задержка1} + \frac{n}{S} \approx \frac{n}{S}, \quad (3)$$

где $T_{задержка1}$ – время, через которое начинается передача первого байта данных. Это время мало по сравнению со временем чтения всего объема данных, поэтому в дальнейшем подобные характеристики учитываться не будут.

Чтение байт массива данных процессорами происходит последовательно (выбрано для снижения числа коллизий). К моменту окончания обработки данных процессором, который получил байты последним, процессоры, принявшие свой сегмент информации раньше, в связи с тем, что чтение и запись происходят примерно с одной скоростью, уже успеют отослать свои результаты. Таким образом,

$$T_{пар.записи}(n) = \frac{n}{P_{node} \cdot S}. \quad (4)$$

Подставив полученные выражения в формулу коэффициента ускорения, получается:

$$K_p(n) = \frac{2 \cdot \frac{n}{W} + T_1(n)}{\frac{n}{S} + \frac{n}{S \cdot P_{node}} + \frac{T_1(n)}{P_{node}}} = \frac{S \cdot P_{node} \cdot (2 \cdot \frac{n}{W} + T_1)}{n \cdot (P_{node} + 1) + S \cdot T_1}. \quad (5)$$

Видно, что при неограниченном росте числа процессоров коэффициент ускорения асимптотически стремится к величине

$$S \cdot \left(\frac{2}{W} + \frac{T_1}{n} \right).$$

Оценить масштабируемость системы можно следующим образом: пока коэффициент ускорения увеличивается, система масштабируется. Как

только ускорение начнет уменьшаться, масштабируемость перестает иметь место. Таким образом, определив число процессоров, соответствующее максимуму коэффициента ускорения, можно оценить масштабируемость программно-аппаратной связи. Экстремум определяется приравнением первой производной коэффициента ускорения по числу процессоров к нулю:

$$\frac{dK_p}{dP} = 0.$$

В данном случае, из формулы видно, что коэффициент ускорения с ростом числа процессоров будет постоянно расти. С другой стороны, эффективность параллельного выполнения будет уменьшаться, поэтому имеет смысл увеличивать количество процессоров до тех пор, пока эффективность будет выше некоторого предела. Эффективность для локального алгоритма определяется выражением:

$$E_p(n) = \frac{K_p(n)}{P_{node}} = \frac{S \cdot (2 \cdot \frac{n}{W} + T_1)}{n \cdot (P_{node} + 1) + S \cdot T_1}. \quad (6)$$

Так как формулы коэффициента эффективности определяются путем простого деления ускорения на число вычислительных узлов или процессоров, поэтому далее они приводиться не будут.

Иногда при распараллеливании обработки данных используют подход, в котором запись осуществляется не параллельно, а обработанные данные передаются управляющему процессу, который производит последовательную запись результата в файл. В этом случае время параллельной записи увеличивается, но отсутствует необходимость синхронизации записи в один файл. Для точечного алгоритма на многопроцессорной системе с распределенной памяти время параллельной записи увеличится до величины

$$\frac{n}{P_{node} \cdot B} + \frac{n}{W_s},$$

где W_s – пропускная способность канала для записи в файл. Если файл сохраняется на узле, собирающем локальные данные, то $W_s = W$, в противном случае, $W_s = S$. Отсюда, для приведенного примера коэффициент ускорения составит:

$$K_p(n) = \frac{2 \cdot \frac{n}{W} + T_1(n)}{\frac{n}{S \cdot P_{node} \cdot B} + \frac{n}{W_s} + \frac{T_1(n)}{P_{node}}} = \frac{S \cdot P_{node} \cdot (2 \cdot \frac{n}{W} + T_1)}{n \cdot (P_{node} + \frac{S}{B} + \frac{P_{node} \cdot S}{W_s}) + S \cdot T_1}. \quad (7)$$

б) Многопроцессорная система с разделяемой памятью.

Расчет модели многопроцессорной системы с разделяемой памятью отличается от первого варианта отсутствием сетевого соединения между узлами и носителем данных:

$$K_p(n) = \frac{P \cdot (2 \cdot n + W \cdot T_1)}{n \cdot (P + 1) + W \cdot T_1}. \quad (8)$$

3.2. Локальный алгоритм

Время последовательного выполнения аналогично определяется выражением (2).

а) Многопроцессорная система с распределенной памятью.

Время параллельного чтения увеличивается на время, необходимое для дополнительного получения процессорами перекрывающихся частей. Возможно перекрытие слева, справа и с обеих сторон, поэтому время параллельного чтения определяется следующим выражением:

$$T_{нар.чтения}(n) = \frac{n + (P_{node} - 1) \cdot (L_l + L_r)}{S}, \quad (9)$$

где L_l – длина перекрытия слева, L_r – длина перекрытия справа (0, если нет). Если предположить, что время обработки прямо пропорционально длине обрабатываемого массива данных, то:

$$T_{нар.обработки}(n) = \frac{T_1(n)}{P_{node}} + T_1(n) \cdot \frac{(L_l + L_r)}{n}. \quad (10)$$

Время параллельной записи, как у точечного алгоритма, равно

$$\frac{n}{P_{node} \cdot S}.$$

Таким образом,

$$K_p(n) = \frac{S \cdot P_{node} \cdot (2 \cdot \frac{n}{W} + T_1)}{P_{node} \cdot n \cdot (1 + (P_{node} - 1) \cdot (M_l + M_r)) + S \cdot T_1 \cdot (1 + P_{node} \cdot (M_l + M_r)) + n}. \quad (11)$$

где $M_l = \frac{L_l}{n}$ – коэффициент перекрытия слева,

а $M_r = \frac{L_r}{n}$ – перекрытия справа.

б) аналогично для многопроцессорной системы с разделяемой памятью получается:

$$K_p(n) = \frac{P \cdot (2 \cdot n + T_l \cdot W)}{P \cdot n \cdot (1 + (P-1) \cdot (M_l + M_r)) + W \cdot T_l \cdot (1 + P \cdot (M_l + M_r)) + n} \quad (12)$$

3.3. Алгоритм параллельной конвейеризации

На рисунке 1 представлено обозначение ступеней конвейеров и массивов данных, пересылаемых ступенями.

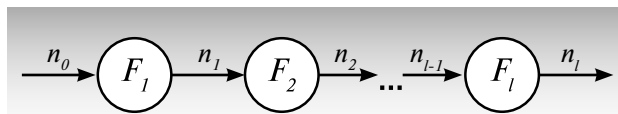


Рис. 1. Конвейерная обработка данных.

Время последовательной обработки данных размером N за $k = \frac{N}{n_0}$ операций можно представить следующим выражением:

$$T_{\text{посл.}}(N) = k \cdot (T_{\text{посл.чтение}}(n_0) + T_l(n_0) + T_{\text{посл.запись}}(n_l)). \quad (13)$$

Времена последовательного чтения и записи равны n_0/W и n_l/W соответственно.

$$T_{\text{посл.}}(N) = \frac{k \cdot n_0 + k \cdot n_l}{W} + k \cdot T_l(n_0). \quad (14)$$

а) Многопроцессорная система с распределенной памятью.

Для оценки эффективности алгоритма предполагается, что одновременно могут приниматься и передаваться данные. В противном случае, пропускная способность коммуникационной среды будет примерно в два раза ниже. Так как время конвейерной обработки k операций складывается из времени одного прохода по конвейеру и последующему $k-1$ срабатыванию самой медленной ступени конвейера, тогда:

$$T_{\text{нар.}}(N) = T_l(n_0) + \frac{(k-1) \cdot n_{x-1} + \sum_{m=0}^l n_m}{S} + (k-1) \cdot T_{fx}, \quad (15)$$

где x – номер самой медленной ступени конвейера, то есть

$$T_{fx} + \frac{n_{x-1}}{S} = \max(T_{fi} + \frac{n_{i-1}}{S}), \quad i = 1, 2, \dots, l.$$

Подставив выражения в формулу коэффициента ускорения, получается:

$$K_p(n) = \frac{\frac{n_0 + n_l}{W} + T_l}{\frac{n_{x-1} - \frac{n_{x-1}}{k} + \sum_{m=0}^l \frac{n_m}{k}}{S} + T_{fx} + \frac{T_l - T_{fx}}{k}}. \quad (16)$$

б) Многопроцессорная система с разделяемой памятью.

В этом случае отсутствует задержка передачи по сети промежуточных данных, поэтому выражение (16) сводится к следующему:

$$K_p(n) = \frac{\frac{n_0 + n_l}{W} + T_l}{\frac{n_0 + n_l}{k \cdot W} + T_{fx} + \frac{T_l - T_{fx}}{k}}, \quad (17)$$

где $T_{fx} = \max(T_{fi} + \frac{n}{S}, T_{fi}, T_{fi} + \frac{n_l}{S}), \quad i = 2, 3, \dots, l-1.$

4. Пример сравнения результата моделирования с практическими значениями

Адекватность разработанной модели была практически проверена в ряде проектов для алгоритмов обработки изображений и сигналов. В работе [5] приводится сравнение модельных и практических значений коэффициента ускорения для КИХ-фильтра цифрового сигнала. КИХ-фильтр был реализован с помощью секционированной свертки и ее вычисления через быстрое преобразование Фурье. Эта операция по степени зависимости данных относится к локальной алгоритмической структуре. Для оценки эффективности алгоритма проводилось таймирование фильтра обработки сигнала размером 227 мегабайт на кластерной системе. Зависимость модельного и практического коэффициента ускорения от количества процессоров показало, что максимальное отклонение аналитической модели от практических значений составляет в данном случае 6,4 %.

5. Заключение

В данной статье кратко представлена методика автоматизированного получения параллельного программного обеспечения. Одним из ключевых моментов, как предложенной высокоуровневой методики, так и других инструментов разработки параллельных решений, должен являться этап предсказания производительности обработки данных для планируемой программно-аппаратной

системы. В данной статье представлена разработанная аналитическая модель для оценки эффективности программно-аппаратной платформы до ее реализации. Для каждой параллельной алгоритмической структуры показан вывод основных характеристик производительности. При добавлении нового параллельного алгоритма в набор необходимо разработать оценку в соответствии с приведенной схемой.

Разработанная аналитическая модель успешно применена в ряде проектов [1,3,5]. Прогнозирова-

ние эффективности программно-аппаратной связки позволило не только оценить ускорение и масштабируемость будущей системы до реализации, но и определить узкие места аппаратной платформы, сделать выбор оборудования до реализации. Максимальное отклонение результатов аналитической модели в проектах составило 11 %. В будущем планируется расширить аналитическую модель новыми параллельными алгоритмическими структурами, отказаться от ряда упрощений для более точного расчета характеристик производительности. ■

Литература

1. Богин И.В., Герценбергер К.В., Чепин Е.В. Реализация параллельной обработки цифровых изображений для библиотеки IPPLab MSCo. // Науч. сессия МИФИ-2006: Сб. науч. Тр. М.: МИФИ, 2004 - Т. 12 - С.139-140.
2. Герценбергер К.В. Методы параллельной обработки изображений и сигналов в зависимости от локальности вычислений // Сб. Тр. Седьмой международной научно-практической конференции «Исследование, разработка и применение высоких технологий в промышленности». Т. 2. СПб.: Изд-во Политехн. ун-та, 2009 – С.49-50.
3. Gertsenberger K.V., Chepin E.V. Implementation parallel processing of digital signals on cluster system // In: Proc. of the International Workshop on Computer Science and information technologies (CSIT'2007), Bashkortostan, 2007. Vol. 1. Ufa State Aviation Technical University, 2007 – P.152-155.
4. Gertsenberger K.V., Chepin E.V. Using a CASE-oriented approach for parallel software development. In: Proc. of the International Workshop on Computer Science and information technologies (CSIT'2008), Antalya, Turkey, 2008. Vol. 1. Ufa State Aviation Technical University, 2008 – P/63-68.
5. Герценбергер К.В., Чепин Е.В. Использование параллельных алгоритмических структур для автоматизации процесса разработки параллельного программного обеспечения. Труды МФТИ.- 2011 (находится в печати).
6. Ошмарин Д. В. Распределение канальных ресурсов в сетях когнитивного радио на основе теории игр. // Бизнес-информатика. - 2010. № 4. С. 38-45.
7. Голубев С. В. Распознавание структурированных документов на основе машинного обучения. // Бизнес-информатика. - 2011. № 2. С. 48-55.
8. Силантьев Д. А. Оценка необходимого размера свертки биометрического образца для обеспечения заданных параметров надежности биометрической системы идентификации. // Бизнес-информатика. - 2009. № 3. С. 21-23.

ПРИМЕНЕНИЕ АЛГОРИТМОВ ДВУМЕРНОГО АНАЛИЗА ГЛАВНЫХ КОМПОНЕНТ ДЛЯ ЗАДАЧ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ ЛИЦ

Н.Л. Щеголева,

кандидат технических наук, доцент кафедры математического обеспечения и применения ЭВМ Санкт-Петербургского государственного электротехнического университета (ЛЭТИ)

Г.А. Кухарев,

доктор технических наук, профессор кафедры математического обеспечения и применения ЭВМ Санкт-Петербургского государственного электротехнического университета (ЛЭТИ), профессор факультета информатики Западно-поморского технологического университета, Щецин, Польша (West Pomeranian University of Technology, Szczecin, Poland)

Адрес: г. Санкт-Петербург, ул. Профессора Попова, д. 5

E-mail: stil_hope@mail.ru, gkukharev@wi.zut.edu.pl

Представлены алгоритмы двумерного анализа главных компонент (Two-dimensional Principal Component Analysis – 2D PCA), ориентированные на обработку цифровых изображений больших размеров в условиях малой выборки. Рассматриваются два способа выполнения 2D PCA, соответствующие параллельной и каскадной формам его реализации. Оценены характеристики представленных алгоритмов. Обсуждаются возможности использования представленных алгоритмов в других областях.

Ключевые слова: двумерный анализ главных компонент, двумерное преобразование Карунена-Лоэва, распознавание изображений лиц, применение в бизнес-приложениях.

Введение

ХХI век ознаменовался переходом от простых информационных систем к использованию интеллектуальных сред — современного поколения вычислительных и информационных технологий. Примерами интеллектуальных сред являются:

- ◆ интернет, через который реализуются общение людей бизнес-сообщества (видеоконференции, консультации, почта), поиск информации и доступ к ней, интернет-маркетинг и интернет-магазины и т.д.;
- ◆ виртуальные предприятия, реализующие проекты по типу «start up», сообщества и ассоциации, реализующие обмен информацией, технологиями,

другими информационными и вычислительными ресурсами;

♦ высокоорганизованные интерфейсы систем «клиент – банк»;

♦ системы класса «Visitor Identification», реализующие распознавание или аутентификацию, например, постоянного покупателя магазина, гостя (посетителя) частного дома или офиса, пассажира международного транспорта и т.д;

♦ системы контроля доступа к информации (базам данных, корпоративным сетям), к средствам (техническим, финансовым), к помещениям, к ресурсам (например, вычислительным).

Перечисленные примеры имеют одну очень важную общую характеристику: здесь все задачи решаются в процессе непосредственного общения человека со средой (в режиме «on-line») и, что самое важное, – этим человеком может быть только зарегистрированный пользователь, подлинность личности которого проверяется на входе в интеллектуальную среду.

Сегодня, например, сложно представить себе успешного бизнесмена без ноутбука, планшетного компьютера, смартфона или коммуникатора. Более того, в ходе круглого стола «Электронный парламент: вопросы правового обеспечения», который прошел 14 апреля 2011 г. было принято решение об обеспечении всех депутатов Госдумы шестого созыва, которые будут избраны в декабре 2011 г., личными интернет-планшетами iPad. Вскоре подобная перспектива ждет и каждое региональное правительство. Все эти мобильные устройства хранят личную и деловую информацию, сведения, содержащие коммерческую тайну, а устройства депутатов Госдумы – сведения, представляющие государственную тайну. Очевидно, что доступ постороннего человека к этой информации может стать причиной материальных потерь и/или нанести ущерб безопасности государства. И здесь одной из наиболее актуальных проблем является проблема идентификации (аутентификации) конечного пользователя подобных устройств.

Задачи идентификации (аутентификации) пользователя успешно решаются в рамках биометрических технологий. Эти идеи отражает статья [1], в которой показано использование методов анализа главных компонент (Principal Component Analysis – PCA) в задачах распознавания изображений лиц.

Однако в современных условиях развития информационных технологий и инженерии программного

обеспечения можно предложить более рациональные алгоритмы для этого метода, позволяющие более детально анализировать многомерные данные, какими являются изображения. Именно эти рациональные алгоритмы и предлагаются в настоящей статье, что отличает ее от работы [1].

Проблемы, связанные с обработкой изображений методами PCA

В задачах обработки изображений методами PCA исходными являются три базовых параметра: $\{M, N, K\}$, где M и N – число строк и столбцов исходных изображений, а K – количество изображений [2 – 4]. Первые два параметра определяют размерность DIM исходного пространства признаков $DIM = MN$. При этом обработке изображений лиц методами PCA сопутствует проблема малой выборки (Small Sample Size – SSS), когда $DIM \gg K$. Размерность исходного пространства признаков DIM определяет вычислительную сложность реализуемых алгоритмов.

Классический метод PCA [2] включает два этапа.

Первый из них – анализ главных компонент, в рамках которого по всем исходным данным определяется матрица ковариации и для нее определяются наибольшие собственные значения (главные компоненты) и матрица собственных векторов (матрица проекции). Второй этап включает одномерное преобразование в собственном базисе (или преобразование Карунена-Ловэа/Karhunen-Loeva Transform/KLT), которое использует найденную матрицу проекции.

Результат KLT представляет исходные данные как их спектр в собственном базисе, поэтому для сохранения основной информации об исходных данных достаточно использовать только несколько главных спектральных компонент [2]. При этом достигается условие:

$$DIM_{new} \ll DIM, \quad (1)$$

где DIM_{new} – размерность нового пространства признаков.

Однако статистически устойчивое решение задачи PCA/KLT в приложении к обработке изображений возможно только в случае отсутствия проблемы SSS и/или выполнения условия:

$$DIM = MN \ll K. \quad (2)$$

Варианты реализации PCA/KLT в условиях проблемы SSS показаны в работах [1, 3]. Они сводятся к предобработке исходных данных с целью уменьшения их размерности до достижения условия (2). Так в [3] решение основано на перемасштабировании каждого исходного изображения до размеров $m < M$ и $n < N$, при которых $DIM = mn \ll K$ и последующего прямого применения PCA/KLT. В [1] решение основано на декомпозиции всего набора исходных данных (общим размером $MN \times K$) в виде нескольких подматриц меньшего размера, для которых выполняется условие (2), нахождения частных решений PCA/KLT для этих подматриц и сведении частных решений к полному решению.

В настоящей статье представлены алгоритмы двумерного анализа главных компонент (2DPCA) и двумерного преобразования Карунена-Лоэва (2DKLT), специально предназначенные для обработки изображений больших размеров в условиях SSS и не требующие предобработки исходных данных с целью достижения условия (2) [4 – 6].

1. Идея и реализация 2D PCA\2D KLT

Пусть нам задан набор X , состоящий из X центрированных и нормированных к «1» изображений $X^{(k)}$ $k = 1, 2, \dots, K$ (для $k = 1, 2, \dots, K$) размером $M \times N$ каждое, и $MN \gg K$:

$$X = [X^{(1)} X^{(2)} \dots X^{(K)}] \quad (3)$$

Целью 2D PCA является определение двух матриц проекции, трансформирующих исходные данные в новое пространство признаков так, чтобы выполнялось условие минимума расстояния:

$$distance(X^{(k)} - W_1^T X^{(k)} W) \rightarrow \min, \forall k = 1, 2, \dots, K, \quad (4)$$

где W_1, W_2 – ортогональные матрицы порядков M и N соответственно.

Параллельный алгоритм. Здесь двумерный анализ главных компонент выполняется независимо по строкам и столбцам исходных данных, и при этом определяются *два независимых собственных базиса*.

Модель вычислений в 2D PCA\2D KLT можно представить следующим образом:

$$\begin{aligned} 1.0. X &\xrightarrow{R} X_1; \\ 1.1. X &\xrightarrow{PCAr} W_1; \end{aligned} \quad (5)$$

$$\begin{aligned} 1.2. X_1 &\xrightarrow{PCAr} W_2; \\ 2. Y^{(k)} &= W_1^T X^{(k)} W_2, \forall k=1, 2, \dots, K. \end{aligned}$$

На этапе 1.0 выполняется реорганизация структуры исходных данных (3). Это реализуется как транспонирование всех матриц размером $M \times N$ в наборе X и составление нового набора данных X_1 . На этапе 1.1 выполняется базовая процедура PCA относительно строк исходных данных (PCAr). На этапе 1.2 выполняется PCA относительно строк (PCAr) данных, входящих в набор X_1 , или, что тоже самое, – относительно столбцов исходных данных. Результат этапов 1.1 и 1.2 – вычисление двух матриц проекции W_1 и W_2 . Этапы 1.1 и 1.2 могут быть выполнены параллельно, поскольку процедура PCAr применяется для независимых наборов данных X и X_1 .

Наконец, этап 2 – это трансформация всех исходных изображений из набора X в новое пространство признаков Y . С учетом этого, на *рис. 1* представлена схема параллельного способа (5) реализации 2D PCA. Блок «R» реализует транспонирование всех матриц размером $M \times N$ в наборе X . Блоки, обозначенные как «⊗», реализуют процедуру проекции исходных данных. Пунктиром выделен блок, определяющий двумерное преобразование Карунена-Лоэва (2D KLT).

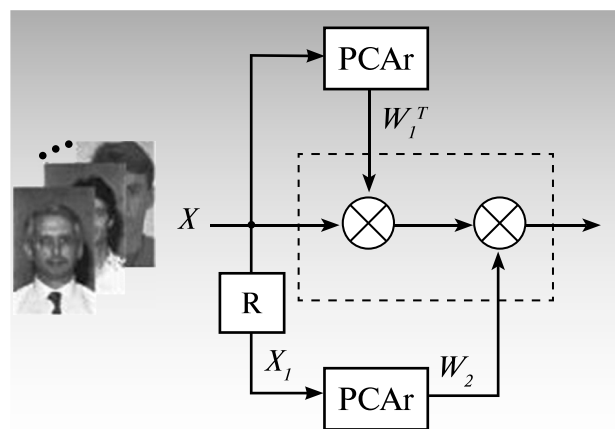


Рис. 1. Параллельный способ реализации 2D PCA.

Заметим, что в представленном методе двумерное 2D KLT реализуется по каскадной схеме: сначала исходное изображение обрабатывается по строкам, а затем полученный результат обрабатывается по столбцам (или наоборот). Это аналогично тому, как могут выполняться многие **двумерные** дискретные ортогональные преобразования изображений

[2], что, однако, противоречит способу получения собственных базисов в (5).

Каскадный алгоритм. Здесь в первом каскаде выполняется анализ главных компонент *по строкам* исходных изображений, а во втором – *по столбцам* результата первого каскада (или наоборот – сначала по столбцам, а потом по строкам результата). При этом определяются два *взаимосвязанных собственных базиса*, и 2D KLT реализуется также по каскадной схеме во взаимосвязанных базисах. Модель каскадного алгоритма запишем следующим образом:

$$X \xrightarrow{PCAr} \{W_1; X_1\} \xrightarrow{R} \tilde{X}_1 \xrightarrow{PCAr} \{W_2; \tilde{Y}\} \xrightarrow{R} Y. \quad (6)$$

Здесь: \tilde{X}_1 – данные X_1 после их реорганизации блоком «R»; выражения в фигурных скобках представляют результаты PCA\KLT по строкам и столбцам исходных изображений.

Схема вычислений, соответствующая модели (6), представлена на рис. 2. Здесь в явной форме отражен каскадный способ реализации 2D PCA\2D KLT.

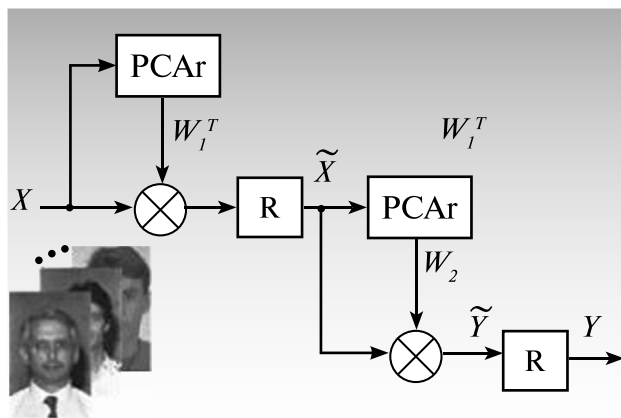


Рис. 2. Структура каскадного алгоритма 2D PCA.

2. Формальное описание алгоритмов

Версия параллельного алгоритма. Вычислим две матрицы ковариации, определив их относительно *строк* («r» – row) и *столбцов* («c» – column) для исходных изображений (3):

$$C^{(r)} = \sum_{k=1}^K X^{(k)} (X^{(k)})^T \quad \text{и} \quad C^{(c)} = \sum_{k=1}^K (X^{(k)})^T X^{(k)}, \quad \forall k \in K, \quad (7)$$

где $C^{(r)}$ и $C^{(c)}$ – матрицы порядков M и N соответственно.

С учетом (7) определим собственные значения и соответствующие им собственные векторы, решив следующие две задачи на собственные значения:

$$W_1^T C^{(r)} W_1 = \Lambda_1; \quad W_2^T C^{(c)} W_2 = \Lambda \quad (8)$$

При этом $W_1^T W_1 = I_1$ и $W_2^T W_2 = I_2$, а I_1 и I_2 – единичные матрицы порядков M и N соответственно; Λ_1 и Λ_2 – диагональные матрицы порядков M и N соответственно.

Трансформация исходных данных в новое пространство признаков реализуется как двумерное преобразование Карунена-Лоэва на основе собственных базисов, определяемых по (8) в следующей форме:

$$Y^{(k)} = (W_1^T X^{(k)}) W_2, \quad \forall k = 1, 2, \dots, K, \quad (9)$$

где $Y^{(k)}$ – матрицы размера $M \times N$, представляющие исходные изображения в новом пространстве признаков.

Версия каскадного алгоритма.

Выполним проекцию исходных данных так, что:

$$X_1^{(k)} = W_1^T X^{(k)}, \quad \forall k = 1, 2, \dots, K, \quad (10)$$

где $X_1^{(k)}$ – матрицы размера $M \times N$, представляющие исходные изображения (3) в новом (промежуточном) пространстве признаков; матрица W_1 была определена в (8).

Далее вычислим матрицу ковариации, определив ее относительно столбцов («c» – column) результата (10):

$$C^{(c)} = \sum_{k=1}^K (X_1^{(k)})^T X_1^{(k)}, \quad (11)$$

где матрица $C^{(c)}$ имеет порядок, равный N .

Вычислим для (11) собственные значения и соответствующие им собственные векторы, решив следующую задачу на собственные значения:

$$W_2^T C^{(c)} W_2 = \Lambda_2, \quad (12)$$

где W_2 и Λ_2 – матрицы порядка N .

Трансформация исходных данных (3) в новое пространство признаков реализуется теперь как двумерное преобразование Карунена-Лоэва (9) на основе собственных базисов, определяемых ортогональными матрицами проекции (8) и (12).

Редукция размерности пространства признаков

Изменим (9) так, чтобы в дальнейших преобразованиях участвовали только те собственные векторы, которые соответствуют « d » главным компонентам ($d < \min\{M, N\}$). Для этого, из выбранных собственных векторов формируются две прямоугольные матрицы: из матрицы W_1^T выбираем « d » строк, а из матриц W_2 выбираем « d » столбцов, соответствующих « d » наибольшему собственным числам, и на их основе формируются две матрицы редукции F_1 и F_2 . Верхние границы параметра « d » выбираются исходя из критерия энергетической значимости собственных значений [5, 6]. Нижние границы параметра « d » выбираются экспериментально с учетом критерия (4). При этом $d < \min\{M, N\}$ или $d_1 < M$; $d_2 < N$, если $d_1 \neq d_2$.

Теперь выполним «усеченное» двумерное преобразование Карунена-Лоэва:

$$\hat{Y}^{(k)} = F_1 X^{(k)} F_2, \text{ где } \hat{Y}^{(k)} = \begin{bmatrix} y(1,1) \dots y(d,1) \\ y(2,1) \dots y(d,2) \\ \vdots \dots \dots \\ y(d,1) \dots y(d,d) \end{bmatrix}^{(k)}. \quad (13)$$

Матрицы F_1 и F_2 имеют размеры $d \times M$ и $N \times d$ соответственно (или $d_1 \times M$ и $N \times d_2$ в общем случае); знак « $\hat{}$ » — определяет отличие результата (13) от (9).

Результирующие матрицы в (13) имеют в общем случае размеры $d_1 \times d_2$ и представляют исходные изображения в новом пространстве признаков.

3. Характеристики разработанных алгоритмов и их отличия от известных

3.1. На реализацию полного 2D KLT (9) потребуется $(NM^2 + MN^2)$ операций, а на реализацию «усеченного» 2D KLT по схеме (13) потребуется $(Md^2 + MNd)$ операций. При этом **сокращение вычислений** на этапе редукции размерности пространства признаков (РПП) можно приблизительно оценить как величину $(M + N)/d$, поскольку:

$$\frac{NM^2 + MN^2}{Md^2 + MNd} = \frac{MN(M + N)}{M(d + N)d} \approx \frac{M + N}{d}, \text{ для } d \ll N. \quad (14)$$

Например, для $M=112$ и $N=92$ (база лиц ORL [9])

и $d=10$ сокращение вычислений составит примерно 20 раз(!) на каждое изображение. С учетом параметра « K » — числа изображений, ускорение вычислений составит величину, равную примерно $K(M + N)/d$ на все исходные данные.

3.2. Результат в (13) содержит d^2 (или $d_1 \times d_2$) элементов, поэтому **степень редукции РПП** определяется соотношением MN/d^2 или $MN/(d_1 d_2)$, если $d_1 \neq d_2$.

Например, для $M=112$, $N=92$ и $d=10$ редукция размерности пространства признаков составит более чем 100 раз(!) на каждое изображение. **Таким образом, использование предложенных в статье алгоритмов ориентировано на сокращение ресурсных затрат в практических приложениях.**

3.3. Наибольший порядок матриц ковариации в (7), (12) составляет величину $DIM = \max\{M, N\}$. Это предопределяет практическую возможность решения задачи на собственные значения и стабильность этого решения даже для изображений больших размеров.

3.4. При представлении исходных изображений как совокупности строк и столбцов общее число получаемых векторов составляет величину $K(M + N)$, а размерность исходного пространства признаков определяется как $DIM = \max\{M, N\}$. Поэтому **при любых значениях M и N** соотношение «размерность/число векторов» всегда будет отвечать условию $\max\{M, N\} \ll K(M + N)$, «обходя» таким образом, проблему SSS.

4. Распознавание изображений лиц

Эксперименты выполнены на базе ORL [9], некоторые изображения которой показаны на рис. 3. База ORL охватывает 40 различных классов образов (изображений лиц людей различных наций), по 10 образов в каждом классе. Все изображения базы ORL относятся к типу «indoog», имеют размер 112 на 92 пикселя и представлены в шкале «GRAY» (256 уровней яркости).

Для лиц базы ORL характерно изменение: контраста изображений и яркости фона вокруг области лица; размера лиц, в том числе не пропорционально по осям X и Y ; поворота головы в пространстве 3D в пределах не менее $\pm 20^\circ$ по оси X , $\pm 30^\circ$ по оси Y и не менее $\pm 10^\circ$ по оси Z ; мимики лиц, прорисовки деталей лиц (их сглаживание).



Рис. 3. Примеры исходных данных.

В первом эксперименте решалась задача поиска изображений лиц по заданному тестовому образцу с помощью представленного в данной статье алгоритма 2D PCA. Эксперимент был выполнен для всех 40 классов с выбором только одного образа из каждого класса для эталона и девяти других образов в каждом классе для теста. Таким образом, в формулировке задачи поиска база системы поиска содержала 40 образов-эталонов (по одному на класс), а в качестве тестовых были использованы 360 образов (по девять образов на каждый из 40 классов). В эксперименте дважды использована процедура кросс-валидации: в каждом эксперименте изменялся номер выбранного образа-эталона, а внутри каждого эксперимента изменялся параметр редукции « d » ($d = 3, 4, \dots, 10$). Всего, таким образом, выполнено $10 \times 8 = 80$ экспериментов. Модель выполненного эксперимента имеет следующий вид [6]:

$ORL(40/1/9/CV)\{2DPCA:112 \times 92 \rightarrow d \times d / KMP/L1/rank = 1\}$,

где определены:

$ORL(40/1/9/CV)$ – база лиц, структура данных и процедура кросс-валидации (CV);

$2DPCA:112 \times 92 \rightarrow d \times d$ – использованный метод и размеры данных до и после редукции размерности пространства признаков;

$KMP/L1/rank = 1$ – тип классификатора, метрика и ранг результата.

Текущие результаты в каждом эксперименте оценивались как отношение количества правильно классифицированных тестовых образов к общему их числу 360. Общее число тестовых образов составило 3600 для каждого значения параметра « d ». Результат считался правильным, если на первом месте ($rank = 1$) в группе результатов находился образ из того же класса, что и тестовый образ.

В эксперименте использован классификатор по минимуму расстояния (KMP) в метрике L1.

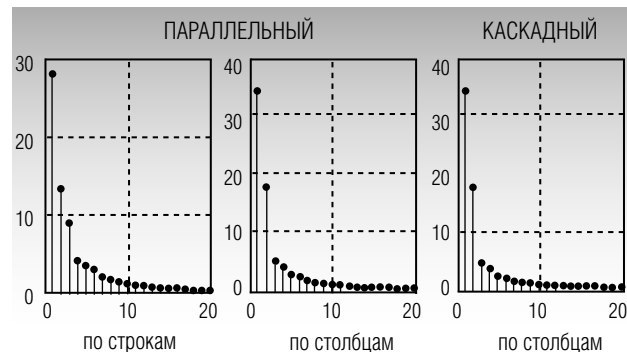


Рис. 4. Собственные значения для метода 2D PCA.

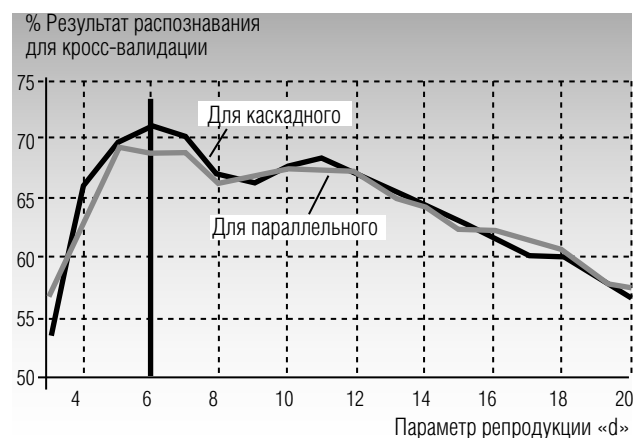


Рис. 5. К выбору параметра редукции « d ».

На рис. 4 показаны 20 наибольших собственных значений (собственные значения, вычисленные по строкам для обоих способов, – одинаковы по определению). При этом главные компоненты ограничены величиной $d \approx 6$, поскольку далее собственные значения практически равны «0».

На рис. 5 показаны результаты решения поставленной задачи поиска для различных значений параметра редукции « d ». Как видно, наилучший результат поиска получен при значении $d = 6$. Степень редукции размерности пространства признаков составила при этом значение, равное 286 (поскольку $112 \times 92 / 36 = 286$).

Подводя итоги эксперимента, необходимо отметить следующее:

♦ При выполнении поставленного эксперимента дважды решалась задача на собственные значения. Порядки матриц ковариации составили соответственно 112 и 92. Общее число «изображений» составило $40 \times (112 + 92)$, поэтому матрицы ковариации были хорошо обусловлены и не потребовалось никаких процедур их регуляризации (в подходе

«Eigenfaces» [1] порядок матрицы ковариации составил бы величину $112 \times 92 = 10304$, что сразу же определило бы невозможность ее дальнейшего использования);

♦ Ярко выраженное наличие «главных компонент» в составе собственных значений гарантирует репрезентативное представление исходных данных в новом пространстве признаков при малом значении параметра « d »;

♦ Решение поставленной задачи поиска выполнено успешно, особенно если учитывать специфику исходных данных – изображений лиц базы ORL и малое число образов-эталонов в системе поиска: по одному на каждый класс. Результаты, представленные в ходе экспериментов, выше результатов, показанных в [5], при решении аналогичной задачи.

Во втором эксперименте решалась задача распознавания изображений лиц при различном числе эталонов в базе системы распознавания (Face Recognition System – FaReS [6]). Модель выполненного эксперимента имеет следующий вид:

$$\text{ORL}(40/L/10-L/R+CV)\{2DPCA:12 \times 92 \rightarrow 8 \times 8\} \\ [KMP/L1/\text{rank} = 1].$$

В базу FaReS выбирается L тренировочных изображений в каждом из 40 классов; выбор организован случайным образом (random – R) и использована процедура кросс-валидации (R+CV). Число тестовых образов составляет $10 - L$ для $L = 1, 2, 3, 4, 5$. Результаты выполненных экспериментов представлены в *таблице 1*.

Таблица 1.

Результаты экспериментов

Э	1/40/360		2/80/320		3/120/280		4/160/240		5/200/200	
	П	К	П	К	П	К	П	К	П	К
1	68	68	78	79	93	93	94	94	97	97
2	68	68	78	78	88	89	90	90	95	96
3	70	70	82	82	90	90	94	94	95	94
4	66	68	83	84	91	90	93	93	95	93
5	71	71	83	83	90	90	91	90	92	91
6	71	71	83	83	90	90	91	92	95	95
7	70	70	87	87	91	92	93	93	96	97
8	64	64	90	90	91	91	94	95	95	95
9	67	67	82	81	89	88	90	89	94	95
10	66	68	82	82	89	89	92	92	96	96
ср	68	69	83	83	90	90	92	92	95	95

Здесь: «Э» – номер эксперимента;

1/40/360...5/200/200 – значение L и общее число изображений-эталонов и тестовых изображений в каждом эксперименте; П и К – способы: параллельный и каскадный.

Из представленных результатов видно, что существенный скачок результативности распознавания возникает уже при $L = 2$, а при значении $L = 5$ результат распознавания составляет (при округлении до наименьшего целого) в среднем 95% и в максимуме 97%. При этом результативность для двух алгоритмов реализации 2D PCA\2D KLT практически одинакова.

Эффективность применения 2D PCA\2D KLT для изображений тестовой базы FERET была показана нами в [10].

Заключение

Укажем наиболее значимые области, в которых применение алгоритмов 2D PCA/2D KLT может дать наиболее значимый эффект.

1. В [11] обосновывается необходимость применения свёртки наименьшего размера, позволяющей достигнуть наибольшей точности и производительности в процессе идентификации. Особенно актуально это для больших баз данных, хранящих информацию о биометрических характеристиках человека. Применение представленных в настоящей статье алгоритмов 2DPCA\2DKLT позволит создавать эффективные биометрические системы, обеспечивающие низкий уровень ошибок.

2. Алгоритмы 2D PCA можно использовать также для анализа и поиска закономерностей в экономических и финансовых данных, представленных многомерной структурой данных – многомерными наблюдениями. Одним из хорошо зарекомендовавших себя методов анализа и прогнозирования финансового рынка является метод «Гусеница» [12]. Метод независимо разрабатывался в России, Великобритании и США (там его аналог получил название SSA – Singular Spectrum Analysis). Суть метода заключается в формировании матриц наблюдений из множества временных рядов и выполнении PCA в приложении к этим матрицам. Применение 2D PCA\2DKLT позволит в этом случае выявить закономерности в исходных данных (тренды и их периоды, сезонные и другие периодические составляющие, взаимосвязь отдельных рядов наблюдений между собой и определить их модели), а также позволит сократить объем вычислений и объем

информации, представив исходные данные в более компактном виде.

3. Развитие методов записи и хранения данных привело к бурному росту объемов собираемой и анализируемой информации. Объемы данных настолько внушительны, что человеку просто не по силам проанализировать их самостоятельно, хотя необходимость проведения такого анализа вполне очевидна, ведь в этих «сырых» данных заключены знания, которые могут быть использованы при

принятии решений. Для того чтобы провести автоматический анализ данных, используется метод «Data Mining». Это процесс обнаружения в «сырых» данных ранее неизвестных нетривиальных практически полезных и доступных интерпретации знаний, поиска закономерностей и представление данных в виде удобном для анализа, делающее возможным построение моделей прогнозирования. Одним из методов, широко используемых в «Data Mining», является PCA. Очевидно, что алгоритмы 2D PCA будут здесь также кстати! ■

Литература

1. Мокеев А.В. О точности и быстродействии метода синтеза главных компонент // Бизнес-информатика, 2010, № 3(13). – С. 65-68.
2. Прэтт У. Цифровая обработка изображений. – М.: Мир, 1982. – 312 с.
3. Turk M., Pentland A. Eigenfaces for Face Detection // Journal of Cognitive Neuroscience, 1991, Vol. 3, №1. – P. 71-86.
4. Tsapatsoulis N., Alexopoulos V., Kollias S.A. Vector Based Approximation of KLT and Its Application to Face Recognition // Proceedings of the IX European Signal Processing Conference EUSIPCO-98, Vol. 3. – P.1581-1584.
5. Кухарев Г.А. Биометрические системы: Методы и средства идентификации личности человека. – СПб.: Политехника, 2001. – 240 с.
6. Kukharev G., Forczmański P. Data Dimensionality Reduction for Face Recognition // Journal MACHINE GRAPHICS & VISION, 2004, Vol. 13, № 1/2. – P. 99-121.
7. Кухарев Г.А., Щеголева Н.Л. Алгоритмы двумерного анализа главных компонент для задач распознавания изображений лиц // Компьютерная оптика, 2010, т.34, № 4. – С. 545-551.
8. Кухарев Г.А. Щеголева Н.Л. Системы распознавания человека по изображению лица. – СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2006. – 176с.
9. Zuo W., Wang K., Zhang H. Subspace Methods for Face Recognition: Singularity, Regularization, and Robustness // State of the Art in Face Recognition. I-Tech, Vienna, Austria, – 2004. – 250 p.
10. Кухарев Г.А. Поиск изображений лиц в больших базах данных // Мир измерений, 2009, № 4(98). – С. 22-30.
11. The ORL database - www.uk.research.att.com/pub/data/att_faces.zip.
12. Shchegoleva N.L., Kukharev G.A. Application of Two-Dimensional Principal Component Analysis for Recognition of Face Images // Pattern Recognition and Image Analysis, 2010, Vol. 20, №4. – P. 513-527. http://www.maik.ru/contents/patrec/patrec4_10v20cont.htm
13. Силантьев Д.А. Оценка необходимого размера свертки биометрического образца для обеспечения заданных параметров надежности биометрической системы идентификации // Бизнес-информатика, 2009, № 3(09). – С. 21-23.

КОМПОЗИЦИЯ СЛУЧАЙНЫХ ИНКРЕМЕНТНЫХ ДЕРЕВЬЕВ И ВОССТАНОВЛЕНИЕ СТРУКТУРЫ ТАБЛИЦ¹

П.Ю. Кудинов,

аспирант отдела вычислительных методов прогнозирования Вычислительного центра им. А.А. Дородницына Российской академии наук (ВЦ РАН)

В.А. Полежаев,

студент кафедры математических методов прогнозирования Московского государственного университета им. М.В. Ломоносова (ВМК МГУ)

Адрес: г. Москва, ул. Вавилова, д. 40

E-mail: pkudinov@gmail.com, valentin.polezhaev@gmail.com

В статье предлагается новый корректный инкрементный алгоритм классификации, основанный на построении композиции случайных деревьев (RIF). Рассмотрена задача распознавания структуры статистических таблиц, предложены постановки задач распознавания и признаки объектов. Проведено сравнение нового алгоритма с известным алгоритмом ITI. Алгоритм RIF показывает лучшие результаты на большинстве задач.

Ключевые слова: инкрементное машинное обучение, корректные алгоритмы, анализ текстов, информационный поиск, классификация, деревья решений, распознавание образов.

1. Введение

Во многих задачах распознавания есть потребность продолжать обучение на новых объектах обучающей выборки. Такой процесс называется динамическим обучением (on-line learning). После распознавания очередного объекта алгоритму передаётся метка класса этого объекта, которую он должен быть способен учесть, т. е. «до-

обучиться». Такого рода алгоритмы используются в полуавтоматических системах обработки информации. Главным преимуществом этих систем является повышение эффективности труда эксперта или оператора, т. к. с течением времени алгоритм совершает меньше ошибок на новых объектах. Во многих приложениях исправления оператора считаются абсолютно верными, поэтому к этим алгоритмам предъявляется *требование корректности*

¹ Работа выполнена при финансовой поддержке РФФИ, проекты №11-07-00480-а и 10-07-00609-а.

[1, 2], т. е. гарантия безошибочной классификации объектов обучающей выборки.

В настоящее время актуальной является задача создания системы информационного поиска по социально-демографической, экономической, финансовой статистической информации, свободно доступной в интернете в виде текстовых таблиц. Основной функцией системы должен быть вывод агрегированных статистических таблиц по запросам пользователей. Первым шагом на пути реализации такой системы является разработка методов извлечения статистических показателей из таблиц [3, 4]. Возникающие при этом задачи распознавания предлагается решать методами динамического обучения. Основной сложностью является большой объём выборки, который будет достигать десятков тысяч таблиц в год. Поэтому к динамическим алгоритмам предъявляется не только требование корректности и малый процент ошибок, но и высокая вычислительная эффективность. В работе [4] авторами исследовались инкрементные алгоритмы для задач классификации, возникающих при извлечении данных из статистических таблиц. В результате проведённых экспериментов оказалось, что наиболее эффективным алгоритмом является Incremental Tree Induction (ИТИ, [5, 6]), который позволяет достичь приемлемого качества классификации и удовлетворяет требованию корректности. Основным недостатком данного алгоритма является необходимость постоянного перестроения и упрощения дерева, без выполнения которых происходит сильное его переобучение. Выполнение этих процедур занимает основное время работы алгоритма. В настоящей работе описываются основные свойства алгоритма ИТИ и предлагается новый корректный инкрементный алгоритм, который строит композицию случайных инкрементных деревьев. Показаны преимущества этого алгоритма перед ИТИ и рассматривается модификация с отбором деревьев.

2. Решающие деревья

Обозначим всё множество объектов через X , а множество признаков – через $F = \{f_1, \dots, f_M\}$, где $f_i: X \rightarrow D_{f_i}$. Если $|D_{f_i}| < \infty$, то f_i – номинальный признак, иначе – числовой. Множество классов обозначим через Y . Пусть дана обучающая выборка $X^l = \{(x_i, y_i)_{i=1}^l\}$, l пар объект-ответ. Решающее дерево – алгоритм классификации $X \rightarrow Y$, который представим в виде множества вершин $T = \{v\}$, корень дерева v_0 . Каждой узловой вершине v соответствует

(L_v, R_v, β_v) , где L_v, R_v – ссылки на левое и правое поддерево, а β_v является логическим условием. Будем рассматривать $\beta_v(x) = \mathbb{I}[f_j(x) < a]$ в случае числового признака, и $\beta_v(x) = \mathbb{I}[f_j(x) = a]$ в случае номинального. Каждой вершине v соответствует метка класса. Классификация осуществляется путём последовательной проверки условий в узлах дерева начиная от корня дерева. Если $\beta_v(x) = 0$ то $v = L_v$, иначе – $v = R_v$. Обучение алгоритма состоит в рекурсивном разбиении выборки на две части по наиболее информативному логическому условию (**Алгоритм 1**). На каждом этапе выбор наилучшего условия осуществляется перебором всевозможных предикатов, то есть перебором всевозможных разбиений выборки и вычислением информативности каждого из них. Информативность может быть вычислена с помощью энтропийного критерия информативности, индекса Джини или других эвристических критериев [7].

Алгоритм 1. Обучение дерева (TrainTree)

Вход: $X^l = \{(x_i, y_i)_{i=1}^l\}$ – множество пар объект-ответ;

Выход: v – корень дерева;

если в X^l представлены объекты как минимум 2 классов **то**

наилучшего $\beta =$ предикат, разделяющий X^l
 $L := \text{TrainTree}(\{x \in X^l \mid \beta(x) = 0\})$; (обучаем левое поддерево)

$R := \text{TrainTree}(\{x \in X^l \mid \beta(x) = 1\})$; (обучаем правое поддерево)

вернуть $v = \langle L, R, \beta \rangle$;

иначе

вернуть $v = \langle y \rangle$, y – класс объектов, представленных в X^l ;

Пусть имеется подвыборка

$$A = \{(a_i, b_i)\} \subseteq X^l, 1, i, |A|$$

и её разбиение на два непересекающихся подмножества $A_1, A_2, A_1 \cap A_2 = \emptyset$. Обозначим через

$$p_y(A) = \frac{1}{|A|} \sum_{i=1}^{|A|} [b_i = y]$$

относительное число объектов класса y .

Инкрементное построение дерева решений

Алгоритм Incremental Tree Induction (ИТИ, [5, 6]) предполагает инкрементное построение бинарного дерева решений. Дерево решений алгоритма ИТИ отличается от дерева решений ID3 тем, что в узловой вершине v помимо тройки (L_v, R_v, β_v) хранится пятёрка $(L_v, R_v, \beta_v, X_v, s_v)$, где L_v, R_v, β_v , как и в обычном дереве решений, левая, правая вершина и пре-

дикат соответственно, $X_v \subset X^l$ – список объектов, прошедших через узел v , s_v – состояние узла. Если через него прошли новые объекты, то $s_v = 1$, иначе $s_v = 0$. Перед началом инкрементного обучения производится обучение на части выборки $X_0 \subset X^l$. Далее начинается этап инкрементного обучения. После классификации очередного объекта x происходит его встраивание в дерево (**Алгоритм 2**). Начиная от корневого узла происходит последовательный переход к дочерним узлам в соответствии с предикатами, пока очередной узел не окажется листом. В каждом посещённом узле v устанавливается состояние $s_v = 1$ и к списку объектов добавляется новый объект x . В случае если класс нового объекта совпадает с классом, установленным в листе, то он добавляется к списку объектов. Иначе происходит поиск наилучшего разбиения объектов, хранимых в листе, и он превращается в узел, для которого строятся потомки. Таким образом, в корне дерева хранится информация обо всей выборке; на следующем уровне в каждом узле или листе хранится информация, которая в совокупности даёт информацию обо всей выборке, и т. д.

Алгоритм 2. Добавление объекта в дерево (AddToTree)

Вход: T, x, y – обученное дерево, объект для классификации, правильный ответ;

Выход: T , – обновленное дерево;

$v := v_0$;

пока v – узловая вершина

$X_v := X_v \cup x$;

$s_v = 1$;

если $\beta^v(x) = 0$, **то**

$v := L_v$; (переход влево)

иначе

$v := R_v$; (переход вправо)

если $y = c_v$, **то**

$X_v := X_v \cup x$;

иначе

$Z := X_v \cup x$; (список объектов в узле)

β = наилучший предикат для Z ;

$L_v := \text{TrainTree}(\{z \in Z \mid \beta(z) = 0\})$; (обучаем левое поддереву)

$R_v := (\{z \in Z \mid \beta(z) = 1\})$; (обучаем правое поддереву)

$v := \langle L, R, \beta, X_v, s_v \rangle$;

вернуть T .

Для обеспечения наилучшего качества при добавлении нового объекта можно было бы перестраивать дерево заново, однако это вычислительно не эффективно. В [6] описывается способ инкрементного преобразования дерева, который заключает-

ся в проверке того, что во всех узлах установлены предикаты, имеющие наилучшую информативность. Если это не так, то происходит поиск нового условия преобразования дерева, называемое трансформацией. Однако поиск лучшего предиката имеет высокую вычислительную сложность – его выполнение занимает большую часть времени работы алгоритма.

3. Алгоритм RIF

Основные затраты при построении инкрементного дерева и выполнении транспозиции приходится на поиск наилучшего предиката, то есть перебора разбиений. Одним из способов существенно сократить эти затраты является сокращение числа признаков, используемых при переборе.

Алгоритм Random Forest [8] представляет собой композицию решающих деревьев. Он основывается на двух принципах: бэггинге и методе случайных подпространств. Применительно к решающим деревьям это означает, что каждое новое дерево строится по новой случайной выборке с повторениями (полученной из исходной) и по случайному поднабору признаков. В работе предлагается новый алгоритм (**Алгоритмы 3, 4**) построения случайного инкрементного леса деревьев (Random Incremental Forest, RIF), результатом работы которого является композиция случайных инкрементных деревьев Random Incremental Tree (RIT). RIF можно представить в виде множества пар «дерево, набор признаков», $RIF = \{\langle RITree_i, M_i \rangle\}_{i=1}^p$, где $RITree_i$ – инкрементное дерево, построенное по случайному набору признаков $M_i \subseteq F$.

Алгоритм 3. Обучение композиции (RIF.Train)

Вход: $X^l = \{(x_i, y_i)\}_{i=1}^l, K, M$ – множество пар объект-ответ, число деревьев, число используемых признаков соответственно;

Выход: $RIF = \{\langle RITree_i, M_i \rangle\}_{i=1}^K$ – множество пар дерево-признаковое подпространство;

для $i = 1, \dots, K$

$M_i :=$ случайное подмножество признаков из

M элементов; $RITree_i := \text{TrainTree}(X^l, M_i)$;

вернуть $\{\langle RITree_i, M_i \rangle\}_{i=1}^K$;

3.1. Обучение

По начальной обучающей выборке строится композиция решающих деревьев по аналогии с Random Forest. Для построения каждого дерева используется одна и та же исходная выборка без перемешива-

ния с повторениями, от принципа бэггинга приходится отказаться для обеспечения корректности композиции.

Для каждого дерева случайно генерируется поднабор из $M_i = |F|$ признаков. Каждое дерево строится по стандартному алгоритму построения дерева решений с небольшим изменением: при поиске наилучшего разбиения выбирается случайный признак из поднабора признаков.

3.2. Классификация и внедрение нового объекта

Объект даётся на классификацию всем деревьям в композиции и относится к тому классу, за который проголосовало большее число деревьев. Внедрение нового объекта происходит во все деревья композиции (Алгоритм 4).

Алгоритм 4. Дообучение композиции новым объектом (RIF.Update)

Вход: $RIF = \{\langle RITree_i, E \rangle\}_{i=1}^K, x, y$ – обученная композиция, новый объект и правильный ответ соответственно;

Выход: $RIF = \{\langle RITree_i, M_i \rangle\}_{i=1}^K$ – обновлённая композиция;

для $i=1, \dots, K$

$RITree_i := AddToTree(RITree_i, x, y)$; – обновление каждого дерева, простое внедрение объекта;

вернуть RIF.

4. Отбор деревьев

Так как набор признаков в каждом дереве случаен, получающиеся в ходе инкрементного обучения деревья могут сильно отличаться по качеству. В случае длинных обучающих выборок можно осуществить отбор деревьев по некоторому критерию качества. Дополним каждое дерево композиции двумя показателями: τ – число объектов, которые были классифицированы деревом, и μ – число ошибок дерева. Эти показатели обновляются после классификации и внедрения каждого нового объекта.

4.1. Процедура отбора деревьев

Введём несколько параметров. K_0 – начальное число деревьев, стоит выбирать разумно большим, в зависимости от темпа отбора и размеров задачи. W – число худших деревьев, которые будут удалены. B – число лучших деревьев, на основе которых будут строиться новые деревья. При построении нового дерева признаки случайно выбираются из подна-

бора признаков, на которых построены лучшие деревья. Будем осуществлять взвешивание признаков – если некоторый признак оказался в признаковых наборах нескольких деревьев, то его вес и соответственно вероятность его выбора в набор для нового дерева увеличивается. ΔK – число новых деревьев, которые будут построены на основе признакового подпространства лучших деревьев.

Процесс отбора деревьев представляет собой последовательность следующих операций (Алгоритм 5) и выполняется с заданным периодом. Значение W – ΔK характеризует темп отбора, значение B характеризует «ширину» отбора. Чем меньше B , тем меньше набор признаков для выбора при построении новых деревьев. Также имеет смысл определить минимальное «время жизни» τ_0 для деревьев и минимальное число деревьев K_{min} . Слишком «молодые» деревья не участвуют в процедуре отбора, так как их статистика малоинформативна, также процедура отбора не выполняется, если число деревьев достигло минимума, то есть существует риск потери в качестве классификации.

Алгоритм 5. RIFTS.SelectTrees

Вход: $RIFTS = \{\langle RITree_i, M_i \rangle\}_{i=1}^K, X^l, W, B, \Delta K, K_{min}, \tau_0$ – обученная композиция, вся доступная обучающая выборка, параметры отбора деревьев соответственно;

Выход: $RIFTS = \{\langle RITree_i, M_i \rangle\}_{i=1}^M$ – обновлённая композиция;

для всех деревьев с номерами $i=1, \dots, K : \tau < \tau_0$

вычислить качество q_i ;

Отсортировать деревья по убыванию $q_i, i=1, \dots, K$;

Удалить $\min(W, K - K_{min})$ последних деревьев;

$A = \bigcup_{i=1, \dots, B} M_i$; – признаки лучших деревьев;

для $j=1, \dots, \Delta K$

$M_j =$ случайные M признаков из A ;

$RIT_j = \text{TrainTree}(X^l, M_j)$;

$RIFTS.Add(RIT_j)$; – добавление нового дерева в композицию;

вернуть RIFTS.

4.2. Оценка качества дерева

Так как деревья строятся в разные моменты и не ошибаются на своих обучающих выборках, то для их сравнения недостаточно учитывать только величину μ (очевидно, построенные позже деревья всегда будут иметь значение μ меньше), также нужно учитывать τ . Деревья можно сравнивать по относительному числу ошибок (величина μ/τ). Рассмотрим два дерева, которые имеют одинаковую

относительную ошибку μ , но одно из деревьев имеет значительно большую величину τ . Такое дерево кажется более надежным, но это никак не отражается в таком критерии. Можно выводить различные эвристические формулы с участием величин τ и μ , но адекватность критериев будет, как правило, зависеть от размеров задачи.

Допустим, что классификация объекта – случайная величина η из распределения Бернулли с параметром p . Пусть имеется n испытаний и m ошибок. Тогда по теореме Муавра-Лапласа можно получить верхнюю оценку вероятности ошибки:

$$\bar{p} = \frac{m}{n} + \Phi_{1-\alpha} \sqrt{\frac{m(n-m)}{n^3}}$$

Полученное выражение для \bar{p} используется в качестве критерия качества дерева (при $n=\tau$, $m=\mu$).

5. Восстановление структуры статистических таблиц

Составители статистических таблиц используют различные средства для разделения таблицы на ячейки и блоки и различные типы логической структуры таблиц. Для её распознавания используются методы динамического обучения. Рассмотрим квадратную сетку $G^{M \times N}$ из M строк и N столбцов и зададим её полное покрытие непересекающимися

прямоугольными областями. Элемент покрытия будем называть *ячейкой*. Множество всех ячеек обозначим через C . Каждой ячейке $c \in C$ поставим в соответствие текстовую строку $text(c)$, возможно пустую, которую назовём *содержимым ячейки* или её *текстом*. Будем полагать, что каждая ячейка c имеет тип $type(c) \in \{data, key, empty\}$. *Ячейка данных (data)* содержит ровно одно числовое значение. *Ячейка описания (key)* содержит текстовую строку, состоящую из словесных описаний одного или нескольких ключей. *Пустая ячейка (empty)* не содержит значимой информации, её содержимое игнорируется. *Статистической таблицей T* будем называть четвёрку $\langle G^{M \times N}, C_V, C_K, R \rangle$, где C_V – множество ячеек данных, C_K – множество ячеек описаний, отображение $R: C_V \rightarrow 2^{C_K}$ ставит в соответствие каждой ячейке данных множество ячеек описания, т. е. задаёт структуру таблицы.

5.1. Логическая структура таблиц

Отображение R определяется взаимным расположением ячеек данных и описаний и стилевым оформлением ячеек. Для таблицы *простой структуры (табл. 1)* оно определяется следующим образом: для каждой ячейки данных выбираются все ячейки описания, пересекающиеся со строкой или столбцом рассматриваемой ячейки.

Первым шагом к построению отображения R яв-

Таблица 1.

Статистическая таблица простой структуры

	2003	2004	2005	2006	2007	2008	2009
Всего в экономике	107,0	106,5	105,5	107,5	107,5	104,8	95,8
Сельское хозяйство, охота и лесное хозяйство	105,6	102,9	101,8	104,3	105,0	110,7	105,0
Рыболовство, рыбоводство	102,1	104,3	98,5	101,6	103,2	95,5	109,2
Добыча полезных ископаемых	109,2	107,3	106,3	103,3	103,1	101,0	96,1
Обрабатывающие производства	108,8	109,8	106,0	108,5	108,4	102,6	96,1
Производство и распределение электроэнергии, газа и воды	103,7	100,7	103,7	101,9	97,5	102,1	96,3
Строительство	105,3	106,8	105,9	115,8	112,8	109,1	91,4
Оптовая и розничная торговля, ремонт автотранспортных средств, мотоциклов, бытовых изделий и предметов личного пользования	109,8	110,5	105,1	110,8	104,8	108,1	92,1
Гостиницы и рестораны	100,3	103,1	108,5	109,2	108,0	109,2	87,1
Транспорт и связь	107,5	108,7	102,1	110,7	107,5	106,5	100,1
Операции с недвижимым имуществом, аренда и предоставление услуг	102,5	101,3	112,4	108,2	117,1	107,9	98,7

ляется классификация типов ячеек, т. е. построение множеств C_V и C_K . Положение ячейки $c \in C$ в таблице описывается координатами левого верхнего $(r_1(c), c_1(c))$ и правого нижнего $(r_2(c), c_2(c))$ углов прямоугольника по сетке $G^{M \times N}$. Для каждой ячейки $c \in C$ генерируются следующие признаки:

1. $f_1(c)$ – количество чисел;
2. $f_2(c)$ – количество слов;
3. $f_3(c)$ – количество символов;
4. $f_4(c) = (r_1(c) + r_2(c)) / 2M$ – вертикальное положение;
5. $f_5(c) = (c_1(c) + c_2(c)) / 2N$ – горизонтальное положение;
6. $f_6(c) = r_2(c) - r_1(c)$ – число вертикально объединённых элементов сетки;
7. $f_7(c) = c_2(c) - c_1(c)$ – число горизонтально объединённых элементов сетки.

5.2 Таблица с суперстроками

Таблица может быть разделена на несколько частей в соответствии со значениями некоторого показателя. Например, данные по мужской и женской занятости, данные за разные годы, абсолютные и относительные данные одних и тех же показателей. Для совмещения таких данных в одной таблице составители часто используют суперстроки (табл. 2).

Таблица 2.

Статистическая таблица с суперстроками

	Общая численность безработных			Численность официально зарегистрированных безработных		
	тыс. человек	в % к		тыс. человек	в % к	
		соответствующему периоду	предыдущего года		соответствующему периоду	предыдущего года
2010 г.						
Январь	6832	105,1	110,7	2202	129,0	102,6
I квартал (в среднем за месяц)	6555	96,2	108,0	2244	114,2	108,5
2011 г.						
Январь	5815	85,1	107,8	1609	73,1	101,2

Рассмотрим задачу классификации строк таблицы на два класса: «обычная строка» и «суперстрока». Для каждой строки будем строить следующий набор признаков:

1. $f_1(c)$ – количество ячеек в строке;
2. $f_2(c) = N$ – ширина таблицы;
3. $f_3(c)$ – высота строки;
4. $f_4(c)$ – количество пустых ячеек;
5. $f_5(c) = (c_1(c) + c_2(c)) / 2N$ – количество не пустых ячеек.

5.3 Вложенные ячейки

Ещё одним часто встречающимся приёмом оформления таблиц является использование вложенных ячеек, когда несколько ячеек сдвигаются на один уровень вправо (табл. 3). Для определения вложенности решается задача классификации, в которой объектами являются пары последовательно идущих ячеек $p = (x_1, x_2)$ в левом блоке ячеек описания. Выделяются три класса: «ячейки находятся на одном уровне», « x_2 сдвинута вправо относительно x_1 » и « x_2 сдвинута влево относительно x_1 ». Для этих объектов вычисляется следующий набор признаков:

1. $f_1(p)$ – текст x_1 заканчивается на «:»;
2. $f_2(p)$ – количество начальных пробельных символов в тексте x_2 ;
3. $f_3(p)$ – тип первого непробельного символа в x_2 : «цифра», «буква» или «знак»;
4. $f_4(p)$ – первая буква в x_1 является прописной;
5. $f_5(p)$ – первая буква в x_2 является прописной;
6. $f_6(p) = r_2(x_1) - r_1(x_1)$ – высота x_1 ;
7. $f_7(p) = c_2(x_2) - c_1(x_2)$ – высота x_2 .

Таблица 3.

Фрагмент таблицы с вложенными ячейками

	2002 г.			
	тыс. человек		распределение по полу, %	
	жен.	муж.	жен.	муж.
Все население	77562	67605	53	47
в том числе в возрасте, лет:				
0-9	6515	6825	49	51
10-19	11390	11817	49	51
20-29	10982	11097	50	50
30-39	10113	9939	50	50
40-49	12575	11578	52	48

6. Эксперименты

Для оценки качества алгоритма RIF использовались задачи из репозитория UCI [9] и реальные выборки статистических таблиц. Ошибка вычислялась как средняя величина по 20 повторениям с перемешиванием обучающей выборки. В качестве критерия информативности разбиения использовался индекс Джини.

6.1. Оптимальное число деревьев

Рассмотрим некоторые задачи, посмотрим зависимость качества от числа деревьев и сравним с результатами, полученными на этих же задачах, алгоритма ITI с транспозицией и без. Период транспозиции выбирался равным 50 или 100 в зависимости от размеров задачи.

В табл. 4 приведены найденные наилучшие пары ошибка-число деревьев для каждой задачи. Эксперименты проводились для значений числа деревьев начиная с 10 и до 250 с шагом 5. На рис. 1 приведён график зависимости процента ошибок от числа деревьев для задачи Heart.

Таблица 4.

Оптимальное число деревьев и ошибка композиции (в процентах)

Задача	Число деревьев RIF	Ошибка RIF	Ошибка ITI с транспозицией	Ошибка ITI без транспозиции
Ionosphere	85	7.6	12.8	12.6
Car	130	9.3	7.3	14.5
German	100	27.4	34.2	34.8
Heart	75	21.3	27.9	30.1
Yeast	115	47	52.1	55.3

Почти на всех задачах заметен выигрыш в качестве алгоритма RIF по отношению к ITI, на задачах Ionosphere и WBC алгоритм RIF работает значительно лучше.

6.2. Отбор деревьев

Эксперименты проводились только на достаточно больших задачах. Композиция без отбора RIF состояла из 50 деревьев.

Результаты эксперимента описаны в табл. 5. Видно, что отбор деревьев даёт улучшения в работе алгоритма не всегда, более того на задачу Pen отбор оказал отрицательное влияние.

Таблица 5.

Результаты отбора деревьев

Задача	Параметры отбора RIFTS ($K_0, B, W, \Delta K, K_{min}$)	Ошибка RIFTS, %	Ошибка RIF (без отбора), %
Car	(160,10,10,7,50)	8.1	10.2
Nursery	(200,8,10,7,50)	4.3	6.4
Optical	(160,8,10,7,50)	13.2	12
Pen	(160,8,10,7,50)	9.9	4.1

6.3. Задачи распознавания структуры таблиц

Алгоритмы RIF и ITI сравнивались на коллекции из 1000 таблиц. Начальная обучающая выборка составляла 50 объектов, после чего запускалось динамическое обучение. Все эксперименты проводились 20 раз со случайно перемешанной выборкой. Композиция RIF состояла из 50 деревьев. График зависимости доли ошибок от числа обучающих объектов для задачи СТ представлен на рис. 1, доверительные интервалы приведены в табл. 6.

Таблица 6.

Доверительные интервалы частоты ошибок алгоритмов (в процентах) на последнем объекте

	ITI			RIF		
	CT	SR	CI	CT	SR	CI
Мин.	0,05	0,02	0,6	0,04	0,02	1,3
Сред.	0,08	0,04	1,7	0,07	0,03	2,8
Макс.	0,13	0,08	2,5	0,11	0,05	4,4

Задача распознавания типа ячеек (СТ) представлена 28624 объектами, 8 числовыми признаками. На этой задаче ITI использовался без транспозиции из-за больших вычислительных затрат. Оба алгоритма справились с задачей достаточно успешно, средняя ошибка составила менее 0,1%. Следует заметить, что доверительный интервал ошибки у RIF меньше, чем у ITI.

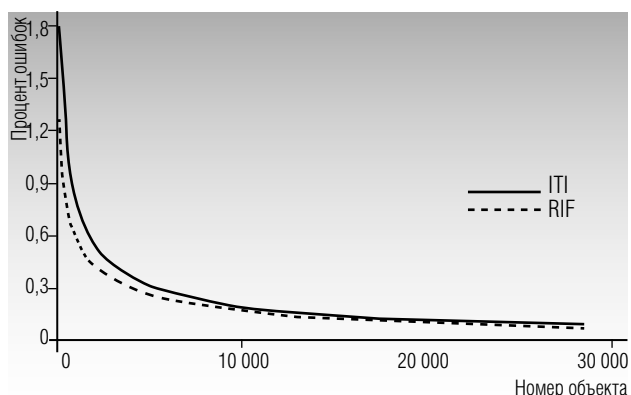


Рис. 1. Задача распознавания типа ячеек (СТ).

Задача распознавания суперстрок (SR) представлена 102217 объектами, 5 числовыми признаками. На ней алгоритм ИТІ также использовался без транспозиции, для этой задачи справедливы выводы, аналогичные выводам по предыдущей задаче. Ошибка составила менее 0,1%.

Задача распознавания вложенных ячеек (CI) представлена 370 объектами, 4 номинальными и 7 числовыми признаками. Относительно небольшая длина выборки позволила использовать ИТІ с транспозицией (с периодом 50), в результате ИТІ справился с задачей лучше – 1,7% ошибок против 2,8% ошибок у RIF. Наличие транспозиции оказалось существенным для этой задачи.

6.4. Выводы

1. Композиция случайных деревьев RIF работает лучше, чем ИТІ, на некоторых задачах качество возрастает в 2 раза, но также есть задачи, где ИТІ значительно опережает RIF.

2. Для таких задач, очевидно, критично наличие транспозиции (или перестроения дерева), то есть этот проигрыш можно устранить добавив какую-нибудь стратегию перестроения деревьев в композицию, например включение отбора деревьев, причем строить и удалять одинаковое число деревьев.

3. Композиция случайных деревьев RIF работает значительно быстрее, чем ИТІ с малыми периодами транспозиции и/или на больших задачах.

4. Композиция случайных деревьев с отбором деревьев RIFTS позволяет улучшить качество.

7. Заключение

В статье описаны особенности и недостатки известного инкрементного алгоритма ИТІ. Предложен новый корректный инкрементный алгоритм RIF, основанный на построении композиции случайных инкрементных деревьев и предложена модификация алгоритма с отбором деревьев (RIFTS). Проведённые вычислительные эксперименты на задачах репозитория UCI показали улучшение качества работы алгоритма на большинстве задач. На некоторых задачах прирост качества весьма значителен, не тратится много времени на операцию транспозиции. Отбор деревьев также, как правило, даёт прирост в качестве, причём величина прироста зависит от времени работы алгоритма – чем продолжительнее отбор, тем больше прирост качества. Рассмотрена задача распознавания структуры статистических таблиц, необходимой для построения поисковой системы статистической информации. Предложены постановки трёх задач распознавания структуры, перечислены признаки объектов, учитывающие специфику статистических таблиц. Эксперименты на реальной выборке показали качество классификации на уровне 96-99% для разных задач.

8. Благодарности

Авторы выражают благодарность своему научному руководителю профессору НИУ ВШЭ, д.ф.-м.н. К.В. Воронцову, а также профессору НИУ ВШЭ, д.т.н. Б.Г. Миркину за советы и конструктивные замечания. ■

Литература

1. Djukova E.V., Zhuravlev J.I., Rudakov K.V. Algebraic-logic synthesis of correct recognition procedures based on elementary algorithms // Computational Mathematics and Mathematical Physics, 1996, 36(8). – P. 1161–1167.
2. Zhuravlev J.I. Algebraic methods in recognition and classification problems // Pattern Recognition and Image Analysis, 1991, 1(1).
3. Кудинов П.Ю., Задача распознавания статистических таблиц // Доклады 14-й Всероссийской конференции «Математические методы распознавания образов» ММРО-2009. – М.: МАКС Пресс, 2009. – С. 552-555.
4. Кудинов П.Ю., Полежаев В.А. Динамическое обучение распознаванию статистических таблиц // Доклады 8-й Международной конференции «Интеллектуализация обработки информации» ИОИ-2010 (Республика Кипр, г. Пафос, 17-24 октября 2010). – М.: МАКС Пресс, 2010. – С. 512-515.
5. Utgoff P.E. An improved algorithm for incremental induction of decision trees // Machine Learning: Proceedings of the Eleventh International Conference, 1994. – P. 318-325.
6. Utgoff P.E., Berkman N.C., Clouse J.A. Decision tree induction based on efficient tree restructuring // Machine Learning, 1997, 29 (1). – P. 5-44.
7. Furnkranz J., Flach P.A. ROC n' rule learning – Towards a better understanding of covering algorithms // Machine Learning, 2005, 58 (1). – P. 39-77.
8. Breiman L., Schapire E. Random forests // Machine Learning, 2001, 45 (1). – P. 5-32.
9. Asuncion A., Newman D.J. UCI Machine Learning Repository. 2007. URL: <http://www.ics.uci.edu/~mllearn/MLRepository.html>.

ИСПОЛЬЗОВАНИЕ БИБЛИОТЕКИ SEDUMI ДЛЯ РОБАСТНОЙ ОПТИМИЗАЦИИ ИНВЕСТИЦИОННОГО ПОРТФЕЛЯ

А.Г. Исавнин,

доктор физико-математических наук, профессор кафедры математических методов в экономике Казанского (Приволжского) федерального университета, филиал в г. Набережные Челны

Д.Р. Галиев,

студент кафедры математических методов в экономике» Казанского (Приволжского) федерального университета, филиал в г. Набережные Челны

Адрес: г. Набережные Челны, бульвар Строителей, д. 1 (5/10), каб. 209

E-mail: isavnin@mail.ru, damir.galiev@mail.ru

В статье раскрываются принципы робастной оптимизации инвестиционного портфеля. Описано использование библиотеки SEDUMI для решения задач робастной оптимизации. Представлены модернизированные версии некоторых моделей выбора оптимального портфеля: модели Марковица, модели Телсера и модели Блэка-Литтермана. Приводятся результаты сравнительного анализа эффективности моделей: до робастной оптимизации и после. Эксперименты проведены при растущем, боковом и понижающемся трендах с некоторыми высоколиквидными акциями, торгующимися на ММВБ. Оценены сильные и слабые стороны разных подходов.

Ключевые слова: инвестиционный портфель, робастная оптимизация, модель Марковица, модель Телсера, модель Блэка-Литтермана, SEDUMI.

1. Введение

Одной из основных проблем в задаче выбора оптимального инвестиционного портфеля является правильная оценка величин ожидаемой доходности μ и матрицы ковариации доходностей Σ . Эти параметры используются практически во всех современных моделях портфельного инвестирования, включая классическую mean-variance модель Марковица [1], модель Блэка-Литтермана [2], «интеллектуальные» модели [3]. На практике их тяжело

оценить правильно, так как значения параметров меняются каждый день. Именно от этих параметров зависит качество инвестиционного портфеля. Под качеством инвестиционного портфеля понимается совокупность параметров риска и доходности.

Существует несколько методов для снижения уровня неопределенности этих параметров. Основная идея этих методов заключается в том, чтобы снизить чувствительность итоговых оптимальных портфелей к неопределенным входным параме-

трам. Другими словами, если значения параметров μ и Σ изменятся незначительно, итоговый портфель не должен кардинально менять свою структуру. Фрост и Саварино [4] предлагают ограничивать веса для каждого актива в портфеле. Чопра [5] предлагает использовать оценку Джеймса-Стейна для ожидаемых средних значений, в то время как Блэк и Литтерман используют Байесовскую оценку μ и Σ (с учетом мнений экспертов). Также существуют методы выборов и сценариев, которые подробно описаны, например, в работе [6].

Принципы робастной оптимизации позволяют снизить влияние описанных проблем. Сначала необходимо определить интервал возможных значений параметров μ и Σ . Интервал значений называется неопределенным множеством этих параметров. Конечная задача будет решена для «наихудшего» случая. В итоге инвестор сможет увидеть уровень дохода портфеля при «наихудшем» развитии событий. Довольно часто в качестве критерия «наихудшего» случая используется показатель *VaR* (Value-at-Risk) [7]. Подобные подходы предложены в работах [8; 9; 10].

2. Математическая постановка задачи и краткое описание библиотеки SEDUMI

Для решения задачи построения оптимальных портфелей применяется метод Лагранжа или теорема Куна-Таккера (при наличии ограничений). При решении робастной оптимизационной задачи для «наихудшего» случая из неопределенного множества использование этих методов неэффективно. Вместо этого задачу можно свести к классу задач SOCP (Second Order Cone Problem):

$$\min_x \{f^T x \mid \|A_i x + b_i\| \leq c_i^T x + d_i, i = 1, \dots, N\}, \quad (1)$$

где $\|\cdot\|$ – Евклидова норма. Так, для вектора u , $\|u\| = \sqrt{u^T u}$, f , x и c – n -мерные векторы. Всего N ограничений. SOCP – класс задач, который находится между линейным программированием (LP – Linear Programming) и полуопределенным программированием (SDP – SemiDefinite Programming) [8].

Для проведения экспериментов была использована библиотека SEDUMI. Эта библиотека является дополнением к комплексу MATLAB для решения оптимизационных задач с линейными, квадратичными и полуопределенными ограничениями. Библиотека SEDUMI разработана Йосефом Штурмом из Тильбургского университета в Нидерландах. С конца 2004 года её развитием и продвижением занимается лаборатория оптимизации канадского университета Макмастер. Библиотека позволяет

использовать комплекснозначные переменные. С применением SEDUMI оптимизационные задачи большой размерности можно решать эффективнее, чем при использовании аналогов (CSDP, SDPLIB). Под эффективность решения понимается скорость нахождения и точность решения.

В настоящей работе для придания моделям свойства робастности был использован метод оптимизации наихудшего случая, а также введен контроль риска потери капитала посредством определения *VaR*-ограничений и ограничений на структуру инвестиционного портфеля. Схема формирования неопределенных множеств S_m и S_v для доходностей и ковариаций следующая:

$$\begin{aligned} \mu_i^L &\leq \mu_i \leq \mu_i^U, \quad \forall i \\ \sigma_{ij}^L &\leq \sigma_{ij} \leq \sigma_{ij}^U, \quad \forall i, j \\ \mu_i^0 &= (\mu_i^L + \mu_i^U) / 2, \quad \beta_i = (\mu_i^U - \mu_i^L) / 2, \\ \sigma_{ij}^0 &= (\sigma_{ij}^L + \sigma_{ij}^U) / 2, \quad \delta_{ij} = (\sigma_{ij}^U - \sigma_{ij}^L) / 2, \\ \mu_i^0 - \beta_i &\leq \mu_i \leq \mu_i^0 + \beta_i, \quad \forall i \\ \sigma_{ij}^0 - \delta_{ij} &\leq \sigma_{ij} \leq \sigma_{ij}^0 + \delta_{ij}, \quad \forall i, j \\ S_m &= \{ \mu : \mu^0 - \beta \leq \mu \leq \mu^0 + \beta, \beta \geq 0 \}, \\ S_v &= \{ \Sigma : \Sigma^0 - \Delta \leq \Sigma \leq \Sigma^0 + \Delta, \Delta \geq 0 \}. \end{aligned} \quad (2)$$

Здесь $\mu_i \in S_m$; $\sigma_{ij} \in S_v$; μ_i^L, μ_i^U – нижние и верхние границы значения параметров μ_i ; $\sigma_{ij}^L, \sigma_{ij}^U$ – нижние и верхние границы значения параметров σ_{ij} . Значения из полученных неопределенных множеств будут использоваться при определении наихудших и наилучших значений риска и доходности. Для получения робастной постановки задачи, необходимо сформулированную задачу оптимизации наихудшего случая привести к SOCP-виду.

3. Робастная модель Марковица

До недавнего времени современная портфельная теория, сформированная Г. Марковицем в 1952 году, оставалась чуть ли не единственным количественным методом решения задачи портфельного анализа. Основная идея этой теории состоит в следующем. Пусть имеются n активов, из которых инвестор может сформировать портфель. Капитал распределяется между активами долями $x_i, 0 \leq x_i \leq 1$. Активы характеризуются эффективностями R_i , которые являются случайными величинами с известными математическими ожиданиями $MR_i = \mu_i$ и матрицей ковариаций Σ . λ – коэффициент неприятия риска [11]. I – вектор-строка, состоящая из единиц. Модель Марковица формулируется следующим образом:

$$\max_x \left\{ \left(\mu^T x - \frac{\lambda}{2} x^T \Sigma x \right) \mid I^T x = 1 \right\} \quad (3)$$

Хотя модель Гарри Марковица может показаться привлекательной и вполне обоснованной с теоретической точки зрения, при её практическом использовании возникает ряд проблем. Некоторые проблемы, с примерами на российском рынке, описаны в [1; 11]. Составим робастную модель, предварительно выполнив ряд преобразований:

$$\begin{aligned} & \max_x \left\{ \min_{\mu, \Sigma} \left[\mu^T x - \frac{\lambda}{2} x^T \Sigma x \right] \mid I^T x = 1 \right\} \\ & \max_x \left\{ \min_{\mu} [\mu^T x] - \frac{\lambda}{2} \max_{\Sigma} [x^T \Sigma x] \mid I^T x = 1 \right\} \\ & \min_{\mu} [\mu^T x] = \min_{\mu} \sum_i \mu_i x_i = \sum_{i: x_i < 0} (\mu_i^0 + \beta_i) x_i + \\ & + \sum_{i: x_i \geq 0} (\mu_i^0 - \beta_i) x_i = \sum_i \mu_i^0 x_i + \sum_{i: x_i < 0} \beta_i x_i - \sum_{i: x_i \geq 0} \beta_i x_i = \\ & = \sum_i (\mu_i^0 x_i - \beta_i |x_i|) = (\mu^0)^T x - \beta^T |x| \quad (4) \\ & \max_{\Sigma} [x^T \Sigma x] = \max_{i, j} \sum \sigma_{ij} x_i x_j = \sum_{i, j: x_i x_j < 0} (\sigma_{ij}^0 - \delta_{ij}) x_i x_j + \\ & + \sum_{i, j: x_i x_j \geq 0} (\delta^0 + \delta_{ij}) x_i x_j = \sum_{i, j} \sigma_{ij}^0 x_i x_j + \sum_{i, j} \delta_{ij} |x_i x_j| = \\ & = \sum_{i, j} \sigma_{ij}^0 x_i x_j + \sum_{i, j} \delta_{ij} |x_i| |x_j| = x^T \Sigma^0 x + |x|^T \Delta |x| \\ & \max_x \left\{ (\mu^0)^T x - \beta^T |x| - \frac{\lambda}{2} x^T \Sigma x - \frac{\lambda}{2} |x|^T \Delta |x| \mid I^T x = 1 \right\} \end{aligned}$$

В итоге робастная задача примет вид:

$$\max_{x, \rho, \tau} \left\{ (\mu^0)^T x - \beta^T |x| - \frac{\lambda}{2} (\rho + \tau) \left| \begin{array}{l} I^T x = 1 \\ \rho \geq x^T \Sigma^0 x \\ \tau \geq |x|^T \Delta |x| \end{array} \right. \right\} \quad (5)$$

4. Робастная модель Телсера

Главное отличие и преимущество модели Телсера [8], в отличие от классической постановки задачи выбора оптимального портфеля, заключается в контроле риска потери капитала с использованием показателя VaR :

$$\max_x \left\{ \mu_p \left| \begin{array}{l} VaR_{\alpha} = -\mu_p - z_{\alpha} \sigma_p \\ \mu_p = \mu^T x \\ \sigma_p^2 = x^T \Sigma x \\ I^T x = 1 \\ x \in R_+^n \end{array} \right. \right\} \quad (6)$$

Рассмотрим вывод формулы VaR , предложенный автором модели. Так, значение VaR уровня $1-\alpha$ (VaR_{α}) определяется как:

$$P(R_p \leq -VaR) = \alpha \quad (7)$$

Совершив ряд преобразований, получим:

$$\begin{aligned} P(R_p \leq -VaR_{\alpha}) = \alpha & \Leftrightarrow P\left(\frac{R_p - \mu_p}{\sigma_p} \leq \frac{-VaR_{\alpha} - \mu_p}{\sigma_p}\right) = \alpha \Leftrightarrow \\ \frac{-VaR_{\alpha} - \mu_p}{\sigma_p} & = z_{\alpha} \Leftrightarrow VaR_{\alpha} = -\mu_p - z_{\alpha} \sigma_p \quad (8) \end{aligned}$$

где R_p – доходность портфеля, μ_p – средняя доходность портфеля, σ_p – риск портфеля (среднеквадратическое отклонение), z_{α} – квантиль стандартного нормального распределения порядка α . При этом полагается, что доходность портфеля имеет распределение, близкое к «нормальному». Так, VaR уровня 95% (т.е. при $\alpha = 0.05$) будет определяться как $VaR_{0.05} = -\mu_p + 1.6449 \sigma_p$.

Эта модель наследует главный недостаток классического подхода – сильную неустойчивость к входным параметрам. Составим робастную модель согласно представленным выше определениям и предположкам:

$$\max_x \left\{ \min_{\mu} [\mu^T x] \left| \begin{array}{l} I^T x = 1 \\ \max_{\mu, \Sigma} [P(R_p \leq -VaR)] \leq \alpha \\ x \in R_+^n \end{array} \right. \right\} \quad (9)$$

$$\min_{\mu} [\mu^T x] = (\mu^0)^T x - \beta^T |x|$$

$$\begin{aligned} \max_{\mu, \Sigma} [P(R_p \leq -VaR_{\alpha})] \leq \alpha & \Leftrightarrow \max_{\mu, \Sigma} \frac{-VaR_{\alpha} - \mu^T x}{\sqrt{x^T \Sigma x}} \leq z_{\alpha} \Leftrightarrow \\ \frac{-VaR_{\alpha} - \min_{\mu} \mu^T x}{\max_{\Sigma} \sqrt{x^T \Sigma x}} & \leq z_{\alpha} \Leftrightarrow -\min_{\mu} \mu^T x - z_{\alpha} \max_{\Sigma} \sqrt{x^T \Sigma x} \leq VaR_{\alpha} \Leftrightarrow \\ -(\mu^0)^T x + \beta^T |x| - z_{\alpha} \sqrt{x^T \Sigma^0 x + |x|^T \Delta |x|} & \leq VaR_{\alpha} \Leftrightarrow \\ -z_{\alpha} \left(\left\| \left(\Sigma^0 \right)^{0.5} x \right\| \right) & \left(\left\| \Delta^{0.5} |x| \right\| \right) \leq (\mu^0)^T x + \beta^T |x| + VaR_{\alpha} \end{aligned}$$

В итоге робастная задача примет вид:

$$\max_x \left\{ (\mu^0)^T x - \beta^T |x| - z_{\alpha} \left(\left\| \left(\Sigma^0 \right)^{0.5} x \right\| \right) \left(\left\| \Delta^{0.5} |x| \right\| \right) \leq (\mu^0)^T x + \beta^T |x| + VaR_{\alpha} \left| \begin{array}{l} I^T x = 1 \\ x \in R_+^n \end{array} \right. \right\} \quad (10)$$

5. Робастная модель Блэка-Литтермана

Модель Блэка-Литтермана была впервые опубликована Фишером Блэком и Робертом Литтерманом из Goldman Sachs [2]. Основой теории является «равновесный подход». Равновесные доходности вычисляются по формуле:

$$\Pi = \lambda \Sigma w_{mkt}, \quad (11)$$

где Π – вектор равновесной доходности; λ – коэффициент неприятия риска; Σ – матрица ковариации исторических доходностей; w_{mkt} – вектор рыночной капитализации каждого из активов относительно суммы капитализации активов в портфеле. Коэффициент λ характеризует готовность инвестора жертвовать величиной ожидаемой доходности портфеля ради снижения его риска:

$$\lambda = \frac{E(r) - r_f}{\sigma^2} \quad (12)$$

где $E(r)$ – ожидаемая доходность рынка, r_f – безрисковая ставка процента, $\sigma^2 = w_{mkt}^T \Sigma w_{mkt}$ – дисперсия рыночного портфеля. Рассмотрим формулу Блэка-Литтермана для апостериорного вектора доходности. Она является ключевым моментом перед расчетом итогового портфеля. Пусть K – количество субъективных мнений, N – количество активов.

$$\mu = [(\tau \Sigma)^{-1} + P' \Omega^{-1} P]^{-1} [(\tau \Sigma)^{-1} \Pi + P' \Omega^{-1} Q] \quad (13)$$

Здесь μ – новый (апостериорный) смешанный вектор доходности ($N \times 1$); τ – масштабирующий фактор; Σ – матрица ковариации доходности с размерностью $(N \times N)$; P – матрица размерности $(K \times N)$, которая идентифицирует активы, насчет которых у инвестора есть субъективное мнение; Ω – диагональная матрица ковариации с уровнями доверия для каждого субъективного мнения, $(K \times K)$; Π – вектор равновесной доходности, $(N \times 1)$; Q – вектор субъективных взглядов, $(K \times 1)$.

Неопределенность субъективных взглядов отражается в векторе ошибок ε , элементы которого нормально распределены со средней 0 и матрицей Ω . Таким образом, итоговые значения субъективных мнений имеет вид $Q + \varepsilon$.

$$Q + \varepsilon = \begin{bmatrix} Q_1 \\ \vdots \\ E_k \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_k \end{bmatrix} \quad (14)$$

Вариации ω элементов вектора ошибок ε формируют диагональную матрицу ковариаций Ω и демонстрируют меру неопределенности субъективных

взглядов. Матрица является диагональной, так как по предпосылкам модели субъективные мнения независимы друг от друга:

$$\Omega = \begin{bmatrix} \omega_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \omega_k \end{bmatrix} \quad (15)$$

Существует несколько методик определения элементов матрицы Ω [2; 10]. Значения доходностей по субъективным взглядам, находящиеся в векторе-столбце Q , вводятся в модель посредством матрицы P . Влияние каждого субъективного мнения отражается в векторе-строке размерностью $1 \times N$. Так, для K взглядов получаем матрицу P размерностью $K \times N$:

$$P = \begin{bmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & \ddots & \vdots \\ p_{k,1} & \cdots & E \end{bmatrix} \quad (16)$$

Итоговая формула выглядит следующим образом:

$$w = \mu (\lambda \Sigma)^{-1} \quad (17)$$

Составим робастную модель. В данном случае, из-за особенностей формирования вектора оценки будущих доходностей, использование предыдущих схем неприемлемо. Для придания робастности введем ограничения на структуру портфеля, а также ограничения по VaR для контроля риска потери капитала:

$$\max_x \left\{ \mu^T x \begin{array}{l} I^T x = 1 \\ AX \leq b \\ x^T \Sigma x \leq s \\ P(Y \leq VaR_\alpha(Y)) > \alpha \\ Y \in R \\ x \in R_+^n \end{array} \right\} \quad (18)$$

Здесь s – приемлемый уровень риска, задаваемый пользователем модели. Y – доходность портфеля. Матрица A и вектор b необходимы для наложения ограничений на структуру портфеля. Предварительно проведем ряд преобразований для показателя VaR , согласно методу, предложенному в [12]:

$$\begin{aligned} P(Y \leq VaR_\alpha(Y)) > \alpha &\Leftrightarrow P(\xi^T x \geq -\chi) \geq \alpha \\ P(\xi^T x \geq -\chi) &= P\left(\frac{\xi^T x - \mu^T x}{x^T \Sigma x} \geq \frac{-\chi - \mu^T x}{\sqrt{x^T \Sigma x}}\right) = \\ &= 1 - F_x\left(\frac{-\chi - \mu^T x}{\sqrt{x^T \Sigma x}}\right) \end{aligned} \quad (19)$$

$$1 - F_x \left(\frac{-\chi - \mu^T x}{\sqrt{x^T \Sigma x}} \right) \geq \alpha \Leftrightarrow F_x \left(\frac{-\chi - \mu^T x}{\sqrt{x^T \Sigma x}} \right) \leq 1 - \alpha \Leftrightarrow \frac{-\chi - \mu^T x}{\sqrt{x^T \Sigma x}} \leq F_x^{-1}(1 - \alpha) \Leftrightarrow \mu^T x + F_x^{-1}(1 - \alpha) \sqrt{x^T \Sigma x} \geq -\chi$$

где F_x – гауссовская функция распределения, ξ – вектор доходностей активов, χ – допустимый уровень VaR для портфеля, α – доверительный уровень и, как уже отмечалось раньше, в наших экспериментах этот показатель принимал значение $\alpha = 0.95$. Подобный подход использован в [13]. В итоге, робастная задача примет следующий вид:

$$\max_x \left\{ \begin{array}{l} \mu^T x \\ I^T x = 1 \\ AX \leq b \\ x^T \Sigma x \leq s \\ \mu^T x + F_x^{-1}(1 - \alpha) \sqrt{x^T \Sigma x} \geq -\chi \\ Y \in R \\ x \in R_+^n \end{array} \right\} \quad (20)$$

В наших экспериментах, в зависимости от требуемого уровня риска, значение χ варьировалось от 0.05 до 0.3. Другими словами, допустимые моделью предельные потери устанавливались от 5 до 30% от стоимости портфеля.

6. Анализ результатов

Доходность является одним из важнейших показателей эффективности управления портфелем, свидетельствующим об эффективности управления. Но нельзя, используя только доходность, судить о качестве управленческой стратегии. Помимо доходности есть и обратная сторона – риск, пренебрежение им в оценке эффективности может исказить реальное положение вещей. В настоящей работе для оценки эффективности инвестиционного портфеля использовался коэффициент Шарпа (Sharpe Ratio) [11]. Этот показатель по-другому называют коэффициент «доходность-разброс» (reward-to-variability ratio) и обозначают как $RVAR$:

$$RVAR = \frac{r_p - r_f}{\sigma_p} \quad (21)$$

где r_p – средняя доходность портфеля за рассматриваемый промежуток времени, r_f – среднее значение безрисковой ставки, σ_p – стандартное отклонение доходности портфеля (общий риск). В качестве безрисковой ставки использовалась доходность государственных облигаций. Всего в рамках работы было проведено несколько экспериментов. Временной промежуток: 01.07.2010 – 01.05.2011. При проведении экспериментов построения оптимальных

портфелей по описанным моделям использовались данные о дневных котировках акций следующих компаний, акции которых торгуются на ММВБ (табл. 1).

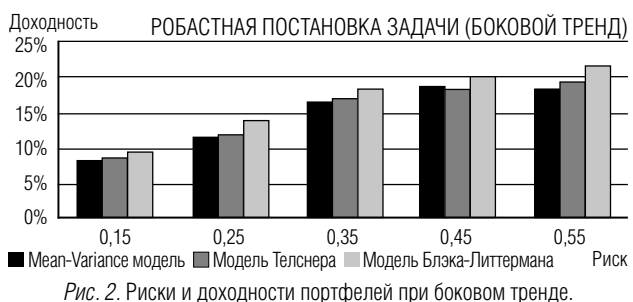
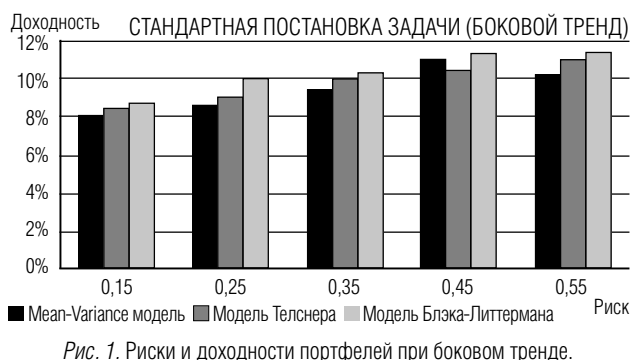
Таблица 1.

Предприятия, данные о котировках которых использовались при проведении экспериментов

№	Предприятие	Капитализация (млрд. руб.)	Количество акций
1	Татнефть	416.57	2 178 690 700
2	Газпром	5 627.19	23 673 512 900
3	Ростелеком	124.97	728 696 320
4	ВТБ	1 049.19	10 460 541 337 338
5	Лукойл	1 752.08	850 563 255
6	Полюс-Золото	321.40	190 627 747
7	Уралсиб	78.30	292 575 808 568
8	МТС	519.06	1 993 326 138
9	Сбербанк	2 340.67	21 586 948 000

Эксперименты проводились на российском рынке при растущем, боковом и понижающемся трендах. Рассмотрим результаты экспериментов при боковом тренде. Видно, что для всех моделей при увеличении риска увеличивается доходность (рис. 1, 2). У робастных моделей наблюдается более высокая доходность при примерно тех же уровнях риска. Следовательно, качество моделей повышается.

Опишем результаты аналогичных экспериментов на российском рынке при растущем тренде. Для всех моделей при увеличении риска увеличивалась доходность. В случае растущего тренда, при стандартной и робастной постановках задачи, наблюда-



лась высокая доходность портфелей. Качество робастных моделей немного выше качества моделей в стандартной постановке (рис. 3, 4).

Еще раз можно заметить, что доминирует модель Блэка-Литтермана, в то время как классическая mean-variance модель и модель Телсера ведут себя примерно одинаково. Это зависит от нескольких причин. Во-первых, были использованы прогнозы от аналитических ведомств с адекватной прогнозной способностью [14]. Во-вторых, при робастной постановке задачи вводились дополнительные ограничения на структуру

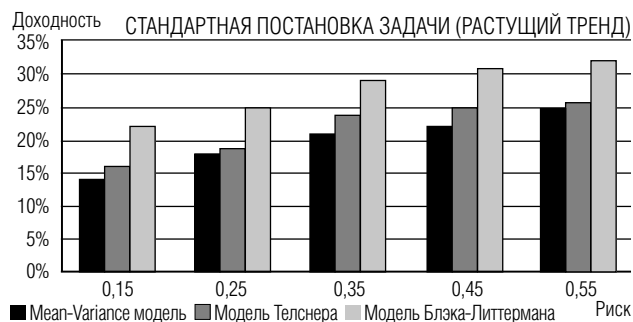


Рис. 3. Риски и доходности портфелей при растущем тренде.

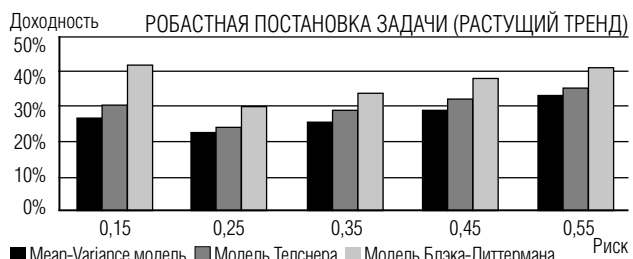


Рис. 4. Риски и доходности портфелей при растущем тренде.

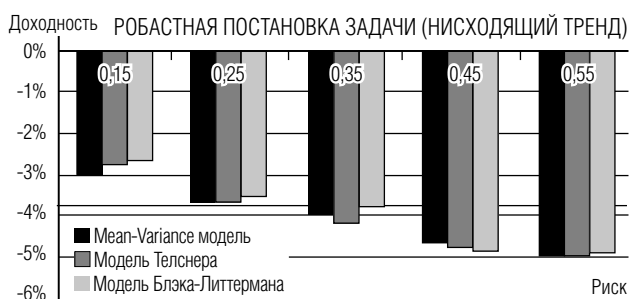


Рис. 5. Риски и доходности портфелей при падающем тренде.

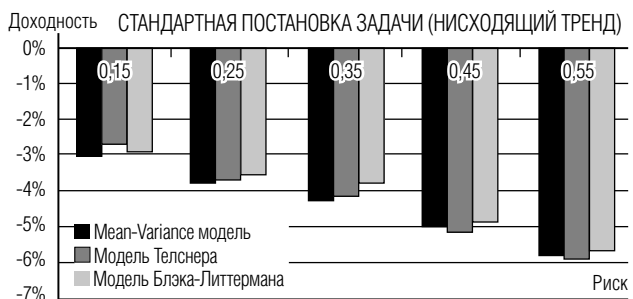


Рис. 6. Риски и доходности портфелей при падающем тренде.

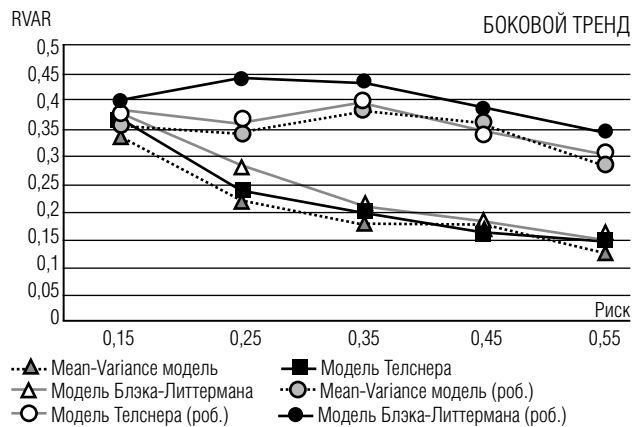


Рис. 7. Оценка качества портфеля при боковом тренде.

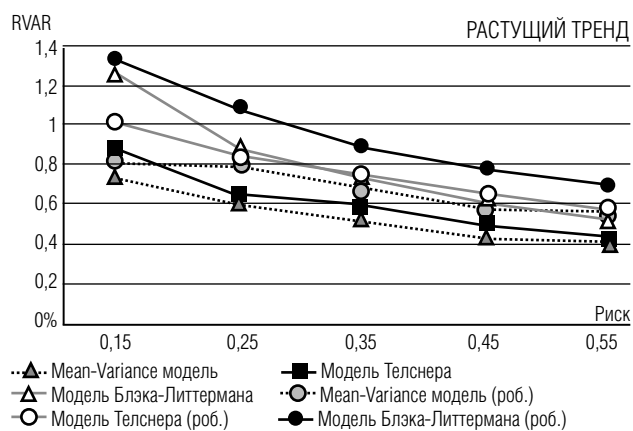


Рис. 8. Оценка качества портфеля при растущем тренде.

портфеля, которые позволили сохранить уровень диверсификации при более высоких рисках.

Обратим внимание на случай, когда рынок, вопреки всем прогнозам, сменил тренд на нисходящий. Такие ситуации характерны при появлении резких негативных новостей в сфере экономики или политики. Ниже показано, как повели себя модели при подобной ситуации (рис. 5, 6). Можно заметить, что модели в робастной постановке во время просядок показывают лучшие результаты.

Теперь перейдем к рассмотрению качества портфелей по критерию риск-доходность. Обратим внимание на поведение коэффициента Шарпа при различных уровнях риска. Ниже отображены значения коэффициента Шарпа при боковом тренде (рис. 7).

Аналогично были рассчитаны коэффициенты Шарпа при растущем тренде для различных уровней риска (рис. 8).

Такие же расчеты были проведены и для случая резкой смены тренда на негативный (рис. 9). В данном случае результат похож на результат эксперимента при растущем тренде.

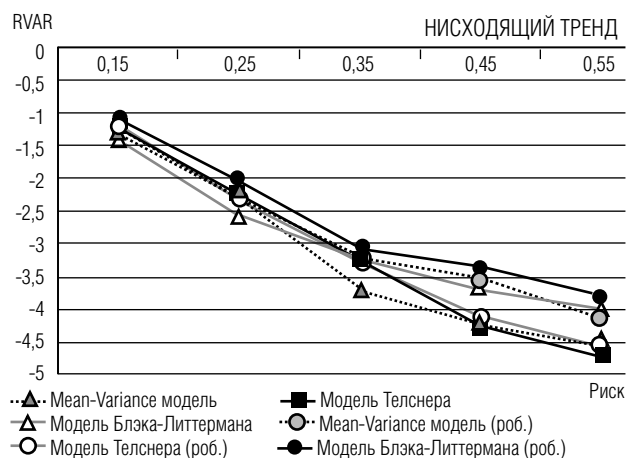


Рис. 9. Оценка качества портфеля при падающем тренде.

7. Заключение

Из результатов экспериментов можно сделать вполне ожидаемый вывод о том, что при позитивных растущих тенденциях на рынке, качество инвестиционного портфеля (совокупность риска и доходности) кардинально от структуры портфеля не зависит. В

периоды неопределенности (боковой тренд, перелом тренда, интервенции) структура портфеля играет решающую роль. В подобные периоды именно от структуры портфеля будет зависеть его качество. Эти выводы характерны для моделей в стандартной и робастной постановках. Робастные модели обладают лучшим качеством по сравнению с моделями в стандартной постановке на всех участках тренда (рис. 1-6).

Итак, в рамках представленного исследования, робастной оптимизации были подвергнуты следующие модели: классическая mean-variance модель, модель Блэка-Литтермана, модель Телсера. Приведен сравнительный анализ эффективности моделей до робастной оптимизации и после. Оценены сильные и слабые стороны разных подходов. О целесообразности использования метода робастной оптимизации говорит тот факт, что качество портфелей увеличилось до 5-21% в зависимости от участков тренда, выбранной модели и значения выбранного риска. Для оценки качества инвестиционных портфелей использовались коэффициенты, отражающие безрисковую ставку, риск и доходность портфелей. ■

Литература

1. Галиев Д.Р. Фундаментальные модели финансовых рынков и факторные модели ценообразования. Проверка применимости для анализа российского фондового рынка. // Итоговая научно-образовательная конференция студентов Казанского университета 2009 г.: Сборник статей. – Казань, 2009. – С. 262-264.
2. Bevan A., Winkelmann K. Using the Black-Litterman Asset Allocation Model: Three Years of Practical Experience // Fixed Income Research, 1998, № 6. – P. 7-18.
3. Недосекин А.О. Методологические основы моделирования финансовой деятельности с использованием нечетко-множественных описаний : дисс. докт. экон. наук. – СПб, 2004. – С. 46-51.
4. Frost P.A., Savarino J.E. For better performance: Constrained portfolio weights // Journal of Portfolio Management, 1988, № 15. – P. 29-34.
5. Chopra V.K. Improving optimization // Journal of Investing, 1993, № 8. – P. 51-59.
6. Michaud R., Richard O. The Markowitz Optimization Enigma: Is «Optimized» Optimal? // Financial Analysts Journal, 1989, № 1. – P. 31-42.
7. Международная конвергенция измерения капитала и стандартов капитала: новые подходы (Базель II). 01.07.2004. URL: <http://www.cbr.ru/today/pk/Basel.pdf> (дата обращения: 04.07.11).
8. Engels M. Portfolio Optimization. PhD's Thesis. – Leiden, 2007. – 120 p.
9. Goldfarb D., Iyengar G. Robust Portfolio Selection Problems // Mathematics of Operation Research, 2003, № 1. – P. 1-38.
10. Schutel A.S. The Black-Litterman Model For Active Portfolio Management // Journal of portfolio management, 2009, № 11. – P. 18-32.
11. Анализ портфельных инвестиций / М.Д. Миссаров, А.Г. Исавнин, И.И. Махмутов, Д.Р. Галиев; фил. Казан. ун-та - Набережные Челны : Лаб. операт. Полиграфии, 2011. – 239 с.
12. Miguel A.L. Stochastic portfolio Optimization with Round Lot Trading Constraints. 02.10.2009. URL: http://www.minlp.org/problems/mod/85/Stoc_Port_with_Round_Lot_-_MODEL.pdf (дата обращения: 05.07.2011).
13. Галиев Д.Р., Исавнин А.Г. Использование VaR-ограничений в модели Блэка-Литтермана при формировании инвестиционного портфеля // В мире научных открытий, 2011, №6. – С. 261-270.
14. Галиев Д.Р. Использование модели Блэка-Литтермана и экспертных оценок комплексного характера для эффективного управления портфелем ценных бумаг. Программная реализация и результаты применения на российском и американском рынках // Итоговая научно-образовательная конференция студентов Казанского университета 2010 г.: Сборник статей. – Казань, 2010. – С. 276-278.

КЛАССИФИКАЦИЯ ОПЕРАЦИОННЫХ РИСКОВ ПРИ СЕРВИСНО-ОРИЕНТИРОВАННОМ ПОДХОДЕ К СОЗДАНИЮ ИНФОРМАЦИОННОЙ СИСТЕМЫ

И.В. Пырлина,

преподаватель кафедры корпоративных информационных систем

Национального исследовательского университета «Высшая школа экономики»

Адрес: г. Москва, ул. Кирпичная, д. 33/5

E-mail: irina.pyrlina@gmail.com

В статье проведено исследование операционных рисков в рамках сервисного подхода создания информационных систем (ИС) и предложен подход к их классификации, основывающийся на ошибках систем, которые являются поставщиками сервисов для типового приложения с сервисно-ориентированной архитектурой. С использованием предложенной классификации собрана статистика сообщений об ошибках за 5 лет по двум предприятиям нефтегазовой и металлургической отрасли. Проведен опрос экспертов с целью определения убытков от исследованных рисков и приведена оценка возможного урона.

Ключевые слова: операционные риски, убытки, классификация ошибок, сервисно-ориентированная архитектура, нефтегазовая и металлургическая отрасль.

Введение

Современная развивающаяся компания с каждым годом совершенствуется и оптимизирует внутренние процессы и процедуры, чтобы достичь наиболее высокого уровня продаж, повысить качество и эффективность работы компании, а также одновременно сократить издержки работы и производства. Для этого необходимо внедрять и использовать новейшие технологии и системы. Причем совершенствуются не только технологии производства продукции и услуг, но и системы, поддерживающие вспомогательные процессы ор-

ганизации, к ним относятся и системы с сервисно-ориентированной архитектурой приложения.

Для того чтобы определить эффективность новой применяемой технологии или архитектуры решения, такой как сервисно-ориентированная архитектура, важно определить не только прибыль, но и стоимость убытков от ее реализации. Обычно к убыткам информационных систем относятся затраты на их внедрение и поддержку (в том числе затраты на серверное оборудование, программные средства, персонал и т.д.)¹. Однако необходимо учитывать не только планируемые издержки, но и внеплановые

затраты, к которым относятся убытки от реализации рисков работы информационных систем. Такие риски называются операционными. Они возникают достаточно часто и суммарно могут представлять ощутимую статью расходов в бюджете ИТ.

Современные информационные технологии позволяют дать более четкую оценку убытков, появившихся в результате реализации операционных рисков на основе статистического анализа. Операционные риски в рамках сервисного подхода создания информационных систем могут возникать из-за недостаточной квалификации персонала (например, при вводе некорректной информации или при неправильной интерпретации результатов) или несанкционированных действий персонала, сбоев в работе информационных систем или сбоев в работе сети, недостатков памяти и перебоев в работе серверного оборудования.

В данной статье автор предлагает подход определения типов операционных рисков от использования приложений с сервисно-ориентированной архитектурой. Проверка эффективности предложенного подхода осуществляется на базе статистики по двум предприятиям. За исследуемый 5-летний период в нефтегазовом предприятии было зафиксировано 820 ошибок в разных компонентах систем, в то время как за 4-летний период (2007-2010 гг.) в металлургическом предприятии было выставлено 916 сообщений об ошибках. На базе полученных ошибок и анализа существующих подходов была разработана классификация ошибок программного обеспечения (ПО). Задача этой работы – создать классификацию операционных рисков компании, внедряющей сервисно-ориентированные системы.

В работе под операционными рисками² понимаются потенциальные убытки, возникающие из-за ошибок работы ресурсов информационных систем с сервисно-ориентированной архитектурой, т.е. ошибок ПО, аппаратного обеспечения и неправильной работы человеческих ресурсов.

В центре сервисно-ориентированной архитектуры приложений лежит понятие сервиса (бизнес-сервиса или технического сервиса). Под бизнес-сервисом в данном случае понимается область деятельности предприятия, где реализация приложений с сервисно-ориентированной архитектурой будет иметь наибольшую выгоду для предприятия. Мы определяем бизнес-сервис как совокупность:

1. функций, объединенных по критичным для бизнеса атрибутам,
2. системных операций, обернутых в web-сервисную оболочку,
3. технических ресурсов, реализующих приложения, на базе которых работают технические сервисы (например, web-сервисы).

Под техническим сервисом, согласно классическому определению, понимаются автономные, модульные, «самоописываемые» приложения, которые предоставляют набор выполняемых функций каждому, кто их запрашивает [9]. В данном подходе технический сервис определяется в виде прикладных компонент или технических компонент, которые поддерживают работу бизнес-сервиса.

Исходя из введенных определений, операционные риски сервисно-ориентированных приложений следует определять на базе используемых ресурсов бизнес-сервиса.

1. Типы операционных рисков

Операционные риски в рамках информационных систем с сервисно-ориентированной архитектурой тесно связаны с ресурсами, потребляемыми этими приложениями, в частности с ресурсами бизнес-сервисов. Среди них стоит выделить (рис. 1) следующие типы:

◆ Человеческие ресурсы – сотрудники, вовлеченные в процесс или в сервис. На рисунке ниже отражены все ресурсы типового бизнес-сервиса. С точки зрения человеческих ресурсов, типовой бизнес-сервис включает представителей бизнес-подразделений и экспертов, отвечающих за ключевые операции сервиса, владельцев процессов и данных, авторов заявок/документов и их согласующих, также администраторов приложений сервиса.

◆ Прикладные ресурсы – прикладные программы, которые имеют возможность публикации/поддержки технических сервисов, к их числу относятся не только ПО, но и системы аппаратного обеспечения, а именно,

– Подсистема доступа – различные технологии пользовательского интерфейса для работы с системой («Тонкий» клиент на базе Web-технологий, и «Толстый» клиент в виде прикладных Windows-приложений),

¹ Для оценки затрат информационных систем используются, например, такие методологии, как анализ Совокупной Стоимости Владения (или TCO) [7].

² Общее определение термина «операционный риск» приводится в [8, стр.185].

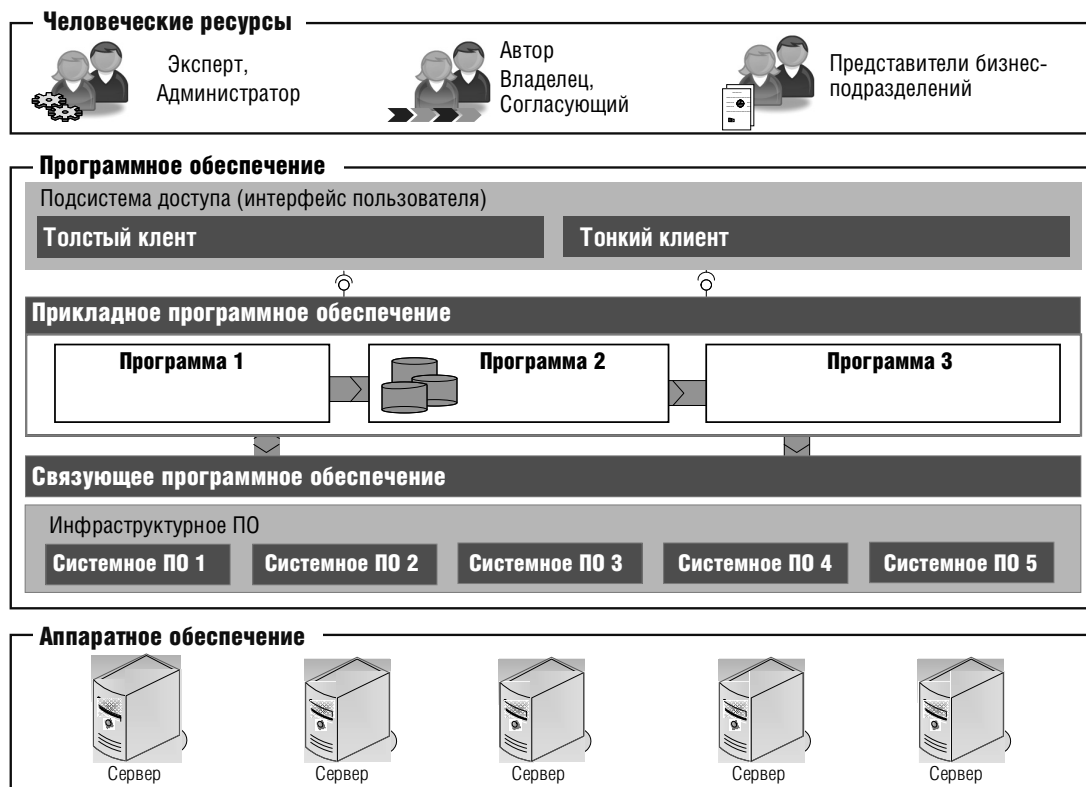


Рис. 1. Ресурсы бизнес-сервиса.

— Прикладное ПО – аналитический аппарат систем,
 — Связующее ПО, предназначенное для организации информационного обмена между всеми системами и подсистемами,

— Инфраструктурное ПО – программы, которые предназначены для организации правильной работы серверного оборудования, а также программного обеспечения, которые устанавливаются на сервере.

◆ Технические ресурсы – серверы, оборудование.

С учетом вышеизложенного выделяются следующие группы операционных рисков:

◆ Риск персонала – риски получения ущерба от некорректной или неадекватной работы персонала, а также риски, связанные с несанкционированной работой персонала,

◆ Риски прикладных систем – риски получения ущерба от выхода из строя любой компоненты программного обеспечения. Данный ущерб получается в результате сбоя работы или ошибок ПО,

◆ Технические риски – риски, связанные с выходом из строя аппаратного обеспечения (т.е. любого оборудования, на котором установлено ПО),

Под ошибкой ПО или программной ошибкой [5] понимается не достижение или отличие от запрограммированного результата, ради которого была создана программа. Ошибки ПО включают в себя:

- дефекты в ходе исполнения программы и определения данных,
- неправильный результат исполнения программы,
- некорректные действия человека, порождающие некорректный результат.

Группа рисков прикладных систем является наиболее крупной из перечисленных выше. Поэтому прежде чем оценивать возможный ущерб от приведенных групп рисков, определим более детально классификацию ошибок ПО, которая расширит общую классификацию операционных рисков приложений с сервисно-ориентированной архитектурой.

2. Обзор классификаций ошибок программного обеспечения

Как известно, качество композитных приложений³, обращающихся к сервисам других систем, зависит от количества ошибок не только самого

³ Мы определяем композитное приложение как приложение, которое использует данные и функции, предоставляемые платформами и приложениями в виде сервисов, комбинируя их в ориентированные на пользователя процессы и представления.

композитного приложения, но качества сервисов, предоставляемых другими приложениями ландшафта. Поэтому достаточно будет сформировать классификацию ошибок программного обеспечения, которое будет использоваться сервисными приложениями.

Существует большое количество подходов классификации ошибок ПО [1,2,3,4,6]. Наиболее распространенные подходы предлагают классифицировать возникающие ошибки

◆ По приоритету и серьезности их ущерба (классификация, применяемая в компании SAP AG)

- Самый высокий приоритет
- Высокий приоритет
- Средний приоритет
- Низкий приоритет

◆ По типу приложения, в котором возникла ошибка (или как указано в [1,2] по «месту их возникновения»)

- Ошибки пользовательского интерфейса
- Ошибки прикладного ПО
- Ошибки связующего ПО (или «ошибки обработки интерпретации данных» по [1])
- Ошибки системного ПО
- Ошибки аппаратного обеспечения (или аппаратные ошибки [2])

◆ По причинам возникновения ошибок (в [3] автор предлагает использовать классификацию Beizer)

- Функциональные ошибки
- Системные ошибки
- Процессные ошибки
- Ошибки данных
- Ошибки кода
- Ошибки документации
- Другие ошибки, причина которых не установлена.

◆ Также есть классификации по месту ошибок в жизненном цикле ПО (в соответствии с ГОСТ 34.601-90 стадии создания АС – формирование требований, разработка концепции АС, техническое задание, эскизный проект, технический проект, рабочая документация, ввод в действие, сопровождение АС). Однако классификация по жизненному циклу приложения не подходит в данном случае, поскольку мы рассматриваем только приложения, находящиеся в эксплуатации. Именно такие приложения как конечный продукт интересны с точки зрения обоснования их эффективности и производительности в рамках построения SOA⁴-стратегии.

⁴ SOA – Service-Oriented Architecture.

3. Новая классификация ошибок программного обеспечения и операционных рисков

Почти все рассматриваемые в рамках анализа методы не имеют статистической информации, отражающей эффективность применяемых классификаций. Только в [4] приводятся классификации с демонстрацией результатов статистических исследований производственного окружения или индустрии ИТ, однако задача этих исследований – проверить собственную классификацию ошибок программного кода и использовать эти данные для дальнейшей разработки систем (в основном являющихся собственными разработками). Классификации авторов [3,4] либо избыточны, либо преследуют специфические цели классификации программ на фазах проектирования, тестирования и реализации. К тому же в [4] приводятся недостатки существующих классификаций ошибок.

В отличие от этого в данном исследовании собиралась информация по типам ошибок систем класса ERP (т.е. крупных стандартных систем) в ландшафте обследуемых предприятий, их приоритетности, частоте возникновения и сложности устранения (т.е. длительности).

Мы предлагаем следующую классификацию ошибок программного обеспечения, которую можно применять ко всем типам программных приложений в рамках сервисно-ориентированной архитектуры

◆ Основная категория классификации

— Ошибки ввода (инициализации данных) – вывода результатов (или ошибки пользовательского интерфейса) – ошибки в определении констант, переменных, в представлении результатов работы системы.

— Функциональные ошибки – ошибки в части программы, где происходит преобразование переменных, реализация логики решения.

— Ошибки связующего ПО (или «ошибки обработки интерпретации данных» по [1]) включают в себя проблемы при передаче данных между подпрограммами, искажение передаваемой информации или ошибки при обмене сообщениями.

— Ошибки данных – ошибки приложения, которые изменяют данные в любом типе хранилища данных (данный класс введен автором [6] для web-приложений, что применимо и для сервисно-ориентированной архитектуры).

— Системные ошибки – ошибки программно-окружения, конфигурации сервера, проблемы тайм-аута, ошибки памяти, ошибки прав пользователей, проблемы производительности, ошибки сети, ошибки баз данных, ошибки операционной системы и аппаратного обеспечения (включают проблемы тайм-аута⁵, ошибки памяти⁶, и также проблемы нехватки ресурсов при повышении нагрузки), а также ошибки при развертывании и сопровождении систем и обеспечении безопасности.

— Не ошибка – к этой группе относятся ошибки, которые были устранены без изменения настройки или изменения кода, ошибки в описании решения, или отсутствие описания решения.

❖ Также при классификации ошибок мы будем указывать приоритет ошибки:

— Очень высокий (1) – ошибки, которые имеют или могут оказать критическое влияние на продуктивную эксплуатацию системы, на бизнес процессы и операции, а также могут привести к краху или полной остановке работы системы;

— Высокий приоритет (2) имеют ошибки, которые оказывают существенное влияние на бизнес-операции, а также ведут к заметному снижению производительности системы;

— Средний приоритет (3) имеют ошибки, которые оказывают влияние на функциональность системы и соответственно могут повлиять на бизнес-операции, однако не являются критическими;

— Низкий приоритет (4) получают ошибки, прозрачные для пользователя. Например, ошибки пользовательского интерфейса не нарушают возможность использования системы, однако не удобны для использования и требуют устранения.

❖ Устранение ошибки или длительность рассмотрения – то, как быстро была решена проблема, характеризует сложность проблемы. Длительность рассчитывается от даты подачи заявления до даты, когда проблема считается решенной, т.е. выставлен статус «completed».

Полученная статистика в результате применения классификации ошибок при внедрении сервисно-ориентированных приложений для двух

предприятий металлургической и нефтегазовой отрасли приведена в таблице (табл. 1. Ошибки ПО нефтегазового клиента за 2006-2010 (выборка)). Здесь приведена лишь выборка наиболее показательных сообщений об ошибках из общей статистической базы.⁷

Таблица 1.

Ошибки ПО нефтегазового клиента за 2006-2010 (выборка)

Но-мер	Короткий текст ошибок	Основная категория	Приоритет
1	Credential for the Adobe Interactive Forms scenario	Ошибки ввода – вывода результатов	4
2	Anonymous users don't see KM content	Ошибки ввода – вывода результатов	3
3	Error with system data editing in support portal	Ошибки ввода – вывода результатов	3
4	Not all components are inserted after BOM explosion	Функциональные ошибки	2
5	Create new cancellation document after cancellation	Функциональные ошибки	2
6	Report 'Price Comparison' fails	Функциональные ошибки	2
7	Different configuration screens in Integration Builder	Ошибки связующего ПО	4
8	Middleware — Bdoc validation error	Ошибки связующего ПО	2
9	Integration Directory-Communication Channel not found	Ошибки связующего ПО	3
10	Authority for master data in BI	Ошибки данных	3
11	Master data time interval error	Ошибки данных	2
12	product catalog replication	Ошибки данных	2
13	Clustering information in SDB	Системные ошибки	2
14	High CPU consumption by a server node	Системные ошибки	2
15	OutOfMemoryError in during export	Системные ошибки	3
16	SAP Solution Manager Preparation Service	Не ошибка	3
17	Incorrect create SD message from ECP system	Не ошибка	3
18	Test message	Не ошибка	4

У каждого сообщения об ошибке собиралась дополнительно к основной категории информация по приоритету и длительности устранения ошибок, чтобы понять важность и критичность возникающих ошибок в каждой группе. Суммарно собрана статистика за 2006-2010 у нефтегазового и металлургического предприятий (табл. 2).

⁵ Мы определяем тайм-аут как перерыв в работе приложений, связанный со слишком высокой нагрузкой на него.

⁶ Ошибки памяти – ошибки, связанные с некорректным обращением к оперативной памяти.

⁷ Автор приносит свои извинения за текст сообщений на английском языке. Система сбора сообщений об ошибках работает только на английском языке и соответственно предъявляет языковые требования к выставляемым сообщениям.

Таблица 2.

Статистика по двум компаниям

Год	Нефтегазовое предприятие	Металлургическое предприятие
2006	106	0
2007	153	21
2008	200	207
2009	166	327
2010	156	264

В таблице (табл. 3) и на диаграмме (диаграмма 1) ниже приведены итоговые результаты по количеству собранных ошибок. Для примера в каждой группе в таблице указано по 3 ошибки. Рисунок же отражает итоговые значения по всем собранным сообщениям по все группам ошибок. На базе приведенных результатов видно, что наибольшее количество ошибок за 5-летний срок наблюдается в категориях «функциональные ошибки» и «системные ошибки». Данная статистика показывает, что внедренные системы всегда имеют недостатки или недоработанные функции сразу после выхода в продуктивную эксплуатацию, а также вероятно были сделаны ошибки при первоначальной оценке необходимого системного ландшафта и при настройке функциональности. Важно также отметить, что статистика демонстрирует первые 5 лет эксплуатации информационных систем сразу после их внедрения.

Таблица 3.

Статистика по каждой группе ошибок

Основная категория (2006-2010)	Количество ошибок						%
	2010	2009	2008	2007	2006	Итого	
1. Ошибки ввода (инициализации данных) – вывода результатов	21	21	95	22	3	168	10
2. Функциональные ошибки	365	269	166	68	62	787	48
3. Ошибки связующего ПО	79	26	21	12	2	110	7
4. Ошибки данных	29	37	19	9	10	92	6
5. Системные ошибки	170	114	86	53	24	410	25
6. Не ошибка	9	26	20	10	5	71	4

На диаграмме (диаграмма 1) отражена проанализированная статистика за 2006-2010 гг. в рамках каждого класса. На первой диаграмме показано количество ошибок в каждом классе и штриховкой обозначены приоритеты полученных заявок. Самое большое количество заявок попадает в средний приоритет, на втором месте высокий приоритет. Крайне редко выставляются заявки низкого или очень высокого приоритета. На второй диаграмме отражена также средняя длительность устранения ошибки для каждой категории ошибок. Общая длительность устранения ошибок в среднем мала. Самыми длительными являются сроки устранения в

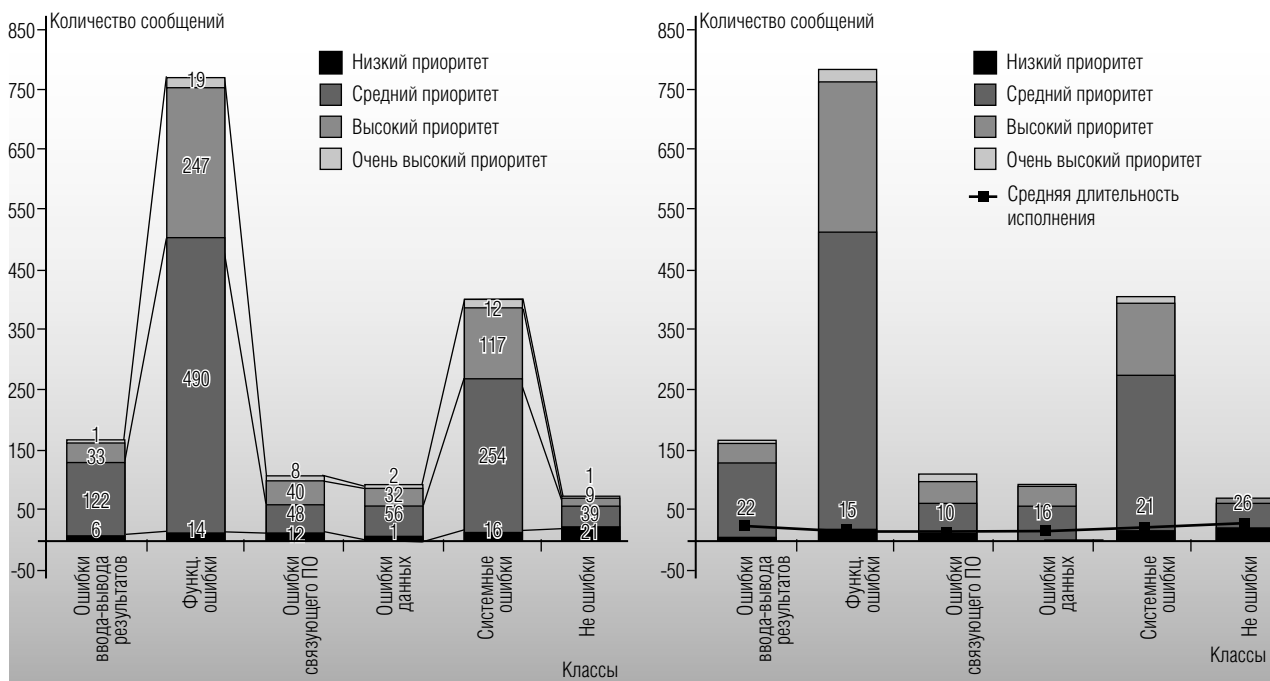


Рис. 1. Диаграмма: основные категории, приоритет и статус (2006-2010).

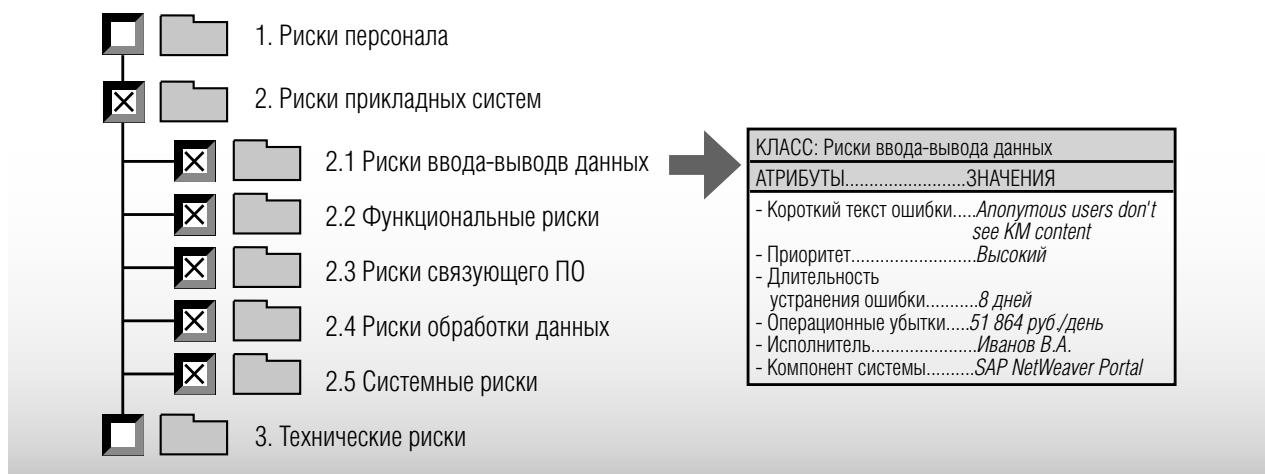


Рис. 2. Классификация операционных рисков при сервисно-ориентированном подходе к созданию ИС.

категории Не ошибка (в среднем 26 дней), Ошибки ввода-вывода результатов (в среднем 22 дня) и Системные ошибки (в среднем 21 день). Минимальные сроки устранения ошибок в категории Ошибки связующего ПО – 10 дней в среднем.

Предлагается детализировать классификацию операционных рисков при сервисно-ориентированном подходе к созданию информационных систем на основе типов ошибок программного обеспечения, приведенных выше. Каждому идентифицированному типу ошибок соответствует класс риска, основные атрибуты которого зависят от приоритета ошибок и длительности устранения ошибок. Соответствующая классификация рисков приобретает следующий вид (рис. 2).

На базе выявленных категорий ошибок (или операционных рисков) можно провести анализ операционных убытков информационных систем с сервисно-ориентированной архитектурой.

4. Операционные убытки при реализации рисков

Автором был проведен опрос с целью оценки убытков при отсутствии решения по выставленным сообщениям об ошибках. Экспертами компании SAP была выставлена оценка времени использования в день систем, на которые были выставлены сообщения об ошибках в соответствующих предприятиях, а также количество пользователей, использующих одновременно данную систему в день. Оценка проводилась по выборке сообщений, представленных выше. На базе полу-

ченных оценок был произведен анализ операционных убытков (в руб. в день, табл. 4)

Кроме того, при расчете величины убытков были использованы три варианта стоимости человеко-дня. Каждый уровень соответствует трем ролям сотрудников (инженер, ведущий инженер, менеджер). Разные приоритеты сообщений об ошибках в среднем по статистике выставляются разными группами пользователей. В среднем здесь наблюдается зависимость приоритета сообщения от роли пользователя

- низкому приоритету ошибок соответствует стоимость дня инженеров,
- среднему и высокому приоритетам мы поставили в соответствие стоимость ведущего инженера,
- очень высокий приоритет оценен стоимостью дня менеджера.

Стоимость системных ошибок будет зависеть от стоимости дня администратора. Также стоит отметить, что стоимость указанных дней включает в себя не только заработную плату соответствующих специалистов, но и административные затраты на сотрудников и налоги. Таким образом, убытки за каждый день не решения проблемы, указанной в сообщении об ошибке, рассчитываются по формуле

$$Убытки = U \cdot Q_u \cdot W_t, \text{ где}$$

- U – % использования функции в день,
- Q_u – количество сотрудников, использующих данную функциональность,
- W_t – стоимость человеко-дня в зависимости от роли и приоритета сообщения.

Таблица 4.

Операционные убытки нефтегазового клиента за 2006-2010 (выборка)

Номер	Короткий текст ошибок	Основная категория	Приоритет	Устранение ошибок (дни)	Операционные убытки (в руб./день)
1	Credential for the Adobe Interactive Forms scenario	Ошибки ввода – вывода результатов	4	1	5 186
2	Anonymous users don't see KM content	Ошибки ввода – вывода результатов	3	8	51 864
3	Error with system data editing in support portal	Ошибки ввода – вывода результатов	3	8	51 864
4	Not all components are inserted after BOM explosion	Функциональные ошибки	2	21	66 682
5	Create new cancellation document after cancellation	Функциональные ошибки	2	11	66 682
6	Report 'Price Comparison' fails	Функциональные ошибки	2	36	133 364
7	Different configuration screens in Integration Builder	Ошибки связующего ПО	4	4	15 559
8	Middleware - Bdoc validation error	Ошибки связующего ПО	2	1	66 682
9	integration Directory-Communication Channel not found	Ошибки связующего ПО	3	2	77 796
10	Authority for master data in BI	Ошибки данных	3	39	25 932
11	master data time interval error	Ошибки данных	2	1	66 682
12	product catalog replication	Ошибки данных	2	5	33 341
13	Clustering information in SDB	Системные ошибки	2	14	51 864
14	High CPU consumption by a server node	Системные ошибки	2	7	518 636
15	OutOfMemoryError in during export	Системные ошибки	3	4	2 593

Дополнительно в таблице отражено общее количество дней, которое потребовалось для устранения ошибки, что дает представление об общей величине операционных убытков ИТ и по сути является примером операционных убытков прикладных систем с сервисно-ориентированной архитектурой.

Заключение

В работе предложен подход к классификации типов операционных рисков от использования приложений с сервисно-ориентированной архитектурой. Установлено, что при их классификации стоит использовать как типы ресурсов, так и категории программного обеспечения. Такой подход расширяет

классификацию операционных рисков и облегчает поиск, сбор и анализ статистики операционных рисков. Кроме того, мы опробовали и привели результаты использования разработанной классификации на статистических данных двух компаний, что позволяет судить о ее эффективности. И, наконец, удалось оценить операционные убытки при отсутствии решения по выставленным сообщениям об ошибках. Определенные в данном исследовании группы операционных рисков послужат основой для модели операционных рисков систем с сервисно-ориентированной архитектурой. Собранные статистика будет использоваться при имитации работы систем данного типа.

Автор благодарен профессору Ф.Т. Алескерову за постановку задачи. ■

Литература

1. Канер С., Фолк Дж., Нгуен Е.К. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. – ДиаСофт, 2001.
2. Майерс Г. Надежность программного обеспечения. – М.: Мир, 1980.
3. Krawczyk H., Wiszniewski B., Mork H. Classification of software defects in parallel programs // HPCTI Progress Report no1, April 1995.
4. Ostrand T.J., Weyuker E.J. Collecting and Categorizing Software Error Data in an Industrial Environment // The Journal of Systems and Software 4, 1984. P. 289-300.
5. IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 729-1983 // Inst. Electrical and Electronics Eng., New York, 1983.
6. Guo Y., Sampath S. Web Application Fault Classification—An Exploratory Study // ESEM'08 Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement ACM New York, NY, USA, 2008. URL: <http://portal.acm.org/citation.cfm?id=1414060>
7. SAP TCO Framework: A Framework to Reduce Total Cost of Ownership // SAP White Paper. 2005. SAP AG.
8. Алескеров Ф.Т., Андриевская И.К., Пеникас Г.И., Солодов В.М. Анализ математических моделей Базель II. – М.: Физматлит, 2010.
9. Haas H., Brown A. Web Services Glossary. W3C Working Group Note 11 February 2004. – W3C. URL: <http://www.w3.org/TR/ws-gloss/>

ЗАДАЧА ВЫБОРА ЧИСЛА И МЕСТ РАЗМЕЩЕНИЯ ЦЕНТРОВ ХРАНЕНИЯ И ОБРАБОТКИ ИНФОРМАЦИИ В КОМПЬЮТЕРНОЙ СЕТИ

О.В. Есиков,

профессор, доктор технических наук, начальник Управления информационных и аналитических технологий аппарата Администрации Тульской области

Д.В. Изотов,

*программист департамента внутренних сервисов
ЗАО «Торговый дом «Перекресток», Москва*

Адрес: г. Москва, ул. Средняя Калитниковская, д. 28 стр. 4

E-mail: dvikti@km.ru

В статье сформулирована задача выбора числа и мест размещения центров хранения и обработки информации в компьютерной сети по критерию максимума интенсивности поступления запросов на информационное обслуживание. Проведен анализ характера информационных процессов в распределённой компьютерной сети.

Ключевые слова: центры хранения и обработки информации, компьютерная сеть, моделирование информационных процессов, центр графа.

Задача выбора числа и мест размещения центров хранения и обработки информации (ЦХИ) в компьютерной сети (КС) по критерию максимума интенсивности поступления запросов на информационное обслуживание может быть сформулирована следующим образом: требуется определить минимальное число ЦХИ, обслуживающих информационные запросы от автоматизированных рабочих мест должностных лиц (АРМ

ДЛ), и такое их размещение в узлах сети, чтобы значение времени задержки передачи сообщения для каждого АРМ ДЛ не превышало допустимой величины, а суммарная приведенная интенсивность поступления запросов на узлы КС, в которых будут размещены ЦХИ, была при этом максимально возможной. Эта задача относится к так называемым задачам о размещении, например, центров скорой помощи, складов и т.п.

Известные математические постановки задач о размещении представляют собой частные случаи классических задач теории графов – «задачи о p -медиане» и «задачи о p -центрах» [1-7], а также «задачи о назначении» [8,9], для решения которых, как правило, используются алгоритмы, основанные на идеях метода ветвей и границ. Этим задачам посвящён ряд работ таких авторов, как: S.L. Hakimi, H. Noltermeier, J. Spoerhose, H. Кристофидес, В.Л. Береснев, Э.Х. Гимади, В.Т. Дементьев, Е.В. Алексеева, Ю.А. Кочетов, Г.Г. Забудский и другие.

В настоящей статье изложен подход к решению задачи, сформулированной применительно к размещению ЦХИ в компьютерной сети на основе известной в теории графов задачи о p -центрах.

С целью формирования математической модели задачи выбора ЦХИ предварительно с применением теории массового обслуживания рассматривается процесс передачи и обслуживания информационных запросов в сети.

Время задержки сообщений в сети можно определить как отрезок времени между моментом начала ввода информации в исходном АРМ ДЛ и моментом получения последнего знака сообщения в узле адресата.

Стохастичность поступления данных и недетерминированный характер их обработки должны быть учтены в ходе моделирования информационных процессов, протекающих в КС. Предположением, необходимым для возможности использования аналитических моделей массового обслуживания в этом случае, можно принять предположение о том, что длительности передачи сообщения (пакета) по разным каналам передачи данных являются независимыми случайными величинами. Для расчета задержек в коммуникационных сетях широкое рас-

пространение получила модель сети массового обслуживания, предложенная в [10]. Суть ее состоит в следующем. В коммуникационной сети с коммутацией сообщений (или пакетов) имеется N узлов и W каналов связи, каждый из которых интерпретируется как система массового обслуживания $M/M/1$ (рис. 1). И каналы связи, и узлы абсолютно надежные. Пропускная способность w -го канала связи C_w , бит/с, $w=1, \dots, W$.

Узлы выполняют операции по коммутации сообщений, включая их редактирование, выбор маршрутов, буферизацию и т.д. Предполагается, что время обработки в узле постоянно и, более того, пренебрежимо мало. Имеются очереди к каналам связи. При передаче сообщений возникают задержки. В узлы поступает пуассоновский поток запросов, который можно определить как поток между каждой парой узлов КС со средней интенсивностью λ_{ij} , $i, j=1, \dots, N$, $i \neq j$. Объемы сообщений независимы и распределены по показательному закону со средним значением $\frac{1}{\mu}$.

Для размещения этих сообщений в узлах сети имеется память неограниченного объема.

Обозначим λ_w – среднюю интенсивность потока сообщений по w -му каналу связи, тогда

$$\lambda_w = \sum_{i=1}^N \sum_{j=1}^N \lambda_{ij} z_{ij}(w), \quad w=1, \dots, W,$$

где $\|z_{ij}(w)\|$ – матрица маршрутов передачи данных; $z_{ij}(w)=1$, если при передаче информации из i -го узла сети в j -й она проходит по w -му каналу передачи данных; 0 – в противном случае.

Время задержки сообщений в w -ом канале связи, в соответствии с моделью $M/M/1$, определяется по следующей формуле

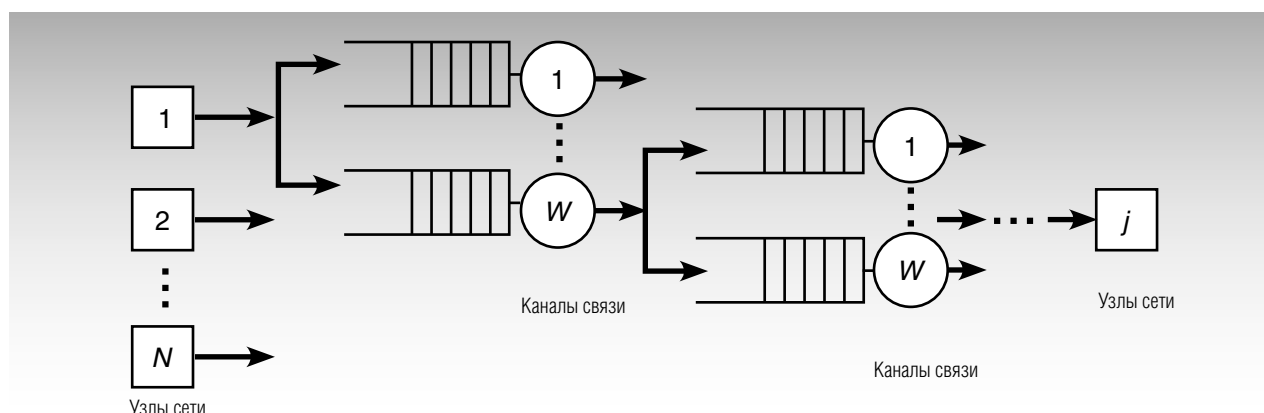


Рис. 1. Сеть массового обслуживания для моделирования информационных процессов в КС.

$$T_w = \frac{1}{\mu C_w - \lambda_w}, w = 1, \dots, W.$$

В реальности технологий обслуживания может быть очень много. Это беспriorитетное и priorитетное обслуживание одним или несколькими приборами, многофазное обслуживание, обслуживание в режиме разделения времени и т.д. Так, например, в [10] показана возможность приближенного расчета задержек в коммуникационных сетях с непоказательными длинами пакетов. Канал связи в этом случае моделируется системой $M/G/1$. Под сообщениями понимаются не только запросы пользователей на получение выходных документов, но и заявки на пересылку файлов или ввод в базу данных информации из входных документов, а также некоторые технологические заявки по управлению вычислительным процессом или контролю над ним.

Для более адекватного отображения функционирования КС и снятия ограничения с учётом введенных выше предположений предлагается использование имитационной модели [11].

Для формализации задачи выбора числа и мест размещения ЦХИ компьютерную сеть представим в виде неориентированного графа $G=(X, \Gamma)$, вершины $x_i, i = 1, \dots, N$, которого соответствуют узлам сети, а дуги $g_w, w = 1, \dots, W$ – каналам связи. Тогда «длины» дуг графа $G - T_w$ – образуют матрицу времен задержки передачи сообщения между соответствующими узлами сети (далее для удобства – «время передачи»). Веса q_j , соответствующие вершинам графа, определяют суммарный объем запросов на соответствующие этим вершинам узлы КС.

Для любой вершины x_i графа $G=(X, \Gamma)$ пусть $R^o(x_i)$ есть множество тех вершин x_j графа G , которые достижимы из вершины x_i с помощью путей с взвешенными «длинами» $q_j T(x_i, x_j)$, не превосходящими величины T_{max} , где $T(x_i, x_j)$ – «длина» кратчайшего пути от вершины x_i до вершины x_j .

Через $R^i(x_i)$ обозначим множество тех вершин графа G , из которых вершина x_i может быть достигнута с использованием путей, имеющих взвешенные длины $q_j T(x_j, x_i) \geq T_{max}$.

Таким образом,

$$\begin{aligned} R^o(x_i) &= \{x_j \mid q_j T(x_i, x_j) \leq T_{max}, x_j \in X\} \text{ и} \\ R^i(x_i) &= \{x_j \mid q_j T(x_j, x_i) \leq T_{max}, x_j \in X\}. \end{aligned} \quad (1)$$

Для каждой вершины x_i , соответственно из множеств $R^o(x_i)$ и $R^i(x_i)$, определим следующие два числа:

$$\begin{aligned} s_o(x_i) &= \max[q_j T(x_i, x_j)] \text{ и} \\ s_i(x_i) &= \max[q_j T(x_j, x_i)]. \end{aligned} \quad (2)$$

Числа $s_o(x_i)$ и $s_i(x_i)$ называются соответственно числом внешнего разделения и числом внутреннего разделения вершины x_i .

Если T_o – наименьшая «длина» T , такая, что для вершины x_i

$$R^o(x_i) = X,$$

(т.е. все вершины графа G достижимы из x_i с использованием путей, взвешенные «длины» которых не превосходят T_o , причем T_o – наименьшее из таких чисел), то из соотношений (1) и (2) следует равенство

$$s_o(x_i) = T_o.$$

Аналогично, если T_i – такая наименьшая длина T , что

$$R^i(x_i) = X, \text{ то}$$

$$s_i(x_i) = T_i.$$

Вершина x_o , для которой

$$s_o(x_o) = \min[s_o(x_i)],$$

(т.е. вершина, соответствующая минимальному из всех чисел $s_o(x_i)$, ранее определённых по указанному выше правилу) называется внешним центром графа G .

Аналогично вершина x_i , для которой

$$s_i(x_i) = \min[s_i(x_i)]$$

называется внутренним центром графа G .

Понятие центра графа допускает следующее обобщение: можно рассматривать не отдельную точку (центр), а множество из p точек, которые образуют кратный центр (p -центр).

Пусть X_p – подмножество (содержащее p вершин) множества X вершин графа $G=(X, \Gamma)$. Через $T(X_p, x_i)$ будем обозначать наикратчайшее из расстояний между вершинами множества X_p и вершиной x_i , т.е.

$$T(X_p, x_i) = \min [T(x_j, x_i)], x_j \in X_p, x_i \in X;$$

Аналогично

$$T(x_i, X_p) = \min [T(x_i, x_j)], x_i \in X_p, x_j \in X.$$

Подобно тому, как определялись числа разделения вершин, можно определить числа разделения для множества вершин:

$$s_o(X_p) = \max [q_j T(X_p, x_j)] \text{ и}$$

$$s_i(X_p) = \max [q_j T(x_j, X_p)], \text{ где}$$

$s_o(X_p)$ и $s_i(X_p)$ – числа внешнего и внутреннего разделения множества X_p .

Множество $X_{p_0}^*$, для которого

$$s_o(X_{p_0}^*) = \min [s_o(X_p)],$$

называется p -кратным внешним центром графа G ; аналогично определяется p -кратный внутренний центр $X_{p_i}^*$ [1].

Следует отметить, что числа внутреннего и внешнего разделения множества X_p рассматриваются при условии достижимости любой вершины графа из множества X_p , что вполне возможно в случае решения задачи для неориентированного графа, соответствующего распределенной компьютерной сети с допустимой передачей данных по каналам связи во всех направлениях.

Очевидно, что внешний и внутренний центры неориентированного графа совпадают, так как в этом случае числа разделения $s_o(X_p)$ и $s_i(X_p)$ равны между собой для любого множества X_p .

Таким образом, исходя из вышеизложенного, задача выбора числа ЦХИ и их размещения в узлах сети будет состоять в нахождении p -центров соответствующего графа G для различных значений p до тех пор, пока число разделения p -центра не станет меньше или равно заданной величине. Полученное (последнее) значение числа p будет наименьшим числом ЦХИ, а p -центр – их оптимальным размещением, удовлетворяющим предъявляемым требованиям.

Исходя из выше изложенного, общая задача определения p -центра применительно к выбору ЦХИ в распределенной компьютерной сети может быть сформулирована следующим образом [12].

Для заданного «критического» расстояния найти такое наименьшее число центров и такое их размещение, чтобы все вершины графа лежали в пределах этого критического расстояния (по крайней мере, каждая вершина – от ближайшего к ней центра).

Очевидно, что центры графа легко могут быть получены из матрицы весов дуг графа. Однако найти полным перебором p -центр можно лишь для небольших графов и для небольших значений величины p . При таком подходе надо построить всевозможные множества вершин $X_p \subseteq X$, содержащие p вершин, а затем непосредственно найти множества $X_{p_0}^*$ и $X_{p_i}^*$. Этот процесс потребует выполнить

$$p \times (N-p) \times \binom{N}{p}$$

сравнений, что даже при небольших значениях N и p неприемлемо много.

Для нахождения p -центра существует ряд эвристических методов, например, метод Сингера [1]. Они имеют низкую эффективность по быстродействию.

Для уменьшения числа переборов предлагается метод, основанный на сведении задачи о p -центрах к задаче о покрытии. Идея такого метода состоит в следующем.

На предварительном этапе матрица весов дуг преобразуется в матрицу $\|d_{ij}\|$, элементами которой являются кратчайшие пути между всеми парами вершин графа G . Затем, исходя из полученной матрицы $\|d_{ij}\|$, составляется бинарная матрица (матрица покрытий) $\|a_{ij}\|$, каждый элемент которой равен 1, если i -я вершина достижима из j -й вершины в пределах критического расстояния, т. е. если $d_{ji} \leq T_{max}$, и 0 – в противном случае. На заключительном этапе решается задача о покрытии, исходными данными для которой будут являться матрица покрытий $\|a_{ij}\|$ и веса q_i , приписанные вершинам графа G , а результатом – искомым p -центр.

Рассмотрим математическую формулировку задачи о покрытии, применительно к которой можно выбрать достаточной эффективный метод решения.

Используя матрицу кратчайших путей $D = \|d_{ij}\|$, перейдем к матрице покрытий $A = \|a_{ij}\|$, которая составляется по следующим правилам:

$$a_{ij} = \begin{cases} 1, & \text{если } d_{ij} \leq T_{max} \\ 0, & \text{в противном случае.} \end{cases}$$

Задача о нахождении p -центра на графе, удовлетворяющего заданным условиям, сводится к нахождению такого наименьшего множества X^* , чтобы из каждой вершины графа была достижима в пределах заданного расстояния T_{max} хотя бы одна вершина, входящая в множество X^* , т.е. чтобы

$$\sum_{i \in X^*} a_{ij} \geq 1, \quad j=1, \dots, N.$$

Сумма $\sum_{i \in X^*} q_i$ должна быть при этом максимально возможной (для обеспечения минимума передаваемой в сети информации).

Таблица 1.

№ уз-ла	1	2	3	4	5	6	7	8	9	10	11	12	13	14	q
1	0	25	6	12	11	4	14	17	21	22	23	24	20	19	36
2	25	0	7	10	12	6	10	10	11	14	15	16	17	11	18
3	6	7	0	7	5	5	9	5	5	7	8	9	10	12	5
4	12	10	7	0	4	3	2	4	3	4	8	7	6	5	6
5	11	12	5	4	0	5	6	7	7	8	9	9	8	7	6
6	4	6	5	3	5	0	3	4	7	8	9	12	18	13	6
7	14	10	9	2	6	3	0	8	8	9	10	10	9	8	9
8	17	10	5	4	7	4	8	0	4	5	7	9	9	8	10
9	21	11	5	3	7	7	8	4	0	7	8	9	7	6	12
10	22	14	7	4	8	8	9	5	7	0	6	11	12	9	12
11	23	15	8	8	9	9	10	7	8	6	0	17	12	13	12
12	24	16	9	7	9	12	10	9	9	11	17	0	11	10	12
13	20	17	10	6	8	18	9	9	7	12	12	11	0	12	6
14	19	11	12	5	7	13	8	8	6	9	13	10	12	0	9

Эта задача может быть сформулирована в виде задачи целочисленного линейного программирования с булевыми переменными следующим образом.

Требуется максимизировать целевую функцию

$$Q = \sum_{i=1}^N q_i x_i \quad (3)$$

при ограничениях

$$\sum_{i=1}^N a_{ij} x_i \geq 1, \quad j=1, \dots, N, \quad (4)$$

$$x_i \in \{0; 1\}, \quad i=1, \dots, N, \quad (5)$$

$$q_i \geq 0;$$

$$a_{ij} = \begin{cases} 1, & \text{если } d_{ij} \leq T_{max} \\ 0, & \text{в противном случае.} \end{cases}$$

Задача (3) – (5) относится к широкому классу задач, которые носят название задач о покрытии. Она является задачей целочисленного программирования с булевыми переменными и, следовательно, может быть решена общими методами целочисленного программирования [13].

Из теории алгоритмов известно, что задача о покрытии относится к классу универсальных переборных задач, или NP-полных задач [14]. Для её решения до сих пор неизвестны, а возможно – и не существуют, полиномиальные алгоритмы. Для всех известных алгоритмов решения универсальных переборных задач время счета растет экспоненциально с ростом размерности задачи. Однако характер

экспоненциальной зависимости существенно зависит от особенностей каждого алгоритма.

В [12] предлагается метод ветвей и границ. Специфические особенности задачи (3) – (5) позволяют значительно упростить процедуру вычисления нижней границы решения на основе использования двойственной задачи [15].

Рассмотрим решение задачи по критерию максимума интенсивности поступления запросов на узлы сети. С помощью имитационного моделирования получены следующие исходные данные, которые приведены в табл. 1, где содержатся времена задержек (в секундах) информации при ее передаче из одного узла сети в другой. Значения q представляют собой «веса» узлов сети.

В табл. 2 приведены результаты решения данной задачи, где для различных значений заданного времени задержки сообщения в сети указаны номера узлов сети, в которых размещаются ЦХИ (р-центр графа).

Таблица 2.

Заданное время задержки передачи сообщения в сети, с	Номера узлов сети, в которых размещаются ЦХИ (р-центр)
1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
2	1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
3	1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14
4	1, 2, 3, 4, 11, 12, 13, 14
5	1, 2, 4, 8, 11, 12
6	1, 2, 7, 8
7	1, 2, 8
8	2, 8
9	2, 8
10	8
15	1
20	1

Полученные данные позволяют определить оптимальную, с точки зрения выбранного критерия, конфигурацию сети для заданного времени задержки передачи сообщения, либо для заданного количества ЦХИ и их размещения в сети определить, какая будет при этом максимальная задержка сообщения. ■

Литература

1. Кристофидес Н. Теория графов. Алгоритмический подход. – М: Мир, 1978. – 432 с.
2. Алексеева Е.В., Орлов А.В. Генетический алгоритм для конкурентной задачи о р-медиане // Труды 14 Байкальской международной школы-семинара «Методы оптимизации и их приложения». Том 1. – Северобайкальск: 2008. – с. 570-585.
3. Забудский Г.Г., Филимонов Д.В. Решение дискретной минимаксной задачи размещения на сети // Известия вузов. Математика, № 5 – Омск: 2004. – с. 33-36.
4. Кочетов Ю.А., Кононов А.В., Плясунов А.В. Конкурентные модели размещения производства // Журнал вычислительной математики и математической физики. Т. 49, № 6. – 2009. – с. 1-17.
5. Nakimi S.L. On locating new facilities in a competitive environment. European J. Oper. Res. 1983. V. 12, P. 29–35.
6. Noltermeier H., Spoerhose J., Wirth H.C. Multiple voting location and single voting location on trees. European J. Oper. Res. 2007. V. 181. P. 65-667.
7. Spoerhose J., Wirth H.C. (r,p)-Centroid problems on paths and trees. Tech. Report 441, Inst. Comp. Science, University of Würzburg, 2008.
8. Береснев В.Л., Гимади Э.Х., Дементьев В.Т. Экстремальные задачи стандартизации. – Новосибирск: Наука, 1978. – 335с.
9. Гимади Э.Х. Обоснование априорных оценок качества приближенного решения задачи стандартизации // Управляемые системы. – Новосибирск, 1987. – Вып. 27. – с. 12-27.
10. Клейнрок Л. Вычислительные системы с очередями: Пер. с англ. – М.: Мир, 1979. – 600 с.
11. Акиншин Н.С., Подчуфаров Ю.Б., Изотов В.Н., Комогорцев П.В. Имитационная модель для решения задачи синтеза физической структуры АСУ нового поколения. // Оборонная техника. 1997. № 3-4. – С. 99-103.
12. Изотов Д.В. Оптимальное размещение центров хранения и обработки информации по критерию максимума интенсивности запросов / Е.В. Ларкин, Д.В. Изотов // Журнал «Прикладная информатика», № 3 (33). – М: Изд-во ООО «Маркет ДС Корпорейшн». – 2011. – с. 37 - 42.
13. Финкельштейн Ю.Ю., Корбут А.А. Дискретное программирование. – М.: Наука, 1969. – 368 с.
14. Современное состояние теории исследования операций / Под ред. Н. Н. Моисеева. – М: Наука, 1979. – 303 с.
15. Алексеев О. Г., Григорьев В. Ф. Некоторые алгоритмы решения задачи о покрытии и их экспериментальная проверка на ЭВМ // ЖВМ и МФ. – 1984. – Т.24, №10. – с. 1565-1570.



INDICATORS OF THE CONDITION AND FACTORS OF DEVELOPMENT OF ICT IN REPUBLIC OF AZERBAIJAN

R. Karayev, R. Gulmamedov, N. Sadikhova, M. Nagiyev

Annotation

Importance of timely and exact data presentation about conditions of ICT for acceptance of directive and investment decisions in modern information economy is marked. The data reflecting indicators of a condition and factors of development of ICT in Republic of Azerbaijan are resulted.

Key words: information-communication technologies, indicators of the condition, factors of the development, Azerbaijan.



DESIGN AND CONCEPT OF AN INTELLECTUAL SITUATIONAL AWARENESS AND SECURITY SYSTEM FOR RUSSIAN RAILWAYS

S. Avdoshin, M. Gorbatovskiy, A. Chernov

Annotation

Architecture and components are offered to design an intellectual system of situational awareness and security for transportation and railroads infrastructure are offered. The concept provides event-based streams processing approach to process the primary data and build a grid application of collecting and analysing railroads data from various sources. An architecture is proposed to support predictive analytics and decision support in real-time on very large data volumes as well as an approach to dynamically optimize the timetables on the railway infrastructure networks.

Key words: Streams processing. Situational awareness. Dynamic optimization. Predictive behavior-based modelling. Sensor networks.



ASSESSMENT OF STRATEGIC DECISIONS OF INFORMATION TECHNOLOGIES SERVICE

T. Kravchenko

Annotation

In the paper an approach to identification of characteristics for assessment of IT strategic decisions is proposed. The main feature of the approach is associated with integration of Balanced Scorecard methodology for IT service (IT Balanced Scorecard) and COBIT standard. Such integration allows to describe a hierarchical structure of characteristics (metrics) for assessment of decisions efficiency in yje field of information technologies.

Key words: Information Technologies service, strategic decisions, Balanced Scorecard, COBIT standard, perspectives, corporate contribution, stakeholders, operational excellence, future orientation, cause-and-effect relationships, decision support systems.



ANALYTICAL MODEL FOR EVALUATING PERFORMANCE MULTIPROCESSING FOR A SET OF PARALLEL ALGORITHMIC STRUCTURE

K. Gertsenberger, E. Chepin

Annotation

The effectiveness of parallel processing depends on the type of algorithm parallelization and hardware architecture. It is important to estimate the projected effectiveness of parallelization of the problem for a particular hardware platform, took place at the earliest stage of development. For this purpose, are encouraged to develop an analytical model predictions of the future performance of hardware-software system. For each parallel algorithmic structures (parallel algorithm) of the submission of the main set is the conclusion the most important performance characteristics: acceleration factor, efficiency and scalability. The adequacy of the developed analytical model for predicting the efficacy of the treatment is confirmed in practice.

Key words: analytical model, the prediction of performance, distributed computing, parallel hardware architecture, parallel algorithmic structure, parallel software.



ALGORITHMS 2DPCA FOR FACE RECOGNITION

N. Shchegoleva, G. Kukharev

Annotation

In article presents algorithms for two-dimensional principal component analysis (Two-dimensional Principal Component Analysis - 2D PCA)-oriented processing of digital images of large sizes in a small sample. Algorithms based on direct calculation of two covariance matrices for all source images without converting them into vectors. Evaluated characteristics of the presented algorithms. We discuss possibilities presented by the use of algorithms in other areas.

Key words: Two-dimensional principal component analysis, a two-dimensional Karhunen-Loeve transformation, the face recognition, the use in the business applications.



RANDOM INCREMENTAL FOREST AND TABLE STRUCTURE RECOGNITION

P. Kudinov, V. Polezhaev

Annotation

The paper proposes a new incremental algorithm for correct classification, based on the construction of a random incremental forest (RIF). The problem of recognizing the structure of statistical tables is considered and classification problems for its resolution are stated. A comparison of the new algorithm with a known algorithm ITI is shown. RIF shows better results for most problems.

Key words: Incremental learning; correct algorithms for classification; text mining; information retrieval; classification; decision trees; pattern recognition.



USING SEDUMI LIBRARY FOR ROBUST PORTFOLIO SELECTION

A. Isavnin, D. Galiev

Annotation

The article presents the principles of robust portfolio optimization. The SEDUMI library was used for problems solution. Modernized versions of the some portfolio selection models are described. The Markowitz, Telser and Black-Litterman models are overviewed. We compare the results of the efficiency analyses before and after robust optimization. Experiments were conducted at side and growing market trends with highly liquid stocks traded at MICEX. The weak and strong features of the different approaches are reviewed.

Key words: Investment portfolio, Robust optimization, Markowitz model, Telser model, Black-Litterman model, SEDUMI.



OPERATIONAL RISKS CLASSIFICATION WITHIN SERVICE-ORIENTED ARCHITECTURE OF INFORMATION SYSTEMS

I. Pyrlina

Annotation

The study identifies operational risks within service-oriented architecture (SOA) of information systems. As a part of operational risks a new error classification scheme is proposed for SOA applications. It is based on errors of the information systems which are service providers for application with service-oriented architecture. The proposed classification approach was used to classify system errors from two different enterprises (oil and gas industry, metal and mining industry). Besides we conducted a research to identify possible losses from operational risks and estimated losses for each error group per day.

Key words: operational risks, losses, error classification, service-oriented architecture, enterprise architecture, oil&gas and metal&mining industries.



THE TASK OF THE CHOICE OF NUMBER AND LOCATION OF THE CENTERS OF DATA STORAGE AND INFORMATION PROCESSING ON THE COMPUTER NETWORK

O. Esikov, D. Izotov

Annotation

In this article the task of a choice of number and location of centers of data storage and information processing on the computer network by criterion of a maximum of intensity of arrival of requests about information service is formulated. The detailed analysis of singularity of information processes has been carried out on the distributed computer network

Key words: centers of data storage and information processing, computer network, modeling of information processes, center of a graph.

**ЖУРНАЛ «БИЗНЕС-ИНФОРМАТИКА»
ОСУЩЕСТВЛЯЕТ РАЗМЕЩЕНИЕ РЕКЛАМНЫХ
И РЕКЛАМНО-ИНФОРМАЦИОННЫХ МАТЕРИАЛОВ**

Расценки:

Обложка: 2, 3, 4 страница обложки, полноцветная печать, полоса 210×290 мм (A4) – 40 тыс. руб.

Текстовый блок, чёрно-белая печать:

- ◆ полоса – 20 тыс. руб.;
- ◆ 1/2 полосы – 15 тыс. руб.;
- ◆ 1/4 полосы – 10 тыс. руб.;
- ◆ меньший объём – 7 тыс. руб.

Вставка (4 полосы, полноцветная печать – 60 тыс. руб.).

Рекламно-информационный блок (8 полос, полноцветная печать) – 80 тыс. руб.

Рекламно-информационный блок (16 полос, полноцветная печать) – 90 тыс. руб.

Корпоративный специальный выпуск – по договоренности.

Материалы принимаются с учётом следующих параметров:

- ◆ дообрезной формат – 215×300 мм;
- ◆ обрезной формат – 210×290 мм;
- ◆ поле набора полосной рекламы – 190×270 мм – с отступом от границ обрезного формата по 10 мм с каждой стороны;
- ◆ файл TIF, EPS, PDF – разрешение не менее 300 dpi.

№	Рубрика
1	Математические модели социальных и экономических систем
2	Программная инженерия
3	Анализ данных и интеллектуальные системы
4	Математические методы и алгоритмы решения задач бизнес-информатики
5	Моделирование и анализ бизнес-процессов
6	Информационные системы и технологии в бизнесе
7	Электронный бизнес
8	Интернет-технологии
Дополнительные рубрики вне номенклатуры	
9	Тематические обзоры
10	Правовые вопросы бизнес-информатики
11	Стандартизация, сертификация, качество, инновации
12	Дискуссионный клуб / Опыт бизнеса

К СВЕДЕНИЮ АВТОРОВ

Редакция просит авторов при оформлении статей и тематических обзоров придерживаться следующих правил и рекомендаций:

1. Предоставляемый авторами материал должен соответствовать рекомендуемой структуре статей журнала.

2. Статья направляется в редакцию в электронном виде (в формате MS WORD версия 2003) и в виде бумажной копии, распечатанной на одной стороне листов А4. Первая страница оригинала подписывается всеми авторами статьи.

3. Ориентировочный объем статьи, предлагаемой к публикации, – 20–25 тыс. знаков (с пробелами) или 30–35 тыс. знаков – для обзорных статей по направлениям.

4. Кегль набора – 12 пунктов с полуторным интервалом. Нумерация страниц – сверху по центру. Поля: левое – 2,5 см, верхнее, нижнее и правое – по 1,5 см.

5. При наборе выключных и строчных формул должен быть использован редактор формул MS Equation. В формульных и символических записях греческие (русские) символы, а также математические функции записываются прямыми шрифтами, переменные аргументы функций в виде английских (латинских) букв записываются наклонным начертанием (курсивом), например, «cos *a*», «sin *b*», «min», «max».

6. Формулы, таблицы и сноски (не концевые) оформляются стандартными средствами редактора MS WORD. Нумерация формул, рисунков и таблиц – сквозная, по желанию авторов допускается двойная нумерация формул с указанием структурного номера раздела статьи и – через точку – номера формулы в разделе.

7. Рисунки (графики, диаграммы и т.п.) оформляются средствами Word, Excel, Illustrator. Ссылки на рисунки в тексте обязательны и должны предшествовать позиции размещения рисунка. Допускается использование графического векторного файла в формате wmf/emf или cdr v.10. Фотографические материалы предоставляются в формате TIF или JPEG с разрешением не менее 300 dpi.

8. Библиографический список составляется в соответствии с требованиями ГОСТ. Нумерация библиографических источников – в порядке цитирования. Ссылки на иностранную литературу – на языке оригинала без сокращений.

Структура статей строится по правилам, рекомендованным журналом «Бизнес-информатика».

**Плата с аспирантов
за публикацию рукописей не взимается.**

Журнал публикует исследовательские научные статьи, размещаемые в рубриках журнала, тематические обзоры, отражающие современное состояние проблем в области бизнес-информатики и сообщения, размещаемые в рубриках «Дискуссионный клуб» и «Опыт бизнеса».

Титульный лист рукописи начинается с указания Ф.И.О. авторов публикации с обязательным указанием учёной степени, учёного звания, должности, основного места работы и e-mail. Титульный лист должен быть подписан всеми авторами статьи.

I. Исследовательские научные статьи (для размещения в тематических рубриках)

Редколлегия рекомендует авторам после названия статьи приводить **аннотацию**, в которой излагается краткое содержание статьи, её основные результаты и область применения. Авторам рекомендуется структурировать статью, выделяя **введение**, содержащее описание проблемы или задачи, обзор существующих подходов или методов решения, их недостатки, и основную цель статьи; **постановку задачи**, включающую допущения и ограничения; **содержательную часть** статьи, в которой предлагаемые решения должны быть аргументированы и сравниваться с существующими подходами или решениями; **заключение**, содержащее краткое изложение новых результатов, полученных в статье и область их применения; **библиографический список**, оформленный в соответствии с ГОСТ. Текст статьи должен содержать нумерованные ссылки на все указанные библиографические источники. Структурирование статьи и нумерация её разделов проводится по усмотрению авторов.

Возможный вариант структуры статьи:

- ◆ Ф.И.О;
- ◆ учёная степень, учёное звание, должность, основное место работы, e-mail;
- ◆ название статьи.
- ◆ аннотация;
- ◆ 1. Введение.
- ◆ 2. Постановка задачи.
- ◆ 3. Основная содержательная часть статьи.
- ◆ 4. Экспериментальные результаты (опционально).
- ◆ 5. Заключение.
- ◆ 6. Библиографический список.

II. Тематические обзоры по направлениям

Редколлегия рекомендует авторам структурировать обзор, выделяя аннотацию, содержащую тематику, краткое содержание обзора и область применения; **введение**, в котором даётся краткий исторический обзор тематики; **содержательную часть** обзора с критическим анализом существующих направлений; **заключение**, в котором отражаются перспективы развития в рамках обзора тематики и наиболее интересные направления с точки зрения научных и практических разработок и методов; **библиографический список**, оформленный в соответствии с ГОСТ.

Текст обзора должен содержать нумерованные ссылки на все указанные библиографические источники. Структурирование обзора и нумерация его разделов проводится по усмотрению авторов.

Возможный вариант структуры обзора:

- ◆ Ф.И.О;
- ◆ учёная степень, учёное звание, должность, основное место работы, e-mail;
- ◆ название обзора;
- ◆ аннотация;
- ◆ 1. Введение.
- ◆ 2. Основная содержательная часть обзора.
- ◆ 3. Заключение.
- ◆ 4. Библиографический список.

Редколлегия журнала проводит обязательное рецензирование рукописей. Статья принимается к публикации только после получения положительного заключения рецензента и одобрения на заседании редакционной коллегии журнала.