

ISSN 1998-0663

№4(06)—2008

www.bi-network.ru/jbi/

БИЗНЕС- ИНФОРМАТИКА

МЕЖДИСЦИПЛИНАРНЫЙ НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ

BUSINESS INFORMATICS

Учредитель:

Государственный университет—
Высшая школа экономики

Главный редактор

Никитин В.В.

Заместители главного редактора

Горбунов А.Р.

Ульянов М.В.

Редакционная коллегия:

Абдульраб А. (Франция)

Авдошин С.М.

Алескеров Ф.Т.

Белов В.В.

Вирин Ф.Ю.

Грибов А.Ю.

Громов А.И.

Понтер Х. (Германия)

Каменнова М.С.

Калягин В.А.

Козырев О.Р.

Кузнецов С.О.

Мальцева С.В.

Миркин Б.Г. (Великобритания)

Моттль В.В.

Пальчунов Д.Е.

Силантьев А.Ю.

Таратухин В.В.

В ЭТОМ НОМЕРЕ:

Информационные технологии
в бизнесе и экономике

Математические модели
социальных
и экономических систем

Анализ данных
и интеллектуальные системы

Математические методы
и алгоритмы решения задач
бизнес-информатики

Анализ данных
и интеллектуальные системы

Программная инженерия

Опыт бизнеса



БИЗНЕС- ИНФОРМАТИКА

<http://www.bi-network.ru/jbi/>

№4(06)-2008

СОДЕРЖАНИЕ

<i>Информационные технологии в бизнесе и экономике</i> <i>А.С. Лихварев, Н.В. Асеева</i> Оценка инвестиционной привлекательности IT-проектов с учетом особенностей современных подходов к архитектуре программных комплексов.....3
<i>Математические модели социальных и экономических систем</i> <i>А.Р. Горбунов</i> Фактор ценового класса выпуска: новые пути исследования и измерения технологического прогресса13
<i>Анализ данных и интеллектуальные системы</i> <i>Д.И. Игнатов, С.О. Кузнецов, В.Б. Лопатникова, И.А. Селицкий</i> Разработка и апробация системы поиска дубликатов в текстах проектной документации21
<i>А.С. Рудаков</i> Подходы к решению задачи прогнозирования временных рядов с помощью нейронных сетей29
<i>Информационный материал о международной конференции ICCS'09</i>35
<i>Математические методы и алгоритмы решения задач бизнес-информатики</i> <i>В.В. Белов, В.И. Чистякова</i> Моделирование и прогнозирование бизнес- процессов с помощью алгоритмов самоорганизации формальных описаний37
<i>Л.А. Демидова</i> Прогнозирование тенденций временных рядов на основе однофакторной нечеткой модели с использованием дискретных нечетких множеств второго типа и генетического алгоритма46
<i>О.Р. Козырев, К.В. Логвинова</i> О фундаментальных свойствах дробных операторов в классе задач нелинейной динамики54
<i>Программная инженерия</i> <i>О.Р. Набиуллин, Э.А. Бабкин</i> Моделирование и автоматическая проверка по AsmL.....56
<i>Опыт бизнеса</i> <i>А. Борщев</i> Применение имитационного моделирования в России – состояние на 2007г.64
<i>Microsoft открыла Центр инноваций</i>69
<i>Реклама</i>71



БИЗНЕС-ИНФОРМАТИКА
№4(06)–2008

Междисциплинарный
научно-практический журнал

Подписной индекс издания
в каталоге агентства
«Роспечать» – 72315

Главный редактор
Никитин В.В.

Учредитель:
Государственный университет-
Высшая школа экономики

Выходит 4 раза в год.

Редакция:

Зам. главного редактора
Горбунов А.Р.
Зам. главного редактора
Ульянов М.В.

Научный редактор
Лычкина Н.Н.

Технический редактор
Осипов В.И.

Дизайн обложки
Борисова С.Н.

Компьютерная вёрстка
Волков А.А.
Бортникене Е.И.

Администратор веб-сайта
Проценко Д.С.

<http://www.bi-network.ru/jbi/>

Адрес редакции:
105679, г. Москва,
ул. Кирпичная, д. 33.
Тел. +7 (095) 771-32-38,
e-mail: bjjournal@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 200 экз.

© Государственный университет-
Высшая школа экономики

ОЦЕНКА ИНВЕСТИЦИОННОЙ ПРИВЛЕКАТЕЛЬНОСТИ ИТ-ПРОЕКТОВ С УЧЕТОМ ОСОБЕННОСТЕЙ СОВРЕМЕННЫХ ПОДХОДОВ К АРХИТЕКТУРЕ ПРОГРАММНЫХ КОМПЛЕКСОВ

А.С. Лихварев,

*аспирант Государственного университета – Высшей школы экономики,
специалист ООО «Теком»*

likhvarev@tecomgroup.ru

Н.В. Асеева,

к.ф.-м.н., старший преподаватель, НФ Государственного университета – Высшей школы экономики,

n_aseeva@mail15.com

Обзор рассматривает существующее отношение бизнеса к оценке инвестиционной привлекательности ИТ-проектов. Приведены основные положения комплексного метода для такой оценки. Показано, что в данной методике не учитываются архитектурные особенности информационных систем. Отмечено, что в данный момент большое распространение получила архитектура на основе сервисов, и учет ее в существующем методе оценки экономической эффективности является актуальной задачей.

Введение

В современном деловом мире ИТ занимает все больше места. Редкая компания обходится без подразделения информационных технологий или, по крайней мере, без программных продуктов, направленных на удовлетворение нужд бизнеса. Даже такая форма предпринимательства как ПБОЮЛ зачастую использует облегченные версии бухгалтерского программного обеспечения.

Тем не менее, внедрение ИТ-технологий в деловом мире редко происходит на основе взвешенных и хорошо просчитанных решений. Зачастую ИТ-проект внедряется по зову интуиции или по принципу копирования действий конкурентов. Все это делается, несмотря на то, что общая методика расчета инвестиционной привлекательности того или иного проекта хорошо развита. Причиной такого положения вещей является отсутствие распространения именно единого прикладного метода оценки инвестиционной привлекательности именно ИТ-проектов.

Отсутствие распространения не говорит о том, что данный прикладной метод не был описан в литературе. Недостаток описания в том, что в нем учитывались особенности архитектуры программного обеспечения, характерные на момент описания метода. Дело в том, что ИТ-отрасль в общем и технологии программного обеспечения в частности динамично развиваются, и новые тенденции в принципах разработки и архитектурных подходах появляются с завидным постоянством.

Так, например, в настоящее время большое распространение получил подход проектирования архитектуры ИС на основе сервисов. Существующий прикладной метод оценки инвестиционной привлекательности ИТ-проекта должен учитывать эти особенности. Так в данном случае, очевидно, их влияние как минимум на затратную часть проекта. Изучить полное влияние архитектурных особенностей систем на основе сервисов и дополнить существующий

прикладной метод является непростой и в то же время актуальной задачей.

В данном реферате формулируется данная задача, рассматривается текущее состояние описанной проблемы, а также делаются предположения о возможных путях решения, то есть о том, как можно усовершенствовать методику оценки инвестиционной привлекательности ИТ-проектов в соответствии с новыми, современными подходами к проектированию программных комплексов. В первом разделе работы рассматриваются существующие проблемы оценки экономической эффективности ИТ-проектов.

Проблемы оценки эффективности ИС

Исторически сложилось, что при внедрении ИТ-проектов редко используется оценка инвестиционной привлекательности таких проектов. Можно выделить две основные группы причин, связанные с отказом от оценки экономической эффективности ИС: связанные с равнодушным отношением к применению оценки инвестиционной привлекательности таких проектов и группа причин, показывающих негативное и даже враждебное отношение к такой оценке. [1]

Типичные аргументы «против» комплексной оценки ИТ-проектов

Рассмотрим первую группу причин, показывающих безразличное отношение к процессу оценки.

1. Практика показывает, что ИТ-проекты считаются достаточно рискованными и нередко не приносят прибыли.

2. Интуитивный подход в принятии решения о внедрении ИТ-проекта. Менеджерам, принимающим решения о необходимости той или иной компьютерной системы в компании, иногда кажется, что это просто «хорошая идея».

3. Решение о внедрении ИТ-проекта кажется иногда очевидным, так как компьютерная система является необходимым условием для определенных видов бизнеса.

4. Издержки самого процесса оценки инвестиционной привлекательности проекта могут значительно превышать предполагаемые издержки самого ИТ-проекта.

5. Используемые в компании компьютерные системы предполагают применение определенного программного обеспечения, поэтому риск внедрения ИТ-проекта, развертывающего это программное обеспечение, невелик, а вероятность успешности — велика.

6. Известно, что оценка инвестиционной привлекательности ИТ-проектов сложнее, чем оценка проектов в других отраслях.

Также существует несколько аргументов, показывающих враждебное отношение к оценке инвестиционной привлекательности ИТ-проектов.

1. Зачем измерять «неизмеримые» составляющие ИТ-проектов (имеются в виду нематериальные выгоды)?

2. Существующие методики ISS (Information System Strategy — Стратегии Информационных Систем) сконцентрированы только на информационном компоненте исследования, при этом не забываются о финансовой стороне.

3. Используемые методы оценки ИТ-проекта учитывают затраты, они достаточно легко отражаются в денежной форме в документации о проекте. Однако, выгоды, которые сулит ИТ-проект, лишь частично являются материальными и оценить их в денежной форме непросто.

4. Особенности существующего бухгалтерского учета на предприятии являются препятствием для применения методики оценки инвестиционной привлекательности ИТ-проекта.

5. Существующие методики оценки проекта «анализ затраты-выгода» применяются каждая по отдельности и не могут быть интегрированы.

6. Выгоды, которые приносит ИТ-проект, не отслеживаются, не производится их мониторинг, учет и сравнение с запланированными значениями. Это приводит к отсутствию подтверждения успешности внедренного проекта, что является барьером к использованию оценки инвестиционной привлекательности будущих ИТ-проектов.

Возможность решения проблемы оценки ИТ-проектов при помощи комплексного метода

Все многообразие аргументов против применения оценки экономической эффективности ИС говорит о существующих проблемах в этой области. Ведь принятие решений о внедрении того или иного ИТ-проекта вслепую, без тщательного расчета по отработанному методу с большой вероятностью может обернуться для компании убытками, и даже ухудшением позиции на рынке. Любой проект, в который предполагает инвестиции, должен быть подвержен комплексной оценке, ИТ-область — не исключение. В целом можно выделить основные причины существующих проблем [1]:

- ✧ инерционное нежелание менеджмента;
- ✧ некомпетентность существующих методик, отсутствие комплексного метода;
- ✧ сложность расчета различных видов рисков и нематериальных выгод, составляющих порой большую часть общего результата проекта;
- ✧ высокая стоимость самого процесса оценки,

связанная с его сложностью, не оправдывающая себя.

С первой и последней причиной достаточно сложно бороться, но две оставшиеся причины существующей проблемы могут быть устранены, если применять комплексный структурированный метод оценки инвестиционной привлекательности IT-проекта, который содержал бы в себе набор взаимосвязанных и уместно примененных методик, учитывающий все компоненты расчета. Это разом устранило бы отсутствие комплексного подхода и снизило сложность благодаря уже выстроенной структуре расчета, включающих методики расчета всех кажущихся невидимыми и не вычисляемыми факторов.

Современные тенденции, располагающие к оценке IT-проектов

Современные тенденции в мире бизнеса и IT лишь усиливают актуальность такого метода. Среди них можно выделить следующие закономерности.

1. Информационные системы становятся сложнее, а их компоненты позволяют быть использованными повторно все с большей гибкостью (развитие технологий БД, ООП, СПД), вследствие чего получили развитие интегрированные системы.

2. Все большие суммы денег привлекаются в разработку IT-проектов.

3. Инвестиции в IT сейчас рассматриваются как часть стратегического инвестиционного портфеля, и, таким образом, должны конкурировать с другими возможностями вложения денег.

Эти современные тенденции все больше меняют изначально негативное отношение к комплексному методу инвестиционной оценки IT-проектов в положительном направлении среди менеджмента, и лишь подчеркивают актуальность применения комплексного структурированного метода оценки инвестиционной привлекательности IT-проекта.

Один из вариантов такого метода был предложен Джоном Харесом (John Hares), профессионалом IT отрасли, и Дунканом Ройл (Duncan Royle). Подробно состояние дел в сфере разработки таких методов и этот метод в частности рассмотрен в следующей главе данной работы.

Существующее состояние методов оценки экономической эффективности IT-проекта

Существующее состояние методов оценки экономической эффективности IT-проекта выглядит не очень радужным по нескольким причинам [1].

1. Существующие структурированные методы для дизайна ИС рекомендуют использовать СВА

для оценки экономической эффективности проекта, прежде чем принять решение о дизайне и разработке ИС, но, к сожалению, СВА обладает отрицательными чертами, не позволяющими с помощью ее получить правдоподобные и точные результаты оценки проекта.

2. Методики, существующие на сегодняшний день, применяются изолированно, таким образом, нет выгод от совместного их применения, и не все аспекты оценки IT-проектов покрываются.

3. Некоторые методики применяются неуместно. Так, например, недисконтированные методики, которые не берут в расчет инфляцию и рыночный риск. Такие методики могут дать неправильные результаты и поспособствовать принятию неверного решения.

4. Некоторые методики могут дать конфликтующие результаты.

5. В методиках, применяемых на практике, не учитываются важные компоненты:

- ✧ издержки – риск проекта и непредвиденных событий. Риски неадекватно покрываются существующими методиками;
- ✧ доходы – гибкость и нематериальные выгоды. Гибкость (если ее себестоимость ниже риска проекта) повышает ценность IT-проекта. Существуют проверенные методики для расчета гибкости проекта.

Таким образом, немало проблем встает перед тем, кто собрался произвести комплексную оценку инвестиционной привлекательности IT-проекта.

Структура метода оценки IT-проекта

Рассмотрим структуру такого метода. Метод оценки инвестиционной привлекательности внедрения IT-проекта состоит из следующих основных частей [1]:

1. Набор интегрированных приемов для определения возможностей для инвестиций, которые могут быть осуществлены в компании в отношении IT.

2. Набор интегрированных методик для стоимостной оценки IT-проектов, входящих в стратегический план. Оценка подвергается следующие составляющие проекта:

- ✧ видимые и скрытые издержки;
- ✧ доходы в виде материальных и нематериальных ценностей;
- ✧ различные виды риска – рыночный, проекта, связанный с непредвиденными событиями;
- ✧ возможность уменьшения риска за счет гибкости проекта;
- ✧ доход, который может принести проект в результате синергии.

3. Механизм для измерения реальной отдачи от внедрения IT-проекта и сравнения этих величин с теми, что были заявлены перед внедрением проекта.

Метод состоит из 4 модулей, пронумерованных от 0 до 3 (рис. 1). «0» означает то, что никакие методы здесь не применяются. Каждый модуль служит для достижения определенной цели метода. Заголовок модуля описывает соответствующую цель.

Каждый модуль состоит из овальных элементов (в дальнейшем - элементы), каждый из элементов представляет собой определенный аспект цели модуля, в котором данный элемент расположен. Внутри каждого элемента может применяться несколько методик для получения результатов, которые в совокупности формируют аспект цели данного модуля. Группировка методик в элементах, касающихся оценки на уровне проекта, показана на рис. 2. На ней намеренно показано, что там, где несколько методик и полученных в результате их применения результатов, покрывают один аспект цели модуля, тогда данные результаты должны быть агрегированы, объединены в один результат.

Модуль стратегии бизнеса

Это начальный модуль в методике. Его цель – определить стратегию развития бизнеса в целом, стратегию вывода определенных товаров и услуг на рынок. Методики для разработки стратегии компании хорошо проработаны и не обсуждаются далее по причине того, что оценка инвестиционной привлекательности имеет дело не со стратегией бизнеса, а с результатами такой стратегии, так как именно результатами стратегии являются потенциальные проекты и возможности для инвестиций в них.

Основные результаты модуля стратегии бизнеса должны включать:

- ✦ миссию компании, включая цели компании в терминах производимых товаров и услугах;
- ✦ сектор рынка, который планирует занимать компания в ближнесрочной и долгосрочной перспективе;
- ✦ задачи, который должны привести к выполнению целей;
- ✦ портфель возможностей для инвестиций, представленный в общих чертах.

Модуль исследования возможностей для инвестиций

Данный модуль связан с определением и планированием возможностей для инвестиций. Инвестиционные проекты должны быть выгодны с финансовой точки зрения, но кроме этого они должны способствовать достижению целей компании

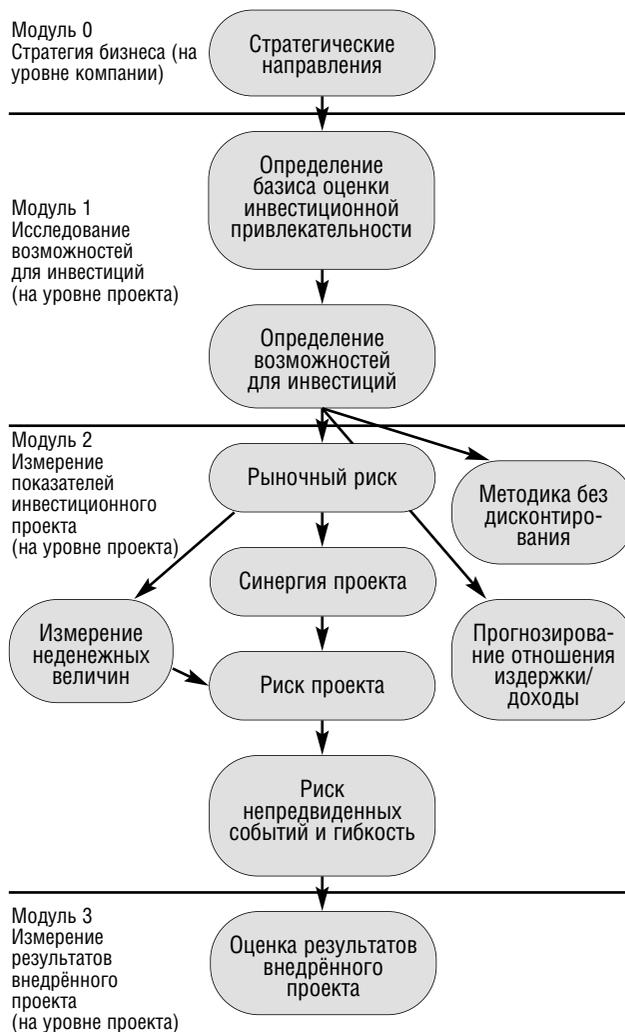


Рис. 1. Структура комплексного метода оценки инвестиционной привлекательности IT-проекта

в рамках ее стратегии. Чем лучше возможность для инвестиции, тем выше приоритет проекта.

Модуль состоит из четырех шагов, которые должны быть предприняты в определенной последовательности, прежде чем производить оценку инвестиционной привлекательности какого-либо из возможных проектов.

1. Основная линейка продуктов или сфера деятельности, с которой связана возможность инвестиций, должна быть согласована с топ-менеджерами компании и руководителями финансового отдела, иначе предложенный компьютерный проект не будет принят представителями менеджмента.

2. Определение возможностей для инвестиций.

3. Определение относительных преимуществ выявленных возможностей для инвестиций.

4. Составление портфеля возможностей для инвестиций.

Совместное использование информационного и экономического компонентов позволяет данному



Рис. 2. Агрегация результатов различных методик

модулю исследованию возможностей для инвестиций быть полным методом ISS (Information System Strategy – Стратегия Информационных Систем). Данный вариант ISS метода, использующего информационный и экономический компоненты, является отличным источником для приоритезации и планирования, необходимых для компании компьютерных систем.

Модуль измерения показателей инвестиционного проекта

Данный модуль является основной частью комплексного модуля оценки инвестиционной привлекательности IT-проекта. Он нацелен на измерение финансовой стороны компьютерных проектов, предложенных в результаты работы предыдущего модуля.

Модуль состоит из методик, традиционно ассоциирующихся с СВА, определения рыночной ставки дисконта, для того чтобы привести денежные потоки в будущем к сегодняшнему моменту. Рыночная ставка дисконта основана на риске, который компания планирует исходя из рынка, на котором она ведет бизнес в соответствии со своей стратегией. Рыночный риск включает эффект инфляции. Методики, измеряющие рыночный риск находятся внутри элемента «рыночный риск». Также туда включены не дисконтируемые методики: «срок окупаемости», «ROI доход с инвестиции».

Не дисконтируемые методики не учитывают негативные эффекты инфляции и рыночного риска, оказываемые на денежные потоки проекта с течением времени. Эти методики были включены в структуру метода не потому, что они приносят пользу (скорее нет), а потому, что они очень широко распространены. Они описаны, их негативные стороны определены и учтены. Их избыточность показана на рис. 3. Также излишними являются балльные методики, используемые там, где издержки и доходы трудно измерить. Баллы невозможно сравнивать с денежными показателями, что не позволяет осуществлять сравнение альтернативных проектов при помощи таких балльных методик.

Элемент «прогнозирование соотношения издержки/доходы» содержит разумные подходы для бухгалтерского дела. Он не дает результатов, которые могли бы использоваться далее в структуре комплексного метода, однако, в целом он полезен.



Рис. 3. Структура комплексного метода оценки инвестиционной привлекательности IT-проекта. Невключенные методики

Рис. 4 показывает методики, которые не являются частью СВА. Именно эти методики позволяют превратить СВА методику в полный комплексный метод оценки инвестиционной привлекательности IT-проектов.

Отдельная группа методик используется для определения и измерения нематериальных выгод, которые компьютерная система может дать компании. Эти методики применяются на ранней стадии модуля, чтобы позволить включить результаты этих методик в дисконтируемые денежные потоки.

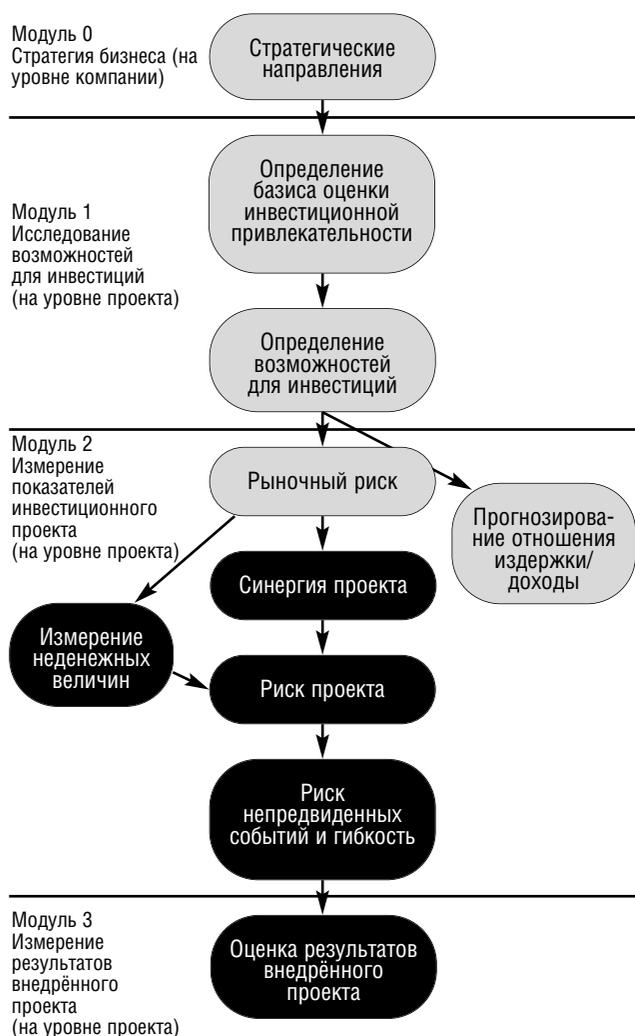


Рис. 4. Структура комплексного метода оценки инвестиционной привлекательности IT-проекта. Новые для СВА методики

Другой рано выполняемый элемент — «синергия проекта», который свидетельствует о том, что компьютерные приложения функционируют не изолированно. Появление таких технологий, как БД, ООП, позволило сделать логику и данные корпоративными ресурсами. Теперь различные приложения могут работать более эффективно, используя в своей работе результаты работы других приложений, используя

совместно данные из БД при помощи единообразной логики разработанной, используя ООП.

Данные положительные тенденции в IT-сфере позволяют все больше выгоды извлекать из совместных друг с другом приложений. Эти тренды используются внутри элемента «синергия проекта». Положительное значение синергии должно быть измерено и включено в дисконтируемые денежные потоки.

Как только рыночная ставка дисконта рассчитана и добавлена к ставке инфляции, риск проекта должен быть рассчитан. Данный риск конвертируется в ставку дисконта и добавляется к рыночной ставке дисконта. Получившаяся ставка дисконта применяется к денежным потокам проекта.

Непредвиденные события могут произойти в период выполнения проекта и повлиять на его успех. Такие события также являются риском, которые необходимо учесть. Вероятности таких событий и их возможное влияние на издержки/доходы должны быть измерены. Так как риски непредвиденных событий снижают ценность проекта, план проекта должен быть скорректирован в направлении поиска возможностей снизить влияние таких событий, используя гибкость проекта, то есть необходимо выявить возможные альтернативные сценарии, которые можно применить в момент возникновения непредвиденных событий, а также оценить издержки/доходы этих сценариев.

Модуль оценки результатов внедренного проекта

Последний модуль метода - «оценка результатов внедренного проекта». Он направлен на сопоставление результатов, которые получены после внедрения компьютерной системы с планируемыми результатами. Реальные финансовые показатели функционирования внедренной компьютерной системы могут быть получены из отчетов менеджмента на местах, где функционирует данная система. На основе данных отчетов можно составить картину, показывающую реальную ценность проекта. Менеджеры могут предпринять оперативные действия в том случае, если обнаружится, что реализуемый проект не приносит тех выгод, что были запланированы.

Недостаточный учет архитектурных особенностей внедряемого IT-проекта в комплексном методе оценки

Комплексный метод оценки инвестиционной привлекательности IT проектов мало внимания уделяет влиянию архитектуре ИС. Основополагающим принципом данный метод считает, что ИС должна помочь компании достигнуть целей, поставленных в рамках стратегии компании. При этом набор

приложений для ИС выбирается на этапе изучения возможностей для инвестиций, на этом же этапе используются методы ISS, то есть определяются общие границы применения и архитектурные особенности приложений, входящих в состав портфеля возможных инвестиций, сами же эти методы выходят из рамок комплексного метода оценки эффективности ИС, так как считаются хорошо проработанными и дающими априори оптимальный результат проектирования. В дальнейшем в структуре метода архитектурные особенности ИС учитываются при измерении показателей экономической эффективности инвестиционного проекта.

Учет архитектуры ИС при измерении «синергии проекта»

На следующем этапе архитектурные особенности ИС учитываются при измерении «синергии проекта», то есть взаимодействия различных приложений внутри компании, и изучения подхода «портфеля приложений», при котором оптимальным образом приложения дополняются друг друга и используют общие данные и бизнес логику.

Утверждается, что около 80% кода независимо от событий, и инкапсулировано внутри объектов благодаря ООП. Эти 80% логики спроектированы как внутренние свойства объектов практически также как данные. Эти 80% логики также как и корпоративные данные являются независимыми от приложений и, таким образом, позволяют получать максимальный эффект синергии при использовании портфеля приложений внутри компании. Только 20% логики является зависит от приложения. Преимущества использования единой БД теперь равностраются еще и на 80% логики. Единая информационная база данных и логики стала реальностью с пришествием технологий БД и ООП.

Главное преимущество, получаемое от синергии проекта, - это возможность ООД поодерживать повторное использование информации и даже частично повторное использование приложений и их модулей. Таким образом, основной подход к дизайну ИС – это использование текущей ИС максимальным образом, добавляя новые данные и функциональность только по мере необходимости. Системная разработка превращается в процесс добавления необходимых модулей в существующую ИС. Это и есть эффект максимальной синергии проекта. При этом перекрытие данных и функциональности компьютерных систем возрастает, так как различные приложения повторно используют данные и логику, используемые другими приложениями.

Синергия проекта и преимущества, которые она дает, вносят изменения в расчет соотношения издержки-доходы. Временные профили издержек изменяются под влиянием синергии. Это видно на рисунке 5. Первый профиль издержек типичен для ИС, архитектура которой основана на CASE/4GL. Издержки низки при использовании системы, но высоки при дизайне, разработке и поддержке. При использовании ООП издержки еще более велики при дизайне (особенно при проектировании первого приложения системы), но ниже при поддержке. Кроме этого максимальный эффект от использования ООП и ООД проявляется при проектировании второго и последующих приложений системы. При этом существующее ядро уже спроектировано и не требует дизайн только для дополнительных функциональных модулей конкретно этих приложений, при этом затраты на дизайн и разработку значительно падают. Все это также видно на рис. 5.

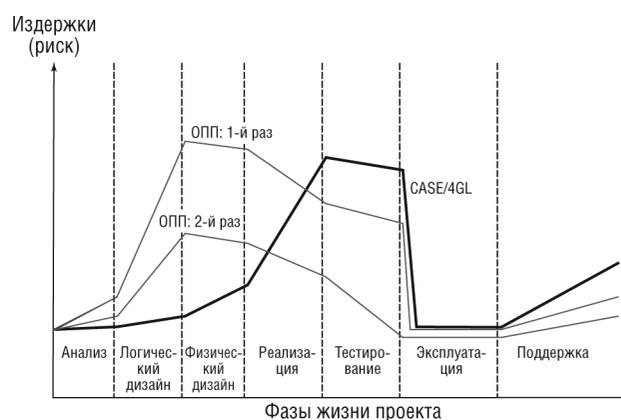


Рис. 5. Профиль издержек для ИС на основе CASE/4GL и ООП

Можно выделить различные степени взаимодействия приложений внутри ИС. Спектр уровней взаимодействия варьируется от проектов, являющимися комплиментарными, до проектов, являющимися взаимно исключающими (отрицательная синергия). Между этими крайними значениями существует много различных промежуточных уровней комплиментарности проектов. Все они показаны на рис. 6.

Инфраструктурные проекты не всегда приносят положительного финансового результата, но они позволяют стартовать другие проекты. Часто бывает так, что только после внедрения проектов, зависящих от этих инфраструктурных проектов, появляется прибыль. Поэтому методики дисконтирования денежных потоков не показывают реальную пользу инфраструктурных проектов (в которых при использовании БД и ООД закладывается 80% в последующем повторно используемых данных и логики). Как

же рассчитать реальную пользу инфраструктурных проектов? Нужно задать вопрос: «Что можно сделать теперь, когда инфраструктура создана, и что нельзя было сделать до этого?» То есть нужно выполнить последовательность: идентифицировать различия, измерить их и представить в денежных единицах для принятия решения.



Рис. 6. Спектр синергии различных проектов

Методики DCF (дисконтирования денежных потоков) могут применяться для определения синергии проектов, но с определенной оговоркой. Невозможно измерить синергию набора приложений, выбрав их из портфеля приложений и суммируя значения эффекта каждого из них. Необходимо измерять эффект синергии для каждого набора. Это делает подход портфеля приложений для измерения синергии слишком трудоемким, так как для определения оптимального набора приложений нужно перебирать все составы n из m приложений, где n – число приложений, которые необходимо выбрать, а m – общее число приложений в портфеле.

В целом архитектурные особенности, рассмотренные в методе, оказывают влияние на профиль издержек и на доходы, так как синергия проекта дает дополнительный доход. То есть архитектурные особенности в целом оказывают сильное влияние на успех проекта, затрагивая и издержки, и доходы, но в методе не учтены новые подходы и концепции в архитектуре современных ИС, такие, как архитектура на основе сервисов.

Учет архитектуры ИС при измерении рисков и гибкости проекта

Метод комплексной оценки инвестиционной привлекательности ИТ-проекта касается архитектурных особенностей ИС при расчете риска и гибкости проекта.

При оценке риска и гибкости при помощи анализа чувствительности ИС к внешним параметрам приводится пример, в котором сравниваются две ИС (рис. 7). Первая основана на мейнфрейме, другая на распределенной ИС-системе, развернутой на множестве персональных компьютеров. Производится анализ чувствительности данных систем на поток входящих заявок (транзакций) на обработку. Показывается, что распределенная система более гибкая и обладает меньшим риском, связанным с уменьшением входящего потока заявок, так как даже при таком потоке благодаря низким издержкам она способна приносить прибыль.

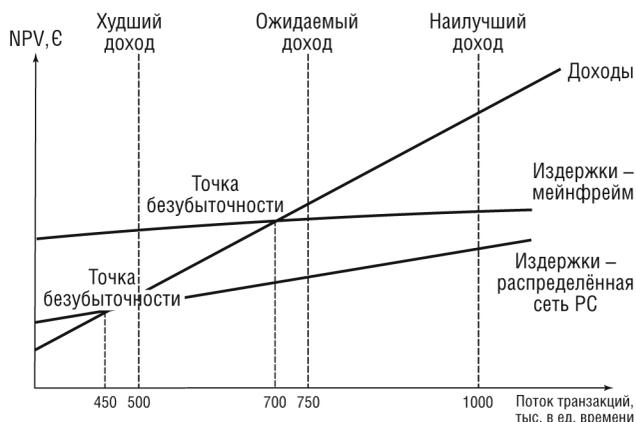


Рис. 7 Анализ чувствительности к изменению параметра количества транзакций в единицу времени

Тем не менее, это лишь общий пример, разница в архитектуре приводится лишь концептуально, без рассмотрения конкретных подходов. Кроме этого в комплексном методе оценки эффективности ИС методика анализа чувствительности приведена лишь как дополнительная, а упор при расчете риска непредвиденных событий и гибкости делается на методику дерева решений (decision tree). Более того, ни слова не сказано про современные подходы в архитектуре, такие как системы на основе сервисов.

Современные тенденции в архитектуре программных комплексов. Популярность систем на основе сервисов

В мире ИТ постоянно происходят изменения. Эволюция языков и технологий разработки, а также концепций разработки архитектуры ИС не останавливается. Такие важные вехи, как внедрение процедурных

языков, смена их ОО-языками, параллельное развитие БД, сетевых технологий и распределенных систем, не являются последними вехами в развитии ИС.

Аспектно-ориентированное программирование

В настоящее время разработка ИС ведется не только с использованием ООП, но и АОП (аспектно-ориентированный подход или программирование). Аспектно-ориентированное программирование (АОП) — парадигма программирования, основанная на идее разделения функциональности, особенно сквозной функциональности, для улучшения разбиения программы на модули [2].

Методология аспектно-ориентированного программирования была предложена группой инженеров исследовательского центра Xerox PARC под руководством Грегора Кикзалеса (Gregor Kiczales). Ими же был разработан первый, и наиболее успешный до сих пор, контекстно-ориентированный язык программирования AspectJ.

Существующие парадигмы программирования, такие как процедурное программирование и объектно-ориентированное программирование, оставляют некоторые способы для разделения и выделения функциональности, например, функции, объекты, классы, пакеты, но некоторую функциональность с помощью предложенных методов невозможно выделить в отдельные сущности. Такую функциональность называют сквозной, так как её реализация разбросана по различным модулям программы. Сквозная функциональность приводит к рассредоточенному и запутанному коду. Запутанным называется такой код, в котором одновременно реализована различная функциональность.

Но АОП подход ближе к дизайну отдельных модулей системы. Среди верхнеуровневых подходов в архитектуре ИС все большую популярность принимают системы на основе сервисов. Появился даже такой термин и аббревиатура SaaS — Software as a Service (программное обеспечение как сервис).

Software as a Service — программное обеспечение как сервис

Software as a service (SaaS) — программное обеспечение как услуга — это модель предложения программного обеспечения потребителю, при которой поставщик разрабатывает веб-приложение, размещает его и управляет им (самостоятельно либо через третьих лиц) с целью и возможностью использования заказчиками через Интернет. Заказчики платят не за владение программным обеспечением как таковым, а за его использование (через API, доступный через веб и часто использующий веб-служ-

бы). Ближким к термину SaaS является термин «приложение по запросу» (On-Demand) [3].

Аббревиатура SaaS существует примерно с 2001 года. На первых порах она не привлекала к себе особого внимания и получила широкое распространение только после 22 марта 2005 года, когда состоялось первое собрание группы единомышленников, назвавшей себя SaaS Special Interest Group. Группа объединяет тех, кого интересует влияние на развитие ИТ-отрасли программ, поставляемых по запросу. Сегодня SaaS SIG наряду со многими другими группами входит в состав общественной организации Software Development Forum — одной из наиболее влиятельных ассоциаций разработчиков, объединяющей более 12 тыс. профессионалов-программистов.

Принципиальным отличием модели SaaS от более ранних (Hosted Applications и Application Service Provider (ASP)) является то, что приобретается именно услуга и интерфейс (пользовательский или программный), то есть некоторая функциональность без жесткой привязки к способу ее реализации.

В модели SaaS:

- ✧ приложение приспособлено для удаленного использования;
- ✧ одним приложением пользуется несколько клиентов (приложение коммунально);
- ✧ оплата взимается как ежемесячная абонентская плата или на основе объема транзакций;
- ✧ поддержка приложения входит в состав оплаты;
- ✧ модернизация приложения происходит плавно и прозрачно для клиентов.

Аналитики о SaaS

Оптимистичней других предсказания Gartner, хотя разные сотрудники этой аналитической компании и дают различающиеся между собой цифры. По мнению одних, рынок SaaS уже сегодня составляет 5 млрд. долл., другие считают, что эта цифра больше, и к 2011 году она может увеличиться до 16–20 млрд. долл., перетянув на себя до 20% от всего рынка программного обеспечения. В IDC тоже неоднозначны в оценках: одни оценивают современное состояние рынка SaaS суммой 2 млрд. долл., другие, напротив, поднимают планку до 8 млрд. долл., в пятилетней перспективе отводя на долю SaaS от 5 до 25% программного рынка. Но в любом случае цифры значительны и серьезно угрожают былой стабильности, присущей рынку программного обеспечения [4].

Аналитики обнаружили, что перспективность SaaS зависит от сферы применения программных систем. Уже сейчас эта схема охватила до 60% рынка продуктов для электронного обучения и до 70%

технологий для Web-конференций. На другом полюсе — средства для поиска и управления корпоративным контентом, где распространенность SaaS не достигает и 2%. Самый яркий пример успешного применения SaaS продемонстрировала компания Salesforce.com, построившая всю свою деятельность на работающих по запросу системах управления отношениями с клиентами [4].

Аналитики сходятся в том, что ключевым для SaaS окажется применение в сфере малого и среднего бизнеса. Помимо очевидных аргументов, связанных с эксплуатационными преимуществами SaaS, есть аргументы, связанные с возможностями более полного удовлетворения потребностей небольших предприятий. Для иллюстрации этого утверждения нередко используется метафора, получившая название «длинный хвост» [4]. Ее в последнее время часто используют для описания того, как распределяется потребительский спрос. В левой части графика находится меньшая часть потребителей, с родственными предпочтениями, по совокупности они составляют примерно половину потенциального потребительского спроса. Производители в первую очередь ориентируются на эту часть потенциальных клиентов. Но есть большое число потребителей с диверсифицированными потребностями, их интегральный потенциал, показанный в правой части, вполне сравним по объему с тем, на что ориентировано консолидированное меньшинство. Развитые рынки (например, автомобильный), построены так, что массовыми продуктами удовлетворяются потребности массового спроса, а специализированные продукты предназначаются для потребителей с повышенной избирательностью. Проблема состоит в высокой себестоимости специализированных продуктов, поэтому во многих случаях спрос представителей «длинного хвоста» остается неудовлетворенным (рис. 8).

Заключение

В данной работе была отмечена проблема практического отсутствия практики применения бизнесом оценки экономической эффективности ИС перед их внедрением. Также освещена ситуация с наличием методов производства оценки инвестиционной привлекательности ИТ-проектов. Отмечен

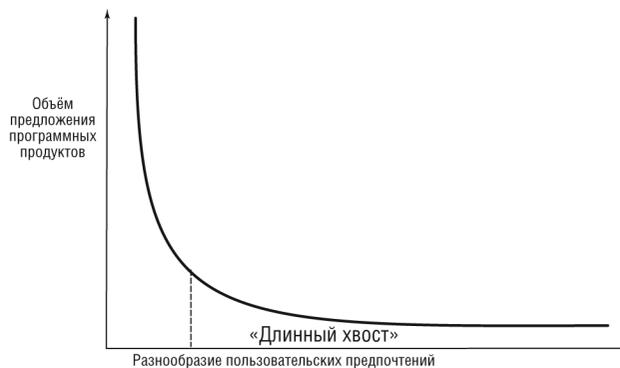


Рис. 8 Диаграмма спрос-предложение на рынке ПО

комплексный метод, предложенный Hares, J., и Royle. Метод несмотря на свою структурированность и проработанность тем не менее не учитывает современные методы и технологии разработки ПО. В работе показаны элементы метода, учитывающие архитектурные особенности ИС и показана из недостаточность.

В настоящее время широкое распространение получила концепция разработки и распространения ПО, по названию SaaS — Software as a Service. Данная концепция имеет принципиальные отличия от традиционной SaaS — Software as a Product. Более того, данная концепция затрагивает не только фазу разработки, но и фазу использования и поддержки компанией, пользующейся данным ПО. Так, например, очевидно, что возможно значительное сокращение издержек за счет того, что само ПО находится физически (и большей частью логически) на стороне провайдера. Причем скорее всего издержки будут сокращаться в большей степени в фазе поддержки, так как она поностью будет на стороне провайдера. Однако есть предположение, что и доходность SaaS по сравнению с SaaS для компаний-пользователей может быть ниже, так как все-таки традиционные приложения под ключ, могут точнее удовлетворять потребности компании и быстрее и эффективнее достигать цели стратегии бизнеса.

Эти предполагаемые закономерности выглядят интересными для изучения и актуальными, так как SaaS по мнению аналитиков набирает обороты и учет этой концепции в методе оценки инвестиционной привлекательности ИТ проектов необходим. ■

Литература

1. Measuring the Value of Information technology/J.Hares, D.Royle – John Wiley and Sons, 1994. – 268с.
2. Р. Ладдад, AOP@Work: Мифы и реальности AOP//IBM – <http://www.ibm.com/developerworks/ru/library/j-aopwork15/>.
3. Software as a Service//Wikipedia – http://en.wikipedia.org/wiki/Software_as_a_Service.
4. Л. Черняк, SaaS – конец начала//Открытые системы –<http://www.osp.ru/os/2007/10/4706040/>.

ФАКТОР ЦЕНОВОГО КЛАССА ВЫПУСКА: НОВЫЕ ПУТИ ИССЛЕДОВАНИЯ И ИЗМЕРЕНИЯ ТЕХНОЛОГИЧЕСКОГО ПРОГРЕССА

А.Р. Горбунов,

к.э.н., старший научный сотрудник Института США и Канады РАН

a_gorbunov07@mail.ru

Измерение технологического прогресса – одна из классических задач эконометрического анализа и стратегических исследований в широком смысле этого понятия. Фактор технологии рассматривается в рамках широкого класса эконометрических моделей, определяемых как производственные функции. Целью статьи является апробация нового подхода – измерения вклада фактора технического прогресса в интегральный экономический результат путем введения в логику эконометрических моделей нового параметра – ценового (капитального) класса выпуска. Анализируется связь эконометрических моделей с методами и моделями системной динамики, различными стратегиями имитационного и математического моделирования экономики и отраслей промышленности в РФ и других странах мира. Ставятся задачи сблизить логику теоретических методов с практикой создания систем поддержки принятия решений. Функция Кобба-Дугласа рассматривается как важный инструмент стратегического анализа.

Введение

Исследования, результаты которых изложены в статье, проводились в рамках определенной научной традиции и научного направления исследования факторов экономического роста и научно-технического прогресса при помощи семейства производственных функций, в основе которых находится функция Кобба-Дугласа. Значительный вклад в теоретические и прикладные исследования, связанные с моделями производственной функции, сделали такие крупные авторитеты как Э.Денисон, Я.Тинберген, Ц.Гриликес, российские исследователи АИ. Анчишкин, Я.З. Демиденко, Б.Н. Михалевский и многие другие. Выбор в качестве базовой модели функции Кообба-Дугласа (ФКД) обусловлен ее простотой и всеобщностью, модифицируемостью, совместимостью с иными подходами и методологиями [1,4].

Все это делает ее универсальным инструментом исследования производительности труда, совокупной факторной эффективности, сравнительных условий производства, типов экономического роста.

Применяется она и в моделях экономического равновесия и экономической динамики. Ниже метод производственной функции совмещен с введенной ранее категорией капитального класса, что открывает новые возможности для исследования и измерения темпов научно-технического прогресса, выявления новых типологий экономического развития и роста. ФКД рассматривается как важный инструмент стратегического анализа.

$$Y = K^{\alpha} \cdot L^{\beta} \cdot C^1$$

Особенностью моделей на основе функции Кобба-Дугласа на наш взгляд является непосредственная связь с концептуальным и управленческим мышлением, тенденцией к исследованию сложных производственно-сбытовых систем как агрегированных объектов. В связи с этим метод аналогичен

¹Функция этого вида предложена американским экономистом Дугласом и математиком Коббом, которые в 1928 году впервые ее использовали для статистического оценивания и экономического анализа.

и совместим с доктринами системной динамики. В частности, она оперирует укрупненными множителями, через которые устанавливается связь со внутренними и внешними переменными моделями и сценариев различного характера. В ряде случаев при помощи ФКД достигается развитие моделей системной динамики, концептуальное обобщение сценариев. Принципы ФКД-анализа относятся к группе базовых идей стратегического анализа, к ядру его «аппарата понимания». Они применимы на предпроектных и предварительных стадиях стратегических исследований, на фазе рабочего моделирования, а также при завершении работ, когда необходимо отрецензировать полученный стратегический результат, дать четкую оценку вероятных решений и сценариев. Некоторые аспекты ФКД позволяют сблизить метод с логикой систем поддержки и принятия решений и лежащих в их основе имитационных моделей. В этой связи она представляется на более предпочтительной, чем более развитие производственные функции семейства CES, «translog» и прочих.

Развитие модели ФКД и тонкие экономические измерения на ее основе осуществляются путем наблюдения за оцениваемыми параметрами при изменении или модифицировании факторных переменных (напр. Э. Денисон, Ц. Грилихес), введением дополнительных факторных переменных (напр. Я. Тинберген, Э. Денисон, Ц. Грилихес), а также путем преобразования выражения ФКД, совместном или параллельном оценивании регрессионных уравнений различного вида (А.И. Анчишкин, И.М. Осадчая, З.Е. Демиденко). Ниже исследования тенденций факторной эффективности, типологий экономического роста и технического прогресса реализованы путем как введения дополнительных факторных переменных, так и модифицированием выражений производственной функции Кобба-Дугласа.

Фактор ценового класса

Классическая традиция ФКД проводит разделение типологий экономического роста, опираясь исключительно на агрегированную стоимостную характеристику всех структурных параметров и величин. Обращает на себя внимание, что здесь отсутствует прямая параметризация облика продукции, структуры выпуска продукции. Любые вводы о ней делаются на основе косвенных данных — анализе параметров и их динамики. Процесс интенсификации производства или научно-технический прогресс выступает как характеристика всей экономической системы, а не как характеристика ее выпуска. Физическое качество продукции здесь исключено, оно как бы «не имеет значения».

В целях сближения логики стратегического анализа производственных систем с логикой компьютерных систем поддержки принятия решений, а также для углубления теоретической базы производственного анализа в структуру экономических моделей фактор облика или физического вида продукции нами был введен напрямую. Для этого была применена категория средней цены рынка или ценового класса продукции, которая стала идентификатором альтернативных выпусков и производственных структур. Она была трансформирована в категорию капитального класса (capital class) — удельной капиталоемкости выпуска с учетом масштаба (серийности) производства, рыночной ниши и позиции в мире товаров и услуг, с учетом межотраслевых условий и альтернатив [2].

Категория ценового (капитального) класса служит для:

- ✧ «шкалирования» стратегических решений и моделей принятия решений в алгоритмах СППР;
- ✧ построения концептуальных моделей о стратегических (продуктовых) альтернативах на начальных фазах стратегических и проектных исследований;
- ✧ введения дополнительных технико-экономических и структурных координат в модели экономической динамики различных типов.

Итак, в целях экспресс-моделирования структурных процессов в экономике и менеджменте был выработан подход прямого введения в модели затраты — выпуск параметра, характеризующего физический облик и структура производства, — ценовый (капитальный) класс продукции (capital class). Для него в базовых моделях придана предельно упрощенная конструкция

$$Yq = f(x)/P \cdot k, \quad k = (P/Pb)^m \text{ и ее варианты} \quad (1)$$

$$Yq = f(x)/Pb \cdot (Pb/P)^m \quad (2)$$

$$Yq = f(x)/Pb \cdot |1 - (P/Pb)^m|, \quad (3)$$

где $f(x)$ — мера емкости рынка, зависящая от различных факторов;

Pb — ценовой (капитальный) класс, средняя цена на рынок;

P — проверяемая цена, фактическая цена;

m — коэффициент эластичности, влияющий на эффект расхождение цены товара от средней цены рынка (капитального класса).

Выражения (1–3), их иные версии обладают рядом удобных для моделирования алгебраических свойств [2].

Модель анализирует соответствие цены капитальному классу, нарушение которого влияет на сбыт. Это сбытовые функции, значение которых определяется двумя величинами – ценой и капитальным классом. Это семейство зависимостей в области производства и потребления было введено нами ранее как *функции капитального класса*.

Продажи (потребление) увязывались с ценой продукции, так и ее структурным классом, представленным средней ценой рынка на продукцию данного класса. Комбинирование уравнений затрат на производство и сбытовой функции позволяет построить модели производственно-сбытовых систем. При этом экономический результат строится от двух величин – затрат на производство продукции и параметра физического облика, то есть выбора вида продукции. На этой основе возможны новые конструкции оптимизации производства и сбыта в зависимости от целевых рынков, их характеристик и конъюнктурной динамики. К примеру, он позволяет провести стратегический анализ производственной программы концерна или отраслевой группы, промышленности государства в целом.

Однако, возвращаясь к модели производственной функции, ценовой класс не может быть непосредственно введен в модель производственной функции без нарушения однородности модели и ущерба ее логике (капитальный класс должен быть соразмерен другим факторам производства). В этой связи фактор капитального класса в модель ФКД был введен иным путем – через показатель натуральной количественной меры выпуска. Коэффициенты переливаются с учетом статистического оценивания на временных рядах и позволяют выявить вклад фактора количественной (натуральной) меры выпуска, то есть в конечном счете – вклад фактора капитального класса.

Мы ввели в модели производственного анализа фактор и параметр средней цены выпуска продукции или по смыслу связанный с ним измеритель натурального (количественного) масштаба выпуска.

Базовые модели производственной функции
Исходная модель Кобба-Дугласа выглядит

$$Y = K^\alpha L^\beta C, \quad (4)$$

где K и L – факторные затраты, труд и капитал;

Y – один из измерителей экономического роста, например выпуск продукции или ВВП.

Обычно (4) оценивается как регрессионное уравнение, приведенное в вид

$$\ln Y = \alpha \ln K + \beta \ln L + \ln C. \quad (5)$$

Здесь результат представлен как сумма вкладов факторов производства. Коэффициенты эластичности α и β показывают зависимость темпов роста результата от темпа роста факторных затрат, исчисленных в индексном или процентном выражении.

(Темп роста продукции равен сумме темпов роста эффекта труда и капитала).

$$y = \alpha k + \beta l. \quad (6)$$

Часто применяется уравнение со связкой, где для $\alpha_1 + \alpha_2 = 1$ оценивается через регрессионное уравнение

$$Y/L = (K/L)^{\alpha_1} \cdot e^{\gamma} \cdot C. \quad (7)$$

Его можно интерпретировать как зависимость между производительностью труда и капиталовооруженностью.

Допускается введение третьего и других ресурсов и соответствующее нормирование по связке эластичностей факторов производства. Для трехфакторной модели с дополнительным ресурсом или иным фактором M оценивание

$$Y = K^{\alpha_1} \cdot L^{\alpha_2} \cdot M^{\alpha_3} \cdot e^{\gamma} \cdot C \quad (8)$$

при ограничении $\alpha_1 + \alpha_2 + \alpha_3 = 1$ обеспечивается простым преобразованием предыдущей формулы.

$$Y/L = (K/L)^{\alpha_1} \cdot (M/L)^{\alpha_3} \cdot C. \quad (9)$$

Следует обратить внимание, что при этом не существует принципиальных препятствий для наложения связей не равных 1, например $\alpha_1 + \alpha_2 + \alpha_3 = 2$.

$$Y/L^2 = (K/L)^{\alpha_1} \cdot (M/L)^{\alpha_3} \cdot C. \quad (10)$$

Представляется, что такой подход навязывает более реалистичное ограничение для оценок параметров $\alpha_1, \alpha_2, \alpha_3$, поскольку на практике наиболее часто они более или менее близки к 1. Последовательное применение связей различной размерности от 1 до 2 и иных в некоторых вариантах статистических исследований способно извлекать дополнительную информацию и знания из временных рядов. Преобразования (8, 10) исключают или понижают степень мультиколлинеарности исходных переменных, которая часто является серьезной проблемой при решении задач статистического оценивания.

Измерение научно-технического прогресса

Основания для применения категории капитального класса в целях измерения экономического роста находятся в классическом аппарате производственных функций. Базовой моделью является производственная функция Кобба-Дугласа, с введенным Я. Тинбергеном фактором технологии e^{gt} .

$$Y = K^\alpha \cdot L^\beta \cdot e^{gt} \cdot C, \quad (11)$$

где K – капитал, привлеченный для производства продукции Y ;

L – трудовые ресурсы привлеченные для производства продукции Y ;

g – темп технологического прогресса, обуславливающий вклад фактора технологии в выпуск.

Для этого рассматривается процедура введения ограничения на коэффициенты эластичности $\alpha + \beta = 1$. Стандартная процедура наложения связи приводит к выражению, связывающему капиталовооруженность и производительность труда.

$$Y/L = (K/L)^\alpha \cdot e^{gt} \cdot C. \quad (12)$$

С помощью данной модели и ее модификаций было установлено, что вклад научно-технического прогресса, то есть совокупность интенсивных факторов экономического роста на протяжении всех последних десятилетий составляла не менее 40–50% экономического результата. Но этот общий вывод дополняется многоуровневым дезагрегированным анализом и дальнейшим развитием концепции производственной функции.

Наличие в модели ФКД коэффициента Тинбергена вносит в нее существенные коррективы. Наложение ограничения $\alpha + \beta = 1$ позволяет по методу совместного оценивания уравнений функции Кобба-Дугласа разделить вклад фактора технологии овеществленный в труде и капитале, а также вклад технологии независимый от материальных факторов и связанный исключительно с фактором времени («темп НТП»).

Академиком А.И. Анчишкиным была предложена схема совместного оценивания параметров производственной функции Кобба-Дугласа, которая в дальнейшем была развита и дополнена многими известными экономистами, включая Е.З. Демиденко и И.М. Осадачаю [1]. Смысл ее заключался в том, что по логике модели при наличии связи коэффициентов эластичности результат всегда адекватен масштабу привлеченной комбинации ресурсов. Соответственно, весь прирост продукции, выходящий за эти

пределы переходит в оценку фактора g , что и является главной предпосылкой оценки факторов интенсификации производства и эффекта технологии. Следует отметить, что в оценку фактора технологии g при подобном оценивании переходит и нелинейный эффект масштаба, который рассматривается как проявление фактора технологии.

Исходя из этого становится очевидным, что при условии связки эффект технологии и масштаба производства останется в коэффициентах α и β . Соответственно, сравнивая эффекты труда и капитала можно оценить компоненты и, прежде всего, вклад научно-технического прогресса, материализованного в фактор труда и капитала, а также эффект технологии G , не связанный непосредственно с ними и связанный исключительно с фактором времени. Эта последняя величина и является «чистой» технологической составляющей выпуска или экономического роста.

Если обозначить эффект капитала, то есть его вклад в конечный результат Y оцененный для (4), то есть без наложения связей K , а полученный аналогично эффект труда L , то

$$G' = (L-L') + (K-K') + G, \quad (13)$$

где G' – вклад технологии, измеренный по формуле (11), то есть без каких-либо связей, L' – вклад в экономический результат фактора труда по формуле со связкой. Вклад фактора капитала в экономический результат по формуле со связкой (12) обозначен K' .

Перелив величин параметров при совместном оценивании выражений и позволяет сделать выводы о соотношении интенсивных ($K-K'$, $L-L'$, $G'-G$, G) и экстенсивных (K' , L') факторов экономического роста, о типологиях экономического роста и производственных систем.

Производственная функция с фактором ценового класса

Фактор капитального класса интерпретирован как фактор масштаба производства серийности, физического размера выпуска и вводится в правую часть уравнения наряду с ресурсами и виртуальным фактором времени как величина Yq^1 . Статистический анализ эффекта этой группы факторов позволяет выявить роль фактора капитального класса, представленного величиной физического масштаба или серийности

¹Фактор может быть перенесен и в левую часть, но при этом изменяется смысл модели, в частности, «снимаются» эффект абсолютных величин. Это сделано ниже в соответствующем контексте (20, 22).

производства (динамику капитального класса как структурную эволюцию символизирует динамика затрат относительно Yq). Соответственно, фактор серийности интерпретируется как фактор капитального класса.

Смысл подхода можно интерпретировать следующим образом: затраченные ресурсы «размещены» в Yq объектов. Тогда эффект производственной серии или эффект Yq становится важной характеристикой данной производственной системы в целом. В зависимости от взаимной динамики факторов производства, его серийности (масштаба) и наблюдаемой эффективности делаются выводы о свойствах и типологии экономического роста и производственных систем.

Фактор капитального класса (как размер производственной серии) может быть введен двумя способами.

$$Yq = K^\alpha \cdot L^\beta \cdot e^{\gamma_1 Yq} \cdot e^{\gamma_2 t} \cdot C, \quad (14)$$

$$Yq = K^\alpha \cdot L^\beta \cdot Yq^{\gamma_1} \cdot e^{\gamma_2 t} \cdot C, \quad (15)$$

В первом случае (14) фактор масштаба производства введен аналогично фактору технологии Тинбергена. Во втором случае (15) фактор технологии введен аналогично фактора ресурсов. По смыслу модели предпочтение было отдано именно этому варианту записи, поскольку виртуальный фактор физического масштаба выпуска вариабелен и соразмерен параметру ресурсов, а не «неограниченному» фактору времени. Предпочтение было отдано второй записи также и потому, что ниже для трёх «ресурсов» предложена процедура наложения связки.

$$Yq = R^\alpha \cdot Yq^{\gamma_1} \cdot e^{\gamma_2 t} \cdot C \quad (16)$$

или еще проще

$$Yq = R^\alpha \cdot Yq^{\gamma_1} \cdot C; \quad (17)$$

$$Yq = R^\alpha \cdot Yq^{m + \gamma_1} \cdot e^{\gamma_2 t} \cdot C, \quad (18)$$

где m – «стандартный» фактор эффекта масштаба¹;
 R – объединённый фактор ресурсов.

Фактор капитального класса «на месте» фактора ресурса является величиной виртуальной, но с алгебраической точки зрения и с точки зрения статистического оценивания выражения (11) и (15) совершенно аналогичны. Факторы затрат, натурального и стоимостного выпуска нередко в высокой степени линейно зависимы. Модели (14–16) наиболее адекватны

¹ Данный вид анализа сфокусирован главным образом на факторе масштаба производства, связанном с изменением структуры производства или переходом к продукции иного ценового класса. – *Прим. автора.*

при переходе к новой продукции – изменению физического содержания выпуска, его структуры, а также параметров эффективности труда и факторной эффективности экономической системы при процессе структурной перестройке экономической системы.

Описание производственной системы (14) дополняет зависимость

$$Yq = f(R) + C, \text{ точнее см. (20)}. \quad (19)$$

При постоянной продуктовой программе данная зависимость близка к линейной. Коллинеарность снижается при нелинейном характере зависимости, в частности, при четких сдвигах в структуре и облике выпуска. Аналогичная ситуация имеет место и в базовой модели Кобба–Дугласа. Факторы труда капитала и выпуска линейно зависимы в стационарных условиях. Но модель наиболее информативна в условия структурной и экономической динамики, для измерения которой она и предназначена. Она «реагирует» на асимметрию факторов и эффекта труда и капитала.

Подобную асимметрию фиксирует и модель (16, 17). Коллинеарность может свидетельствовать о тривиальном характере экономической динамики. При этом неколлинеарность и устойчивость статистических оценок свидетельствует о значимых изменениях в производственной системе. В этом случае связь между факторными параметрами (с учетом логарифмирования) обычно является нелинейной. В этой связи определенный интерес представляет сплайн-анализ – разбиение временных рядов и соответствующих аппроксимирующих трендов (производственных функций) на кусочно-линейные (и кусочно-нелинейные) участки, в рамках которых статистические оценки наиболее устойчивы и достоверны.

В производственном потоке происходит наложение различных процессов, событий и тенденций, поэтому связь количественной меры выпуска и затрат носит индетерминированный, стохастический характер. Метод выявляет действие суммы производственных процессов с различными характеристиками на основе статистической информации. Связь выпуска и затрат в (физическом выражении) характеризует запись

$$Yq = f(R) + C + E, \quad (20)$$

где E – величина статистической ошибки.

Следует обратить внимание, что, как и выше, мультиколлинеарность для (16) по смыслу модели полностью устраняется наложением связки на

параметры эластичности факторов производства. Для $\alpha + \gamma_1 = 1$ оценивалось регрессионное уравнение

$$Y/R = (Yq/R)^{\gamma_1} \cdot e^{\gamma_2 t} \cdot C. \quad (21)$$

Аналогичным образом при перестановке переменных уравнение выглядит как

$$Y/Yq = (R/Yq)^{\alpha} \cdot e^{\gamma_2 t} \cdot C, \quad (22)$$

согласно чему капиталоемкость выпуска является функцией ресурсоемкости затрат.

Если возможна связка, то возможен и совместный анализ по общей схеме А.И. Анчишкина. Модели (16, 22) позволяют реализовать известную программу совместного оценивания производственной функции с наличием связки эластичностей по единице и без нее. «Перетекание» параметров позволяет оценить нелинейную составляющую (эффект масштаба производства) по эффекту масштаба R , нелинейный эффект изменения фактора капитального класса (через эффект Yq). Очевидно, существует и независимый от фактора масштаба производства и фактора капитального класса вклад совокупной эффективности факторов производства (через эффект времени). В его состав может входить эффект более совершенных производственных технологий, накопление производственного опыта (опытная кривая). Метод позволяет выявить экономический рост, ориентированный на повышение (понижение) ценового (капитального) класса выпуска и экономический рост, связанный исключительно с масштабом производства, экономический рост, независимый от того и другого и различные комбинации. В целом технический прогресс и динамика эффективности характерным образом сопровождается динамикой ценового класса и структуры выпуска продукции. На выявление этих зависимостей и нацелен приведенный метод. Наложённая связка вызывает перелив эффекта серии Yq относительно затрат R в оценку фактора технологии G_2' .

$$G_2' = (Yq - Yq') + (R - R') + G_2. \quad (23)$$

Наиболее надежно «чистый» фактор капитального класса выявляется через $(Yq - Yq') + (R - R')$, поскольку параметры затрат и выпуска в той или иной степени содержат линейно-зависимую компоненту, которую однозначно устранить обычно нельзя. Соответственно нельзя точно разделить капитальный фактор между R и Yq . Следующий шаг заключается в дезагрегировании фактора интегральных затрат R и возвращении к комбинации затрат труда L и капитала K .

Степень коллинеарности понижается наложением связки эластичностей при трех факторах производства. Для $Y = K^{\alpha_1} \cdot L^{\alpha_2} \cdot Yq^{\gamma_1} \cdot e^{\gamma_2 t} \cdot C$ при ограничении $\alpha_1 + \alpha_2 + \gamma_1 = 1$ обеспечивается как и в (8, 9) преобразованием предыдущей формулы

$$Y/Yq = (K/Yq)^{\alpha_1} \cdot (L/Yq)^{\alpha_2} \cdot e^{\gamma_2 t} \cdot C. \quad (24)$$

Приведенная выше схема Анчишкина допускает дальнейшее дезагрегирование путем совместного оценивания и замен переменных, из которых вытекают дополнительные типологии экономического роста и производственных систем.

Новые типологии экономического роста и производственных систем

Анализ производственных систем по схемам функции Кобба-Дугласа позволяет выявить интенсивные и экстенсивные типы производственных систем и экономического роста, их трудо- и капиталоемкие подразделения. «Овеществление» (см. выше) технического прогресса в факторах труда или капитала указывает на дальнейшие модификации экономического роста в виде его трудо- и (или) капиталоемких версий, придает этим характеристикам количественную форму [напр. 6].

Продемонстрированное выше видоизменение схем и концепции производственной функции, включение в рассмотрение параметра капитального класса позволяет существенно расширить подход к классификации экономических систем. Оно идентифицирует производственные стратегии и системы, тяготеющие либо к повышению капитального класса выпуска либо его снижению, к производству продукции высокого и низкого капитального класса и их комбинациям. По связи основных факторов с эффектом фактора капитального класса (серийности) выпуска идентифицируются «капитал-серийные» (*capitalserial*) и «трудо-серийные» (*laborserial*) модификации экономического роста и производственных систем, а также их интенсивные и экстенсивные версии и их комбинации.

Международная специализация в страны (в самой общей форме и на фоне отраслевых и технологических особенностей производства) также сводится к выпуску относительно более или менее тиражированных (и соответственно более или менее «дорогих», капиталоемких видов продукции). Не только потребительские товары, но промышленное оборудование и транспортные средства могут идентифицироваться как объекты более или менее капиталоемкие. Этому подразделению полегит большинство отраслевых производственных и сбытовых номенклатур.

Установлено, что существуют тонкие различия в специализации США и, например, Японии, стран Европы. Торговые дефициты США на протяжении десятилетий складывались за счет «неприоритетных» для стратегических позиций страны групп товаров (потребительская электроника, бытовая электротехника, стандартные виды электронных компонентов). Они балансировались наиболее наукоемкой продукцией высокой меры капиталоемкости (капитального класса) – высокотехнологичным промышленным оборудованием, промышленной электроникой, продукцией авиакосмического комплекса, боевой техникой и вооружениями [3]. Если США специализируются на промышленной электронике, то для Японии и стран ЮВА более характерно производство электроники потребительской сферы, электронных устройств и компоненты массовых стандартных типов.

Детали специализации стран уточняются при помощи приведенных схем анализа, которые доступны для всех ситуаций, где присутствует статистика не только по стоимостному, но и физическому объему выпуска.

Аналогичные фундаментальные особенности производства и выпуска характерны для корпоративных индустриальных систем. Метод расширяет возможности описания стратегического маневрирования корпораций. Переход к продукции иного капитального класса, сопряженный с новым комплексом маркетинговых и технико-экономических характеристик, становится предметом сценариев и стратегических планов.

Сценарии стратегической конкуренции, стратегической игры тоже обобщаются как борьба за выгодные и рынки и рыночные сегменты по параметрам удельной капиталоемкости и масштаба операций. Очевидно, лидирующей стратегией является производство объектов максимального капитального класса глобальными сериями (серийное производство аэробусов). Стратегией высокого качества может являться производство продукции не только максимального, но малого капитального класса. К таковой, к примеру, относится производство, пальчиковых батареек глобальными сериями. С этих позиций «арьергардная» абстрактная стратегия состоит в производстве продукции низкого капитального класса малыми масштабами (впрочем кустарные цеха и микробюджетная индустрия Китая обеспечила стране видные экономические позиции – по целому ряду товаров низкого ценового класса – игрушкам, предметам обихода, многим видам продукции легкой промышленности).

Таким образом, корпоративные индустриальные системы, концерны и отраслевые группы имеют

свои индивидуальные «спектры» производственных возможностей и особенности научно-промышленного потенциала. Продуктовые линейки и модельные ряды распределены по параметру ценового класса продукции. Такие различия становятся доступны для методов количественного анализа, непосредственным образом регистрирующих различия по структурным характеристикам затрат и выпуска.

Следует отметить, корпоративных структуры создают глобальные комбинации производственных систем, совмещающие сравнительные преимущества производственных систем различных типов. К ним относятся стратегические партнерства корпораций Запада и Востока, трансатлантические, панамериканские и трансевразийские структурные стратегические альянсы.

Из всего этого следует, что каждая корпорация, отраслевая группа или национальная экономика должна стремиться выйти на наилучшие позиции с точки зрения капитального класса и масштаба выпуска и его перспектив. В этом во многом заключается стратегия роста меры добавленной стоимости (Value Adding Strategy), в абстрактное описание которой включен параметр обобщенного физического облика выпуска (как параметр среднего капитального класса для отраслей и экономики в целом, для производства и потребления).

В соответствии с принципом капитального класса сегментировано не только производство, но и потребление, сбыт товаров. Более того, капитальная структура подвержена циклическим колебаниям. Хорошо известно, что в период спадов экономической конъюнктуры из продуктовых линеек исключаются в первую очередь относительно дорогие позиции. В период экономических подъемов происходит вымывание дешевого ассортимента, повышение структурного класса товаров и услуг. Структурные характеристики производства и потребления находятся в сложном взаимодействии, динамике, сопровождающие колебания совокупного спроса и предложения, условий реализации ВВП. Особый интерес представляет исследование с точки зрения роли структурных и статусных условий глубокого мирового экономического кризиса, стартовавшего в 2008 г. Кризис сопровождается признаками демонтажа целых экономических комплексов и систем различного капитального класса (начиная с наиболее капиталоемких).

Оценки фактора капитального класса применимы везде, где есть статистика натурального или штучного объема производства и потребления. Измерения производятся на основе данных в постоянных ценах поскольку, внимание сфокусировано на процессах объективного роста (изменения) «средних»

цен до учета инфляции и влияния денежной составляющей. Обобщенные натуральные измерители применимы в теоретических моделях, изучающих принципиальные и теоретические процессы структурной эволюции экономических систем.

Отраслевые и национальные сценарии

Метод производственной функции имеет большое прикладное значение. Дело в том, что в основе отраслевого или национального сценария находится укрупненная предварительная гипотеза и сокращенная модель, выражающая основной смысл и основные процессы, имеющие место в экономической системе. Модель производственной функции – это обобщенная характеристика производственной системы, конкретизируемая в различных сценариях различными методами. Производственная функция с учетом обратных связей и ограничений развивается в конечном счете «сжатую» модель производства ВВП страны или отрасли ее экономики. Она определяет общие ориентиры по ресурсам и эффективности. Из общей установки выводится абстрактная задача о распределении ресурсов, а также конкретные отраслевые планы, политические концепции и платформы.

Обобщенная производственная функция сопровождает весь цикл стратегических исследований, экономической политики и планирования развития отраслей и регионов страны. Она определяет направленность конкретных сценариев. В некоторых случаях производственная функция вводится в непосредственном виде как математическая и алгоритмическая основа модели или сценария.

Международная специализация на продукции высокого капитального класса промышленного оборудования и энергетических машин, промышленной электроники вооружения боевой техники, узлов и компонентов для этих групп – основа про-

изводственной стратегии РФ. В системе приоритетов свое место занимают изделия иных структурных капитальных классов (автомобилестроение, бытовая и потребительская электроника и электротехника, офисная и персональная электроника и пр). Для автомобилестроения – производство дешевых автомобилей национального бюджетного класса и сближение с международными стандартами в другой части продуктовых линеек.

Свое место в этой схеме занимает и «малая экономика» – продукция артелей микрохозяйств, самозанятость. Утилитарные номенклатуры, внося свой вклад в производственную стратегию РФ.

Производства различных отраслей различаются по физическому (штучному) масштабу выпуска, характерной ценой, условиями сбыта и производства, особенностями жизненного цикла. Отраслевые приоритеты находятся как минимум в русле экономических измерений с фактором капитального класса (серийности выпуска).

Экономика России должна не только потреблять, но и производить ресурсы собственного развития. Ее необходимо нацеливать на дешевое производство, в том числе продукции высокого капитального класса. Развитая «избыточная» сфера НИОКР способна укрепить преимущества РФ в ключевой группе производств. Политика доходов и издержек, структурная политика – регулирование форм и методов хозяйственной деятельности развитие приоритетных направлений отраслей способно «улучшить» интегральную производственную функцию РФ.

Нечеткость сегментации сектора высоких технологий, расплывчивость отраслевой структуры и межотраслевой коопераций, размывание специализаций и компетенций, неэффективность уровней управления. На все эти аспекты выводит концепция «стратегической» производственной функции и построенных на ее основе сценариев. ■

Литература

1. Научно-технический прогресс и капиталистическое воспроизводство / С.М. Никитин, И.М. Осадчая, Э. Рехтциглер и др. отв. ред. С.М. Никитин и др. – М.: Мысль, 1987–207 с.
2. Горбунов А.Р. Системная динамика моделирование принятия стратегических и оперативных решений (метод функции капитального класса) Бизнес-информатика N2(04)-2008.
3. А.Р. Горбунов. Машиностроительный комплекс и внешняя торговля., США-ЭПИ, N1 за 1990.
4. Леонтьев В. Экономические эссе. Теории, исследования, факты и политика. пер с англ.– М. Политиздат, 1990–415с.
5. Анчишкин А.И. Прогнозирование социалистической экономики. М., 1973.
6. Productivity Growth and US Competitiveness. Edited by William J. Baumol and Kenneth McLennan. A Supplementary Paper of the Committee for Economic Development, Oxford University Press, 1985.

РАЗРАБОТКА И АПРОБАЦИЯ СИСТЕМЫ ПОИСКА ДУБЛИКАТОВ В ТЕКСТАХ ПРОЕКТНОЙ ДОКУМЕНТАЦИИ

Д.И. Игнатов,

д.ф.-м. н., Государственный Университет – Высшая Школа Экономики

С.О. Кузнецов,

Государственный Университет – Высшая Школа Экономики

skuznetsov@hse.ru

В.Б. Лопатникова, к.т.н., ООО «Кварта ВК»

И.А. Селицкий, ООО «Мастерхост»

В статье рассмотрена система поиска (почти) дубликатов в текстах проектной документации. Описаны ее архитектура, математические модели и алгоритмы поиска документов-дубликатов, а также их реализация. Предложены методики подбора оптимальных параметров методов и тестирования системы. Обозначены актуальные для подобных систем исследовательские задачи.

Постановка задачи и актуальность

Выявление дублирования в текстах проектной документации особенно важно при анализе эффективности результатов выполнения научно-исследовательских и опытно-конструкторских работ (НИОКР), финансируемых за счет бюджетных средств. Автоматизация поиска дубликатов проектных документов позволяет повысить качество приемки работ за счет упрощения деятельности экспертов при анализе результатов выполнения НИОКР. Внедрение автоматизированной системы поиска дубликатов обеспечит формирование фонда уникальных разработок, полученных в ходе выполнения НИОКР. Исключение дублирующихся разработок является неотъемлемой частью патентных исследований.

В настоящее время широкое распространение получила система «Антиплагиат» [AntiPlagiat, 2008], предназначенная для обнаружения результатов плагиата в курсовых, реферативных и дипломных работах студентов. На сайте разработчика [Forecsys, 2008], компании «Форексис», в 2007 году сообщалось также о создании пакета «Антиплагиат.ВАК»,

предназначенного для выявления авторства и фактов плагиата в диссертационных работах. Потребность решать похожие задачи с помощью такого рода поисковых систем возникает и в других областях, связанных с научной и проектной деятельностью.

Цель нашего проекта состоит в разработке методологии и программного инструментария для анализа дублирования в текстах проектной документации НИОКР.

Основная задача проекта – отслеживать недобросовестное копирование документов других авторов, в том числе и своих собственных текстов.

Уточним, что в данной работе под дублированием мы будем понимать значительное совпадение фрагментов текстов.

Для достижения поставленной цели нами были отобраны представительные методы, наиболее подходящие для решения проблемы сравнения слабоструктурированных текстов разной тематической направленности. Был создан программный комплекс, основой которого является библиотека настраиваемых алгоритмов, реализующих выбранные методы анализа дублирования. Затем были проведены

эксперименты по подбору параметров алгоритмов, позволяющих с наименьшей потерей точности отнести анализируемый документ к классам «уникальный» или «дубликат». В результате нами предложена технология оптимизации анализа текстов, позволяющая исключить из поискового пространства те документы, которые заведомо не могут быть дубликатами. По сравнению с системой «Антиплагиат», использующей по существу синтаксические методы, разработанный программный продукт обладает рядом ценных для аналитика преимуществ. Во-первых, в системе используются известные модели и методы поиска дубликатов (см. раздел 3), поведение которых изучено для различных типов и коллекций документов, а их достоинства и недостатки хорошо известны. Аналитик (пользователь системы) может самостоятельно выбирать один или несколько методов для поиска сходных документов, изменять параметры, установленные для каждого из них по умолчанию. Тем самым снимается эффект «черного ящика» при использовании системы. Во-вторых, аналитику предоставляется возможность указать уровень текстуального сходства документов и даже выбрать вид агрегированной меры сходства (в случае если используется несколько методов одновременно). Это позволяет учесть специфику методов и изменить вес того или иного из них для достижения более точных результатов. В работе предложена авторская методика автоматической калибровки методов по полноте и точности поиска, которая также представлена в системе в виде отдельного модуля.

Описание системы

Система представляет собой комплекс программных средств, предназначенных для анализа дублирования текстов проектной документации. Система также реализует поддержку жизненного цикла НИОКР за счет контроля этапов подготовки и проведения работ. В состав Системы входят следующие функциональные модули: модуль нормативной базы, библиотека НИОКР, модуль справочников и классификаторов, аналитический модуль, модуль выполнения отчетов.

Модуль нормативной базы реализует функции ввода и хранения нормативно-методических документов, регламентирующих порядок выполнения НИОКР.

Библиотека НИОКР реализует функции ввода и хранения данных о НИОКР в виде карточек НИОКР и текстов проектной документации как результата выполнения НИОКР.

Модуль справочников и классификаторов предна-

значен для ведения справочников юридических лиц, являющихся заказчиками и исполнителями НИОКР, а также справочников видов документов. В Системе доступен также справочник ГРНТИ.

Аналитический модуль реализует проверку текстов проектных документов НИОКР в форматах TXT, DOC, RTF, PDF на повторяемость в рамках заданной коллекции документов. Модуль позволяет задавать различные параметры анализа для каждой коллекции документов.

Модуль выполнения отчетов предназначен для просмотра информации о состоянии НИОКР в различных разрезах (тематика, состояние выполнения, исполнители, наличие документов-дубликатов и др.).

Система реализована по принципу клиент-серверной архитектуры, с инкапсуляцией ядра системы на уровне СУБД с доступом через Web-браузер. База данных и ядро системы реализуются на основе СУБД Microsoft SQL Server 2005, а дополнительные библиотеки – на языке C# и работают в среде .NET Framework CLR.

Архитектурно база данных Системы построена на основе объектно-реляционной модели и делится на три логических фрагмента:

- ✧ мета-данные, описывающие информационные объекты, их взаимосвязи и объекты визуальных представлений данных (экранные и печатные формы);
- ✧ внесённые данные о НИОКР;
- ✧ программное ядро, состоящее из хранимых процедур и функций на языке T-SQL и дополнительных библиотек на языке C#.

Web-клиент Системы реализован в виде ASP.NET приложения, работающего под управлением Microsoft Internet Information Services.

Методы поиска дубликатов

При нахождении множеств (почти) дубликатов документов основными являются следующие этапы:

1. Представление документов множеством признаков;
2. Составление образа документа путем выбора подмножества признаков;
3. определение отношения сходства на образах документов.

Затем, в зависимости от конкретной задачи, могут проводиться и комбинироваться следующие этапы:

4. Вычисление кластеров сходных документов;
5. Слияние кластеров сходных документов из различных коллекций;
6. Принятие решений о дублировании и компиляции.

На первом этапе, после снятия разметки (например, HTML), документы, как линейные последовательности слов (символов), преобразуются во множества слов, возможно с указанием кратности вхождения. Здесь двумя основными типами методов, определяющими весь возможный спектр смешанных стратегий, являются:

1. Синтаксические методы (в которых осуществляется выбор последовательностей символов, слов, или предложений);

2. Лексические (семантические) методы (в которых происходит выбор представительных языковых единиц).

Основным синтаксическим методом является шинглирование, когда документ, очищенный от разметки, пробелов и знаков препинания, сперва представляется набором всех подцепочек последовательных слов (символов) определенной длины. Такие цепочки, выбираемые с определенным сдвигом по линейной структуре текста, называют шинглами (от англ. shingle — черепица, чешуйка). Каждой цепочке сопоставляется хеш-код, при выборе которого обеспечиваются следующие важные свойства: равенство цепочек гарантирует равенство кодов (т.е. кодирование есть хеш-функция), а равенство кодов говорит о высоком сходстве цепочек. Наиболее распространенными являются хеш-коды SHA1 [NIST, 1995] и Rabin [Broder, 1997]. Необходимым условием является минимальное число коллизий для хеш-функций. Из множества хеш-кодов цепочек, в соответствии с некоторой схемой рандомизации, выбирается подмножество, которое и служит т.н. «отпечатком» (образом) документа. Данный метод используется во многих системах определения сходства документов, а также в таких поисковых системах как Google и AltaVista.

Среди способов выбора подцепочек используются следующие методы выбора подцепочек: фиксированный, или логарифмического от длины текста, выбор каждой k -й цепочки и т.д.

В методах лексического типа реализуется отбор множества представительных слов исходя из показателей значимости этих слов. В множество значимых слов не включаются слова из заранее фиксированного списка стоп-слов. Список стоп-слов для каждого языка является стандартным и включает в себя предлоги, артикли, вводные слова и т.п. Показателями значимости служат частотные характеристики: для дальнейшего анализа отбираются слова, чьи частоты лежат в некотором интервале, так как высокочастотные слова могут быть неинформативными, а низкочастотные — опечатками или случайными словами.

В лексических методах, таких как, в известном методе I-Match [Chowdhury et al., 2002], используют большой текстовый корпус для порождения лексикона, то есть набора представительных слов. Документ представляется множеством тех его слов, которые входят в лексикон. При порождении лексикона отбрасываются самые низкочастотные и самые высокочастотные слова. I-Match порождает сигнатуру документа (множество слов-термов), а по ней — хэш-код документа, причем два документа получают один хэш-код с вероятностью равной их мере сходства (по метрике косинуса). I-Match порой неустойчив к изменениям текста [Kocuzetal., 2004], например, к рандомизации по существу одних и тех же спамерских сообщений. Для устранения этого недостатка, помимо стандартной сигнатуры, создается еще K сигнатур, каждая из которых получается случайным удалением некоторой доли всех термов из исходной сигнатуры (таким образом, все новые сигнатуры являются подмножествами исходной). Два документа можно считать очень сходными, если их наборы из $K+1$ сигнатуры имеют большое пересечение хотя бы по одной из сигнатур. Такой подход сходен с подходом на основе супершинглов (конкатенации шинглов), когда сходство документов определяется как совпадение хотя бы одного супершингла [Kocuz et al., 2004].

В лексическом методе [Plyinsky et al., 2002] большое внимание уделяется построению словаря — набора дескриптивных слов, который должен быть небольшим, но хорошо покрывать коллекцию, а присутствие каждого из слов в образе документа устойчиво по отношению к малым изменениям документов. Проблема автоматического порождения адекватного словаря для анализа сходства документов по определенной теме связана с составлением представительной коллекции документов по данной теме — корпуса текстов, что затрудняет применение лексических методов без постоянной поддержки такого корпуса.

На первом этапе можно учесть структуру текста и его шаблонов, представляя исходный документа в виде кортежа разделов определенных шаблонами и производя шинглирование внутри компонент кортежа.

На втором этапе из документа, представленного множеством синтаксических или лексических признаков, выбирается подмножество признаков, образующее краткое описание (образ) документа. В синтаксических методах такого рода отбор чаще всего осуществляется с помощью схем рандомизации [Broder, 1997, Broder et al., 1997, 1998], в лексических методах — с помощью методов выбора

существенных слов, например, на основе заранее созданных словарей [Chowdhury et al., 2002], или на основе какой-либо меры существенности слова для текста [Plyinsky et al., 2002].

Техника отбора подстрок, начиная с некоторого определенного места в документе, также позволяет учесть структуру документа с использованием так называемых «якорей» — особых мест документа: начала абзаца, раздела, ключевого слова и т.п. [Hoad et al., 2003]. Являясь одной из самых эффективных, эта техника может потребовать много времени для тонкой ручной настройки по каждой коллекции документов.

На третьем этапе определяется отношение сходства на документах. Для этого используется определенная числовая мера сходства, сопоставляющая двум документам число на отрезке $[0, 1]$, которое характеризует сходство, и некоторый параметр — порог, превышение которого свидетельствует о большом сходстве документов или о том, что документы являются (почти) дубликатами друг друга [Broder, 1997, Broder et al., 1997, 1998].

На четвертом этапе, на основе отношения сходства документы могут объединяться в кластеры (почти)дубликатов. Определение кластера также может варьироваться. Самый частый используемый на практике подход [Broder, 1997]: если документам сопоставить граф, вершины которого соответствуют самим документам, а ребра — отношению «быть (почти) дубликатом», то кластером объявляется компонента связности такого графа. Достоинством такого определения является эффективность вычислений: компоненту связности можно вычислить за линейное время от числа ребер. Недостаток такого подхода: отношение «быть (почти) дубликатом» не является транзитивным, поэтому в кластер сходных документов могут попасть абсолютно разные документы. Противоположным — «самым сильным» — определением кластера, исходя из отношения «быть (почти) дубликатом», является его определение через клики графа (максимальные по вложению полные подграфы) коллекции документов. При этом каждый документ из кластера должен быть сходным со всеми другими документами того же кластера. Такое определение кластера более адекватно передает представление о групповом сходстве, но, может встретить трудности при масштабировании системы поиска документов-дубликатов в силу того, что поиск клик в графе — классическая труднорешаемая задача.

Указанные два определения кластера задают спектр промежуточных формулировок, в которых можно находить необходимый баланс между точно-

стью и полнотой определения кластеров, с одной стороны, и сложностью вычисления кластеров с другой стороны.

Другие методы определения кластеров основаны на вариациях стандартных методов кластерного анализа, например, когда при отнесении очередного объекта к кластеру используется расстояние до центров масс кластеров. Такого рода методы существенно зависят от последовательности поступления объектов, образующих кластеры. Это означает, что документы, попавшие в коллекцию раньше, сильнее определяют структуру кластера, чем документы, поступившие позднее.

Недостатки методов кластеризации, основанных на мерах сходства между документами, является частая возможность объединения в один кластер документов лишь попарно сходных друг с другом, но не имеющих общекластерного сходства. Альтернативой такому подходу служат методы, в которых кластер сходных документов определяется как множество документов, у которых число общих элементов описания превышает определенный порог. Такие методы основаны на би-кластеризации [Mirkin et al., 1995] и решетках формальных понятий [Ganter et al., 1999].

На пятом этапе работы системы возможен учет работы с распределенными коллекциями документов. Здесь возможны две противоположные стратегии (задающий спектр промежуточных между ними): рассмотрение дублирования по отношению к каждой коллекции по отдельности и рассмотрение дублирования по представителям кластеров из разных коллекций, например, в работе [Yang et al., 2006] предлагается использовать документ источник в качестве представителя кластера.

Шестой этап работы системы предусматривает принятие решений о дублировании и компиляции. Здесь необходимо создание удобного интерфейса, позволяющего ЛПР просматривать документы, чье сходство было установлено автоматически и принятие окончательного решения о дублировании или плагиате, в том числе возникшего путем компиляции нескольких документов из разных коллекций. Так же на этом этапе определяется документ-источник для каждого кластера.

Реализация поиска дубликатов в системе

Для реализации в Системе отобраны следующие модификации методов:

- ✧ модификация метода I-Match со сравнением сходства по метрике косинусов;
- ✧ модификация метода I-Match со сравнением сходства по метрике TF-IDF;

- ✧ DSC (полнотекстовый поиск);
- ✧ DSC (первый в сперестановках);
- ✧ DSC-SS.

Пользователями настраиваются следующие параметры методов:

- ✧ для I-Match: нижний частотный порог слов α , попадающих в словарь (редкие слова отбрасываем);
- ✧ для I-Match: верхний частотный порог слов β , попадающих в словарь (частые слова отбрасываем);
- ✧ для DSC, DSC-SS – сдвиг (расстояние между шинглами);
- ✧ для DSC, DSC-SS – размер шингла (число слов в одном шингле);
- ✧ для DSC-SS – размер супершингла;
- ✧ пороговое значение, при превышении которого документ становится кандидатом в дубликаты.

Приведем краткую схему алгоритма анализа документа на дублирование перечисленными выше методами (рис. 1).

Далее приведем сценарий, согласно которому пользователь выполняет анализ проектных документов на дублирование в Системе.

Проведение анализа документов в Системе

Система анализирует документы, представленные в форматах TXT, DOC, RTF и PDF. В процессе проверки выполняются следующие действия.

ШАГ 1. Пользователь загружает файлы проектной документации НИОКР в Систему. Документам, впервые поступившим в систему, присваивается статус «Не проверен».

ШАГ 2а. Система выполняет автоматическую проверку всех непроверенных документов с коллекцией документов, уже хранящихся в системе и признанных уникальными. Автоматическая проверка организована в виде назначенного задания ОС Windows и запускается в установленное время (например, ночью).

ШАГ 2б. Пользователь выполняет проверку всех непроверенных документов в ручном режиме.

ШАГ 3. По результатам автоматической проверки документы, получившие значение сходства выше установленного порогового значения, считаются «подозрительными», система присваивает им статус «кандидат в дубликаты». Документы со значением сходства, не превышающим установленный порог, признаются оригинальными, им присваивается статус «уникальный». Для каждого документа,

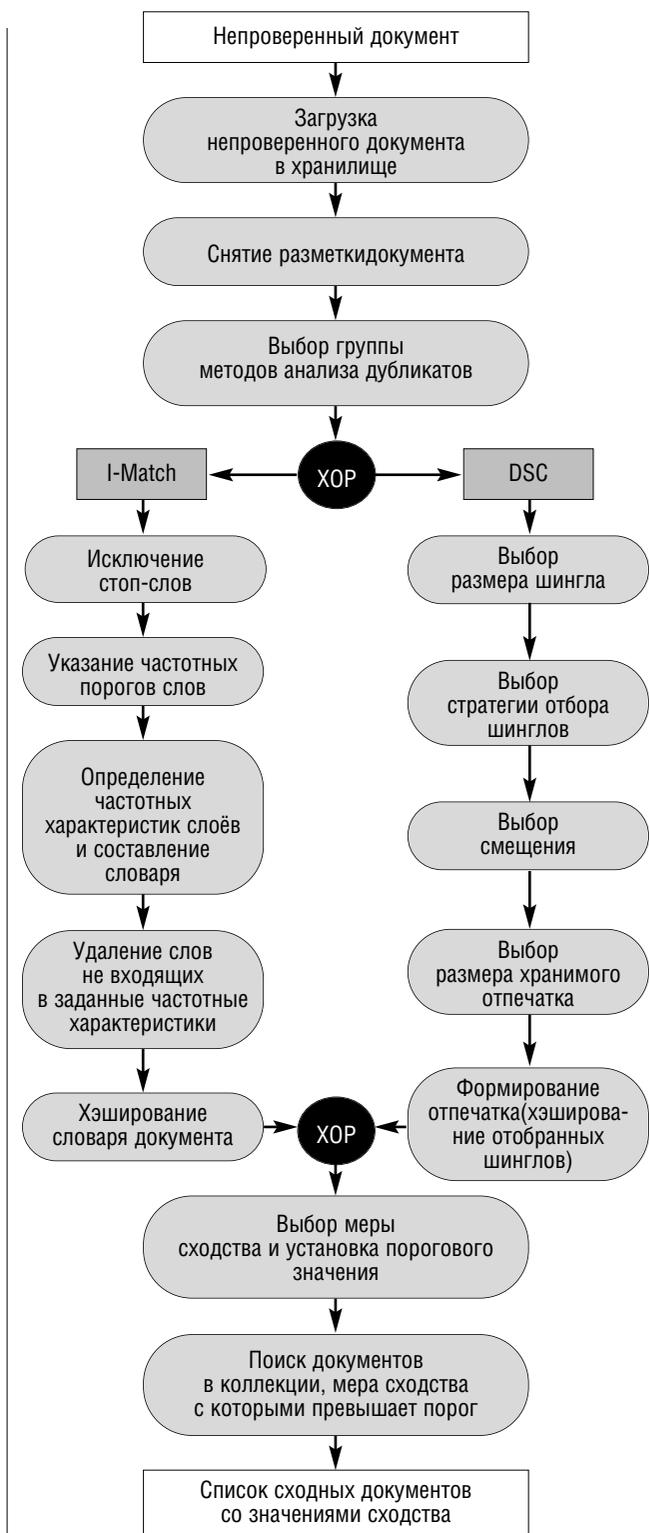


Рис.1 Схема алгоритма анализа документа на дублирование

признанного кандидатом в дубликаты, указывается источник сходства (или несколько источников – документов из уникальной коллекции).

ШАГ 4. Пользователь принимает решение – является ли документ уникальным или дубликатом – после просмотра двух документов с выделенными фрагментами совпадающих текстов. Документам,

признанным аналитиком дубликатами, Система присваивает статус «дубликат».

ШАГ 5. В случае необходимости пользователь может изменить параметры анализа и провести повторный анализ и в зависимости от результатов принять решение о статусе документа.

ШАГ 6. Пользователь может выполнить отчет о проверенных документах за определенный период.

Подбор параметров и тестирование

Авторами была разработана и апробирована методика подбора параметров работы алгоритмов поиска дубликатов. Были выбраны следующие способы модификации документов, см. табл. 1. Суть тестирования заключается в проверке способности методов выявлять (почти) дубликаты полученные из исходных указанными способами. Кроме того авторы произвели подбор наилучших параметров методов для различных типов документов. Для подбора параметров методов, реализованных в Системе, и проведения тестирования авторами была разработана специальная утилита – «Генератор тестов». Программа «Генератор тестов» позволяет автоматизировать создание тестовых изменённых документов и расчёт параметров сходства и показателей работы методов.

Таблица 1

Способы генерации тестовых данных

№	Название метода	Параметры метода
1	Перестановка параграфов	Доля переставляемых параграфов
2	Удаление параграфов	Доля удаляемых параграфов
3	Добавление параграфов	Доля добавляемых параграфов
4	Замена слов	Доля заменяемых слов
5	Добавление повторяющихся абзацев	Количество абзацев и количество повторений каждого
6	Замена букв	Множество пар букв: (<исходная буква>, <новая буква>)

В табл. 1 приведены способы создания тестовых данных на основе исходных документов. Для оценки качества нахождения документов-дубликатов используются стандартные для информационного поиска меры: полнота, точность и *F*-мера. В ходе проведения тестирования выяснилось, что даже при сохранении малой доли (до 10%) исходного документа в тестовой коллекции удается адекватно выявлять такие документы как дубли при малом пороге сходства. Относительное число «ложных дубликатов» в худшем случае оказывается невелико (≤30%).

Таблица 2

Оптимальные параметры тестируемых методов

Метод анализа	Интервал для частотных порогов	Размер шингла, слов	Смещение, слов	Размер супершингла
I-Match(cos)	(0.35, 0.85)	–	–	–
I-Match (TF-IDF)	(0.35, 0.85)	–	–	–
DSC (Fulltext)	–	10	1	–
DSC	–	10	1	–
DSC-SS	–	10	1	5

Для метода I-match с помощью алгоритма оптимизации Хука-Дживса были найдены верхняя и нижняя частотные границы для построения словаря по исходной коллекции из 13 документов. Для методов группы DSC проводился подбор оптимальных значений параметров – размер шингла, размер супершингла, величина сдвига для слов русского языка и размер образа документа (для неполнотекстовых методов размера образа в 100–150 шинглов вполне достаточно).

Следующая проблема, которую приходилось решать – это способ агрегирования значений сходства, полученных всеми реализованными в Системе методами.

Для проведения соответствующих экспериментов в генератор тестовых документов загружались два документа одного из рассматриваемых типов, далее номера файлов документов №0 и №10 соответственно. Документы №0 и №10 являются документами тестовой коллекции. На их основе с использованием способов из табл. 1 были сгенерированы документы, которые затем сравнивались с документами данной коллекции. В нашем случае было порождено 9 документов с номерами №1, ..., №9. Генерация производилась следующим образом: сначала удалением 10%, 20%, ..., 90% абзацев из файла №0, а потом добавлением 10% к 90%, 20% к 80%, 30 к 70%, 40% 60%, ..., 90% к 10% частей файла №10 к оставшейся части файла №0. Все изменения, указанные в табл. 1, производились случайным образом, например, добавление случайных абзацев в случайное место, замена случайных слов и т.д.

При этом истинное значение сходства двух документов вычислялось по мере Жаккара:

$$Sim(A, B) = \frac{|A \cap B|}{|A \cup B|},$$

где *A* и *B* – представление документов в виде цепочек слов, а не в виде множеств.

Далее использовался подход из области машинного обучения, который называется бустингом (boosting). Предполагается, что мы оцениваем значение некоторой величины, в данном случае сходства. При этом мы имеем оценку сходства для каждого из методов I-Match, DSC, DSC-SS и DSC-Fulltext для наблюдений с 1 по 20. Используя парную регрессию со свободным членом как линейный классификатор, мы строим линейную модель для каждого из методов:

$$y = c_{I-match} + \alpha x_{I-match} + \epsilon_{I-match}$$

$$y = c_{DSC} + \beta x_{DSC} + \epsilon_{DSC}$$

$$y = c_{DSC-SS} + \gamma x_{DSC-SS} + \epsilon_{DSC-SS}$$

$$y = c_{DSC-Fulltext} + \delta x_{DSC-Fulltext} + \epsilon_{DSC-Fulltext}$$

где $c_{[название метода]}$ – свободный член регрессии;
 α, β, γ и δ – коэффициенты при значении сходства $x_{[название метода]}$, найденного конкретным методом;
 $\epsilon_{[название метода]}$ – остатки регрессии.

Нами использовались 4 типа сверток (рис. 2), первая из которых представляла собой среднеарифметическое значение сходства. В терминах бустинга необходимо построить так называемый сильный классификатор (свертку) на основе нескольких слабых в предположении, что взвешенные значения сходства, полученные разными методами, компенсируют недостатки каждого из методов в отдельности.

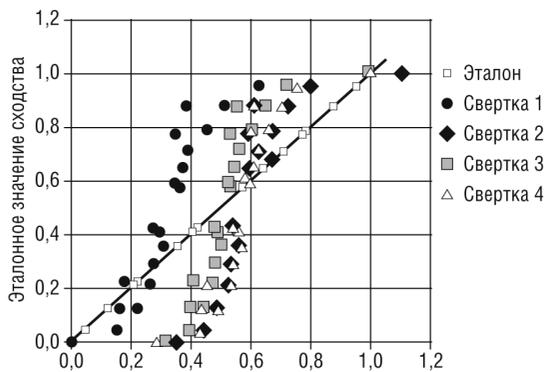


Рис. 2. Графики значений сверток

Наилучший результат показала свертка 4 типа, представляющая собой среднее нормированных регрессий.

$$S(x_{I-match}, x_{DSC}, x_{DSC-SS}, x_{DSC-Fulltext}) = \frac{1}{4} \left(\frac{\alpha x_{I-match} + c_{I-match}}{\alpha + c_{I-match}} + \frac{\beta x_{DSC} + c_{DSC}}{\beta + c_{DSC}} + \frac{\gamma x_{DSC-SS} + c_{DSC-SS}}{\gamma + c_{DSC-SS}} + \frac{\delta x_{DSC-Fulltext} + c_{DSC-Fulltext}}{\delta + c_{DSC-Fulltext}} \right)$$

Для этой свертки наблюдалось наименьшее сходства $S_{min}=0,11$ при эталонном значении $S_{Эт}=0$, и наибольшее значение положительного сходства $S_{max}=0,77$ при $S_{Эт}=0,95$. Применение свертки необходимо для сглаживания эффектов завышения и/или занижения значений сходства, выдаваемые отдельными методами. При этом эксперт обязан установить минимальный порог сходства для кандидатов в дубликаты несколько выше S_{min} для того чтобы уменьшить число «ложных срабатываний».

Направления дальнейшей работы

Важной проблемой для дальнейшего развития продукта является возможность выявления степени компиляции документов, т.е. определения источников из которых получен документ (например, как результат множественного плагиата). Немаловажной проблемой является также учет структуры документа при анализе. Для обработки документов с учетом их структуры должен использоваться специальный конвертер (парсер) документов, который не обязательно входит в состав аналитического модуля системы. Конвертер должен предоставлять аналитику-эксперту возможность самостоятельной настройки фильтра шаблонных фраз. Для представления документа рекомендуется использовать древесную структуру, т.о. документ после обработки хранится в виде дерева разделов. В качестве технологии реализации (представления) рекомендуется использовать XML или SGML. В состав конвертера необходимо включить метод автоматического построения структуры дерева для данного типа документов, с возможностью предварительного задания шаблонов. После построения древесного представления документов их попарное сходство рассчитывается покомпонентно. Корневой узел содержит название документа, промежуточные узлы – названия разделов, листья – содержания разделов нижнего уровня. При построении кластеров сходных документов мы предлагаем использовать подход, описанный нами в [Кузнецов и др., 2005] и [Игнатов и др., 2006], основанный на использовании частых замкнутых множеств признаков (frequent closed itemset mining). При этом в роли объектов выступают элементы описания (шинглы или слова), а в роли признаков – документы. Для такого представления «частыми замкнутыми множествами» являются замкнутые множества документов, для которых число общих единиц описания в образах документов превышает заданный порог. Таким образом, имея набор частых множеств признаков – документов по некоторой коллекции, можно судить о степени сходства конкретного документа с определенной группой документов коллекции. Такой мерой может выступать относительное

число общих шинглов некоторой группы документов и вновь внесенного документа. Еще одним важным вопросом при выявлении дублирования является учет типологии документов, а именно формальных признаков, таких, как:

- ✧ тип документа;
- ✧ научная область;
- ✧ область применения;
- ✧ стандарты (ГОСТ).

Очевидное решение — использование существующих древесных классификаторов и перечней. Такие классификаторы имеют ряд недостатков. Например, невозможно повторение одного и того же раздела на различных уровнях иерархии. Предлагаемый подход опирается на прикладную алгебраическую дисциплину — анализ формальных понятий (АФП) [Ganter et al., 1999]. В рамках АФП мы предлагаем использовать решеточную классификацию. Преимущества решеточного классификатора состоят в том, что в нем снимается проблема множественности наследования, когда один и тот же документ относится к разным типам. Другими словами, при использовании древовидного

представления возможны только вкладывающиеся друг в друга надклассы, а в решетке классы могут пересекаться. Таким образом, документ не обязательно имеет одного родителя. Это свойство обеспечивает гибкость решеточного классификатора.

Благодарности

Авторы статьи выражают благодарность за активное участие в обсуждении математических моделей поиска документов-дубликатов и алгоритмических аспектов разработанной системы ведущему научному сотруднику ВИНТИ РАН Виноградову Д.В. и доценту кафедры анализа данных и искусственного интеллекта ГУ-ВШЭ Объедкову С.А. Авторы выражают большую признательность коллективу разработчиков ООО «Кварта ВК» — Калинкиной Ю.А., Звездиной Е.А., Кузнецову А.С. и особенно его руководителю — Еськину И.Ю. за успешную реализацию программного инструментария. Авторы также благодарят Научный фонд ГУ-ВШЭ, предоставивший грант в рамках проекта «Учитель-ученики» для разработки алгоритмов бикластеризации, необходимых для дальнейшего развития проекта. ■

Литература

- [AntiPlagiat, 2008] <http://www.antiplagiat.ru/> — сайт Интернет-сервиса AntiPlagiat.ru компании ЗАО «Анти-Плагиат».
- [Forecsys, 2008] <http://forecsys.ru/> — сайт компании ЗАО «Форексис», официального разработчика Интернет-сервиса AntiPlagiat.ru.
- [NIST, 1995] NIST, “Secure Hash Standard”, Federal Information Processing Standards Publication 180-1, 1995.
- [Broder, 1997] A. Broder, On the resemblance and containment of documents, in Proc. Compression and Complexity of Sequences (SEQS: Sequences’97).
- [Chowdhury et al., 2002] A. Chowdhury, O. Frieder, D. Grossman, and M. McCabe. Collection statistics for fast Duplicate document detection. In ACM Transactions on Information Systems (TOIS), Volume 20, Issue 2, 2002.
- [Kołcz et al., 2004] A. Kołcz, A. Chowdhury, J. Alspector. Improved Robustness of Signature-Based Near-Replica Detection via Lexicon Randomization. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, page 605–610, Seattle, WA, USA, 2004.
- [Ilyinsky et al., 2002] S. Ilyinsky, M. Kuzmin, A. Melkov, I. Segalovich. An efficient method to detect duplicates of Web documents with the use of inverted index. WWW Conference 2002.
- [Broder et al., 1998] A. Broder, M. Charikar, A.M. Frieze, M. Mitzenmacher, Min-Wise Independent Permutations, in Proc. STOC, 1998.
- [Broder et al., 1997] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. In Proceedings of WWW6’97, pages 391–404. Elsevier Science, April 1997.
- [Hoad et al., 2003] T. Hoad and J. Zobel. Methods for identifying versioned and plagiarized documents. In Journal of the American Society for Information Science and Technology, Vol 54, I 3, 2003.
- [Mirkin et al., 1995] B. Mirkin, P. Arabie, L. Hubert (1995) Additive Two-Mode Clustering: The Error-Variance Approach Revisited, Journal of Classification, 12, 243–263.
- [Ganter et al., 1999] B. Ganter and R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer, 1999.
- [Yang et al., 2006] H. Yang and J. Callan. Near-Duplicate Detection by instance-level constrained clustering. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in information retrieval. Pages 421–428. Seattle, Washington 2006.
- [Кузнецов и др., 2005] Кузнецов С.О., Игнатов Д.И., Объедков С.А., Самохин М.В. Порождение кластеров документов дубликатов: подход, основанный на поиске частых замкнутых множеств признаков. Интернет-математика 2005. Автоматическая обработка веб-данных. Москва: Яндекс, 2005, стр. 302–319.
- [Игнатов и др., 2006] Игнатов Д.И., Кузнецов С.О. О поиске сходства Интернет-документов с помощью частых замкнутых множеств признаков// Труды 10-й национальной конференции по искусственному интеллекту с международным участием (КИИ’06). — М.: Физматлит, 2006, Т.2, стр.249–258.

ПОДХОДЫ К РЕШЕНИЮ ЗАДАЧИ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ

*А. С. Рудаков,
аспирант Московского государственного университета печати
E-mail: shmel_ras@mail.ru*

В статье предлагается методика выбора рационального подхода к решению задачи прогнозирования временных рядов с помощью нейронных сетей между существующим подходом аппроксимации и предлагаемым подходом кластеризации, который только косвенно упоминается в области прогнозирования. Предлагаемая методика позволяет повысить эффективность проектирования и применения нейронных сетей для решения задач прогнозирования временных рядов путем определения на начальном этапе проектирования наиболее подходящей для решения поставленной задачи структуры сети и алгоритма обучения. Для выполнения этой задачи в основу методики положен анализ исходных обучающих данных, известных из условия поставленной задачи. В статье предлагается алгоритм для реализации методики.

Введение

Одной из наиболее распространенных областей применения нейронных сетей в настоящее время является область экономических и финансовых систем [1]. Это обусловлено одним из основных достоинств нейронных сетей — возможностью имитации процессов с большим количеством влияющих параметров, что является очень сложным или невозможным для традиционных методов в подобных условиях, например методов оптимизации. Однако, следствием широкого распространения нейронных сетей стала разработка большого количества разнообразных алгоритмов обучения нейронных сетей и их модификаций. Таким образом, актуальной является задача разработки метода выбора рационального подхода к решению задачи прогнозирования временных рядов с помощью нейронных сетей.

В большинстве случаев для решения задач прогнозирования с помощью нейронных сетей в настоящее время применяют подход аппроксимации функции. При использовании подхода аппроксимации функции, настраиваемые параметры сети

в результате обучения принимают вид, соответствующий некоторой функции, представленной входными и выходными векторами обучающего множества [2]. Данный подход в основном применяется в задачах прогнозирования, в которых каждому конкретному входному вектору, представленному входными параметрами нейронной сети, соответствует конкретное значение прогнозируемого вектора, представленного выходными параметрами нейронной сети:

$$y_i = f(x_i), \quad (1)$$

где x_i — i -ый входной вектор, y_i — соответствующее значение прогнозируемого вектора;
 $f(x_i)$ — прогнозирующая функция.

Подход аппроксимации функции имеет свои недостатки. В случае непериодической зависимости входных и выходных данных не исключена ситуация отрицательного результата обучения, из-за сложной формы аппроксимируемой функции.

Отрицательный результат обучения возможен также в условиях неполных данных, необходимых для успешной аппроксимации функции. Избежать подобных результатов можно, например, путем ввода в систему некоторой допустимой погрешности обучения.

Для решения задач прогнозирования временных рядов с помощью нейронных сетей предлагается также применять еще один подход, который только косвенно упоминается в области прогнозирования, — подход кластеризации. Применение подхода кластеризации предпочтительно в случаях, когда все множество входных векторов можно разбить на отдельные непересекающиеся подмножества — классы, или кластеры, и поставить в соответствие каждому подмножеству отдельный выходной вектор:

$$y^j = \delta(x_v), x_v \in X_j, j \in [1, k], \quad (2)$$

где x_v — v -ый входной вектор;

y^j — выходной вектор;

$\delta(x_v)$ — прогнозирующая функция;

X_j — подмножество обучающих входных векторов — класс, которое соответствует выходному вектору;

k — количество непересекающихся подмножеств входных векторов — классов и, соответственно, количество выходных векторов.

Таким образом, задача прогнозирования временных рядов с применением подхода кластеризации заключается в нахождении класса, к которому в большей степени подходит входной вектор, представленный значениями входных параметров нейронной сети. Подход кластеризации упрощает задачу прогнозирования, так как в нем происходит аппроксимация подмножеств обучающих входных векторов, а не отдельных обучающих входных векторов, что увеличивает скорость обучения, уменьшает время проектирования и повышает эффективность применения нейронной сети. Одним из недостатков подхода кластеризации является конечное число возможных прогнозируемых классов, которое зависит от количества выходных параметров проектируемой нейронной сети. Для увеличения числа возможных прогнозируемых классов можно увеличить размеры проектируемой нейронной сети.

Часто главная роль в постановке задачи прогнозирования отводится человеку: он определяет цель задачи, ее исходные данные и подход к решению. Положительный результат в решении поставленной задачи в большей степени определяется опытом и знаниями разработчика. Однако проектирование

нейронной сети даже опытным специалистом не гарантирует рационального выбора подхода к решению задачи прогнозирования, из-за возможной человеческой ошибки, что подтверждает актуальность разработки метода выбора рационального подхода к решению задачи прогнозирования, позволяющего более обоснованно принять решение по выбору подхода. Выбор рационального подхода влияет на эффективность проектирования и применения нейронной сети для решения задачи прогнозирования временных рядов, так как на начальном этапе определяет наиболее подходящую для поставленной задачи структуру сети и алгоритм обучения.

Структуры и алгоритмы обучения нейронных сетей для реализации подходов к решению задач прогнозирования временных рядов

Каждый из подходов к решению задачи прогнозирования требует соответствующей структуры и алгоритма обучения нейронной сети. В большинстве случаев для решения задачи аппроксимации функции применяют алгоритм обратного распространения ошибки [2, 3, 4].

Алгоритм обратного распространения ошибки представляет собой итеративный градиентный алгоритм обучения, который используется с целью минимизации среднеквадратичного отклонения текущего выхода от желаемого выхода многослойной нейронной сети [2]. Обобщенная структура двухслойной нейронной сети для реализации алгоритма обратного распространения ошибки представлена на рис. 1 [5], где w_{nm} — весовые коэффициенты связей между нейронами, OUT_m — выходные сигналы нейронов выходного слоя, n — количество нейронов в скрытом слое, m — количество нейронов в выходном слое. Входной слой выполняет только распределительные функции.

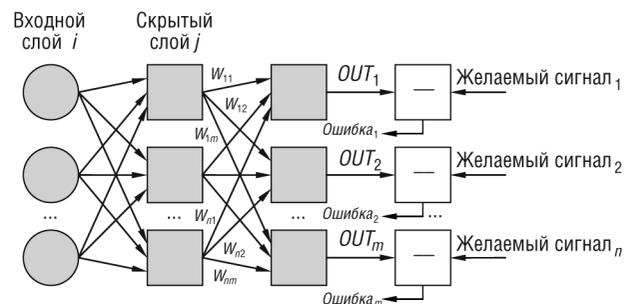


Рис.1. Обобщенная структура двухслойной нейронной сети для реализации алгоритма обратного распространения ошибки

Для подхода кластеризации предлагается применять структуру нейронной сети и алгоритм обучения Кохонена, так как данная структура и алгоритм

применяются в основном для кластеризации входных образов [2, 3, 4]. Алгоритм обучения Кохонена также называют алгоритмом конкурентного обучения, а нейронные сети, использующие такой алгоритм — конкурентными. Конкурентные нейронные сети относятся к самоорганизующимся нейронным сетям. Самоорганизующиеся нейронные сети характеризуются обучением без учителя, в результате которого происходит адаптация сети к решаемой задаче. Структура конкурентной нейронной сети в общем случае представляет собой двухслойную нейронную сеть с прямыми связями. Первый слой выполняет распределительные функции, причем каждый нейрон первого слоя имеет связи со всеми нейронами второго слоя, который является выходным. Второй слой осуществляет конкуренцию между нейронами, в результате которой определяется нейрон-победитель. Победителем в конкуренции считается нейрон, который в результате подачи на вход сети определенного входного вектора имеет максимальную взвешенную активность. Для нейрона-победителя весовые коэффициенты усиливаются, а для остальных нейронов не изменяются или уменьшаются. По мере поступления входных векторов посредством обучения происходит разбиение n -мерного входного пространства на различные области решений, каждой из которых соответствует отдельный нейрон выходного слоя [2]. Обобщенная структура конкурентной нейронной сети представлена на рисунке 2 [2], где w_{nm} — весовые коэффициенты связей между нейронами, OUT_m — выходные сигналы нейронов выходного слоя, n — количество нейронов в распределительном слое, m — количество нейронов в выходном слое.

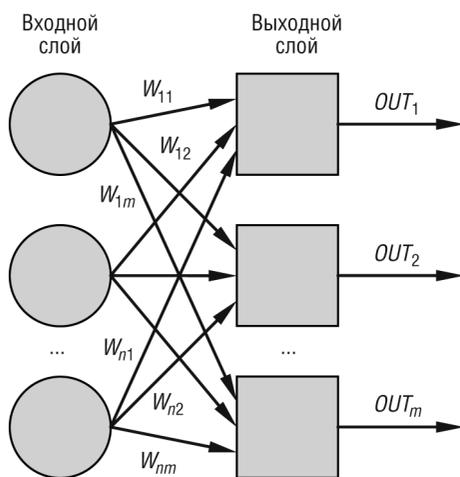


Рис.2. Обобщенная структура конкурентной нейронной сети

Методика выбора рационального подхода к решению задачи прогнозирования с помощью нейронных сетей

В основу предлагаемой методики выбора рационального подхода к решению задачи прогнозирования с помощью нейронных сетей между существующим и предлагаемым подходами положен анализ исходных обучающих данных, известных из условия поставленной задачи.

Пусть γ — вектор, размерность которого равна размерности входного вектора, содержащий информацию о минимально возможном расстоянии между подмножествами обучающих входных векторов — классами, и пусть выполнены следующие условия:

1. Подмножества X_j создают разбиение множества X , для которого верно:

$$X = \{X_1, X_2, \dots, X_k\}, \tag{3}$$

$$X_j \cap X_l = \emptyset, j \neq l, j \in [1, k], l \in [1, k], \tag{4}$$

$$\bigcup_{j=1}^k X_j = X, \tag{5}$$

где X — множество обучающих входных векторов; X_j — j -ое подмножество обучающих входных векторов — класс; k — количество подмножеств обучающих входных векторов — классов.

2. В каждом из подмножеств расстояние между наиболее близкими друг к другу соседними обучающими входными векторами — x_b^j и x_p^j — не превышает γ :

$$|x_b^j - x_p^j| < \gamma, \tag{6}$$

где $j \in [1, k]$, k — количество подмножеств обучающих входных векторов — классов; $b \in [1, z_j]$, $p \in [1, z_j]$, z_j — количество обучающих входных векторов в j -ом подмножестве обучающих входных векторов — классе; x_b^j и x_p^j — наиболее близкие друг к другу соседние обучающие входные вектора j -го подмножества обучающих входных векторов — класса.

3. Расстояние между соседними подмножествами обучающих входных векторов — классов превышает или равно γ :

$$|x_d^j - x_r^h| \geq \gamma, \tag{7}$$

где $j \in [1, k]$, $h \in [1, k]$, $j \neq h$, k — количество подмножеств обучающих входных векторов — классов; x_d^j и x_r^h — соответственно наиболее близкие друг к другу обучающие входные вектора соседних подмножеств обучающих входных векторов — классов; $d \in [1, z_j]$, z_j — количество обучающих входных векторов в j -ом подмножестве обучающих входных векторов — классе; $r \in [1, z_h]$, z_h — количество обучающих входных векторов в h -ом подмножестве обучающих входных векторов — классе.

4. Каждому обучающему входному вектору из j -го подмножества обучающих входных векторов — класса, где $j \in [1, k]$ и k — количество подмножеств обучающих входных векторов — классов, соответствует один выходной вектор — y^j :

$$\alpha_w^j = (x_w^j, y^j), \quad (8)$$

где $w \in [1, z_j]$, z_j — количество обучающих входных векторов в j -ом подмножестве обучающих входных векторов — классе; $j \in [1, k]$, k — количество подмножеств обучающих входных векторов — классов; α_w^j — w -ая обучающая входная выборка j -го подмножества обучающих входных векторов — класса.

Тогда, при одновременном выполнении всех приведенных выше условий, все множество обучающих выборок представляется в виде отдельных подмножеств, каждому из которых соответствует определенный выходной вектор. Следовательно, в этом случае, для решения задачи прогнозирования с помощью нейронных сетей целесообразно применять подход кластеризации.

Таким образом, предлагается следующая методика выбора рационального подхода к решению задачи прогнозирования с помощью нейронных сетей, представляемая в виде последовательности следующих этапов:

1. Проектированием нейронной сети, на основе своих знаний и опыта, задается вектор — вектор, размерность которого равна размерности входного вектора, содержащий информацию о минимально возможном расстоянии между подмножествами обучающих входных векторов — классами.

2. Осуществляется определение подмножеств обучающих входных векторов — классов X_j , создающих разбиение множества обучающих входных векторов X :

2.1. Для рассмотрения выбирается очередной обучающий входной вектор x_m , где $m \in [1, z]$, z — общее количество обучающих входных векторов.

2.2. Для рассмотрения выбирается очередной обучающий входной вектор x_q , где $q \in [1, z]$, z — общее количество обучающих входных векторов, $q \neq m$.

2.3. Проверяется выполнение условия на принадлежность пары обучающих входных векторов x_m и x_q к одному подмножеству обучающих входных векторов — классу:

$$|x_m - x_q| < \gamma. \quad (9)$$

Если условие (9) выполняется, идентификатор подмножества обучающего входного вектора x_m определен, $k_{x_m} \neq 0$, который вводится для каждого обучающего входного вектора и используется для хранения информации о принадлежности обучающего входного вектора к определенному подмножеству, и идентификатор подмножества обучающего входного вектора x_q не определен, $k_{x_q} = 0$, то $k_{x_q} = k_{x_m}$ и осуществляется переход к этапу 2.7.

Иначе, если условие (9) выполняется, $k_{x_m} = 0$ и $k_{x_q} \neq 0$, то $k_{x_m} = k_{x_q}$ и осуществляется переход к этапу 2.7.

Иначе, если условие (9) выполняется, $k_{x_m} = 0$ и $k_{x_q} = 0$, то увеличивается значение счетчика подмножеств k , содержащего информацию о количестве подмножеств обучающих входных векторов, $k_{x_m} = k$, $k_{x_q} = k$ и осуществляется переход к этапу 2.7.

Иначе, если условие (9) выполняется, $k_{x_m} \neq 0$, $k_{x_q} \neq 0$, то уменьшается значение счетчика подмножеств k и осуществляется переход к этапу 2.4.

2.4. Для рассмотрения выбирается очередной обучающий входной вектор x_r , где $r \in [1, z]$, z — общее количество обучающих входных векторов, или выборок.

2.5. Если $k_{x_m} > k_{x_q}$ и $k_{x_i} = k_{x_m}$, то $k_{x_i} = k_{x_q}$.

Иначе, если $k_{x_m} > k_{x_q}$ и $k_{x_i} > k_{x_m}$, то $k_{x_i} = k_{x_i} - 1$.

Иначе, если $k_{x_m} < k_{x_q}$ и $k_{x_i} = k_{x_q}$, то $k_{x_i} = k_{x_m}$.

Иначе, если $k_{x_m} < k_{x_q}$ и $k_{x_i} > k_{x_q}$, то $k_{x_i} = k_{x_i} - 1$.

2.6. Если рассмотрены все обучающие входные вектора x_r , то осуществляется переход к этапу 2.7. Иначе осуществляется переход к этапу 2.4.

2.7. Если рассмотрены все обучающие входные вектора x_q , то осуществляется переход к этапу 2.8. Иначе осуществляется переход к этапу 2.2.

2.8. Если рассмотрены все обучающие входные вектора x_m , то осуществляется переход к этапу 3. Иначе осуществляется переход к этапу 2.1.

3. Осуществляется определение возможности применения обучающих входных выборок подмножеств обучающих входных векторов — классов в подходе кластеризации:

3.1. Для рассмотрения выбирается очередное подмножество обучающих входных векторов — класс X_j с номером j .

3.2. Для рассмотрения выбирается очередная обучающая входная выборка (x, y_i) .

3.3. Если значение идентификатора подмножества выбранного обучающего входного вектора x_i равен значению номера выбранного подмножества j и данное равенство выполнилось впервые для выбранного подмножества обучающих входных векторов — класса, то $y^j = y_i$, где y^j — выходной вектор для j -го подмножества обучающих входных векторов — класса, и устанавливается флаг выбранной обучающей входной выборки (x_i, y_i) , показывающий возможность ее применения в подходе кластеризации, $f_i = 1$, иначе, если значение идентификатора подмножества выбранного обучающего входного вектора x_i равен значению номера выбранного подмножества j и $y_i = y^j$, то $f_i = 1$.

3.4. Если рассмотрены все обучающие входные выборки (x_i, y_i) , то осуществляется переход к этапу 3.5, иначе — к этапу 3.2.

3.5. Если рассмотрены все подмножества обучающих входных векторов X_j , то осуществляется переход к этапу 3.6, иначе — к этапу 3.1.

3.6. Если все обучающие входные выборки (x_i, y_i) имеют возможность применения в подходе кластеризации: $f_i = 1$, то для данного вектора u целесообразно применять подход кластеризации, иначе — подход аппроксимации.

Изменяя значения параметров вектора u , проектировщик нейронной сети имеет возможность получать различное количество подмножеств обучающих входных векторов. Чем меньше вектор u , тем большее количество подмножеств обучающих входных векторов можно получить. При выборе подхода кластеризации следует учитывать количество полученных подмножеств обучающих входных векторов, так как чем больше количество подмножеств, тем большим должно быть число выходных параметров проектируемой нейронной сети.

Алгоритм для реализации методики выбора рационального подхода к решению задачи прогнозирования с помощью нейронных сетей

Для реализации методики выбора рационального подхода к решению задачи прогнозирования с помощью нейронных сетей предлагается следующий алгоритм, который включает в себя следующие шаги:

1. Ввод проектировщиком нейронной сети вектора γ — вектора, размерность которого равна размерности входного вектора, содержащего информацию о минимально возможном расстоянии

между подмножествами обучающих входных векторов — классами.

2. Инициализация переменных:

$$m = 1, q = 1, t = 1, i = 1, j = 1, k = 1, n = 0,$$

где $m \in [1, z]$, $q \in [1, z]$, $t \in [1, z]$, $i \in [1, z]$, z — общее количество обучающих входных выборок;

$j \in [1, k]$, k — счетчик, содержащий информацию о количестве подмножеств обучающих входных векторов — классов;

n — флаг, устанавливаемый при первом нахождении обучающей входной выборки определенного подмножества обучающих входных векторов.

где $i \in [1, z]$, k_{x_i} — идентификатор подмножества обучающего входного вектора, который вводится для каждого обучающего входного вектора и используется для хранения информации о принадлежности обучающего входного вектора определенному подмножеству;

где $i \in [1, z]$, f_i — флаг i -ой обучающей входной выборки, показывающий возможность ее применения в подходе кластеризации.

3. Выбор очередного обучающего входного вектора x_m .

4. Выбор очередного обучающего входного вектора x_q .

5. Проверка условия (9). Если условие (9) выполняется, $k_{x_m} \neq 0$ и $k_{x_q} = 0$, то $k_{x_q} \leftarrow k_{x_m}$ и осуществляется переход к шагу 9.

Иначе, если условие (9) выполняется, $k_{x_m} = 0$ и $k_{x_q} \neq 0$, то $k_{x_m} \leftarrow k_{x_q}$ и осуществляется переход к шагу 9.

Иначе, если условие (9) выполняется, $k_{x_m} = 0$ и $k_{x_q} = 0$, то $k \leftarrow k + 1$, $k_{x_m} \leftarrow k$, $k_{x_q} \leftarrow k$ и осуществляется переход к шагу 9.

Иначе, если условие (9) выполняется, $k_{x_m} \neq 0$, $k_{x_q} \neq 0$, то $k \leftarrow k - 1$ и осуществляется переход к шагу 6.

6. Выбор очередного обучающего входного вектора x_i .

7. Если $k_{x_m} > k_{x_q}$ и $k_{x_i} = k_{x_m}$, то $k_{x_i} \leftarrow k_{x_q}$.

Иначе, если $k_{x_m} > k_{x_q}$ и $k_{x_i} > k_{x_m}$, то $k_{x_i} \leftarrow k_{x_i} - 1$.

Иначе, если $k_{x_m} < k_{x_q}$ и $k_{x_i} = k_{x_q}$, то $k_{x_i} \leftarrow k_{x_m}$.

Иначе, если $k_{x_m} < k_{x_q}$ и $k_{x_i} > k_{x_q}$, то $k_{x_i} \leftarrow k_{x_i} - 1$.

8. Если $t = z$, то осуществляется переход к шагу 9.

Иначе $t \leftarrow t + 1$ и осуществляется переход к шагу 6.

9. Если $q = z$, то осуществляется переход к шагу 10.

Иначе $q \leftarrow q + 1$ и осуществляется переход к шагу 4.

10. Если $m = z$, то осуществляется переход к этапу 11.

Иначе $m \leftarrow m + 1$ и осуществляется переход к шагу 3.

11. Выбор очередной обучающей выборки (x_i, y_i) .

12. Если $k_{x_i} = j$ и $n = 0$, то $n \leftarrow 1$, $y' \leftarrow y_i$ и $f_i \leftarrow 1$.

Иначе $m \leftarrow m + 1$, если $k_{x_i} = j$, $n = 1$, $y_i = y^j$, то $f_i \leftarrow 1$.

13. Если $i = z$, то осуществляется переход к шагу 14.

Иначе $i \leftarrow i + 1$ и осуществляется переход к шагу 11.

14. Если $i = k$, то осуществляется переход к шагу 15.

Иначе $i \leftarrow 1$, $n \leftarrow 0$, $j \leftarrow j + 1$ и осуществляется переход к шагу 11.

15. Если $\forall i, f_i = 1$, где $i \in [1, z]$, z — общее количество обучающих входных выборок, f_i — флаг обучающей входной выборки, показывающий возможность ее применения в подходе кластеризации, то для данного вектора γ целесообразно применять подход кластеризации, иначе — подход аппроксимации.

Алгоритм выбора рационального подхода к решению задачи прогнозирования с помощью нейронных сетей считается завершенным.

Выводы

Предлагаемая в статье методика позволяет осуществить выбор рационального подхода к решению задачи прогнозирования временных рядов с помощью нейронных сетей между существующим подходом аппроксимации и предлагаемым подходом кластеризации, который только косвенно упоминается в области прогнозирования. Выбор рационального подхода позволяет повысить эффективность проектирования и применения нейронных сетей для решения задач прогнозирования временных рядов путем определения на начальном этапе проектирования наиболее подходящей для решения поставленной задачи структуры сети и алгоритма обучения. В основу методики положен анализ исходных обучающих данных, известных из условия поставленной задачи, что позволяет более обоснованно принять решение по выбору подхода к решению задачи прогнозирования временных рядов с помощью нейронных сетей. ■

Литература

1. Данько, Т.П. «Системы искусственного интеллекта в разработке корпоративных маркетинговых стратегий», Журнал «Маркетинг в России и за рубежом», № 5 / Т.П.Данько, М.А.Ходимчук. М.: Финпресс. 2000.
2. Головкин, В.А. «Нейронные сети: обучения, организация и применение» / В.А.Головкин. М.: ИПРЖР. 2001.
3. Круглов, В.В. «Нечеткая логика и искусственные нейронные сети» / В.В.Круглов, М.И.Дли, Р.Ю.Голунов. М.: ФИЗМАТЛИТ. 2001.
4. Крисилов, В.А. «Представление исходных данных в задачах нейросетевого прогнозирования» / В.А.Крисилов, К.В.Чумичкин, А.В.Кондратюк. Одесса: ОНПУ. 2003.
5. Уоссермен, Ф. «Нейрокомпьютерная техника: теория и практика» / Ф.Уоссермен. Пер. с англ. Ю.А.Зуев, В.А.Точенов. 1992.

ISBN 978-5-7598-0461-1
296 с.
60x88/16
Переплёт
2007 г.

Мальцева И.О., Роцин С.Ю.

ГЕНДЕРНАЯ СЕГРЕГАЦИЯ И ТРУДОВАЯ МОБИЛЬНОСТЬ НА РОССИЙСКОМ РЫНКЕ ТРУДА

Второе издание



В книге рассматриваются особенности положения мужчин и женщин на российском рынке труда, исследуется феномен гендерной сегрегации занятости, начиная с советских времён и до настоящего времени, обсуждаются причины и последствия гендерной сегрегации на рынке труда, проводятся международные сопоставления. На основе данных российской статистики и Российского мониторинга экономического положения и здоровья населения проводятся эмпирические оценки интенсивности и направлений трудовой мобильности по гендерным группам, влияние мобильности на гендерные различия в заработной плате и сегрегацию. Отдельно исследуются вертикальная сегрегация и проблема «стеклянного» потолка в российской экономике.

Для экономистов, социологов, специалистов в области занятости и управления человеческими ресурсами.

Уважаемые коллеги!

Приглашаем Вас принять участие в международной конференции

ICCS'09 CONCEPTUAL STRUCTURES: LEVERAGING SEMANTIC TECHNOLOGIES

26-31 июля 2009 г.,

**Москва, Государственный Университет
Высшая Школа Экономики (ГУ-ВШЭ)**

<http://iccs09.hse.ru>

Труды конференции будут опубликованы в серии Lecture Notes in Artificial Intelligence, Springer к началу конференции

Срок подачи текстов докладов – до 12 января 2009 г.

Язык конференции – английский

Участникам конференции – гражданам Союзного Государства будет предоставлена значительная скидка в уплате конференционного взноса.

Следите за информацией на сайте конференции

С.О. Кузнецов, председатель оргкомитета ICCS'09

ICCS'09 CONCEPTUAL STRUCTURES: LEVERAGING SEMANTIC TECHNOLOGIES

26–31 July 2009

Moscow, Russia

State University Higher School of Economics (SU HSE)

<http://iccs09.hse.ru>

The 17th International Conference on Conceptual Structures (ICCS 2009) is the latest in a series of annual conferences that have been held in Europe, Australia, and North America since 1993. The focus of the conference has been the representation and analysis of conceptual knowledge for research and practical application. ICCS brings together researchers and practioners in information and

computer sciences as well as social science to explore novel ways that conceptual structures can be deployed.

Arising from the research on knowledge representation and reasoning with Conceptual Graphs, over the years ICCS has broadened its scope to include innovations from a wider range of theories and related practices, among them other form of graph-based reasoning systems like RDF or

Existential Graphs, Formal Concept Analysis, Semantic Web Technologies, Ontologies, Concept Mapping and more. Accordingly, ICCS represents a family of approaches related to conceptual structures that build on the successes with techniques derived from artificial intelligence, knowledge representation and reasoning, applied mathematics and lattice theory, computational linguistics, conceptual modeling and design, diagrammatic reasoning and logic, intelligent systems and knowledge management. The activity of the field is witnessed by two recently published books («Conceptual Structures in Practice», ed. by Hitzler and Schärfe and «Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs» by Chein and Mugnier) as well as by an ISO standard («Common Logic», ISO/IEC 24707) which originated in this community.

The 2009 ICCS's theme «leveraging semantic technologies» hints to the large overlap of the research fields of semantic technologies and conceptual structures, and emphasizes the goal of closer connecting these two areas in order to obtain a mutual benefit.

Papers for ICCS 2009 are invited on, but not restricted to, the following topics:

- ◆ conceptual structures - theory and applications
- ◆ semantic technologies and conceptual structures [FD6]
- ◆ the interplay of conceptual structures with language, semantics, semiotics and pragmatics
- ◆ conceptual data processing, analysis and conceptual logic
- ◆ modeling, representation and visualization of concepts
- ◆ conceptual knowledge acquisition
- ◆ knowledge representation and reasoning with conceptual structures

- ◆ applied conceptual structures including data mining and knowledge discovery
- ◆ theory and applications of formal ontologies
- ◆ graph-based knowledge representation and reasoning

Authors are invited to submit papers describing both theoretical and practical research outcomes. Papers accepted or under review by other conferences or journals are not acceptable as submissions to ICCS. The language of the conference will be English.

SUBMISSION DETAILS

Papers are limited to 14 pages in Springer's LNCS format. We recommend the use of LaTeX2e for the final version. For more details see <http://www.springer.de/comp/lncs>. Position papers (up to 4 pages) are also welcome.

DEADLINES

◇ Monday January 5, 2009: a one page abstract submitted via conference website (Abstracts should clearly state the purpose, results and conclusions of the work to be described in the final paper.)

◇ Monday January 12, 2009: full paper (not more than 14 pages) in PDF format submitted via conference website

Final acceptance will be based on the full-length paper, which if accepted, must be presented at the conference. Papers accepted for publication will appear in the Lecture Notes in Artificial Intelligence Series by Springer-Verlag (see <http://www.springeronline.com/lncs>). A precondition for publication is that the final version is in full compliance with Springer's format.

CONFERENCE CHAIRS

General Chair :

Sergei Kuznetsov (SU Higher School of Economics, Moscow, Russia)

Program Chairs:

Frithjof Dau (SAP Research CEC Dresden, Germany)

Sebastian Rudolph (Institute AIFB, University of Karlsruhe, Germany)

МОДЕЛИРОВАНИЕ И ПРОГНОЗИРОВАНИЕ БИЗНЕС- ПРОЦЕССОВ С ПОМОЩЬЮ АЛГОРИТМОВ САМООРГАНИЗАЦИИ ФОРМАЛЬНЫХ ОПИСАНИЙ

В.В. Белов,

д.т.н., профессор кафедры вычислительной и прикладной математики Рязанского государственного радиотехнического университета,
e-mail vvbeloff@yandex.ru

В.И. Чистякова,

к.т.н., доцент кафедры вычислительной и прикладной математики Рязанского государственного радиотехнического университета,
e-mail compvv@mail.ryazan.ru

Предлагаются направления развития метода группового учёта аргументов, разрабатываются новые алгоритмы, отличающиеся применением процедуры оптимизации частных полиномов в последовательных рядах приближений. Разработанные алгоритмы могут применяться для моделирования и прогнозирования показателей производственно-экономической деятельности предприятий. Приводятся примеры описаний, полученных в процессе разработки прогноза социально-экономического статуса тепловой электростанции.

Введение

Предметом рассмотрения являются производственно-экономические процессы, представляющие собой изменение во времени показателей производственно-экономической деятельности предприятия. Стимулом к рассмотрению явились исследование текущей ситуации и прогноз социально-экономического статуса крупной тепловой электростанции, выполненные в рамках хоздоговорной НИР.

В настоящей статье приводятся результаты разработки формализованных описаний, позволяющих прогнозировать значения проблемных показателей и моделировать зависимости результативных показателей от факторных. Проблемными названы показатели, представляющие интерес для исследователя, результативными – те показатели, непосредственное управление которыми практически невозможно, факторными – показатели, на значения которых

возможны прямые или косвенные управляющие воздействия.

Предлагаемые алгоритмы осуществляют, по сути дела, поиск наилучшей в некотором смысле нелинейной множественной регрессии. Они могут применяться во всех случаях, когда возникает необходимость синтеза модели, описывающей зависимость одних количественных дискретных величин от других. Наибольший эффект достигается при достаточно большом количестве потенциальных аргументов синтезируемого формального описания и отсутствии сколь-нибудь надёжных предположений о статистических свойствах изучаемых процессов. Естественно, существуют и значительные ограничения:

- 1) объём статистического материала должен быть адекватным размерности решаемой задачи;
- 2) зависимость результативных показателей от факторных должна быть близка к детерминированной.

Указанные ограничения весьма нечётки, проверить их выполнение практически невозможно. Выручает следующее: в процессе работы алгоритмов качество построенных прогнозирующих описаний тестируется самым надёжным способом – тестовой последовательностью.

В качестве базовой нелинейной модели используется полином Колмогорова – Габора (дискретный вариант функционального ряда Вольтерра)

$$\hat{y} = a_0 + \sum_{i=1}^p a_i x_i + \sum_{i=1}^p \sum_{j=1}^i a_{ij} x_i x_j + \sum_{i=1}^p \sum_{j=1}^p \sum_{k=j}^p a_{ijk} x_i x_j x_k + \dots$$

Конкретная структура полинома не задаётся – она является предметом поиска в процессе самоорганизации модели и, в частности, определяет состав эффективных аргументов описания, которые автоматически выбираются из множества учитываемых факторных показателей. Способность алгоритма выделять эффективные аргументы, – оказывающие в своей совокупности наиболее существенное влияние на результативный показатель, – называется его селективирующей способностью. Она, конечно же, легко проверяется на искусственных, специальным образом синтезированных данных.

Постановка задачи

Формально рассматриваемая задача сводится к задаче построения аппроксимации (модели) зависимости скалярной функции от векторного аргумента со следующим требованием: получаемая аппроксимация должна обеспечивать приемлемые значения погрешности экстраполяции, то есть должна обладать достаточно существенной прогностической силой. Функция представляет собой некоторый результативный показатель, а элементы векторного аргумента являются факторными показателями. Исходными данными являются одновременные наблюдения скалярной функции и векторного аргумента $\vec{x} = (x_1, x_2, \dots, x_p)$:

$$(y_i, \vec{x}_i), i = 1, 2, \dots, m; \quad \vec{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip}).$$

Известны многочисленные подходы и методы решения указанной формальной задачи. Наиболее часто используются регрессионный анализ [1], алгоритмы поиска «наилучшей» регрессии [2] и искусственные нейронные сети [3]. В излагаемых исследованиях в качестве классической платформы использован метод эвристической самоорганизации [4] в конкретной его форме, называемой методом группового учёта аргументов [5]. Обоснование целесообразности такого выбора можно найти, например, в [6] и [7].

Разработка прогноза осуществляется методом сценарных условий: прогнозные значения факторных показателей находятся по экспертным оценкам и/или методом выявления и экстраполяции тенденций; будущие значения результативного признака вычисляются путём подстановки найденных значений в полученную ранее модель. В ряде приложений хорошо срабатывает метод лагированных (задержанных) переменных: в качестве аргументов модели используются не текущие, а предыдущие значения факторных показателей. При этом глубина прогноза равна величине задержки аргументов. Указанный приём применим в тех случаях, когда имеет место определенная инерция в изменении результативного показателя под влиянием изменений факторных показателей.

Новые алгоритмы эвристической самоорганизации, отличающиеся структурой интегрирующего ядра

Анализ известных методов построения математических моделей по результатам эксперимента показал, что метод группового учёта аргументов (МГУА) остается наиболее эффективным средством создания математического описания процессов. Однако классические варианты МГУА базируются на фиксированных структурах опорных полиномов. Обычно это квадратичные полиномы. При переходе на новый ряд в качестве аргументов используются лучшие приближения предыдущего ряда. Нетрудно убедиться, что такой способ организации селекции изначально имеет методическое ограничение точности восстановления функций. Действительно, пусть неизвестная функция $y = f(x)$ представляет собой просто x^3 . Если организовать восстановление этой функции по совокупности дискретных отсчетов (y_i, x_i) , $i = 1, 2, \dots, m$ в соответствии с классическим МГУА, то первое приближение будет иметь вид

$$y^{(1)} = a_1 x^2 + b_1 x + c_1,$$

а второе

$$\begin{aligned} y^{(2)} &= a_2 (y^{(1)})^2 + b_2 y^{(1)} + c_2 = \\ &= a_2 a_1^2 x^4 + 2a_2 a_1 b_1 x^3 + (a_2 b_1^2 + 2a_2 a_1 c_1 + b_2 a_1) x^2 + \\ &+ (2a_2 b_1 c_1 + b_2 b_1) x + a_2 c_1^2 + b_2 c_1 + c_2. \end{aligned}$$

Очевидно следующее:

1) желательно, чтобы выполнялись соотношения:

$$2) 2a_2 a_1 b_1 = 1;$$

$$\begin{aligned} a_2 a_1^2 &= a_2 b_1^2 + 2a_2 a_1 c_1 + b_2 a_1 = 2a_2 b_1 c_1 + b_2 b_1 = \\ &= a_2 c_1^2 + b_2 c_1 + c_2 = 0; \end{aligned}$$

3) указанные соотношения несовместны;

4) дальнейшие итерации в построении модели не приведут к улучшению точности приближения.

Таким образом, даже простейшая функция не восстанавливается абсолютно точно классическим вариантом МГУА. Причиной этому является фиксированность структуры опорных полиномов, которая обуславливает следующий факт: в процессе самоорганизации описания принципиально отсутствует возможность синтеза произвольного полинома Колмогорова – Габора, в процессе эволюционного отбора порождаются и участвуют в конкуренции только некоторые из всех возможных полиномов.

Для устранения указанного недостатка и существенного улучшения свойств восстановления функций предлагается модифицировать метод группового учета аргументов следующим образом. Для обеспечения возможности быстрого восстановления зависимостей от одного аргумента необходимо ввести дополнительный этап формирования приближений от одиночных аргументов. Для обеспечения возможности точного восстановления простейших зависимостей от одиночных и парных аргументов, а также для повышения точности приближения функции необходимо осуществлять подбор оптимальной степени полинома частного приближения, а не использовать «опорный» (например, квадратичный или иной) полином.

Тестирование классического МГУА путем решения контрольных задач с искусственно формируемыми исходными данными показывает, что его селективные способности не достаточно высоки: в некоторых примерах аргументы, не входящие в формулу, определяющую процесс, оказывались в списке аргументов модели процесса.

Для варианта алгоритма МГУА, описанного ниже, таких явлений на тех же тестовых данных не наблюдается, что свидетельствует о его более высоких селективных способностях. Кроме того, при обработке больших объемов исходных данных заметно различие во времени работы программы: классический вариант алгоритма обрабатывает одинаковые данные почти вдвое дольше предлагаемого варианта.

В общем случае различные варианты алгоритмов, реализующих рассматриваемый метод построения модели, могут отличаться следующими элементами:

✧ структурой интегрирующего ядра; при этом количество аргументов в модифицированном векторе факторов может быть произвольным; на начальном шаге алгоритма может исполь-

зоваться и один аргумент, алгоритм при этом становится более унифицированным; степень первичного полинома также может быть произвольной; кроме того, в качестве частичных приближений могут использоваться не только полиномы, но и другие функции;

✧ видом коэффициентов первичного полинома и аргументов процесса – они могут быть либо скалярными, либо интервальными величинами;

✧ методом вычисления значений коэффициентов первичного полинома – это может быть метод наименьших квадратов или линейное программирование (в зависимости от объема исходных данных и вида аргументов);

✧ способом деления выборки на рабочую и контрольную группы; контрольная группа может предшествовать рабочей, следовать за ней, или данные групп могут чередоваться.

Критерий выбора наилучшего приближения на каждом шаге алгоритма определяется методом вычисления коэффициентов первичного полинома. В методе наименьших квадратов – это сумма квадратов отклонений расчетных значений от экспериментальных. В задаче линейного программирования с интервальной линейной регрессией – это сумма длин интервалов и отклонения средних значений от экспериментальных. При использовании интервальных величин каждый элемент вектора коэффициентов A представляет собой интервал $A_i = (a_i, \alpha_i)$, где a_i – среднее значение интервала, α_i – половина длины интервала.

В тех случаях, когда предполагается использование синтезируемой модели для решения задачи прогнозирования, оценку точности или соответствующей погрешности формального описания целесообразно осуществлять по тестовой, или проверочной последовательности, – части исходных данных, не использованных в процессе оценки параметров.

Мы будем использовать запись $y^{*(l)} = \text{best}(y^{*(l,j)})$ для отражения факта выбора наилучшего приближения на конкретном этапе серии последовательных приближений, которая эквивалентна записи $y^{*(l)} = y^{*(l,k)}$, где $k = \arg(\min Q^{(l,j)})$; $Q^{(l,j)}$ – критерий, описывающий погрешность приближения $y^{*(l,k)}$ к y .

В общем случае для определения наилучшего приближения возможно использование человеческого фактора в интерактивном режиме, при этом процесс выбора наилучшего приближения не может быть строго формализован.

**Алгоритм автопостроения модели
без использования селекции
в последовательных рядах приближения**

На начальном (нулевом) этапе построения приближения каждый из аргументов функции используется для формирования частного аппроксимирующего полинома. Причем степень полинома постепенно увеличивается до тех пор, пока показатель погрешности приближения проверочной последовательности не перестанет уменьшаться. Все полученные таким образом частные приближения включаются в список аргументов. Расширенный набор аргументов используется для реализации следующего шага построения модели.

На этапе №1 алгоритма формируются полиномиальные приближения для всех возможных пар аргументов. При этом степень полинома постепенно увеличивается по той же схеме, что и на этапе №0. Все полученные новые частные приближения также включаются в список аргументов последующего этапа.

На этапе №*r* алгоритма (*r* = 2, 3, ...) формируются полиномиальные приближения для следующих пар аргументов – первый элемент пары берется из списка аргументов этапа №*r*-2, а второй – из списка результатов (приближений) этапа №*r*-1; после исчерпания пар указанного типа формируются все возможные пары из списка результатов этапа №*r*-1. Степень полинома подбирается по схеме этапа №0. Все полученные новые частные приближения включаются в список аргументов последующего этапа.

На каждом этапе алгоритма оценивается показатель погрешности приближения ϵ_r , равный наименьшей погрешности среди погрешностей всех частных приближений этапа:

$$\epsilon_r = \min_{v_k} (\epsilon_r^{(k)})$$

Последовательные этапы выполняются до тех пор, пока наименьшая ошибка приближения тестовой последовательности не прекратит уменьшаться, т.е. $\epsilon_r \leq \epsilon_{r+1}$. Конечным результатом алгоритма является лучшее приближение предпоследнего этапа.

Формирование набора аргументов описанного алгоритма схематично можно изобразить в следующем виде:

$$\begin{aligned} \text{этап №0} & \quad \vec{x} \rightarrow \vec{P}^{(0)} \\ \text{этап №1} & \quad \vec{P}^{(0)*} \vec{P}^{(0)} \rightarrow \vec{P}^{(1)} \\ \text{этап №2} & \quad (\vec{P}^{(0)*} \vec{P}^{(1)}) \& (\vec{P}^{(1)*} \vec{P}^{(1)}) \rightarrow \vec{P}^{(2)} \\ \dots & \\ \text{этап №}r & \quad (\vec{P}^{(r-2)*} \vec{P}^{(r-1)}) \& (\vec{P}^{(r-1)*} \vec{P}^{(r-1)}) \rightarrow \vec{P}^{(r)} \\ \text{финальная модель: } & \hat{y} = \text{best}_{\epsilon_r}(\vec{P}^{(r)}), \end{aligned}$$

где \vec{x} – начальный вектор аргументов синтезируемой модели;

$\vec{P}^{(i)}$, – вектор частных приближений, порождаемый на *i*-м шаге построения описания, являющийся вектором аргументов *i*+1-го шага;

$\text{best}_{\epsilon_r}(\vec{P}^{(r)})$ – наилучшее по ошибке восстановления тестовой последовательности

из частных приближений, полученных на последнем шаге работы алгоритма;

символ \rightarrow означает отображение вектора аргументов в вектор частных приближений функции;

$v_1 * v_2$ – операция прямого умножения векторов, результатом которой является упорядоченное множество упорядоченных пар, первый элемент которых принадлежит вектору v_1 , а второй – вектору v_2 ;

$A \& B$ – операция объединения упорядоченных множеств (множество *A* дополняется справа множеством *B*).

Алгоритм автопостроения модели с использованием механизмов эвристической селекции в последовательных рядах приближения

Как показывают результаты практических расчетов, рассмотренный в предыдущем подпункте алгоритм автопостроения модели обеспечивает достаточно точное восстановление законов и приемлемую для многих приложений аппроксимацию процессов. Однако он имеет следующие существенные недостатки. Во-первых, он требует большого объема памяти для хранения нарастающего объема частных приближений, используемых в качестве аргументов последующих шагов алгоритма. Во-вторых, он не отсеивает несущественные факторы. Случайные (за счет шумов в исходных данных) проявления их влияния на изучаемый процесс могут негативно влиять на последующие приближения, понижая точность аппроксимации. Избавиться от указанных недостатков можно с помощью механизмов отбрасывания плохих промежуточных приближений [8].

На *i*-м этапе построения модели в качестве аргументов используются результаты всех предыдущих этапов алгоритма, однако не все, а только лучшие из них, – удовлетворяющие некоторому селективирующему требованию. В частности, можно использовать следующее требование: в состав аргументов *i*-го этапа включаются частные приближения предыдущих этапов, погрешность которых не превышает удвоенной погрешности (*i*-1)-го этапа.

Операцию выбора лучших приближений обозначим как $\text{best}_{\epsilon}(\vec{P})$,

где \vec{P} – вектор приближений;

ε – отбраковывающее значение погрешности приближения.

Как в алгоритме без селекции последовательные этапы выполняются до тех пор, пока наименьшая ошибка приближения тестовой последовательности не прекратит уменьшаться, т.е. $\varepsilon_r \leq \varepsilon_{r+1}$.

С учетом изложенного схему формирования набора аргументов улучшенного алгоритма можно изобразить в следующем виде:

$$\text{этап № 0: } \vec{x} \rightarrow \vec{P}_{k_0}^{(0)};$$

$$\text{этап № 1: } \vec{P}^{(0)} * \vec{P}^{(0)} \rightarrow \vec{P}_{k_1}^{(1)};$$

$$\text{этап № 2: } \vec{P}_{sel}^{(0)} = \text{best}_{2\varepsilon_1}(\vec{P}_{k_0}^{(0)}); \vec{P}_{sel}^{(1)} = \text{best}_{2\varepsilon_1}(\vec{P}_{k_1}^{(1)});$$

$$(\vec{P}_{sel}^{(0)} * \vec{P}_{sel}^{(1)}) \& (\vec{P}_{sel}^{(1)} * \vec{P}_{sel}^{(1)}) \rightarrow \vec{P}_{k_2}^{(2)};$$

...

$$\text{этап № } r : \vec{P}_{sel}^{(0)} = \text{best}_{2\varepsilon_{r-1}}(\vec{P}_{k_0}^{(0)}); \dots \vec{P}_{sel}^{(r-2)} = \text{best}_{2\varepsilon_{r-1}}(\vec{P}_{k_{r-2}}^{(r-2)});$$

$$\vec{P}_{sel}^{(r-1)} = \text{best}_{2\varepsilon_{r-1}}(\vec{P}_{k_{r-1}}^{(r-1)});$$

$$(\vec{P}_{sel}^{(0)} * \vec{P}_{sel}^{(r-1)}) \& (\vec{P}_{sel}^{(1)} * \vec{P}_{sel}^{(r-1)}) \& \dots$$

$$\dots (\vec{P}_{sel}^{(r-2)} * \vec{P}_{sel}^{(r-1)}) \& (\vec{P}_{sel}^{(r-1)} * \vec{P}_{sel}^{(r-1)}) \rightarrow \vec{P}_{k_r}^{(r)};$$

$$\text{финальная модель: } \hat{y} = \text{best}_{\varepsilon_r}(\vec{P}_{k_r}^{(r)}),$$

где $\vec{P}_{k_i}^{(i)}$, $i = \overline{0;r}$ – вектор частных приближений, порождаемый на i -м шаге построения описания, состоящий из элементов;

$\vec{P}_{sel}^{(i)}$, $i = \overline{0;r-1}$ – вектор лучших по ошибке восстановления тестовой последовательности частных приближений, полученных на i -м шаге, – результат выполнения процедуры селекции.

Заметим, что с ростом номера этапа i количество аргументов предыдущих этапов, вовлекаемых в построение модели, имеет тенденцию к уменьшению, поскольку погрешность этапов ε_i уменьшается и селектирующее требование становится более жёстким.

В качестве иллюстрации построения формального описания по предложенному модифицированному алгоритму МГУА ниже приведены модели зависимостей *нагрузки на собственные нужды и условно-постоянных затрат* от технико-экономических показателей, описывающих условия производственно-экономической деятельности тепловой электростанции.

Представление формального описания в виде псевдоформул

Результат поиска формального описания методом самоорганизации с использованием полиномов в качестве интегрирующего ядра представляет

собой некоторый алгоритм, позволяющий вычислять модельные значения результирующего показателя для конкретных значений факторных показателей. Естественно, это формальное описание может быть представлено и в виде конкретного полинома Колмогорова – Габора. Конкретика полинома определяется набором членов, значениями коэффициентов и составом аргументов. Однако, получение классического полиномиального представления найденной модели особого практического смысла не имеет по следующим причинам:

1) необходимы не совсем тривиальные символьные преобразования;

2) переход от алгоритмического описания к полиномиальному сопряжён с дополнительными ошибками округления в значениях коэффициентов;

3) сам получаемый полином Колмогорова – Габора выполняет чисто декоративную роль, поскольку его применение для вычислений нерационально из-за указанных выше дополнительных ошибок округления;

4) даже традиционные коэффициенты эластичности могут быть более точно вычислены по алгоритмической модели как отношение приращения моделируемой величины к приращению одного из аргументов в окрестности интересующего набора значений факторных показателей.

Для получения общего аналитического представления о финальной алгоритмической модели более целесообразным является использование предлагаемых ниже двух вариантов псевдоформул. Псевдоформула состоит из композиционной формы записи полинома и двух ассоциированных таблиц значений коэффициентов.

Первый вариант композиционной формы представляет собой символическое обозначение полинома вида $P_n(u, v)$, аргументами которого являются другие символические обозначения полиномов того же вида или символическое обозначение $P_n(x_i)$ полинома с одним аргументом – результирующим показателем x_i . Полиномы с одним аргументом являются терминальными элементами композиции. Они вычисляются все одновременно на нулевом этапе алгоритма. Очередности вычисления остальных полиномов явно не указываются и определяются косвенно по расположению скобок с аргументами.

Второй вариант композиционной формы содержит символические обозначения только терминальных полиномов вида $P_n(x_i)$. Вместо полиномов с двумя аргументами используется символ конструктора \bullet_n^k , в котором n – это порядок полинома, а k – порядок (очередь) срабатывания конструктора.

Очерёдности вычисления можно рассматривать как номера полиномов в композиции и как номера пар их аргументов. Формально конструктор определяется так:

$$u \cdot_k^n v = \sum_{i=0}^n \sum_{j=0}^{n-i} a_{(2n-i+3)i/2+j} u^i v^j,$$

где u, v – аргументы полинома.

Заметим, что номер полинома не влияет на формулу его вычисления.

Индекс коэффициента a определяется выбранным способом нумерации пар индексов i и j : пары при их естественном рассмотрении (j изменяется быстрее i), нумеруются последовательно друг за другом, начиная с нуля. При этом, поскольку верхнее значение j является переменным и равно $n-i$, пара (i, j) получает линейный номер, вычисляемый по формуле:

$$index(i, j) = \begin{cases} j, & \text{если } i = 0; \\ \sum_{l=1}^i (n-l+2) + j, & \text{если } i \geq 1. \end{cases}$$

Далее после применения формулы членов арифметической прогрессии получаем:

$$index(i, j) = (2n - i + 3)i / 2 + j.$$

Нетрудно убедиться в том, что указанный индекс последовательно пробегает значения от 0 до $N = (n+3)n/2$, где N – общее число всех возможных слагаемых полинома n -й степени с двумя аргументами.

Композиционная форма записи полинома наглядно представляет финальный набор эффективных аргументов и очерёдности вычисления полиномов с двумя аргументами. Необходимая числовая информация (значения коэффициентов и погрешности восстановления тестовой последовательности) размещается в ассоциированных таблицах. Образуемая композиционной формой и таблицами псевдоформула полностью описывает алгоритмическую модель, полученную в процессе самоорганизации, и легко воспроизводится программно.

Модель зависимости мощности, необходимой для собственных нужд, от технико-экономических показателей ГРЭС

Часть электроэнергии, производимой на электростанции, используется самой электростанцией для подачи питательной и циркуляционной воды, работы мельниц, дымососов, дутьевых вентиляторов и т.д. Расход энергии на эти потребности электростанции, называемые собственными нуждами, составляет 3–10 % производимой электроэнергии.

Параметры, определяющие изменение расхода мощности на собственные нужды ($N_{\text{собст}}$):

- ✧ $N_{\text{макс1}}, N_{\text{макс2}}, N_{\text{макс}}, N_{\text{ср1}}, N_{\text{ср2}}, N_{\text{ср3}}, N_{\text{ср4}}, N_{\text{ср5}}, N_{\text{ср6}}$ – электрические характеристики вспомогательного оборудования; с увеличением нагрузки энергоблоков увеличивают потребляемую мощность, при этом наиболее существенно увеличение потребляемой мощности при максимуме нагрузки;
- ✧ $N_{\text{соб1}}, N_{\text{соб2}}, N_{\text{соб3}}, N_{\text{соб4}}, N_{\text{соб5}}, N_{\text{соб6}}$ – составляющие мощности на собственные нужды станции, отражают потребность в мощности на собственные нужды каждого энергоблока в отдельности;
- ✧ W_t – затраты электроэнергии оборудования, участвующего только в процессе производства теплоты;
- ✧ $W_{\text{хн}}, W_{\text{пр}}$ – затраты электроэнергии на хозяйственные нужды и собственных потребителей;
- ✧ $W_{\text{пот}}, W_{\text{небал}}$ – потери при передаче и учете энергии;
- ✧ $T_{\text{раб}}$ – параметр нормативного метода учета электроэнергии;
- ✧ $B_{\text{пуск1}}, K_{\text{пуск1}}, B_{\text{пуск2}}, K_{\text{пуск2}}$ – увеличение затрат мощности на пуски энергоблоков по разному типу оборудования;
- ✧ $t_{\text{взд}}$ – изменение потребления мощности как от температуры наружного воздуха в котло-турбинном цехе, так и от периода года.

Исходные данные для анализа представляют собой статистические материалы по Рязанской ГРЭС. Результаты предварительного корреляционного анализа позволяют сделать следующие выводы:

- ✧ не все факторные признаки коррелированы с результативным признаком;
- ✧ коррелированность $N_{\text{собст}}$ с $N_{\text{макс}}$ и $N_{\text{макс1}}$ подтверждает сильную зависимость затрат на собственные нужды станции от пиковых нагрузок; некоррелированность $N_{\text{собст}}$ с $N_{\text{макс2}}$ связана с тем, что блоки второй очереди экономичней блоков первой очереди и работают на оптимальных режимах;
- ✧ коррелированность $N_{\text{собст}}$ с W_t и $t_{\text{взд}}$ отражает тот факт, что потребление мощности на собственные нужды носит сезонный характер;
- ✧ коррелированность $N_{\text{собст}}$ с $T_{\text{раб}}$ указывает на преобладание в фактическом учете электроэнергии нормативного метода, при котором отнесение затрат электроэнергии на объект в связи с отсутствием приборов учета производится по нормативу, а не по фактическим затратам;

- ✧ коррелированность $N_{\text{собст}}$ с $W_{\text{пот}}$ отражает физический процесс – с увеличением нагрузки увеличиваются потери в трансформаторе;
- ✧ факторные признаки коррелированы между собой.

Зависимость мощности, необходимой для собственных нужд, от факторов, перечисленных выше и пронумерованных от x_1 до x_{26} в порядке их перечисления, в виде композиционной формулы первого вида записывается следующим образом:

$$y = P_1(P_1(P_1(P_2(P_4(x_{16}), P_3(x_{26})), P_1(P_2(x_{16}), P_3(x_1))), P_1(x_{21})), P_1(P_2(P_3(x_{19}), P_3(x_2)), P_2(P_4(x_3), P_2(x_{14}))))).$$

В виде композиционной формулы второго вида эта зависимость имеет вид:

$$y = (((P_4(x_{16}) \cdot P_1^2(P_3(x_{26}))) \cdot P_1^3(P_2(x_{16}) \cdot P_2^1(P_3(x_1)))) \cdot P_1^4(P_1(x_{21})) \cdot P_1^8((P_3(x_{19}) \cdot P_2^2(P_3(x_2))) \cdot P_1^7(P_4(x_3) \cdot P_2^2(P_2(x_{14}))))).$$

Параметры модели представлены в табл. 1 и 2, где ϵ – погрешность аппроксимации тестовой последовательности; k – номер объединения полиномов в пары операций. В первой таблице представлены параметры полиномов с одним аргументом, полученные на нулевом шаге алгоритма синтеза описания. Во второй таблице приведены параметры полиномов с двумя аргументами, полученные на последующих шагах. Заметим, что процедура селекции привела к тому, что из 26 исходных факторов в финальную модель вошло только 8 из них (№№1, 2, 3, 14, 16, 19, 21, 26).

Таблица 1

Параметры терминальных полиномов с одним аргументом

Номер аргумента	Порядок полинома	$\epsilon, \%$	Кoeffициенты a_i при x^i
16	4	15	$a_0=3,5; a_1=-233,5; a_2=0,345; a_3=0,089; a_4=96,5$
26	3	16	$a_0=-31,78; a_1=0,67; a_2=-0,96; a_3=32,57$
1	3	19	$a_0=562,76; a_1=0,02; a_2=27,74; a_3=-0,06$
3	4	22	$a_0=3,59; a_1=-454,98; a_2=90,76; a_3=925,62; a_4=-0,35$
21	1	26	$a_0=13,61; a_1=32,25$
19	3	27	$a_0=-835,51; a_1=0,73; a_2=-43,09; a_3=178,54$
2	3	27	$a_0=-0,36; a_1=0,89; a_2=8,14; a_3=38,29$
14	2	29	$a_0=-36,95; a_1=3298,53; a_2=-27,68$

Таблица 2

Параметры полиномов с двумя аргументами

k	Порядок полинома n	$\epsilon, \%$	Кoeffициенты $a_{(2n-i+3)/2+j}$ полинома, i - степень 1-го аргумента, $i=0, \dots, n$; j - степень 2-го аргумента, $j=0, \dots, n-i$
1	2	10	$a_0=0,01; a_1=-233,5; a_2=0,345; a_3=0,089; a_4=96,5; a_5=96,5$
2	2	9	$a_0=43,5; a_1=-4,23; a_2=-78,35; a_3=12,08; a_4=0,005; a_5=9,56$
3	1	8	$a_0=-0,0013; a_1=963,85; a_2=-1233,72$
4	2	7	$a_0=38,45; a_1=-835,5; a_2=0,31; a_3=1,078; a_4=166,5; a_5=-26,35$
5	2	6	$a_0=788,5; a_1=-2,05; a_2=0,0003; a_3=-0,189; a_4=45,3; a_5=966,85$
6	2	6	$a_0=31,25; a_1=-23,5; a_2=1,345; a_3=7,059; a_4=823,43; a_5=16,83$
7	1	5	$a_0=5,15; a_1=-3373,35; a_2=-247,12$
8	1	5	$a_0=3347,5; a_1=-0,0028; a_2=-2534,5$

Модель зависимости условно-постоянных затрат от технико-экономических показателей работы ГРЭС

К условно-постоянным затратам относятся затраты, которые не изменяются в зависимости от роста или сокращения объема производства. В соответствии с положением о калькулировании себестоимости электрической и тепловой энергии к условно-постоянным затратам относятся все затраты, связанные с производством, за исключением затрат на топливо. Факторы, влияющие на уменьшение или увеличение условно-постоянных затрат:

- ✧ W_1, W_2, W_p – объемы производства электроэнергии на оборудования первой и второй очереди электростанции и объем производства тепловой энергии;
- ✧ $M_{пер}$ – объем перевозок железнодорожным транспортом;
- ✧ Z^* , ТП – налоги, уплачиваемые от себестоимости продукции и объема товарной продукции;
- ✧ $S_{фонд}$ – амортизационные отчисления и налоги, уплачиваемые от стоимости основных фондов;
- ✧ $Ч, S_i$ – затраты на оплату труда и налоги, уплачиваемые от заработной платы;
- ✧ $K_{тл}$ – общая обеспеченность предприятия оборотными средствами для ведения хозяйственной деятельности;
- ✧ $K_{осс}$ – наличие собственных оборотных средств у предприятия, необходимых для финансовой устойчивости;

- ⇨ $K_{\text{впс}}$ — отражает возможность предприятия восстановить свою платежеспособность; коэффициенты $K_{\text{гл}}$, $K_{\text{осс}}$, $K_{\text{впс}}$ в своей совокупности отражают внешние и внутренние факторы, влияющие на производство;
- ⇨ $t_{\text{нв}}$ — сезонность затрат;
- ⇨ D — уровень инфляции в рассматриваемом периоде; за базовый период принят 1995 г.; коэффициенты приведены к базовому периоду;
- ⇨ $K_{\text{с}}$ — влияние курса рубля на стоимость материальных ресурсов и услуг производственного характера.

Исходные данные для анализа представляют отчетные материалы по Рязанской ГРЭС. Результаты предварительного корреляционного анализа показывают:

- ⇨ не все факторные признаки коррелированы с результативным признаком;
- ⇨ коррелированность УП с Z^* , ТП и $S_{\text{фонд}}$ указывает на наличие влияния налогов на УП;
- ⇨ коррелированность УП с $S_{\text{фонд}}$ подтверждает зависимость УП от стоимости основных фондов;
- ⇨ коррелированность УП с Ч и S_t отражает влияние трудозатрат на УП;
- ⇨ коррелированность УП с Ч и S_t подтверждает существование инфляции и зависимость производства от курса рубля к доллару;
- ⇨ отсутствие коррелированности УП с $K_{\text{гл}}$, $K_{\text{осс}}$, $K_{\text{впс}}$ означает, что внешние факторы и финансовое состояние предприятия не влияют на УП; это может быть следствием того, что электростанции являются предприятиями-монополистами, кроме того, производство электроэнергии и теплоэнергии имеет исключительную особенность — его продукция всегда реализуется на 100 %;
- ⇨ факторные признаки коррелированы между собой.

Зависимость условно-постоянных затрат от факторов, перечисленных выше и пронумерованных от x_1 до x_{18} в порядке их перечисления, в виде композиционной формулы первого вида записывается следующим образом:

$$y = P_1(P_1(P_1(P_2(P_3(x_5), P_3(x_6)), P_1(P_2(x_9), P_2(x_2))), P_2(P_3(x_{14}), P_3(x_{15}))), P_2(P_2(x_2), P_3(x_5)))$$

В виде композиционной формулы второго вида эта зависимость имеет вид:

$$y = (((P_3(x_5) \cdot P_1^2(P_3(x_6)) \cdot P_1^3(P_2(x_9) \cdot P_2^1(P_2(x_2)))) \cdot P_1^5(P_3(x_{14}) \cdot P_3^2(x_{15}))) \cdot P_1^7(P_2(x_2) \cdot P_3^2(x_5)))$$

Параметры модели представлены в табл. 3 и 4. Заметим, что процедура селекции привела к тому, что из 18 исходных факторов в финальную модель вошло только 6 из них (№№2, 5, 6, 9, 14, 15).

Таблица 3

Параметры терминальных полиномов с одним аргументом

Номер аргумента	Степень полинома	$\epsilon, \%$	Коэффициенты a_i при x_i
5	3	8	$a_0=17,35; a_1=-3,15; a_2=41,35; a_3=8,09; a_4=31,69$
6	3	9	$a_0=12,17; a_1=29,83; a_2=-0,08; a_3=-7,50; a_4=0,91$
9	2	11	$a_0=32,96; a_1=-456,2; a_2=442,74; a_3=-0,006$
2	2	17	$a_0=3212,56; a_1=-43,76; a_2=0,077; a_3=57,02$
14	3	18	$a_0=0,043; a_1=4444,55; a_2=0,001; a_3=40,09; a_4=-906,03$
15	3	19	$a_0=-11,03; a_1=-185,61; a_2=0,21; a_3=20,94; a_4=91,31$

Таблица 4

Параметры полиномов с двумя аргументами

k	Степень полинома n	$\epsilon, \%$	Коэффициенты $a_{(2n-i+j)/2+j}$ полинома, i - степень 1-го аргумента, $i=0, \dots, n$; j - степень 2-го аргумента, $j=0, \dots, n-i$
1	2	14	$a_0=44,14; a_1=-17,24; a_2=33,31; a_3=77,019; a_4=61,89; a_5=-262,03$
2	1	13	$a_0=-1,101; a_1=3,07; a_2=-187,92$
3	2	10	$a_0=88,01; a_1=-204,79; a_2=412,59; a_3=-29,48; a_4=76,15; a_5=-9,03$
4	2	10	$a_0=88,81; a_1=-2,19; a_2=-42,19; a_3=-1799,08; a_4=0,004; a_5=6,013$
5	1	7	$a_0=-94,33; a_1=376,94; a_2=-244,24$
6	1	6	$a_0=-10,03; a_1=8,012; a_2=-13,30$
7	1	6	$a_0=0,093; a_1=-3,26; a_2=771,25$

Заключение

Получены следующие основные теоретические результаты:

- 1) определены направления развития метода группового учёта аргументов, обеспечивающие повышение его селектирующих способностей и точности аппроксимации функции векторного аргумента;

2) разработаны модифицированные алгоритмы метода группового учёта аргументов, повышающие его эффективность посредством оптимизации частных полиномов; высокое качество предложенных алгоритмов (способность надёжно выделять эффективные аргументы, приемлемая точность интерполяции и экстраполяции процессов и высокая производительность) подтверждено методом моделирования; хорошие интерполяционные и экстраполяционные свойства предложенных алгоритмов позволяют использовать их не только для прогноза методом сценарных условий и лагированных переменных, но и для анализа по принципу «что – если»;

3) предложена оригинальная легко реализуемая программно нотация записи алгоритмических описаний, получаемых в результате реализации разработанных алгоритмов, в виде псевдоформул с композиционной формой записи полиномов двух видов, позволяющих наглядно представлять состав актуальных аргументов, порядки частных полиномов, вид частных приближений и последовательность их формирования.

Указанные результаты пополняют арсенал эффективных средств специалистов, занятых разработкой и анализом прогнозов социально-экономических процессов.

Основным практическим результатом является следующее:

1) построены алгоритмические модели, позволяющие исследовать взаимосвязь между результативными и факторными показателями социально-экономической деятельности крупного энергогенерирующего предприятия ОАО «Рязанская ГРЭС» (две модели приведены в данной статье в качестве примеров);

2) построенные модели использованы для разработки прогноза социально-экономического статуса ОАО «Рязанская ГРЭС»;

3) результаты анализа полученных моделей и прогнозных значений проблемных показателей использованы для коррекции и планирования социально-экономической политики предприятия ОАО «Рязанская ГРЭС». ■

Литература

1. Айвазян С.А., Мхитарян В.С. Прикладная статистика и основы эконометрики. М.: ЮНИТИ, 1998. 1024 с.
2. Себер Дж. Линейный регрессионный анализ: Пер с англ. М.: Мир, 1980. 456 с.
3. Уоссерман Ф. Нейрокомпьютерная техника: Теория и практика: Пер. с англ. М.: Мир, 1992. 240 с.
4. Габор Д. Перспективы планирования // Автоматика. 1972. № 2.
5. Ивахненко А. Г. и др. Принятие решений на основе самоорганизации. М.: Сов. радио, 1976. 280 с.
6. Тамура, Кондо. Современная методология групповой обработки данных и ее приложения // Оперсэндзу рисати. 1987. № 2. С. 104 – 111.
7. Прикладные нечеткие системы: Пер. с япон. / К. Асаи, Д. Ватада, С. Иваи и др. Под редакцией Т. Тэрано, К. Асаи, М. Сугано. М.: Мир, 1993. 368 с.
8. Белов В.В., Васильев С.В., Наумкина С.Г. Модифицированный метод группового учета аргументов на основе процедуры оптимизации частных полиномов // Вычислительные машины, комплексы и сети: Межвуз. сб. науч. трудов. Рязань: РГРТА, 1999. С. 95–99.



ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ – ВЫСШАЯ ШКОЛА ЭКОНОМИКИ

представляет свои периодические издания

ВОПРОСЫ ОБРАЗОВАНИЯ
ЕЖЕКВАРТАЛЬНЫЙ НАУЧНО-ОБРАЗОВАТЕЛЬНЫЙ
ЖУРНАЛ

Издается с 2004 г.

Главный редактор –
Ярослав Иванович Кузьминов

Издание освещает теоретические и прикладные проблемы российского образования. Содержит статьи ведущих российских и зарубежных ученых и экспертов. В каждом номере – дискуссии, рецензии, обзоры публикаций и законодательства в области образования.

Каталог Агентства «Роспечать» – индекс 82950 Объединенный каталог «Пресса России» – индекс 15163

Координаты редакции:
101990 Москва, ул. Мясницкая, 20, офис 308
E-mail: edu.journal@hse.ru
Тел: (495) 628-5102, 621-8523

ПРОГНОЗИРОВАНИЕ ТЕНДЕНЦИЙ ВРЕМЕННЫХ РЯДОВ НА ОСНОВЕ ОДНОФАКТОРНОЙ НЕЧЕТКОЙ МОДЕЛИ С ИСПОЛЬЗОВАНИЕМ ДИСКРЕТНЫХ НЕЧЕТКИХ МНОЖЕСТВ ВТОРОГО ТИПА И ГЕНЕТИЧЕСКОГО АЛГОРИТМА

Л.А. Демидова,

к.т.н., доцент кафедры вычислительной и прикладной математики Рязанского государственного радиотехнического университета,
e-mail Liliya.Demidova@rambler.ru

Рассматривается применение дискретных нечетких множеств второго типа для разработки однофакторных нечетких моделей прогнозирования. Предлагается генетический алгоритм, обеспечивающий выбор оптимальных параметров модели прогнозирования – действительных чисел для корректировки границ универсума, числа интервалов разбиения универсума и степеней принадлежности элементов дискретных нечетких множеств второго типа.

Введение

Анализ временных рядов играет важную роль в решении многих актуальных задач, например, при краткосрочном прогнозировании тенденций рынка труда в России. В настоящее время существует необходимость в разработке методов прогнозирования, которые бы обеспечили получение адекватной оценки предстоящих изменений политики и принятия решений в региональных органах управления на основе известных показателей развития регионов. Так как большинство реальных событий характеризуются некоторой неопределенностью, то каждому наблюдению временного ряда (фактора) можно поставить в соответствие нечеткую переменную с некоторой функцией принадлежности.

Модель прогнозирования на основе нечетких множеств первого типа

Нечеткие временные ряды могут быть представлены с помощью нечетких множеств первого или второго типа [1, 2, 3].

Дискретное нечеткое множество первого типа (ДНМТ1) A , определенное на универсуме может быть определено в виде:

$$A = f_A(u_1)/u_1 + f_A(u_2)/u_2 + \dots + f_A(u_n)/u_n, \quad (1)$$

где $f_A(u)$ – функция принадлежности ДНМТ1 A , $f_A(u): U \rightarrow [0, 1]$, $f_A(u_r)$ определяет степень принадлежности элемента u_r ДНМТ1 A , $r = \overline{1, n}$.

Пусть $Y(t)$ ($t = \dots, 0, 1, 2, \dots$) – универсум, определенный на множестве действительных чисел, а $F(t)$ – набор функций $f(t)$ ($i = 1, 2, \dots$), определенных на универсуме $Y(t)$. Тогда $F(t)$ называется нечетким временным рядом на универсуме $Y(t)$.

Пусть

$$F(t) = F(t-1) \circ R(t, t-1),$$

где $R(t, t-1)$ – нечеткое отношение и \circ – операция *max-min* композиции.

Обозначим зависимость $F(t)$ от $F(t-1)$ как $F(t-1) \rightarrow F(t)$, где $F(t-1)$ и $F(t)$ – нечеткие множества.

$$C_{\bar{A}} = \int \dots \int_{z_1 \in Z_1} \int_{z_n \in Z_n} \int_{w_1 \in W_1} \int_{w_n \in W_n} 1 / \left(\sum_{r=1}^n w_r \cdot z_r / \sum_{r=1}^n w_r \right) = [y_{left}, y_{right}], \quad (7)$$

где Z_r ($r = \overline{1, n}$) в (7) представляет собой ДНМТ1, имеющее центр c_r и протяженность s_r ($s_r \geq 0$), а W_r ($r = \overline{1, n}$) представляет собой ДНМТ1, имеющее центр h_r и протяженность Δ_r ($\Delta_r \geq 0$).

Так как $Z_1, \dots, Z_n, W_1, \dots, W_n$ представляют собой ДНМТ1, то и $C_{\bar{A}}$ является ДНМТ1. Для вычисления центра $C_{\bar{A}}$ необходимо найти две конечные точки интервала: y_{left} и y_{right} и, рассмотрев задачи минимизации и максимизации функции (положив $z_r = c_r + s_r$ и $z_r = c_r - s_r$ соответственно) [3]:

$$y(w_1, \dots, w_n) = \sum_{r=1}^n w_r \cdot z_r / \sum_{r=1}^n w_r \quad (8)$$

при условии $w_r \in [h_r - \Delta_r, h_r + \Delta_r], h_r \geq \Delta_r, r = \overline{1, n}$.

Продифференцируем функцию $y(w_1, \dots, w_n)$ по w_k :

$$\begin{aligned} \frac{\partial}{\partial w_k} y(w_1, \dots, w_n) &= \frac{\partial}{\partial w_k} \left(\sum_{r=1}^n w_r \cdot z_r / \sum_{r=1}^n w_r \right) = \\ &= (z_k - y(w_1, \dots, w_n)) / \sum_{r=1}^n w_r. \end{aligned} \quad (9)$$

Так как $\sum_{r=1}^n w_r > 0$,

то из (9) следует, что:

$$\frac{\partial}{\partial w_k} y(w_1, \dots, w_n) \begin{cases} \geq 0, & \text{если } w_k \leq y(w_1, \dots, w_n) \\ \leq 0, & \text{если } w_k \geq y(w_1, \dots, w_n) \end{cases} \quad (10)$$

Так как из $y(w_1, \dots, w_n) = z_k$ следует, что

$$\begin{aligned} \sum_{r=1}^n w_r \cdot z_r / \sum_{r=1}^n w_r &= z_k, \text{ то} \\ \sum_{r=1, r \neq k}^n w_r \cdot z_r / \sum_{r=1, r \neq k}^n w_r &= z_k. \end{aligned} \quad (11)$$

Из (10) видно, что если $z_k > y(w_1, \dots, w_n)$, то $y(w_1, \dots, w_n)$ увеличивается при увеличении w_k ; а если $z_k < y(w_1, \dots, w_n)$, то $y(w_1, \dots, w_n)$ уменьшается при уменьшении w_k . Для вычисления центра $C_{\bar{A}}$ можно использовать итерационный алгоритм Карника-Менделя [3].

Итерационный алгоритм Карника-Менделя

Пусть $h_r \geq \Delta_r$ так, что $w_r \geq 0$ для $r = \overline{1, n}$. Максимальное (минимальное) значение, которое может принимать w_k ($k = \overline{1, n}$), равно $h_k + \Delta_k$ ($h_k - \Delta_k$).

Функция $y(w_1, \dots, w_n)$ достигает своего максимального значения, если:

♦ $w_k = h_k + \Delta_k$ для тех значений k , для которых $z_k > y(w_1, \dots, w_n)$;

♦ $w_k = h_k - \Delta_k$ для тех значений k , для которых $z_k < y(w_1, \dots, w_n)$.

Функция $y(w_1, \dots, w_n)$ достигает своего минимального значения, если:

♦ $w_k = h_k - \Delta_k$ для тех значений k , для которых $z_k > y(w_1, \dots, w_n)$;

♦ $w_k = h_k + \Delta_k$ для тех значений k , для которых $z_k < y(w_1, \dots, w_n)$.

Максимум функции $y(w_1, \dots, w_n)$ может быть определен с помощью следующей итерационной процедуры. Пусть $z_r = c_r + s_r, (r = \overline{1, n})$. Предположим, что все z_r упорядочены по возрастанию, то есть $z_1 \leq z_2 \leq \dots \leq z_n$.

1. Пусть $w_r = h_r$ для $r = \overline{1, n}$. Вычислим $y' = y(h_1, \dots, h_n)$ по формуле (8).

2. Определим такое $k (1 \leq k \leq n-1)$, что $z_k \leq y' \leq z_{k+1}$.

3. Пусть $w_r = h_r - \Delta_r$ для $r \leq k$ и $w_r = h_r + \Delta_r$ для $r \geq k+1$. Вычислим $y'' = y(h_1 - \Delta_1, \dots, h_k - \Delta_k, h_{k+1} + \Delta_{k+1}, \dots, h_n + \Delta_n)$ по формуле (8). Так как все z_r упорядочены по возрастанию, с учетом формул (10) и (11) и неравенства $z_k \leq y' \leq z_{k+1}$, можно утверждать, что $y'' \geq y'$, поскольку w_r выбраны так, что w_r уменьшены для $r \leq k$ и w_r увеличены для $r \geq k+1$.

4. Если $y' = y''$, то вычисления заканчиваются, а y'' представляет собой максимум функции $y(w_1, \dots, w_n)$. Если $y' \neq y''$, то осуществляется переход к шагу 5.

5. Полагаем $y' = y''$ и осуществляем переход к шагу 2.

Алгоритм требует не более n итераций, где одна итерация состоит из шагов 2–5 [3].

Минимум функции $y(w_1, \dots, w_n)$ может быть определен с помощью аналогичной итерационной процедуры, где $z_r = c_r + s_r, (r = \overline{1, n})$; а на Шаге 3 для вычисления $y' = y(h_1 + \Delta_1, \dots, h_k + \Delta_k, h_{k+1} - \Delta_{k+1}, \dots, h_n + \Delta_n)$ полагается, что $w_r + \Delta_r$ для $r \leq k$ и $w_r = h_r - \Delta_r$ для $r \geq k+1$.

Генетический алгоритм, обеспечивающий повышение точности прогнозирования на основе нечетких множеств второго типа

Самостоятельной задачей при прогнозировании на основе нечетких временных рядов является определение оптимальных параметров модели, обеспечивающих максимальную точность прогнозирования: действительных чисел D_1, D_2 , используемых при корректировке универсума U , количества интервалов разбиения $D_3 = n$ универсума U и степеней принадлежности элементов D_4 (α_{upper}), D_5 (α_{lower}) для ИДНМТ2. Применение генетического алгоритма (ГА) позволяет значительно сократить время

поиска оптимальных значений параметров D_1, D_2, D_3, D_4, D_5 [5]. При этом хромосома s будет иметь вид: $s = (D_1, D_2, D_3, D_4, D_5)$. Для каждого элемента хромосомы следует задать диапазоны их изменения: для $D_1 - [-d_1; 0]$, для $D_2 - [0; d_2]$, для $D_3 = n - [2; n_{\max}]$, для $D_4, D_5 - [0; 1]$, где d_1, d_2 , – положительные действительные числа, равные, например, $d_i = D_{\max} - D_{\min}$, $i = 1, 2$; n_{\max} – натуральное число, $n_{\max} \leq m - 1$ (m – количество значений временного ряда). Также при формировании начальной популяции, при выполнении операций скрещивания и мутации необходимо следить за выполнением требования: $D_4 \geq D_5$ так как элементы D_4, D_5 , в хромосоме определяют верхнее и нижнее значение функций принадлежности ИДНМТ2 соответственно.

Для каждого набора параметров D_1, D_2, D_3, D_4 , и D_1, D_2, D_3, D_5 необходимо вычислить функцию соответствия. В качестве функции соответствия можно выбрать функцию (5). Однако при вычислении функции соответствия по формуле (5) для хромосом как начальной популяции размера P , так и расширенной популяции размера $(P + P * P_c)$ (где P_c – коэффициент скрещивания) может быть получено значение вида «0/0» (если имеются группы нечетких зависимостей с неопределенными правыми частями, и поэтому невозможно вычислить прогнозное значение по формулам (3), (4), так как значение приращения фактора y_{i+1} определяется как «0/0»). Поэтому при удалении из популяции размером $(P + P * P_c)$ хромосом с худшими значениями функций соответствия следует предварительно оценить хромосомы со значением функции соответствия «0/0» как наихудшие (например, таким хромосомам можно поставить в соответствие максимально возможное значение ошибки, равное 100%). При сортировке хромосом по возрастанию значений функций соответствия хромосомы со значением функции соответствия, равным «0/0», будут занимать последние места в списке и в результате отбраковки $P * P_c$ худших хромосом будут исключаться из популяции [1]. Для обеспечения гарантированного выполнения прогноза следующего значения временного ряда необходимо видоизменить функцию соответствия в ГА следующим образом. Если для каждого набора D_1, D_2, D_3, D_4 , и D_1, D_2, D_3, D_5 некоторой хромосомы s определены все правые части групп нечетких зависимостей, то функция соответствия для этих наборов вычисляется по формуле (5). Если для любого из наборов D_1, D_2, D_3, D_4 , и D_1, D_2, D_3, D_5 не определена хотя бы одна правая часть в группах нечетких зависимостей, то значение функции соответствия находится как сумма средней относительной ошибки прогноза по формуле (5) и числа 100.

Таким образом, видоизмененная функция соответствия имеет вид:

$$J_V(s) = \begin{cases} \text{если определены все} \\ \text{правые части в группах} \\ \text{логических зависимостей} \\ AFER, \\ \text{если не определена хотя бы} \\ \text{одна правая часть в группах} \\ \text{логических зависимостей} \\ AFER + 100, \end{cases} \quad (12)$$

где $AFER$ определяется по формуле (5), $V = \alpha_{lower} \alpha_{upper}$.

В результате, набор D_1, D_2, D_3, D_4 или D_1, D_2, D_3, D_5 , несмотря на то, что для него средняя относительная ошибка прогноза по формуле (5) может быть минимальной (если значение функции соответствия не определяется как «0/0»), будет признан одним из худших в процессе реализации ГА, и, возможно, соответствующая ему хромосома будет исключена из популяции.

На основе наборов D_1, D_2, D_3, D_4 и D_1, D_2, D_3, D_5 , вычисляются два значения функций соответствия $J_{\alpha_{upper}}$ и $J_{\alpha_{lower}}$. Если хотя бы для одного из наборов значение функции соответствия оказалось больше 100, то соответствующую ему хромосому следует признать «нежизнеспособной» и положить значение ее функции соответствия J_s равным наибольшему из двух значений функций соответствия наборов D_1, D_2, D_3, D_4 и D_1, D_2, D_3, D_5 . При этом нет необходимости в вычислении функции соответствия для хромосомы с использованием алгоритма Карника-Менделя для определения центроида ИДНМТ2. В противном случае для хромосомы s вычисляется средняя относительная ошибка прогнозирования $AFER$ по формуле (5) с использованием алгоритма Карника-Менделя для определения центроида ИДНМТ2. Если значение $AFER$ для s (и ИДНМТ2) окажется меньше, чем значения функций соответствия $J_{\alpha_{upper}}$ и $J_{\alpha_{lower}}$ и для наборов D_1, D_2, D_3, D_4 и D_1, D_2, D_3, D_5 соответственно, то такую хромосому следует считать «жизнеспособной», а значение ее функции соответствия J_s положить равным $AFER$, иначе необходимо положить значение ее функции соответствия J_s равным $AFER + 100$ (для возможного исключения этой хромосомы из популяции). Таким образом, функцию соответствия хромосомы s следует вычислять по формуле:

$$J_s(s) = \begin{cases} AFER, \text{ при } AFER < J_{\alpha_{upper}}, \text{ } AFER < J_{\alpha_{lower}} \\ AFER + 100, \text{ иначе} \end{cases} \quad (13)$$

Отметим, что необходимо затратить дополнительное время на формирование начальной популяции, чтобы она состояла только из «жизнеспособных»

хромосом (у которых значение функции соответствия меньше 100). Тогда выполнение операций скрещивания и мутации будет более эффективным и результативным (иначе вся популяция может быть с самого начала «нежизнеспособной»). Хромосома, минимизирующая функцию (13), имеет больше шансов быть признанной лучшей. Выбор родителя будет состоять в выборе лучшей хромосомы из двух случайно выбранных. Затем две выбранные таким образом хромосомы-родителя используются для скрещивания, при этом выбирается коэффициент скрещивания R_c и генерируется число $N_c = \text{rahdrom}([0, 1])$. Если $R_c > N_c$, то случайным образом выбирается точка скрещивания z и выполняется скрещивание. При выполнении мутации выбирается коэффициент мутации R_m и генерируется число $N_m = \text{rahdrom}([0, 1])$. Если $R_m > N_m$, то случайным образом выбирается точка мутации z .

Тогда генетический алгоритм имеет вид:

1. Создается начальная популяция размера P из случайным образом выбранных хромосом s .

2. При $g < G$ (G – количество генераций) вычисляется функция соответствия для каждой хромосомы, затем создается $P/2$ пар хромосом-родителей и осуществляется переход к шагу 3. При $g > G$ осуществляется переход к шагу 5.

3. Выполняются операции скрещивания и мутации для текущей популяции.

4. Создается новая популяция размера P , дополненная хромосомами-детьми, а хромосомы с худшими значениями функции соответствия отбрасываются.

5. Выбирается лучшая хромосома, которая минимизирует функцию соответствия.

Пример прогнозирования

На примере данных по фактору «численность занятого населения» (в России) для периодов 2-1999 – 5-2004, полученных от Госкомстата (табл. 1) была построена нечеткая модель прогнозирования на основе ИДНМТ2. Представление выходных значений фактора на основе ИДНМТ2 приведено в табл. 1. При этом были получены группы нечетких логических зависимостей:

Группа 1: $FOU_1 \rightarrow FOU_2$.

Группа 2: $FOU_2 \rightarrow FOU_5, FOU_6$.

Группа 3: $FOU_3 \rightarrow FOU_2, FOU_3, FOU_4 FOU_7$.

Группа 4: $FOU_4 \rightarrow FOU_3$.

Группа 5: $FOU_5 \rightarrow FOU_1, FOU_3, FOU_5$.

Группа 6: $FOU_6 \rightarrow FOU_5$.

Группа 7: $FOU_7 \rightarrow FOU_4$.



Рис. 2. Графические зависимости для реальных и прогнозируемых значений

В табл. 2 приведены параметры трех однофакторных нечетких моделей и результаты прогнозирования. «Модель 1» основана на ДНМТ1 при заранее заданном значении степени принадлежности элементов нечеткому множеству. Параметры определялись с помощью ГА. «Модель 2» основана на ДНМТ1. Параметры определялись с помощью ГА. «Модель 3» основана на ИДНМТ2. Параметры определялись с помощью ГА. На рис. 2 приведены графические зависимости для реальных и прогнозируемых значений на основе ИДНМТ2. Анализ результатов моделирования показывает уменьшение средней относительной ошибки прогнозирования за счет применения ИДНМТ2 (при этом относительная ошибка прогнозирования для периода 5-2004 на основе «Модели 3» равна 1,173505 %).

Заключение

Предлагаемый метод прогнозирования тенденций рынка труда обеспечивает получение более высоких результатов прогноза, чем предложенный в [4], и может быть рекомендован для проведения краткосрочных прогнозов. Применение ИДНМТ2 и ГА для поиска оптимальных параметров нечеткой модели обеспечили более высокую точность прогноза. Для повышения точности прогноза можно представлять значения временных рядов с помощью непрерывных нечетких множеств второго типа, что приводит, однако, к соответствующему увеличению вычислительной сложности и временных затрат. ■

Таблица 1

Фактор «численность занятого населения»

Месяц-год	Фактор, тыс. ч.	Приращение, тыс. ч.	Входное ИДНМТ2	Выходное ИДНМТ2 ($V = \alpha_{lower} \alpha_{upper}$)
2-99	60614	–	–	–
5-99	62462	1848	FOU_6	–
8-99	63742	1280	FOU_5	$0/A_1+0/A_2+0/A_3+V/A_4+1/A_5+V/A_6+0/A_7$
11-99	63082	-660	FOU_3	$1/A_1+V/A_2+1/A_3+V/A_4+1/A_5+V/A_6+0/A_7$
2-00	62439	-643	FOU_3	$V/A_1+1/A_2+1/A_3+1/A_4+V/A_5+V/A_6+1/A_7$
5-00	64961	2522	FOU_7	$V/A_1+1/A_2+1/A_3+1/A_4+V/A_5+V/A_6+1/A_7$
8-00	65154	193	FOU_4	$0/A_1+0/A_2+V/A_3+1/A_4+V/A_5+0/A_6+0/A_7$
11-00	64465	-689	FOU_3	$0/A_1+V/A_2+1/A_3+V/A_4+1/A_5+V/A_6+0/A_7$
2-01	62953	-1512	FOU_2	$V/A_1+1/A_2+1/A_3+1/A_4+V/A_5+V/A_6+1/A_7$
5-01	64542	1589	FOU_6	$0/A_1+0/A_2+0/A_3+V/A_4+1/A_5+1/A_6+V/A_7$
8-01	65459	917	FOU_5	$0/A_1+0/A_2+0/A_3+V/A_4+1/A_5+V/A_6+0/A_7$
11-01	64664	-795	FOU_3	$1/A_1+V/A_2+1/A_3+V/A_4+1/A_5+V/A_6+0/A_7$
2-02	65021	357	FOU_4	$V/A_1+1/A_2+1/A_3+1/A_4+V/A_5+V/A_6+1/A_7$
5-02	65962	941	FOU_5	$0/A_1+V/A_2+1/A_3+V/A_4+1/A_5+V/A_6+0/A_7$
8-02	67502	1540	FOU_5	$1/A_1+V/A_2+1/A_3+V/A_4+1/A_5+V/A_6+0/A_7$
11-02	65766	-1736	FOU_1	$1/A_1+V/A_2+1/A_3+V/A_4+1/A_5+V/A_6+0/A_7$
2-03	64104	-1662	FOU_2	$V/A_1+1/A_2+V/A_3+0/A_4+0/A_5+0/A_6+0/A_7$
5-03	65528	1424	FOU_5	$0/A_1+0/A_2+0/A_3+V/A_4+1/A_5+1/A_6+V/A_7$
8-03	66674	1146	FOU_5	$1/A_1+V/A_2+1/A_3+V/A_4+1/A_5+V/A_6+0/A_7$
11-03	66496	-178	FOU_3	$1/A_1+V/A_2+1/A_3+V/A_4+1/A_5+V/A_6+0/A_7$
2-04	64941	-1555	FOU_2	$V/A_1+1/A_2+1/A_3+1/A_4+V/A_5+V/A_6+1/A_7$
5-04	67271	2330	FOU_6	$0/A_1+0/A_2+0/A_3+V/A_4+1/A_5+1/A_6+V/A_7$

Таблица 2

Параметры нечётких моделей

Параметры	«Модель 1»	«Модель 2»	«Модель 3»
D_1	816,486940898299	818,938883168293	818,914669508277
D_2	662,918661869601	656,198590769605	656,765458625010
$D_3 = n$	7	7	7
$D_0 = \alpha$	0,5	0	–
$\alpha_{upper} (\alpha_{lower})$	–	–	1 (0)
$AFER$ (%)	1,22965304295085	1,22676137780468	1,22528803913897
$AFER_{upper}$ (%)	–	–	1,24229944946879
$AFER_{lower}$ (%)	–	–	1,22678992137390
t (с)	36,782	91,203	187,734

Литература

1. Демидова Л.А. Прогнозирование тенденций рынка труда на основе однофакторных нечетких временных рядов // Системы управления и информационные технологии, 2007, № 3.2(29). – С. 241–246.
2. Chen S.M. Forecasting enrollments based on fuzzy time series // Fuzzy Sets Systems, 1996, vol. 81, no. 3, pp. 311–319.
3. Mendel J. M. Type-2 fuzzy sets and systems: an overview // IEEE Computational intelligence magazine. 2007, vol. 2, № 1, pp. 20–29.
4. Ярушкина Н.Г. Основы теории нечетких и гибридных систем: Учеб. пособие. – М.: Финансы и статистика, 2004, 320 с.: ил.



ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ – ВЫСШАЯ ШКОЛА ЭКОНОМИКИ

представляет свои периодические издания

**ЭКОНОМИЧЕСКИЙ ЖУРНАЛ
ВЫСШЕЙ ШКОЛЫ ЭКОНОМИКИ**
ЕЖЕКВАРТАЛЬНЫЙ НАУЧНО-ИНФОРМАЦИОННЫЙ
ЖУРНАЛ

Издается с 1997 г.

Главный редактор –
Евгений Евгеньевич Гавриленков

Журнал освещает теоретические и прикладные проблемы экономической науки. В каждом* номере – статьи ведущих российских экономистов. Рецензии, методологические и лекционные материалы. Эксклюзивные статьи зарубежных экономистов. Данные официальной статистики по широкому кругу вопросов.

Каталог Агентства «Роспечать» – индекс 79264
Объединенный каталог «Пресса России» – индекс 29233

Координаты редакции:
101990 Москва, ул. Мясницкая, 20, офис 235
e-mail: redact@hse.ru, тел./факс: (495) 628-0442

**РОССИЙСКАЯ ЭКОНОМИКА:
ПРОГНОЗЫ И ТЕНДЕНЦИИ**
ЕЖЕМЕСЯЧНЫЙ СПРАВОЧНО-АНАЛИТИЧЕСКИЙ
ЖУРНАЛ

Издается с 1993 г.

Главный редактор –
Елена Анатольевна Иванова

Журнал освещает состояние, динамика и дает прогноз основных социально-экономических индикаторов. В каждом номере – хроника событий экономики. Результаты конъюнктурных опросов предприятий. Самые свежие данные. Аналитический материал представлен с использованием таблиц, графиков и диаграмм.

Каталог Агентства «Роспечать» – индекс 79275
Объединенный каталог «Пресса России» – индекс 40548

Координаты редакции:
117312 Москва, ул. Вавилова, 7, офис 203
E-mail: id.hse@mail.ru Тел./факс: (495) 772-9571

РАСПРОСТРАНЯЮТСЯ ПО РОССИИ И ДРУГИМ СТРАНАМ СНГ. ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ – НА САЙТЕ: www.hse.ru



ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ – ВЫСШАЯ ШКОЛА ЭКОНОМИКИ

представляет свои периодические издания

**ВЕСТНИК МЕЖДУНАРОДНЫХ
ОРГАНИЗАЦИЙ:**
образование, наука, новая экономика
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКИЙ ЖУРНАЛ

Издается с 2006 г.

Главный редактор –
Марина Владимировна Ларионова

Освещает деятельность ведущих международных организаций и объединений в области образования, науки, новой экономики, а также в области международной и социально-экономической политики, решения вопросов глобального развития. Содержит информацию о международных конференциях, форумах и семинарах, проектах и новых публикациях.

Каталог Агентства «Роспечать» – индекс 20054

Координаты редакции:
101000 Москва, ул. Мясницкая, 20
Тел.: (495) 621-4464, факс: (495) 621-8711
E-mail: iori@hse.ru

О ФУНДАМЕНТАЛЬНЫХ СВОЙСТВАХ ДРОБНЫХ ОПЕРАТОРОВ В КЛАССЕ ЗАДАЧ НЕЛИНЕЙНОЙ ДИНАМИКИ

О.Р. Козырев,

профессор Государственного университета – Высшей школы экономики, Москва
okozyrev@hse.ru

К.В. Логвинова,

профессор Государственного университета – Высшей школы экономики, Нижний Новгород
klogvinova@hse.nnov.ru

Исследованы свойства одного из дробных операторов, встречающихся в задачах гетерогенной динамики в средах со стохастическим поведением. Рассмотрено существование соответствующего фундаментального решения.

Фундаментальные свойства дробного оператора

Рассмотрим дробный дифференциальный оператор

$$L = \frac{1}{2} (D_+^{2-\alpha} + D_-^{2-\alpha}).$$

Он встречается в некоторых задачах диффузии в случайных средах и определяется так же как в [1–2], заданный на функциональном пространстве для элементов, достаточно быстро убывающих на бесконечности и имеющих поведение в нуле вида . Здесь, согласно определению [3], имеем

$$D_+^{2-\alpha} = \frac{1}{\Gamma(\alpha)} \frac{d^2}{dt^2} \int_0^t \frac{f(\tau)}{(t-\tau)^{1-\alpha}} d\tau,$$

$$D_-^{2-\alpha} = \frac{1}{\Gamma(\alpha)} \frac{d^2}{dt^2} \int_t^0 \frac{f(\tau)}{(\tau-t)^{1-\alpha}} d\tau,$$

где $0 < \alpha \leq 1$. Интеграл понимается в смысле Римана, причём предполагается существование и при $t \rightarrow \infty$.

Образ

$$Lf \in [C^\infty(\mathbb{R}_+) \cap C^\infty(\mathbb{R}_-)]$$

будет достаточно быстро убывать на бесконечности вместе со всеми своими производными.

Мы можем определить L на обобщённом функциональном пространстве. Для всех обобщённых функций f , имеющих указанные выше асимптотические и дифференциальные характеристики результат действия Lf также является регулярной функцией, совпадающей с обычной, получающейся при стандартном определении.

Утверждение. Оператор L с указанными свойствами существует.

Покажем это.

Пусть

$$f \in C^\infty(\mathbb{R}); \quad f(x) = O(|x|^{2-\alpha}), \quad \text{если } x \rightarrow 0.$$

Тогда при $x \rightarrow 0$ для $\forall n$ имеем

$$f(x) = O(x^{-n}).$$

В этом случае:

$$\begin{aligned} (Lf)(x) &= \frac{1}{2\Gamma(\alpha)} D^2 \left[\int_x^\infty (y-x)^{\alpha-1} f(y) dy + \int_{-\infty}^x (x-y)^{\alpha-1} f(y) dy \right] = \\ &= \frac{1}{2\Gamma(\alpha)} D^2 \left[\int_{-\infty}^\infty (|y-x|^{\alpha-1} f(y) dy \right] = \frac{1}{2\Gamma(\alpha)} D^2 [|x|^{\alpha-1} \cdot f]. \end{aligned}$$

Рассмотрим свойства (1), используя преобразование Фурье F [3].

Пусть

$$\varphi(s) = F[f(x)].$$

Исходя из свойств f , очевидно, что:

- ◆ для $x \rightarrow \infty$ имеем $x^n f(x) \rightarrow 0$;
- ◆ для $k \leq n$ имеем $x^2 D^k[x^n f(x)] \rightarrow 0$ и $D^k[x^n f(x)]$ непрерывна и ограничена.

В этом случае интеграл Фурье для $x^{n-2} f(x)$ сходится абсолютно.

Для $\varphi(s)$ это означает, что существуют непрерывные и ограниченные:

$$\left(\frac{d}{ds}\right)^{n-2} F[f(x)], \quad s^k \left[\left(\frac{d}{ds}\right)^n F(f(x))\right]. \quad (2)$$

Как известно из свойств преобразования Фурье, при $|s| \rightarrow \infty$ мы имеем:

$$\varphi^k(s) = 0 \left(|s|^{-k-\text{асимптотика}} \cdot f(x) \right).$$

Всё вышеуказанное означает, что существует

$$\begin{aligned} F[|x|^{\alpha-1} \cdot f(x)] &= F[|x|^{\alpha-1}] \cdot F[f(x)] = \\ &= 2\Gamma(\alpha) \cdot \cos \frac{\pi\alpha}{2} |s|^{-\alpha} \varphi(s) \end{aligned} \quad (3)$$

Первый сомножитель распределения обращается в регулярную функцию.

Исходя из свойств $\varphi(s)$, результат ограничен даже при умножении на s^2 , поэтому выражение $|x|^{\alpha-1} \cdot F$ дважды дифференцируемо почти всюду.

Отметим, что

$$F\left(\frac{1}{2\Gamma(\alpha)} D^2[|x|^{\alpha-1} \cdot f(x)]\right) = |s|^{2-\alpha} \cos \frac{\pi\alpha}{2} \varphi(s).$$

поэтому линейный оператор L в пространстве функций порождает такой оператор \tilde{L} в пространстве Фурье, что для всех $\varphi(s)$ с указанными выше свойствами мы имеем:

$$(\tilde{L}\varphi)(s) = |s|^{2-\alpha} \cos \frac{\pi\alpha}{2} \varphi(s).$$

В этом случае \tilde{L} продолжаем во всём пространстве обобщённых функций и

$$\tilde{L} = |s|^{2-\alpha} \cos \frac{\pi\alpha}{2} \varphi(s) + \sum_{j=0}^N c_j \delta^{(j)}(s).$$

Здесь N определяется поведением $\varphi(s)$ в нуле.

Окончательно. Если нам известен оператор \tilde{L} , мы всегда можем определить, на какое пространство следует распространить оператор L , чтобы Lf было регулярной функцией.

Существование фундаментального решения

Рассмотрим дифференциальное уравнение вида [2]:

$$\frac{\partial}{\partial t} \Theta(x, t) = L\Theta(x, t). \quad (4)$$

Применение преобразования Фурье приводит к равенству:

$$\begin{aligned} \frac{\partial}{\partial t} \tilde{\Theta}(s, t) &= \tilde{L}(s) \cdot \tilde{\Theta}(s, t) = \tilde{L}_0(s) \cdot \tilde{\Theta}(s, t) + \\ &+ C_0 \tilde{\Theta}(0, t) + C_1 \frac{\partial}{\partial t} \tilde{\Theta}(0, t) + \dots + C_N \left(\frac{\partial}{\partial t}\right)^N \tilde{\Theta}(0, t). \end{aligned}$$

Здесь $\tilde{L}(s)$ – распределение, а все C_i зависят от граничных условий (источники и т.п.). Если все они нулевые, мы можем определить

$$\tilde{\Theta}(s, t) = \exp\left(t |s|^{2-\alpha} \cos \frac{\pi\alpha}{2}\right) \cdot \tilde{\Theta}(s, 0),$$

$$\Theta(x, t) = F^{-1}\left(\exp\left(t |s|^{2-\alpha} \cos \frac{\pi\alpha}{2}\right)\right) * \Theta(x, 0).$$

Фундаментальное решение в точке x_0 выглядит следующим образом:

$$\Theta_{x_0} \sim F^{-1}\left(\exp\left(isx_0 - \lambda t |s|^{2-\alpha} \cos \frac{\pi\alpha}{2}\right)\right).$$

Литература

1. Логвинова К.В., Морозов В.П. Уравнения переноса в случайных средах. Известия АИН РФ, сер. Прикладная математика и механика, 2002, Т. 3, С. 69–73.
2. Логвинова К.В. Диффузия в дельта – коррелированной гауссовой случайной среде. Известия АИН РФ, сер. Прикладная математика и механика, 2003, Т. 4, С. 88–91.
3. Самко С.Г., Килбас Л.А., Маричев О.И. Интегралы и производные дробного порядка и некоторые их приложения. Минск: наука и техника, 1987, 688 С.

МОДЕЛИРОВАНИЕ И АВТОМАТИЧЕСКАЯ ПРОВЕРКА ПО AsmL

О.Р. Набиуллин,

аспирант, Нижегородский филиал ГУ ВШЭ,
nabiulin@tecomgroup.ru

Э.А. Бабкин,

доктор PhD, профессор, заведующий кафедрой информационных систем и технологий,
Нижегородский филиал ГУ ВШЭ,
babkin@hse.nnov.ru

Целью данной статьи является описание подхода связанного с использованием формализма Абстрактных Машин Состояний (ASM) и среды SpecExplorer, разработанной в Microsoft Research.

Введение

С усложнением программного обеспечения усложняются и подходы к разработке и анализу корректности создаваемых продуктов. Одним из новых подходов является использование моделей не только для анализа требований и архитектуры, но и для анализа непосредственно продукта. От статических моделей (UML, Booch и т.д.) происходит переход к исполняемым (executable) моделям. Целью данной статьи является описание подхода связанного с использованием формализма Абстрактных Машин Состояний (ASM) и среды SpecExplorer, разработанной в Microsoft Research [1].

Исполняемые модели не являются совершенно новым явлением, к таким моделям можно отнести сети Петри [2], модели на языке Promela [3], и многие другие. Абстрактных Машин Состояний (в дальнейшем просто ASM) это математический формализм, разработанный [4]. Одной из имплементаций этого формализма в виде языка программирования является AsmL [5; 6]. По синтаксису этот язык является некоторой смесью Python и Visual Basic. Это объектно-ориентированный, динамический слабо типизированный язык. Отдельно стоит отметить, что AsmL являясь .Net-языком имеет доступ к любому совместимому с .Net коду в т.ч. ко всем стандартным сервисам .Net Framework.

В первой части данной статьи описывается формализм абстрактных машин состояний и его реализация AsmL. Во второй части описывается методология предлагаемая средой SpecExplorer. Третья часть содержит пример AsmL-модели и её детальное описание. В четвертой части авторы приводят заключение и обзор альтернативных инструментов.

Исполняемые спецификации

Формализм абстрактных машин состояний, описанный в [4], является чисто математическим объектом. Однако структура ASM такова, что возможна его реализация в виде языка программирования. Существует несколько альтернативных реализаций, например [7] или [8]. Основной идеей ASM является хранение состояния объекта как некоторого набора значений ассоциированных с локациями (location)¹. Также ASM определяет механизм частичных обновлений, как основной способ эволюции системы, и математическую нотацию правил, по которым изменяются значения. Одним из ключевых моментов является одновременность выполнения действий.

Среда SpecExplorer, разработанная в Microsoft Research [1], содержит реализацию ASM в виде

¹ На данный момент не существует, канонического перевода терминов связанных с ASM. В спорных случаях английский вариант термина приводится в скобках.

языка AsmL [.Net]. С точки зрения синтаксиса AsmL представляет собой некий синтез языков Python, Delphi и Visual Basic, и при этом содержит высокоуровневые конструкции, близкие к математическим (определение множеств; квантификация выражений, pattern matching). С точки зрения парадигмы программирования AsmL является объектно-ориентированным языком, в тоже время допуская как функциональное, так и императивное программирование.

AsmL входит в семейство .Net-языков и имеет доступ к любому .Net-совместимому коду. Тем не менее предполагается, что модели написанные на AsmL минимально пользуются средствами не входящими в язык и его стандартную библиотеку, т.к. использование внешних API сокращает возможности по анализу модели. Более подробно анализ AsmL моделей будет рассмотрен ниже. Помимо AsmL, SpecExplorer предлагает методологию автоматической проверки соответствия модели проектируемой системе.

AsmL [6] это язык для создания спецификаций для программного обеспечения, основанный на формализме абстрактных машин состояний. Он используется для создания понятных (human-readable), исполняемых моделей работы системы таким образом, чтобы обеспечить минимальность и завершенность, с учетом заданного уровня абстракции. Спецификации на AsmL называются *исполняемыми спецификациями*.

Так же, как и традиционные спецификации, исполняемые спецификации являются описанием того, как работают компоненты программного обеспечения. В отличие от традиционных спецификаций исполняемые спецификации имеют единственное, недвусмысленное значение. Это значение проявляет себя в виде абстрактной машины состояний (ASM), математической модели изменения системы, состояния времени выполнения (runtime state).

Спецификации на AsmL могут быть запущены как программы, например, чтобы симулировать поведение некоторой системы или проверить поведение реализации на соответствие спецификации. Однако предполагается, что в отличие от обычных программ, исполняемые спецификации должны быть минимальными. Другими словами, несмотря на то, что они верны в описании, без упущений, всего, что является частью на выбранном уровне детализации, они эквивалентно верны в том оставлении неспецифицированным того, что осталось за рамками этого уровня детализации.

Таким образом, в отличие от программ, исполняемые спецификации ограничивают себя теми

ограничениями (constraints) и поведением, какие должны быть присущи всем корректным реализациям системы. Другими словами, исполняемые спецификации должны быть так же ясны относительно свободы данной конкретным реализациям системы, которая описывается, как и относительно ограничений.

В качестве примера, исполняемые спецификации не ограничивают порядок выполнения операций до тех пор, пока он не является важным, в то время как сегодняшние программы реализуют последовательный порядок выполнения операций, как решение уровня реализации (в отличие от уровня проектирования).

Это можно проследить на примере:

```
var A = [3, 10, 5, 7, 1]
indices = {0, 1, 2, 3, 4}

Main()
step until fixpoint
choose i in indices, j in indices
  where i < j and A(i) > A(j)
  A(i) := A(j)
  A(j) := A(i)
step
WriteLine(A) // prints [1, 3, 5, 7, 10]
```

Листинг 1 Сортировка

Эта исполняемая спецификация использует абстрактную машину состояний для сортировки через алгоритм обмена.

Машина выполняет последовательные шаги, которые меняют местами значения A , элементы которого обозначаются индексами i и j , такими что i меньше j , и значения $A(i)$ и $A(j)$ не совпадают с порядком сортировки. Это продолжается до тех пор, пока никаких дополнительных обновлений не возникает, то есть, до тех пор, пока последовательность не станет отсортированной. Как последний шаг, печатается отсортированная последовательность. Состояние машины на каждом этапе полностью характеризуется значением последовательности A на этом шаге.

Эта спецификация минимальна. Первый момент заключается в том, что выражение choose не говорит о том, как два индекса выбираются, только то, что выбранные значения должны быть различными индексами элементов, нарушающих порядок сортировки. Таким образом, многие алгоритмы сортировки, в том числе и quicksort, и пузырьковая сортировка, будут соответствовать тому, что мы указали.

Кроме того, наш пример не говорит о том, как происходит операция обмена значениями. Значения переменных изменяются, как атомарная транзакция. Это оставляет реализации решение о том, как осуществлять последовательный обмен, например, с использованием копирования во временный элемент.

Методология

Подход, предлагаемый средой SpecExplorer (Microsoft), состоит из следующих шагов:

1. Создание высокоуровневой модели системы с использованием языка AsmL (также возможно использование Spec#, но его рассмотрение выходит за рамки данной работы).
2. Исследование поведения модели путем прогонки по фиксированным (пользовательским) и произвольным (автоматическим) сценариям.
3. Автоматическая генерация сценариев с целью их дальнейшего воспроизведения.
4. Автоматическая проверка продукта на соответствие модели.

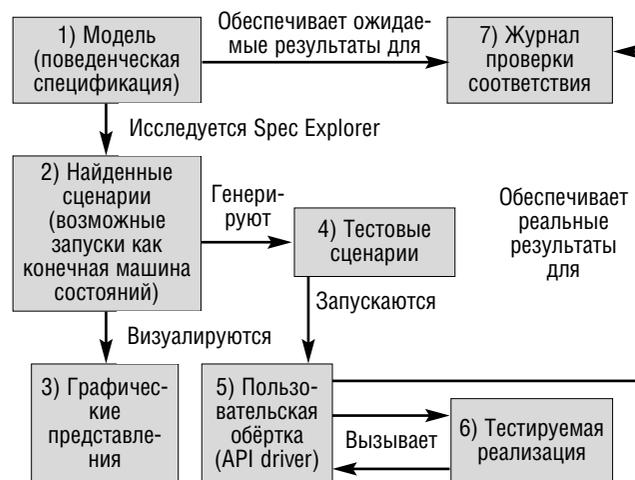


Рис. 1 Методология SpecExplorer

Сквозной пример

В качестве сквозного примера в данной статье будет рассматриваться следующая задача: есть несколько компонентов (программ), которые обладают определенными зависимостями. Требуется запустить их некоторое подмножество из них с учетом зависимостей. Предполагается, что старт компонента может завершиться неудачей. В этом случае необходимо пропустить все компоненты, которые явно или косвенно зависят от компонента, вызвавшего сбой. Исходная задача – сделать подобие windows services, но с учетом специфики предметной области.

Рассмотрим пример. Есть три компонента *A*, *B*, *C*. Исходные зависимости:

- A*: []
- B*: [*A*]
- C*: [*A*, *B*]

Предположим, что, пользователь пытается запустить компонент *C*. Система должна запустить последовательно компоненты *A*, *B* и *C*. В случае если все компоненты запущены, и пользователь пытается остановить компонент *B*, система должна сначала остановить компонент *C*, так как в его список зависимостей входит *B*, и только потом *B*. В реальной задаче компонентом являются как отдельные программы, так и совокупности программ, но с точки зрения предлагаемой модели данное различие несущественно. Выбранный уровень абстракции предполагает работу программы как взаимодействие с некоторым числом абстрактных компонент. В дальнейшем модель может быть уточнена с учетом этих различий. Механизм итеративного уточнения моделей описан в (Borger E., 2003).

Целью моделирования на первоначальном этапе является фиксация требований к системе в виде формальной спецификации и проверка результата на логическую непротиворечивость. Следующим этапом будет являться автоматическая проверка соответствия модели и реальной программы, что выходит за рамки данной статьи.

Модель

Рассмотрим модель проектируемой системы:

Класс *Comp* является абстракцией компонента системы. Его характеристиками являются: имя компонента, список зависимостей (имен), а также текущий статус (запущен/нет).

```
class Comp
var name as String
const deps as Seq of String
var running as Boolean = false
Start()
require not running
require forall d in deps holds FindComp(d).running
choose x in {true, false}
if x then
    running := x
else
    throw new Exception(name)
WriteLine(«Started « + name + « successfully»)
Stop()
require running
```

```

require forall d in FindDependants(name) holds not
FindComp(d).running
running := false
Check() as Boolean
return running
Depends(name as String) as Boolean
return name in deps

```

Листинг 2. Класс Comp

Язык AsmL поддерживает программирование по контракту, что выражается с помощью конструкций `require`. В случае если произойдет вызов метода `Comp.Stop` в тот момент, когда компонент еще не запущен, система сгенерирует ошибку. Такая ошибка означает, что модель внутренне не согласована, и, либо ограничение установлено неверно, либо существует такая последовательность действий, которая приводит к «незаконному» вызову `Stop`. Каждый компонент однозначно определяется своим именем, причем предполагается, что двух компонентов, обладающих одним именем нет. Состояние компонента определяется булевой переменной `running`, означающей запущен компонент в данный момент или нет. Допущение данной модели состоит в том что, предполагается если сбой возникает, то он возникает в момент запуска компонента. Удачно стартовавший компонент в дальнейшем работает стабильно. Как видно из кода функции `Comp.Start` при запуске компонента возможен сбой. В этом случае генерируется исключение. Предусловиями для запуска компонента являются утверждения:

1. Он не запущен.
2. Все компоненты, от которых он зависит, работают.

Известные компоненты описываются следующей последовательностью:

```

var components as Seq of Comp = [
new Comp(«core», []),
new Comp(«mv», [«core»]),
new Comp(«logo», [«core», «mv»]),
new Comp(«automation», [«core»]),
new Comp(«pxos», [«core»])]

```

Листинг 3. Список компонентов

Теперь определим высокоуровневые действия, применимые к системе:

```

[Action]
StartA(name as String)
let c = FindComp(name)
step
forall n in c.deps + [name]

```

```

SetWillStart(n, true)
step
Start(name)

```

Листинг 4. Запуск компонента

Запуск компонента — это двухфазная операция. На первой фазе мы помечаем компоненты, предназначенные к запуску, на втором непосредственно стартуем всю цепочку.

```

Start(name as String)
let c = FindComp(name)
if not c.running then
step
WriteLine(«Starting « + c.name)
step
try
step
StartSeq(c.deps)
step
if GetWillStart(name) then
c.Start()
else
WriteLine(«Skipped « + name + « (failed dependency)»)
catch
e as Exception:
Failed(name)

StartSeq(l as Seq of String)
require AllowedSequence(l)
var i = 0
step while i < Size(l)
Start(l(i))
i += 1

```

Листинг 5. Запуск компонента (продолжение)

Как видно из метода `Start(name as String)` на высоком уровне состояние компонента проверяется во время исполнения (runtime). Вкупе с требованием чтобы `Comp.Start` вызывался для не запущенных компонентов, обеспечивается проверка утверждения что ни один компонент не будет запущен дважды. В противном случае в процессе автоматического исследования (exploration) модели будет сгенерировано исключение. Причем исключение сгенерированное нарушением контракта всегда останавливает выполнение модели, в отличие от пользовательских исключений, таких как в методе `Comp.Start`.

Вспомогательная функция `AllowedSequence` проверяет упорядоченную последовательность компонентов на предмет непротиворечивости, т.е. отсутствие компонентов, которые могли бы быть запущены прежде своих зависимостей.

```
function AllowedSequence(l as Seq of String) as Boolean
return forall i in Indices(l), j in Indices(l)
  where i < j
  holds not (l(j) in GetDependencies(l(i)))
```

Листинг 6 Функция AllowedSequence

Как можно видеть из кода этой функции язык AsmL допускает формулирование условий практически в математической форме (для любых i и j из множества Indices(l), таких что $i < j$, выполняется условие).

Чтобы завершить тему запуска компонентов приведем два оставшихся высокоуровневых метода, которые являются отражением исходных требований к системе. Метод StartAll предназначен для запуска всех известных компонентов. Метод StartDefault предназначен для запуска некоторого подмножества компонентов, которое в реальных условиях будет задаваться конфигурационным файлом.

```
[Action]
StartAll()
require forall c in components holds not c.running
step
forall c in components
  SetWillStart(c.name, true)
step
var i = 0
step while i < Size(components)
  Start(components(i).name)
  i += 1
[Action]
StartDefault()
require forall c in components holds not c.running
step
forall ci in Indices(components) where start_ini(ci)
  SetWillStart(components(ci).name, true)
step
StartSeq([components(c).name | c in IndexRange(components) where start_ini(c)])
```

Листинг 7. Групповой запуск компонентов

Помимо запуска компонентов система должна предоставлять возможность остановки уже запущенных компонентов причем, в случае если останавливается компонент от которого зависят другие запущенные, остановить нужно всю цепочку.

```
StopSeq(l as Seq of String)
var i = 0
step while i < Size(l)
  let c = FindComp(l(i))
```

```
if c.running then
  Stop(l(i))
  i += 1
[Action]
Stop(name as String)
let c = FindComp(name)
require c.running
step
  StopSeq(FindDependants(name))
step
c.Stop()
```

Листинг 8. Остановка компонентов

В данной спецификации использовались вспомогательные функции, приведенные в Листинг 9.

```
var start_ini as Seq of Boolean = [
  true, // core
  true, // multiviewer
  false, // logo_inserter
  false, // automation
  true] // nxos

var will_start as Seq of Boolean = [
  false,
  false,
  false,
  false,
  false]

function FindComp(name as String) as Comp
return the c | c in components where c.name = name

function IsRunning(name as String) as Boolean
return FindComp(name).running

function GetDependencies(name as String) as Seq of String
return FindComp(name).deps

function FindDependants(name as String) as Seq of String
return [c.name | c in components
  where exists dep in c.deps
  where dep = name]

SetWillStart(name as String, val as Boolean)
let index = the ci | ci in Indices(components) where components(ci).name = name
will_start(index) := val

function GetWillStart(name as String) as Boolean
let index = the ci | ci in Indices(components) where components(ci).name = name
return will_start(index)
```

```
PrintComponents(l as Seq of Comp)
Write(«Components: «)
Write([c.name | c in l])
WriteLine(«.»)
```

Листинг 9. Вспомогательные функции. Исследование модели

При исследовании модели Spec Explorer был сконфигурирован следующим образом:

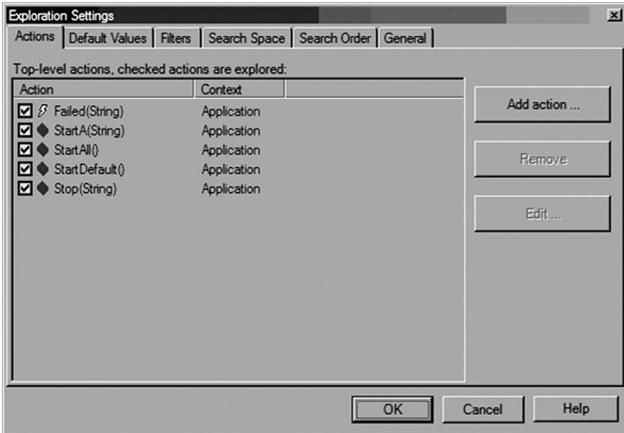


Рис. 2 Настройка действий

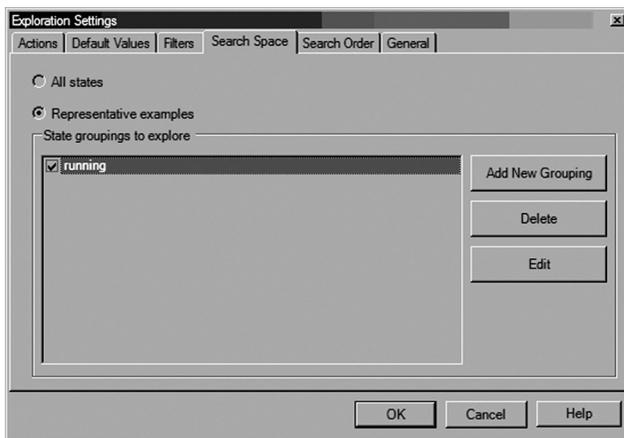


Рис. 3 Настройка области исследования

Группировка running определяется следующим AsmL выражением:

```
[c.name | c in components where c.running]
```

Листинг 10 Группировка состояний

Автоматическое исследование модели привело к следующему графу состояний (рис. 4):

Количество компонентов в данном графе сокращено до трех с целью повышения наглядности,

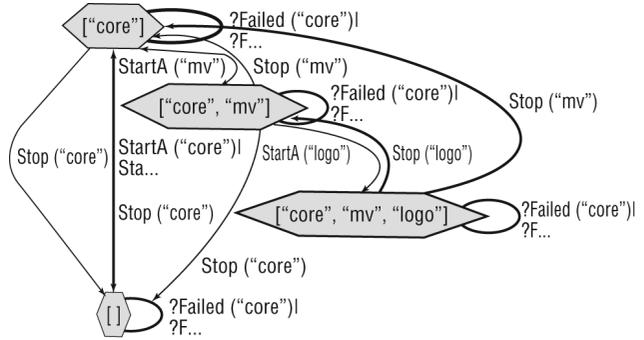


Рис. 4 Граф состояний

в реальной системе участвует большее количество компонентов и результирующий граф слишком велик, чтобы быть приведенным в иллюстрации. Несмотря на небольшое число компонентов граф, представленный на рисунке «Граф состояний», является срезом полного графа. Полный граф состояний, сгенерированный SpecExplorer приводится на рис. 5 «Полный граф состояний»

Заключение

Приведенные выше выкладки демонстрируют, как формализм ASM может быть применен для фиксации требований и проверки их на непротиворечивость. Построена модель программной системы и продемонстрированы способы автоматического исследования (exploration) этой модели. Выделены некоторые ключевые моменты подхода ASM, такие как одновременность вычисления выражений, последовательные обновления, кодирование состояния с помощью значений переменных. Также продемонстрированы некоторые возможности среды SpecExplorer, разработанной в Microsoft Research. Следующим шагом будет являться построение обертки (API Driver) для реальной системы с целью автоматической проверки соответствия модели системе.

Язык AsmL существует не только в виде части SpecExplorer, но также доступен в виде open-source компилятора, расположенного на ресурсе CodePlex [9].

Существуют также альтернативные реализации ASM такие как CoreASM [7] и его предшественник ASM-SL (реализованный в инструменте AsmWorkbench [8]), а также AsmGofor [10], реализующий семантику ASM как подмножество языка Haskell.

Исследования проводились при поддержке фонда РФФИ, грант 07-07-00058-а. ■

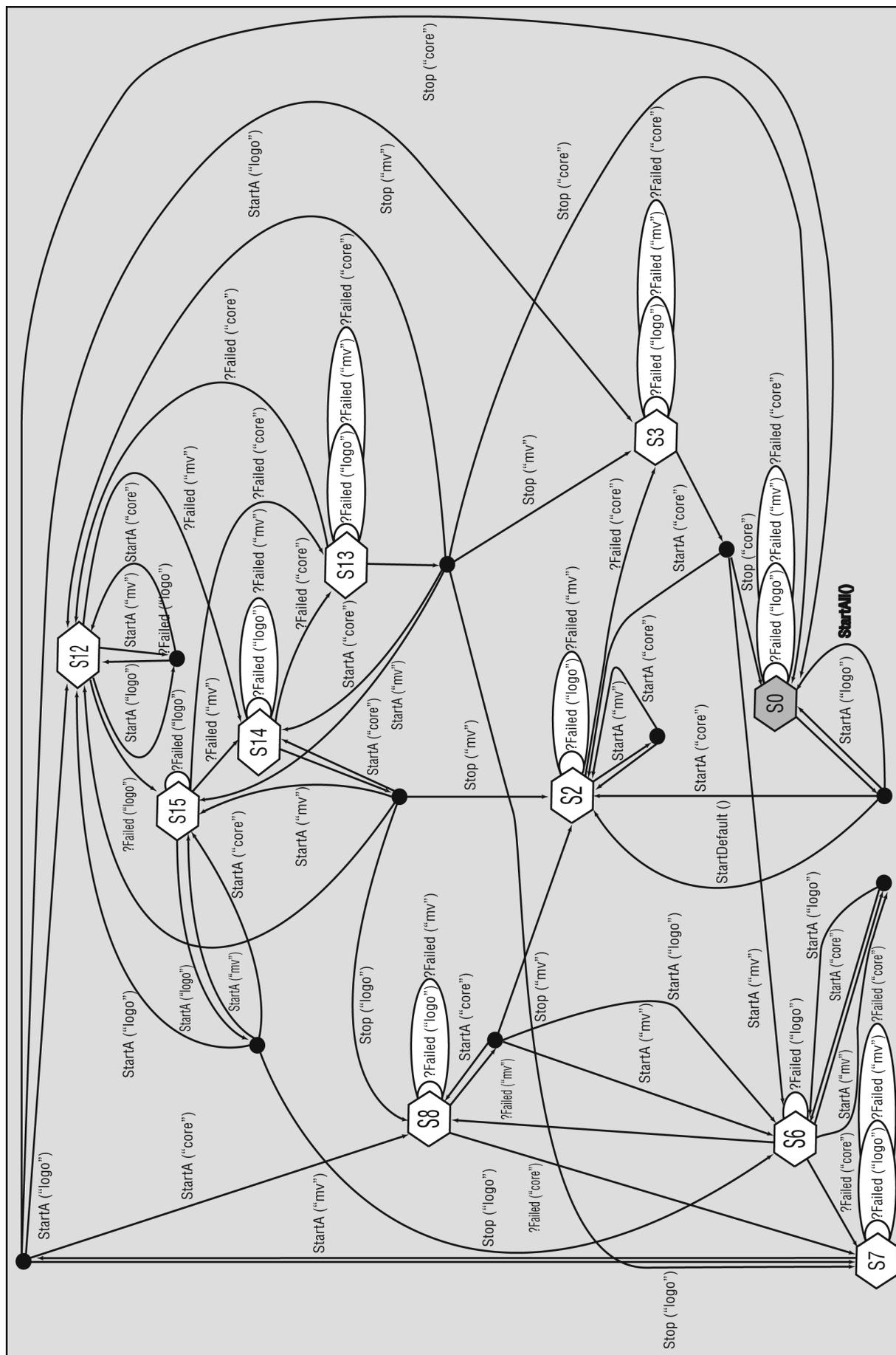


Рис. 5. Полный граф состояний

Литература

1. Foundations of Software Engineering. Online: <http://research.microsoft.com/fse/>. // 2008.
2. Cheung K, Cheung T, Chow K, A petri-net-based synthesis methodology for use-case-driven system design//Journal of Systems and Software, 2006, 79 Issue 6, C. [772–790].
3. Del Mar Gallardo M, Merino P, Pimentel E, A generalized semantics of PROMELA for abstract model checking//Formal Aspects of Computing, 2004, 16 Issue 3, C. [166–193]
4. Börger E, Stark R, Abstract State Machines. A Method for High-Level System Design and Analysis, Springer-Verlag, 2003, 448 p.
5. Gurevich Y, Rossman B, Schulte W. Semantic Essence of AsmL: Extended Abstract. In Springer Lecture Notes in Computer Science. Vol. 3188. 2004.
6. The AsmL webpage. Online: <http://research.microsoft.com/foundations/AsmL/>. // 2008.
7. Farahbod R, Gervasi V, Glasser U, Memon M, Design and Specification of the CoreASM Execution Engine // 2006, SFU-CMPT-TR-2006-09, [58]
8. Del Castillo G. The ASM Workbench. A Tool Environment for Computer-Aided Analysis and Validation of Abstract State Machine Models. Heinz Nixdorf Institut, Universität Paderborn. 2001.
9. AsmL Compiler. Online: <http://www.codeplex.com/AsmL/>. // 2008.
10. AsmGofer project page. Online: <http://www.tydo.de/Doktorarbeit/AsmGofer/index.html>. // 2008.

Яковлев А.А.

АГЕНТЫ МОДЕРНИЗАЦИИ

Второе издание

ISBN 978-5-7598-0460-4

432 с.

60x88/16

Переплёт

2007 г.



Экономическое развитие и процессы модернизации, в конечном счёте, зависят не от темпов инфляции, валютного курса, размеров Стабилизационного фонда и даже не от цен на нефть. Все эти макроэкономические параметры важны постольку, поскольку они могут повлиять на поведение экономических агентов. Их предпочтения, их желание (или нежелание) инвестировать в развитие бизнеса и повышать эффективность собственных компаний определяют конкурентоспособность собственной экономики и динамику уровня жизни населения.

В этой книге на примере четырех явлений, которые до недавнего времени были своего рода символами российской экономики («черный нал», бартер и неплатежи, массовые нарушения прав акционеров и «челночная» торговля), показано, почему российские предприятия в 1990-е гг. вели себя совсем не так, как того ожидали российское правительство и эксперты из МВФ и Всемирного банка. На обширном эмпирическом материале в книге объясняются мотивы действий экономических агентов, которые внешнему наблюдателю часто казались иррациональными, но на самом деле были вполне логичными в условиях сложившихся иррациональных правил игры.

Что изменилось в 2000-е гг.? Кто сегодня заинтересован в создании «правильных» стимулов для предприятий? Как можно ускорить модернизацию предприятий и тем самым способствовать модернизации российской экономики? Ответы на эти вопросы даются с учётом «дела «ЮКОСа»» и иных последних событий в социально-политической жизни России. Большое внимание в книге уделено возможностям применения в наших условиях опыта других стран, имеющих сопоставимый с Россией уровень развития институтов государства и рынка и сумевших добиться значимых успехов в своём социально-экономическом развитии.

Для широкого круга читателей, интересующихся проблемами экономической политики и управления предприятиями в переходной экономике.

ПРИМЕНЕНИЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ В РОССИИ – СОСТОЯНИЕ НА 2007г.

А. Борщев,

к.т.н., ген. директор ООО «Экс Джей Текнолоджис» (XJ Technologies)
andrei@xjtek.com

Рассматривается практическое применение имитационного моделирования у нас в стране, в частности: динамика спроса на ИМ со стороны бизнеса и государства за последние годы, спектр областей, где ИМ сейчас наиболее востребовано, факторы, сдерживающие рост. Кроме того, даётся обзор типичных сценариев (форм) внедрения имитационных моделей, а также технологий, используемых при их построении. Статья имеет обзорный характер и основана на данных компании XJ Technologies – разработчика продукта AnyLogic и лидера российского рынка ИМ.

О каком ИМ мы говорим

Этот обзор посвящён бизнес-приложениям ИМ, в частности в логистике, производстве, сфере услуг, энергетике, финансах, IT и т.д., в также приложениям, где конечным заказчиком является (или должно бы являться) государство: здравоохранение, оборона, чрезвычайные ситуации, социальная динамика. Мы не рассматриваем моделирование «физического» уровня («MATLAB'овское»), используемое при проектировании, скажем, систем управления или корпуса самолёта или корабля. Статья должна помочь всем участникам российского ИМ-сообщества определить вектор развития, идентифицировать ограничивающие факторы и выбрать направление усилий.

ИМ в России сегодня – обзор приложений

За последние 3–4 года картина рынка услуг, связанных с применением ИМ, изменилась кардинально. Если в 2003 г. (первый ИММОД) спрос на ИМ со стороны бизнеса и государства только робко намечался, а в 2005 г. (второй ИММОД) происходило интенсивное знакомство потенциальных заказчиков с подобными технологиями, то сейчас можно с уверенностью полагать, что ИМ уже обосновалось

в арсенале средств прогноза, анализа, и оптимизации. Свидетельство этому – множественные success stories в производстве, транспорте, обслуживании, маркетинге и т.д. Применяется ИМ пока не очень широко, но рост очевиден и в ближайшей перспективе он не прекратится: нынешнее состояние российского бизнеса и хозяйства вообще – это огромное пространство для улучшения, а, значит, и для применения наших с Вами умений и технологий.

В качестве иллюстрации к этим утверждениям приведём структуру оборота компании XJ Technologies как российского лидера в области ИМ (рис. 1).

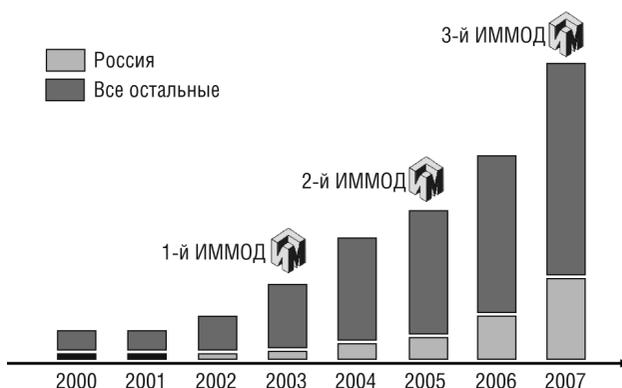


Рис. 1. Рост доли российских пользователей и заказчиков в обороте XJ Technologies

Интересной является структура спроса по областям применения. Диаграмма на *рис. 2* составлена по совокупным данным отделов консалтинга, продаж и технической поддержки XJ Technologies. Абсолютно полной она не является, но в большой мере соответствует действительности, так как спектр применений AnyLogic совпадает со спектром применений ИМ вообще. Картина спроса в России несколько отличается от общемировой, что, впрочем, вполне объяснимо, так как она отражает относительные уровни развития различных типов бизнеса и государственных структур. Области применения расположены сверху вниз по убыванию «градуса интереса».

Безусловным лидером по «осознанному спросу» и внедрениям ИМ является область логистики: перевозки, работа склада, политики закупок, и, шире, функционирование цепочек поставок. Объясняется это во-первых тем, что логистика в России переживает невероятный подъём, а во-вторых – сложным динамическим характером логистических процессов, обилием временных и причинно-следственных связей, размерностью задач. Невозможность оптимизировать логистические системы «на коленке» (= в Excel'e) настолько очевидна, что заставляет сами компании искать более продвинутые технологии.

Если брать производство, то ИМ наиболее активно интересуются в металлургии, нефтегазовой отрасли, производстве строительных материалов, пищевых продуктов, то есть опять же в наиболее «горячих» отраслях. Потребность в моделировании возникает при модернизации производств, то есть при необходимости оценить и сравнить ещё не реализованные варианты, а также при желании оптимизировать текущие процессы.

Анализ производительности компьютерных систем и сетей при помощи ИМ был известен у нас давно, так что наблюдающийся спрос на это сейчас со стороны телекоммуникационных компаний вполне предсказуем, хотя и не очень велик.

Более или менее массовый спрос ограничивается тремя перечисленными областями и, пожалуй, моделированием разного рода систем обслуживания и связанных с ними бизнес-процессов. Что касается таких традиционных (в мире) приложений как управление активами, портфелями проектов, моделирование потребительского рынка и конкуренции, управление персоналом в больших организациях, то здесь российские проекты с применением ИМ иницируются единичными «продвинутыми» энтузиастами из менеджмента компаний или банков. Успешные внедрения есть, но массового характера они не имеют.

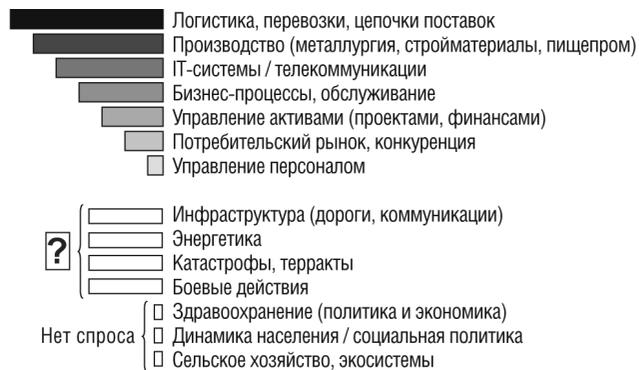


Рис. 2. Структура спроса на ИМ в России по областям применения (сверху вниз по убыванию градуса интереса)

Далее в диаграмме следует группа приложений, о которых мы имеем неполное представление в силу их несколько закрытого характера. Речь идёт о применениях ИМ, где заказчиком выступает государство: инфраструктурные проекты от городского до федерального уровня, моделирование внештатных ситуаций, требующих государственного вмешательства, военные применения ИМ. В России работы этого типа ведутся, их немало, количество их растёт, но оценить объём мы сейчас не можем.

Наконец, последняя группа – это области, где, в отличие от мировой практики, интерес к ИМ в России близок к нулю. Причём если в моделировании, скажем, различных политик в области социальной сферы и здравоохранения, в демографическом и эпидемиологическом моделировании наблюдается хоть какая-то активность, о проектах в области сельского хозяйства или экосистем неизвестно ничего. Повторюсь, я говорю о практических (то есть решающих чьи-то реальные проблемы) проектах, а не об академических упражнениях.

Как может выглядеть внедрённое ИМ-решение

Прежде всего заметим, что сама по себе имитационная модель очень редко является целью проекта. Для конечного пользователя важно решение конкретной задачи, например: оптимизировать количество персонала, определиться с планированием территории, управлением парком транспортных средств, политикой закупок и т.д. И естественно, модель одного и того же объекта будет выглядеть совершенно по-разному при разных целях моделирования. А иногда при анализе задачи становится ясно, что изначально планировавшееся ИМ вовсе не необходимо, и можно обойтись более простыми методами (например, линейным программированием). Это важно отследить в самом начале проекта.

Если же говорить о сценариях использования имитационных моделей в практике, то наиболее типичные из них таковы:

- ✧ модель полностью встроена в производственный или бизнес-процесс, запускается автоматически при выполнении соответствующих операций;
- ✧ модель оформлена в виде утилиты, регулярно запускающейся вручную при принятии оперативных решений;
- ✧ модель создаётся для оценки и сравнения вариантов предполагаемых изменений (модернизации), или для выработки оптимальной стратегии;
- ✧ модель создаётся ради динамической визуализации (демонстрации) работы проектируемого объекта — как дополнительный аргумент;
- ✧ модель оформлена в виде (распределённой) игры, использующейся для обучения сотрудников.

В первых трёх случаях модель используется в составе так называемой системы поддержки принятия решений (СППР).

Наиболее часто встречающаяся архитектура СППР показана на рис. 3. Собственно имитационная модель, как видно, — это только часть СППР, причём совсем необязательно самая большая и часто далеко не самая трудоёмкая. Роль оптимизатора — выбрать близкое к оптимальному решение из числа возможных, запуская модель много раз с различными параметрами. Эвристики «скрывают» от оптимизатора большое число прямых параметров модели, заменяя их небольшим числом «настроечных» параметров, с которыми оптимизатор должен легко справляться. Эвристики могут быть самым интеллектуальным модулем во всей системе, для их разработки часто приглашаются профессиональные математики. Модель часто конфигурируется из базы данных, например, читает оттуда наиболее свежую информацию. Для пользователей создаются специализированные интерфейсы с возможностью задавать параметры, запускать эксперименты, просматривать результаты в виде различных графиков, таблиц, печатать отчёты и т.д. Программирование пользовательских интерфейсов иногда занимает до 60% всего времени разработки.

XJ Technologies использовала подобные архитектуры при разработке СППР для оперативного управления парком транспортных средств компаний ЕВРОСИБ, Балтика, Русский Алюминий. База данных в этом случае содержит текущую дислокацию

вагонов (БД МПС), эвристики с оптимизатором могут быть настроены, например, на минимизацию порожнего пробега, пользователи получают конкретные рекомендации по управлению вагонами; то, что эти рекомендации выработаны при помощи ИМ, им может быть и неизвестно.

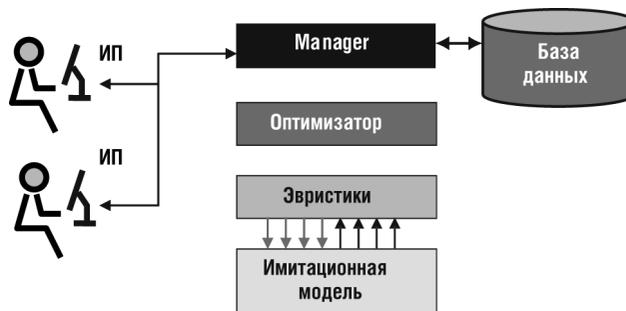


Рис. 3. Типичная архитектура системы поддержки принятия решений

Применяемые методы ИМ

Поскольку большая доля приложений ИМ приходится на оперативно-тактические задачи в сфере логистики, производства и систем обслуживания, основным используемым методом является дискретно-событийное моделирование либо в традиционном его понимании (то есть взгляд на систему как на процесс и задание процесса как последовательности операций и множества ресурсов), либо, реже, в агентном исполнении (взгляд на систему как на совокупность активных объектов и задание их индивидуальных поведений и взаимодействия). Даже в ряде стратегических задач (например: где построить контейнерный терминал?) требуемый уровень абстракции часто опускается до рассмотрения индивидуальных объектов (вагонов, единиц груза и т.п.) и не позволяет использовать методы системной динамики.

При моделировании потребительского рынка, в особенности динамичных рынков с высокой степенью конкуренции (например, сотовая связь, страхование, банковские услуги), где выбор потребителя зависит от его индивидуальных особенностей, истории, внутренней динамики, сети знакомств, внешних воздействий, агентное моделирование позволяет достичь максимальной адекватности. Построение таких агентных моделей облегчается сейчас тем, что требуемые для этого данные уже накоплены у большинства компаний в CRM-системах, а также в результатах опросов и тому подобных исследований.

В общем, при выборе между «процессным» дискретно-событийным моделированием и агентным

моделированием, надо иметь в виду, что агентные модели обычно более трудоёмки в построении и калибровке, поэтому если система хорошо описывается в виде последовательности операций над (пассивными) объектами, разделяющими ресурсы, нет смысла строить агентную модель.

Особые технологии «физического уровня» применяются при моделировании пешеходной и дорожной динамики, боевых действий, терактов, эвакуации. Здесь физическое пространство — это важный разделяемый ресурс и взаимодействие в нём объектов уже не может быть упрощено до уровня процесса «встал в очередь — обслужился — пошел к выходу». Вот в этой области в ряде случаев может быть оправдано построение специализированных «движков» и инструментов.

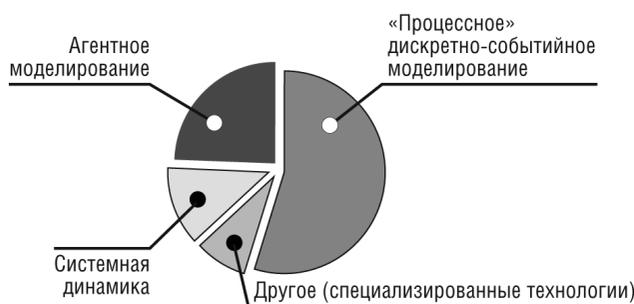


Рис. 4. Применяемые методы ИМ (по частоте использования)

В области российской системной динамики в последнее время наблюдается некоторая активность, в частности образована российское отделение (Russian Chapter) международного Общества системной динамики (International System Dynamics Society). Про практическое применение системной динамики в России известно мало, успех здесь будет зависеть от взаимодействия специалистов, владеющих методом, с людьми, принимающими стратегические решения — ведь система динамика в основном применима к долгосрочным процессам, интересным только со стратегических позиций.

Отдельно следует сказать о методах оптимизации, использующихся вместе с имитационными моделями. Поскольку результат выполнения имитационной модели обычно зависит от её параметров сложным нелинейным образом, оптимизатор так или иначе должен исследовать большую часть пространства параметров, чтобы не ограничиться локальными минимумами и максимумами. Из универсальных оптимизаторов такого рода в России часто используется встроенный в AnyLogic OptQuest американской фирмы OptTek. Мы часто облегчаем OptQuest'у задачу, дописывая эвристики, позволяющие вести поиск более эффективно за счёт знания

специфики проблемы. Иногда используются свои алгоритмы оптимизации, написанные под конкретное приложение.

Факторы, ограничивающие рост применения ИМ

Прежде, чем говорить о том, почему рост применений ограничен, мы хотим обратить Ваше внимание на следующий факт: основную долю в российской части оборота XJ Technologies составляет консалтинг и гораздо меньшую — продажа продукта, см. рис. 5. В зарубежной же части эти составляющие более сбалансированы с перевесом продаж продукта (то есть лицензий AnyLogic). Почему это происходит? Почему в России заказчики не берутся сами создавать имитационные модели?

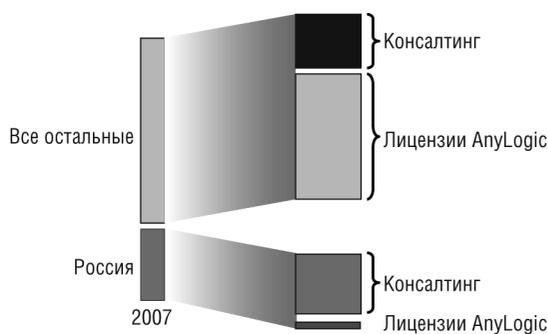


Рис. 5. Структура продаж XJ Technologies в России и за рубежом

Одним из основных препятствий роста практического применения ИМ в России была и остаётся нехватка квалифицированных кадров, что может звучать странно для страны с такими университетскими традициями и наконец-то прекратившимся оттоком мозгов. Действительно, ИМ в том или ином виде преподаётся во многих вузах, но преподаватели часто не могут (или не сильно хотят) вывести студента за рамки чисто «научных» или игрушечных проблем в пространство задач, востребованных в реальной жизни. Вместо этого преподаватели, студенты и аспиранты часто увлекаются, например, созданием собственных симуляторов, или, хуже того, разводят ложное (заумное и занудное) теоретизирование вокруг понятия «модель». Вопрос — зачем? Вокруг полно задач, решив — или просто попытавшись решить — которые, Вы не просто получите читаемую и цитируемую публикацию, но и выйдете на новый уровень понимания того, куда должны развиваться Ваши (прикладные!) исследования. Я уже не говорю про возможные материальные выгоды от сотрудничества с реальным заказчиком и уверенное трудоустройство выпускников.

В каждом городе, где есть университет, обязательно есть:

- ✧ склад;
- ✧ транспорт;
- ✧ супермаркет;
- ✧ банк;
- ✧ служба такси;
- ✧ дорожная инфраструктура;
- ✧ производство.

Господа преподаватели, Вы пробовали говорить с людьми, которые отвечают за ежедневное функционирование, ремонт, модернизацию, перспективное планирование этих элементов хозяйства? Попробуйте. Узнайте, что их волнует, где, по их мнению, возможна оптимизация, добейтесь от них хотя бы примерной постановки задачи, подумайте, поможет ли им ИМ. Если да – вот Вам, как минимум, готовая тема диплома. Пошлите студента делать измерения и собирать статистику на реальном объекте с секундомером, рулеткой, камерой – он сразу поймёт, что такое экспоненциальное распределение и почему длину очереди не всегда можно рассчитать по формуле Литтла.

Другой аспект кадровой проблемы следующий. Люди, в обязанности которых входит анализ и улучшение текущих процессов, а также планирование изменений, имеют обычно экономическое образование. В стандартный набор их умений входит владение Excel и, скажем, каким-нибудь статистическим пакетом, а динамическое (то есть имитационное) моделирование им незнакомо. Поэтому модели, которые так или иначе создаются в аналитических отделах компаний по большей части статические, основаны на формулах; время, причинность, смена состояний в них обычно не присутствует. При необходимости повысить адекватность модели такие специалисты иногда пытаются ввести элементы имитации во времени, оставаясь в рамках Excel-подобных инструментов, результатом чего является многократное усложнение моделей, их нечитаемость и, в конечном счёте, невозможность дальнейшей поддержки.

Корни проблемы опять-таки в образовании: ИМ у нас исторически было ближе к инженерно-техническим специальностям (в большой мере это так до сих пор), а экономические программы вузов, за редким исключением, его игнорируют.

Надо понимать, что университеты сами по себе достаточно инерционны, кроме того даже при не медленном изменении программ должно пройти 3–4 года, прежде, чем студенты, по ним учившиеся,

пойдут работать, и ещё некоторый срок, прежде чем они смогут принимать какие-то решения, так что кадровая проблема в любом случае быстро не решится. А вот над популяризацией ИМ, над разъяснением того, что оно может дать практически, можно и нужно работать уже сейчас. Это, кстати, одна из целей Российского Общества ИМ. Важны не просто выступления в своей академической «тусовке», а там, где Вас может услышать потенциально заинтересованный практик – то есть на отраслевых конференциях, и не разговоры про то, как сети Петри с раскрашенными фишками лучше, чем с нераскрашенными (да простят мне почитатели этой замечательной математической конструкции), а про то, на сколько процентов сократится время ожидания грузовика на въезде в контейнерный терминал, если принять такие-то и такие-то решения, подсказанные имитационной моделью.

К сожалению, российские специалисты практически не посещают основных конференций по ИМ (это Winter Simulation Conference, INFORMS Annual Meeting, IIE Annual Conference, International System Dynamics Conference), а это нужно, чтобы быть включённым в мировое ИМ-сообщество. Впрочем, материалы большинства этих конференций, к счастью, находятся в публичном доступе.

Заключение

Мы наблюдаем рост спроса на системы поддержки принятия решений, основанные на имитационных моделях со стороны как бизнеса, так и государства. Структура спроса повторяет общую по России картину сравнительного развития отраслей с явным доминированием всего, что связано с логистикой. Для эффективного удовлетворения этого спроса необходимы, прежде всего грамотные специалисты, с одной стороны владеющие различными методами ИМ, а с другой – умеющие быстро вникнуть в предметную область и понять суть проблемы. Нынешние студенты, изучающие ИМ, нуждаются в ориентации на решение практических задач, а студенты экономических и управленческих специальностей – хотя бы в простом знакомстве с ИМ. С другой стороны, необходимо знакомить сотрудников компаний и госорганизаций, ответственных за оптимизацию и вырабатывающих стратегические решения, с тем, какие проблемы позволяет решать ИМ. Недавно созданные Российское Общество ИМ и Russian Chapter of the International System Dynamics Society должны рассматривать эти задачи как приоритетные. ■

Microsoft открыла Центр инноваций в Государственном университете – Высшей школе экономики

28 октября Microsoft, мировой лидер в производстве программного обеспечения, предоставлении услуг и разработке интернет-технологий, открыл в Высшей школе экономики Центр инноваций (Microsoft Innovation Center, MIC), который будет предоставлять студентам ВУЗа и молодым специалистам возможности для разработки инновационных проектов.

Центр также будет содействовать комплексному развитию в вузе информационных технологий. Открытие Центра инноваций на базе нашего университета позволит расширить возможности ГУ-ВШЭ по обучению студентов современным технологиям Microsoft, а также, в перспективе, реализовывать научно-исследовательские проекты в области экономических и социальных наук при поддержке исследовательского подразделения корпорации Microsoft – Microsoft Research.

По мнению ректора ГУ-ВШЭ Ярослава Кузьмина, «инновационный центр Microsoft в Высшей школе экономики откроет новые возможности для развития интенсивных научных исследований в социально-экономической сфере, таких, например, как стратегическое моделирование деятельности субъектов рынка в турбулентных экономических условиях. Эти исследования дадут толчок для активного освоения студентами передовых инновационных идей и технологий, их предпринимательской активности».

Центр инноваций Microsoft в ГУ-ВШЭ позволит сформировать среду, в которой будут создаваться инновации во многом благодаря активному вовлечению в его деятельность студентов, преподавателей, ИТ-специалистов и лидеров ИТ-индустрии.

На пресс-конференции с докладами выступили: генеральный директор Microsoft в России Биргер Стен, проректор, директор института развития бизнес-информатики Виктор Никитин и директор департамента технологий и экономического развития Microsoft в России Олег Сютин.

«Открытие Центра инноваций Microsoft в Государственном университете – Высшей школе экономики стало новым этапом долгосрочного сотрудничества между нашей компанией и вузом, – сказал Биргер Стен,

генеральный директор Microsoft в России. — Мы уверены, что совокупность возможностей и ресурсов центра в сочетании с фундаментальной и прикладной научной базой ГУ-ВШЭ позволит создать благоприятные условия для развития инновационных проектов».

В рамках Центра инноваций Microsoft в ГУ-ВШЭ будет проводиться дополнительное обучение студентов технологиям Microsoft и принципам технологического предпринимательства. Для тех студентов, которые хотят создать собственный ИТ-бизнес, в Центре инноваций запланированы специальные тренинги по технологическому предпринимательству. Первым шагом к созданию собственной компании может стать и участие в конкурсе Imagine Cup, на котором студенты имеют возможность представить свои инновационные технологические проекты и, в случае успеха, получить признание и инвестиции. На базе Центра Инноваций организована подготовка команд к участию в Imagine Cup. В 2007 году Майкрософт подал идею и поддержал создание образовательной программы по предпринимательству в сфере разработки ПО — Start in Garage (www.startingarage.com). Образовательная программа дает возможность ее участникам получить знания об особенностях создания и развития ИТ-бизнеса в жестких рыночных условиях, пообщаться с коллегами по индустрии, увидеть и обсудить актуальные проблемы софтверного бизнеса, обменяться опытом, а также наладить отношения с потенциальными партнерами. Представители Центра Инноваций имеют возможность бесплатно пройти курс обучения по такой программе.

Особую роль в процессе обучения будут играть научные исследования в прикладной части информационно-коммуникационных технологий (ИКТ). Через Центр Инноваций Microsoft, а также интернатуру в лабораториях Microsoft Research в Редмонде и Кембридже, планируется создать международную сеть обмена передовыми знаниями, основанную на индивидуальном развитии студентов и аспирантов ГУ-ВШЭ и других вузов России. Важная роль в развитии Центра отводится интеграции студентов и преподавателей ГУ-ВШЭ в международное научное сообщество в области экономики и информационных технологий. Как перспективное направление развития сотрудничества, обсуждается возможность реализации силами студенческой лаборатории в рамках Центра инноваций Microsoft и при поддержке Microsoft Research научно-исследовательских проектов в области экономических и социальных наук.

Центры инноваций Microsoft работают в 60 странах мира. В России открыто уже 6 таких Центров на базе крупнейших вузов в Москве, Санкт-Петербурге, Нижнем Новгороде, Томске и Казани. Открытие Центров инноваций осуществляется в рамках Инициативы Microsoft по ускорению развития российской индустрии программного обеспечения.

**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.

Тематические рубрики журнала
«БИЗНЕС-ИНФОРМАТИКА»

№	Специальность номенклатуры	Рубрика
1	05.13.10	Математические модели социальных и экономических систем
2	05.13.11	Программная инженерия
3	05.13.17	Анализ данных и интеллектуальные системы
4	05.13.18	Математические методы и алгоритмы решения задач бизнес-информатики
5	05.13.18	Моделирование и анализ бизнес-процессов
6	05.25.05	Информационные системы и технологии в бизнесе
7	05.25.05, 05.13.11	Электронный бизнес
8	05.25.05, 05.13.17	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 – вверху 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 представляются 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».

Журнал «БИЗНЕС-ИНФОРМАТИКА»
Рекомендуемая структура статей

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 обозреваемой 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.