

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Advanced science

Научный журнал

№ 2 (2)

Киров

2012

Главный редактор:

Пугач В. Н.,

кандидат экономических наук,
ректор ФГБОУ ВПО «ВятГУ»

Заместитель главного редактора:

Литвинец С. Г.,

кандидат сельскохозяйственных наук, доцент,
проректор по НИИ ФГБОУ ВПО «ВятГУ»

Редакционная коллегия:

Частиков А. В.,

доктор технических наук, профессор,
декан ФПМТ ФГБОУ ВПО «ВятГУ»

Князьков В. С.,

доктор технических наук, профессор кафедры ЭВМ ФГБОУ ВПО «ВятГУ»

Шатров А. В.,

доктор физико-математических наук, профессор,
заведующий кафедрой ММЭ ФГБОУ ВПО «ВятГУ»

Черепанов В. В.,

доктор технических наук, профессор кафедры ЭПС ФГБОУ ВПО «ВятГУ»

Эфрос Е. И.,

доктор технических наук, профессор кафедры ТиГ ФГБОУ ВПО «ВятГУ»

Прозоров Д. Е.,

доктор технических наук, профессор кафедры РЭС ФГБОУ ВПО «ВятГУ»

Петров Е. П.,

доктор технических наук, профессор кафедры РЭС ФГБОУ ВПО «ВятГУ»

Решетников С. М.,

доктор технических наук, профессор, заведующий кафедрой физики ФГБОУ ВПО
«ВятГУ»

Тюкалов Ю. Я.,

доктор технических наук, профессор кафедры СК ФГБОУ ВПО «ВятГУ»

Красных А. А.,

доктор технических наук, профессор, заведующий кафедрой ЭиЭ ФГБОУ ВПО
«ВятГУ»

Ковалевский А. В.,

доктор технических наук, профессор, заведующий кафедрой ОХ ФГБОУ ВПО
«ВятГУ»

Флегентов И. В.,

доктор технических наук, профессор, заведующий кафедрой ПЭБ ФГБОУ ВПО
«ВятГУ»

Бывалов А. А.,

доктор биологических наук, профессор кафедры БТ ФГБОУ ВПО «ВятГУ»

Дармов И. В.,

доктор медицинских наук, профессор, заведующий кафедрой
микробиологии ФГБОУ ВПО «ВятГУ»

Скопина И. В.,

доктор экономических наук, профессор, заведующая кафедрой менеджмента и
маркетинга ФГБОУ ВПО «ВятГУ»

Палкина М. В.,

доктор экономических наук, профессор, заведующая кафедрой РЭУ ФГБОУ ВПО
«ВятГУ»

Печенкин А. А.,

доктор исторических наук, профессор, декан ГФ ФГБОУ ВПО «ВятГУ»

Ненашев М. И.,

доктор философских наук, профессор кафедры ФСП ФГБОУ ВПО «ВятГУ»

Стяжкина С. А.,

начальник редакционно-издательского отдела ФГБОУ ВПО «ВятГУ»

Редакторы: Куликова А. В., Мухачева М. С.

Верстка: Мухачева М. С.

«Вятский государственный университет» ПРИП ФГБОУ ВПО «ВятГУ» 610000,
Киров, ул. Московская, 36 Тел.: (8332) 64-23-56, <http://vyatsu.ru>

Свидетельство о регистрации средства массовой информации
(Министерство по делам печати, телерадиовещания и средств массовых
коммуникаций)

Эл. № ФС77-50605

Содержание

**Во второй номер журнала «Advanced science» вошли статьи и доклады с
VII Всероссийской научной конференции «Математическое
моделирование развивающейся экономики, экологии и биотехнологий»
(ЭКОМОД 2012)**

Поспелов И. Г., Оленев Н. Н., Шатров А. В.

Вступительная статья. VII Всероссийская научная конференция
«Математическое моделирование развивающейся экономики, экологии и
биотехнологий» (ЭКОМОД-2012)

Авдеенко А. М.

Модели адаптивного поведения на финансовом рынке

Апраушева Н. Н.

О достаточных условиях унимодальности двухкомпонентной гауссовой
смеси

Белянков А. Я.

Обобщение операции умножения матриц

Бродский Ю. И.

О концепции декларативно-поведенческого программирования
в имитационном моделировании сложных распределенных систем

Воротынцев А. В.

Графы работ для некоторых вычислимых моделей с рекурсиями

Дикусар В. В., Оленев Н. Н.

Оптимальное управление линейными системами с нерегулярными смешанными ограничениями и определение геометрии оптимальной траектории при помощи распределенных вычислений

Ефимов Д. А., Никулин В. Н.

Предсказание биологического состояния молекул исходя из их химических свойств

Козлов М. В., Костюк Ф. В., Сорокин С. В., Тюленев А. В.

Решение задачи коммивояжера методом целочисленного линейного программирования с последовательным исключением подциклов: описание и алгоритмическая реализация

Козлов М. В., Костюк Ф. В., Сорокин С. В., Тюленев А. В.

Решение задачи коммивояжера методом целочисленного линейного программирования с последовательным исключением подциклов: обоснование, тестовые испытания, применение

Кошкин Ю. Л.

Парадоксы системной регрессии

Лопаткина М. А., Зобнин К. В., Ростовцев В. С.

Многокритериальная система поддержки принятия решений

Макарова М. А.

Исследование равновесия в задаче заемщика и кредитора

Сорокин С. В.

Линейная зависимость плотности почвы и высоты рельефа в полупустыне
Северного Прикаспия и прогнозирование её свойств

Шатров А. В.

Новые применения асимптотических методов в краевых задачах
математической физики

Поспелов И. Г., Оленев Н. Н., Шатров А. В.

Pospelov I. G., Olenev N. N., Shatrov A. V.

**VII ВСЕРОССИЙСКАЯ НАУЧНАЯ КОНФЕРЕНЦИЯ
«МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ РАЗВИВАЮЩЕЙСЯ
ЭКОНОМИКИ, ЭКОЛОГИИ И БИОТЕХНОЛОГИЙ»
(ЭКОМОД-2012)**

**VII ALL RUSSIAN SCIENTIFICALLY CONFERENCE
«MATHEMATICAL MODELING OF EVOLUTIONARY ECONOMICS,
ECOLOGY AND BIOTECHNOLOGY» (EKOMOD-2012)**

Со 2 по 8 июля 2012 г. в г. Кирове на базе ФБГОУ ВПО «ВятГУ» была проведена VII Всероссийская научная конференция «Математическое моделирование развивающейся экономики, экологии и биотехнологий» (ЭКОМОД-2012). Конференция была организована Федеральным государственным бюджетным учреждением науки «Вычислительный центр им. А.А. Дородницына» Российской академии наук, Вятским государственным университетом и Московским физико-техническим институтом (Национальным исследовательским университетом) при поддержке Российского фонда фундаментальных исследований. Сопредседатели организационного комитета конференции: И. Г. Поспелов, член-корреспондент РАН, заведующий отделом ВЦ РАН и В. Н. Пугач, ректор ФБГОУ ВПО «ВятГУ». На конференции обсуждались теоретические аспекты системного анализа развивающейся экономики, исследования операций, экологии и биотехнологий, а также новые приложения этих актуальных направлений исследований. Данная конференция продолжает традицию летних школ академика АН СССР Н. Н. Моисеева, организованных им в 1960-1980-х годах и возрожденных

его учениками в 2006 году научных конференций серии ЭКОМОД – академиком РАН А. А. Петровым и членом-корреспондентом РАН И. Г. Поспеловым. К работе в конференции ЭКОМОД-2012 были привлечены ведущие ученые в области моделирования сложных систем и решений, которые обстоятельно осветили современное состояние научных достижений в этих областях. Молодые ученые, аспиранты и студенты, составляющие около половины от числа участников конференции, представили свои результаты ведущим специалистам на секционных заседаниях конференции. Были также организованы лекции на сопутствующих семинарах молодежной школы. В работе конференции участвовали более 90 человек из 12 регионов РФ, а также зарубежные участники из Украины, Вьетнама, Ирана, Монголии. Избранные доклады, представленные на конференции, рекомендованы Программным комитетом конференции к публикации в данном выпуске научного журнала Вятского государственного университета «Advanced Science». Проведено шесть пленарных заседаний, на которых заслушаны и обсуждены 18 докладов, часть которых представлены в данном номере журнала. На четырех секциях были заслушаны 47 докладов, посвященных широкому спектру вопросов математического моделирования экономических, экологических и биологических систем.

Секция I. «Системный анализ развивающейся экономики». Председатель: член-корр. РАН Поспелов И. Г. На секцию отбирались доклады, развивающее основанное акад. РАН А. А. Петровым научное направление «Системный анализ развивающейся экономики», особенностью которого является учет структурных и институциональных особенностей изучаемой экономической системы. Результатом исследований становятся динамические модели процессов воспроизводства и денежного обращения на национальном, региональном или отраслевом уровне. Эти модели должны в явном виде включать

описание процессов государственного регулирования экономики и давать оценки последствий реализации крупных проектов или экономических реформ. Развитие указанного направления в настоящее время привело к появлению весьма сложных и разнообразных как с концептуальной, так и с математической точки зрения моделей, а также к появлению новых методов анализа экономической статистики. Приветствовались также необычные подходы к математическому моделированию экономики, поскольку складывающаяся сейчас в мире экономическая ситуация явно требует привлечения свежих идей для своего осмысления. Было заслушано и обсуждено 11 докладов, посвященных новым теоретическим подходам к математическому моделированию экономики, развитию методов идентификации и верификации моделей и методов обработки статистических данных, а также опыту моделирования конкретных региональных и отраслевых экономических систем.

Секция II. «Методы оптимизации и параллельные вычисления». Председатели секции профессор Князьков В. С., доцент Оленёв Н. Н. Методы оптимизации, включая многокритериальную оптимизацию, являются мощным быстро развивающимся средством изучения математических моделей технических и социально-экономических систем. Применение параллельных вычислений открывает новые возможности анализа сложных математических моделей таких систем. На секции заслушаны и обсуждены 11 докладов, посвященных новым подходам к анализу сложных систем, которые основаны на использовании методов оптимизации, а также на использовании параллельных вычислений на кластерных и многоядерных архитектурах. Обсуждены последние достижения методов оптимизации, включая применение параллельных вычислительных технологий, позволяющих значительно сократить время поиска оптимального решения. Особое внимание уделено последним достижениям в области многокритериальной оптимизации нелинейных

систем, идентификации сложных нелинейных экономических моделей, содержащих большое число параметров, разработке систем распределенного имитационного моделирования сложных систем, приложению методов моделирования региональных экономических систем.

Секция III «Имитационное моделирование». Председатель секции доцент Бродский Ю. И. На секции заслушаны и обсуждены 12 докладов, так или иначе опирающихся на методы имитационного моделирования. Современное имитационное моделирование возникло на основе соединения методов математического моделирования с новыми информационными и компьютерными технологиями. Это соединение математических методов с современными компьютерными технологиями сделало практически реализуемыми и экономически целесообразными гораздо более сложные и подробные модели, чем это было возможно в рамках старой, докомпьютерной технологии. Основа метода имитационного моделирования – максимально точное воспроизведение поведения изучаемой сложной системы с помощью компьютерных технологий. В дальнейшем на такое воспроизведение поведения могут накладываться различные аналитические задачи такие как оптимизация, исследование устойчивости траекторий, исследование игровых взаимодействий, системы поддержки принятия решений.

Секция IV. «Информационно-математическое моделирование экономики, экологии и биотехнологий». Председатель секции профессор Шатров А. В. Информационные технологии важны в моделировании сложных систем. Инструментальные системы математического моделирования поддерживают разработку проблемно-ориентированных программных сред. Такие системы поддерживают процедуры управления, планирования, проектирования при разработке математических моделей в экономике, экологии и биотехнологиях. Применение информационно-

математических технологий в различных областях стимулирует появление новых задач, решаемых с помощью высокопроизводительных вычислений на современных кластерных системах. На двух заседаниях секции было заслушано 13 докладов 16 авторов, из них 4 студентов и 2 аспиранта, представлявших 4 субъекта РФ.

Кроме того, для студентов и аспирантов были организованы сопутствующие семинары: Н. Н. Оленев, к.ф.-м.н. (ВЦ РАН) провел семинары на тему «Параллельные вычисления в математическом моделировании экономики». Для участников конференции был организован круглый стол по мониторингу окружающей среды, на котором был представлен доклад директора ОАО «Кировгипрозема» Созинова О.Г. по мониторингу окружающей среды в Кировской области. Кроме доклада на круглом столе обсуждался широкий круг проблем математического моделирования эколого-экономических систем и применения моделей для анализа и прогнозирования экологических последствий экономической деятельности.

Информация об авторах

Data on authors

Поспелов Игорь Гермогенович, д.ф.-м.н., член-корреспондент РАН, заведующий отделом «Математическое моделирование экономических систем» ВЦ РАН, г. Москва.

Тел.+7(499) 783-33-28

Электронная почта: pospeli@ccas.ru

Pospelov Igor, Prof., PhD of Mathematics & Physics, Member-Corr. of RAS, Head of Department of Mathematical Modeling for Economical Systems CCAS, Moscow

Tel: +7(499) 783-33-28

E-mail: pospeli@ccas.ru

Оленёв Николай Николаевич, к.ф.-м.н., доцент, старший научный сотрудник ВЦ РАН, доцент кафедры математического моделирования в экономике ФГБОУ ВПО «ВятГУ».

Тел.+7(499) 783-33-28

Электронная почта: nolenev@mail.ru

Olenev Nicholas, docent, senior researcher at CCAS, Associate Prof. on Math Modeling in Economics FGEI HPE «VyatSU».

Tel: +7(499) 783-33-28

E-mail: nolenev@mail.ru

Шатров Анатолий Викторович – докт. физ.-мат. наук, заведующий кафедрой ММЭ ФГБОУ ВПО «ВятГУ».

Тел.: +79127096301

Электронная почта: avshatrov1@yandex.ru

Shatrov Anatoly – Prof., PhD of Mathematics & Physics, Head of Department of Mathematical Modeling in Economics FGEI HPE «VyatSU».

Tel.: +79127096301

E-mail: avshatrov1@yandex.ru

МОДЕЛИ АДАПТИВНОГО ПОВЕДЕНИЯ НА ФИНАНСОВОМ РЫНКЕ

MODELS OF ADAPTIVE BEHAVIOR IN FINANCIAL MARKET

Работа посвящена созданию мультивалютного советника на основе модели адаптивного поведения. Обсуждаются результаты тестирования модели и перспективы ее развития.

The work is dedicated to development of multicurrency adviser algorithm based on the model of adaptive behavior. Model test results and possibilities of its progress are discussed.

Ключевые слова: адаптивное поведение мультивалютный советник, тестирование

Keywords: adaptive behavior, multicurrency adviser, model test

Введение

В основе моделей адаптивного поведения лежит теория функциональных систем крупнейшего советского физиолога П. К. Анохина, разработанная в 1930-1970 гг.

Основная особенность теории функциональных систем заключается в том, что решение, принимаемое системой, направлено не на действие непосредственно (в нашем случае на вход-выход в короткую или длинную позицию), а на адаптивные результаты этих действий. В прикладном аспекте в этой области широко известны работы по теории обучения с подкреплением Sutton R, Barto A (MIT, 1998) [1] и др.

1. Модель адаптивного поведения

Модель адаптивного поведения основывается на результатах предшествующих работ по моделированию случайных или квазислучайных финансовых последовательностей на примере рынка Форекс [2-5].

Схема состоит из трех основных блоков: модель (M), контроллер 1 (C_1) и контроллер 2 (C_2) – рис. 1. На вход модели подается исходная информация – дискретная финансовая последовательность $X(t_n)$, на выходе модели – предсказание будущего поведения системы. В нашем случае тот или иной вариант реализации модели нелинейных стохастических вейвлетов [2-4] или элементарные динамические или статистические критерии.

Например, для модели вейвлетов Хаара – динамический критерий – это предсказание усредненной скорости изменения котировки валютной пары, статистический критерий – это оценка в вероятностном пространстве с заданным уровнем риска моментов переполучки или перепродажи. Как уже говорилось ранее, в качестве элементарных критериев могут использоваться стандартные модели – полосы Боллинджера, MACD и т. д.

Предсказания модели подаются на вход контроллера 1 совместно с исходной последовательностью. Цель контроллера 1 – сформировать обобщенный критерий на основе элементарных критериев, максимально приближенный к реально наблюдаемой тенденции изменения последовательности $X(t)$.

Здесь используются элемент булевой алгебры [5] либо система многослойных персептронов, оптимизируемая методом обратного распространения.

Адаптивное поведение реализуется в контроллере 2. Его цель – максимизировать сумму прибыли (плавающей либо зафиксированной).

Если L_n – решение о входе-выходе в момент времени t_n , принятое контроллером 2, то оптимизируется величина

$$r_n + br_{n+1} + b^2r_{n+2} + \dots$$

где r_n текущая (плавающая) прибыль (убыток) по открытым позициям, $0 < b < 1$ – дисконтный фактор (доллар сегодня важнее двух долларов завтра).

Пусть δ – среднеквадратичное колебание валютного курса в масштабе предсказания $t_{n+1} - t_n$. Не ограничивая общности, можно положить $\delta = 1$.

Если положить, что решение контроллера 2 формируется системой многослойных персептронов, то оптимизация его коэффициентов λ_n имеет вид

$$\lambda_{n+1} = \lambda_n + g \nabla_{\lambda} L_n (r_n \text{sign}(L_n) + bL_{n+1} - |L_n|) \quad (1)$$

где $\text{sign}(x) = -1, x < 0; \text{sign}(x) = 1; x \geq 0$ знаковая функция

Смысл этого выражения вполне очевиден: система получает прибыль, если знак предсказания $\text{sign}(L_n)$, совпал со знаком реального изменения цены r_n , оценка будущей прибыли δL_{n+1} , δ полагалась равной 1. Иными словами модель адаптирует свое поведение таким образом, чтобы максимизировать будущую (!) прибыль.

Кроме того, в модель адаптивного поведения заложены еще два эффекта – «долговременная память» и «креативность». Суть эффекта долговременной памяти в уменьшении величины g – скорости оптимизации вблизи качественного описания «будущего», определяемого долей совпадения предсказанного и реального поведения. Если эта доля велика, коэффициенты стабилизируются, т. е. изменяются мало (запоминаются или переходят в область долговременной памяти); если доля правильных предсказаний мала; то величина g растет, коэффициенты λ_n изменяются быстро (кратковременная память).

Эффект «креативности» реализуется следующим образом: направление входа-выхода в короткую или длинную позиции, предсказанные контроллером 2, реализуется случайным образом с вероятностью $1-\alpha$, где $\alpha \ll 1$. Величина α характеризует степень свободы, предоставленную алгоритму для принятия «неожиданных» решений. Значение α так же связано с качеством принятия решений контроллером 2, при высокой степени адаптации системы к принятию адекватных решений α мало, в условиях плохой адаптации и высокой степени неопределенности α увеличивается.

Наконец, допускается возможность двухконтурной схемы реализации адаптивного поведения: на один контур подаются результаты предсказания поведения системы в масштабе «быстрого» времени (например, решения уравнения Ито в масштабе области определения вейвлета Хаара), на другой – результаты анализа в «медленном» времени – статистическое описание (например, моменты переполки или перепродажи). Двухконтурная система, как показал анализ на реальных финансовых данных, дает более устойчивый результат.

Перечисленные особенности модели адаптивного поведения справедливы как для советника, работающего на одной валютной паре, так и для мультивалютных систем. Для мультивалютной системы существует ряд особенностей. Простейший вариант воспроизводит сильно связанную самосборку в модели, описанной в работах [4, 5] – рис.1а (другая валютная пара указана пунктиром). Решения принимаются независимо, а затем суммируются с весом, равным парной корреляционной функции различных валютных пар. Альтернативный вариант – результаты с контроллера C_1 подаются на один из контуров контроллера C_2 . На коротких последовательностях (8 валютных пар, две недели торгов) преимущество того или иного способа в настоящее время не выявлено.

2. Тестирование модели адаптивного поведения и реализация алгоритма в режиме реального времени

Модель адаптивного поведения тестировалась на валютной паре eur/usd с использованием тестера стратегий MT 4 с таймфреймом 1 мин на данных реальных торгов в период с апреля 2011 по март 2012. Случайным образом выбиралось 100 непересекающихся временных интервалов продолжительностью 48 часов. Торговля осуществлялась в автоматическом режиме без предварительной настройки с постоянным лотом 0.1 стандартной величины с ограничением на открытие не более двух ордеров и на величину свободных средств не менее 0.5 исходного депозита. Использовалась схема двухконтурной сети (рис. 1) и адаптивная подстройка в соответствии с уравнением (1). Был так же реализован алгоритм «долговременной памяти» и «креативности».

Оценивались доля прибыльных двухсуточных временных интервалов P , среднесуточная прибыльность, отнесенная к исходному депозиту s при плече 10, и нормированный среднеквадратичный разброс прибыльности по всем реализациям d . Соответствующие результаты приведены в табл. 1:

Таблица 1

Результаты тестирования модели адаптивного поведения;
100 реализаций с 04.2011 по 03.2012; eur/usd

P	s	d
0.67	0.0184	3.02

В среднем за сутки совершалось 8-12 сделок, причем средняя абсолютная величина прибыли интервала в 1.4-1.6 раз превышала средний убыток для неудачной реализации.

С учетом воспроизводимости результатов при большом количестве повторений, тестирование свидетельствует о фантастической эффективности алгоритма адаптивного поведения.

Анализ эффективности алгоритма в режиме реальных торгов проводился по старой схеме [5] при одновременной торговле восьмью валютными парами в соответствии со схемой 1а, т. е. реализацией варианта взаимодействия между валютными парами на основе быстрой горизонтальной самосборки.

В целом в реальном времени алгоритм показал хороший результат, во всяком случае для короткого шестинедельного интервала. Подробная информация представлена в [2-5]. Однако по-прежнему, в ряде случаев, наблюдался эффект, описанный в прежней работе, а именно: вход в короткие или длинные позиции осуществляется корректно, после этого для данной валютной пары система балансирует на небольших прибылях-убытках, недостаточных для закрытия позиции. Если далее начинается резкий рост или падение котировок, происходит аварийное закрытие позиции с существенным убытком. Для борьбы с этим эффектом в настоящее время применяется следующий алгоритм: позиция закрывается, если средняя прибыльность позиций, открытых и закрытых после нее, превышает плавающий убыток от данной позиции. Для оценки эффективности данного метода в модели адаптивного поведения в настоящее время не хватает статистических данных.

3. Перспективы развития модели адаптивного поведения

Модель адаптивного поведения имеет интересные перспективы для развития, в частности, в направлении усложнения топологии связей между контроллерами. Например, можно рассматривать топологию типа «Ленты Мебиуса», когда выход $J(t)$ подается на вход контролера C_2 (ответственного за адаптивное поведение) с противоположным знаком.

Предварительные оценки показывают возможность существования аттракторов с интересными свойствами. Кроме того, в настоящий момент адаптивное поведение рассчитано на максимизацию перспективной прибыли по открытым позициям альтернатива-минимизация плавающих убытков по тем же позициям. Так же представляет интерес включение в параметры модели адаптивного поведения временного масштаба p_1 – базы определения вейвлетов [2-5], задаваемого в настоящее время извне. Все это – предмет следующей работы.

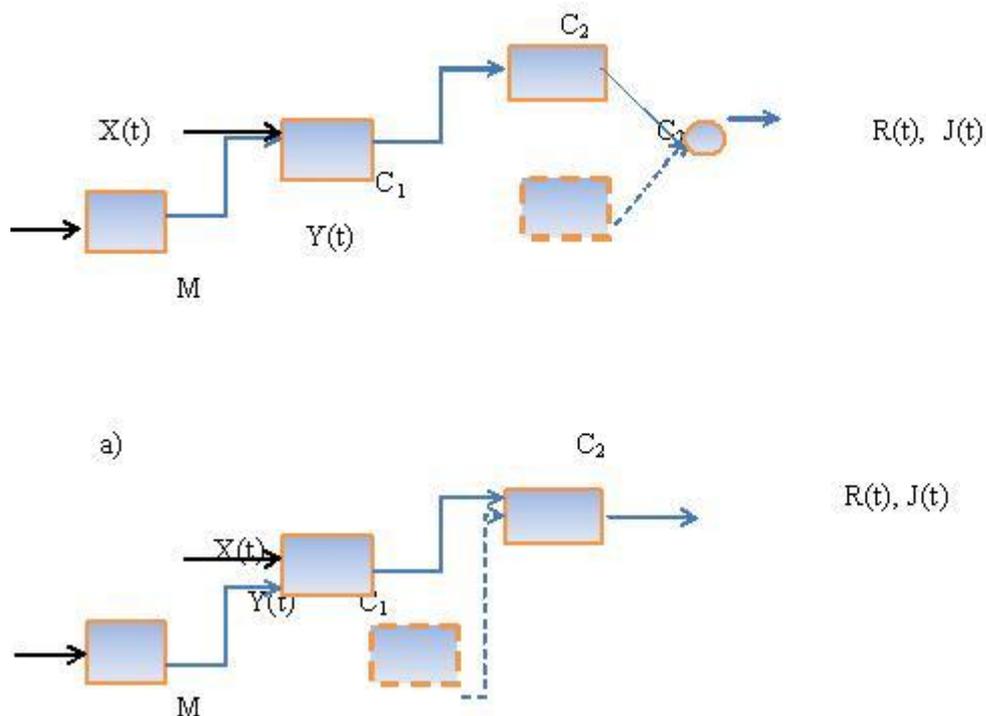


Рис.1. Блок-схема алгоритма адаптивного поведения:

M – модель, $C_{1,2}$ – контроллеры, Q – контур долговременной памяти, $X(t)$ – входной сигнал, $Y(t)$ – смоделированное будущее, $R(t)$ – решение модели адаптивного поведения, $J(t)$ – решение для мультивалютной модели.

Библиографический список:

1. Sutton R, Barto A. Reinforcement learning: An Introduction. Cambridge: MIT Press, 1998
2. Авдеенко А. М.. «Стохастический анализ сложных динамических систем. Рынок Forex». // Нелинейный мир. – 2010. – №6
3. Avdeenko A. M. Optimization of Financial Instrument Parcels in Stochastic Wavelet Model. 2010arXiv1007.5413A
4. Avdeenko A. M. Chaos structures. Multicurrency adviser on the basis of NSW model and social-financial nets 2011 arXiv:1106.4502
5. Avdeenko A. M. Multicurrency adviser based on the NSW model. Detailed description and perspectives 2011, arXiv: 1111.5726

Информация об авторах

Data on authors

Авдеенко Алексей Михайлович – д.ф.-м.н., профессор,
Национальный исследовательский технологический университет,
г. Москва

Тел. +7(495)230-46-86

E-mail: alexei-avdeenko@mail.ru

Avdeenko Alexey – PhD of Mathematics & Physics, professor of
Department of Mathematical Modeling National Researches Technological
University, Moscow

Tel: +7(495)230-46-86

E-mail: alexei-avdeenko@mail.ru

**О ДОСТАТОЧНЫХ УСЛОВИЯХ УНИМОДАЛЬНОСТИ
ДВУХКОМПОНЕНТНОЙ ГАУССОВОЙ СМЕСИ**

**ON A SUFFICIENT UNIMODALITY CONDITION OF TWO
COMPONENT GAUSSIAN MIXTURE**

Исследовалась двухкомпонентная гауссова смесь с различными дисперсиями. Для этой смеси получены два достаточных условия её унимодальности на основе принципа сжимающих отображений.

Two component Gaussian mixture with different variances is investigated. The two sufficient unimodality conditions of this mixture are obtained on the base of the contracted mapping principle.

Ключевые слова: гауссова смесь, различные дисперсии, сжатое отображение, унимодальность.

Key words: Gaussian mixture, different variances, contracted mapping, unimodality.

Введение

Гауссовы смеси нашли широкое применение в качестве универсального аппроксиматора в различных областях науки и практики: математическом моделировании, спектроскопии, распознавании образов, биологии, метеорологии, гидрологии и др. [1-6]. Полезность гауссовых смесей обусловлена такими их свойствами, как гладкость, идентифицируемость, разрешение, полнота [7-9]. Популярность этих

смесей вызывает необходимость решения таких задач, как определение их мод и предварительное оценивание их числа.

Предварительное оценивание числа мод смесей тесно связано с задачей отыскания условий их унимодальности и мультимодальности. В общем случае задача об унимодальности и мультимодальности гауссовых смесей не решена. В частном случае, для одномерных двухкомпонентных гауссовых смесей с равными дисперсиями, получено исчерпывающее решение этой задачи в [10, 11]. Достаточные условия унимодальности многомерных двухкомпонентных гауссовых смесей специального вида сформулированы в [12, 13].

В этой работе на основе принципа сжимающих отображений [14, 15] получено два достаточных условия унимодальности двухкомпонентной гауссовой смеси с различными дисперсиями. Одно из них совпадает с результатом, сформулированным в [16].

1. Постановка задачи и методы решения

Плотность вероятности исследуемой смеси $f(x)$ представима в виде:

$$f(x) = (2\pi)^{-\frac{1}{2}} \sum_{i=1}^2 \pi_i \sigma_i^{-1} \zeta_i(x), \quad (1.1a)$$

$$\zeta_i(x) = \exp\left[-(x - \mu_i)^2 (2\sigma_i^2)^{-1}\right], \quad (1.1b)$$

μ_i – математическое ожидание i -ой компоненты, σ_i^2 – её дисперсия, π_i – её априорная вероятность, $\pi_i \in (0,1)$, $\pi_1 + \pi_2 = 1$, $\mu_1 \neq \mu_2$, $\sigma_1^2 \neq \sigma_2^2$, $x \in (-\infty, \infty)$. Параметры распределения смеси μ_1 , μ_2 , σ_1^2 , σ_2^2 , π_1 известны.

Задача состоит в отыскании достаточных условий унимодальности функции $f(x)$. По определению [17], мода плотности вероятности $f(x)$ – это точка её локального максимума, являющаяся корнем уравнения

$$f'_x(x)=0, \quad (1.2)$$

$$f'_x(x)=(2\pi)^{-\frac{1}{2}}\sum_{i=1}^2\alpha_i(\mu_i-x)\zeta_i(x), \quad (1.3)$$

где $\zeta_i(x)$ выражается формулой (1.1b),

$$\alpha_i = \pi_i\sigma_i^{-3}, \quad i = 1, 2. \quad (1.4)$$

Тогда уравнение (1.2) принимает вид:

$$\sum_{i=1}^2\alpha_i(\mu_i-x)\zeta_i(x)=0. \quad (1.5)$$

Для определённости положим

$$\mu_1 < \mu_2. \quad (1.6)$$

и докажем теорему.

Теорема 1. Все корни уравнения (1.5) лежат в интервале (μ_1, μ_2) . Если это уравнение имеет единственный корень, то он является модой функции $f(x)$.

Доказательство. В силу выражений (1.3), (1.4), (1.1b) и (1.6) имеем

$$f'_x(\mu_1) > 0, \quad f'_x(\mu_2) < 0. \quad (1.7)$$

По теореме Больцано-Коши непрерывная функция $f'_x(x)$ имеет хотя бы один корень $\tilde{x} \in [\mu_1, \mu_2]$, $f'_x(\tilde{x})=0$. Но так как $f'_x(x) > 0$ при всех $x \in (-\infty, \mu_1]$ и $f'_x(x) < 0$ при всех $x \in [\mu_2, \infty)$, то все корни уравнения (1.5) лежат в интервале (μ_1, μ_2) . Если уравнение (1.5) имеет один корень, то в силу неравенств (1.7) он является модой функции $f(x)$.

Если число корней уравнения (1.5) $s = 3$ и все действительные корни различны, то гладкая функция $f(x)$ бимодальна. Если для числа действительных корней имеет место неравенство $1 \leq s \leq 2$, то функция $f(x)$ унимодальна, в случае $s = 2$ функция $f(x)$ имеет одну моду и одну вырожденную точку перегиба.

Для решения поставленной задачи от уравнения (1.5) перейдём к уравнению, ему равносильному

$$x = \varphi(x), \quad (1.8a)$$

$$\varphi(x) = \left(\sum_{i=1}^2 \alpha_i \mu_i \zeta_i(x) \right) \left(\sum_{i=1}^2 \alpha_i \zeta_i(x) \right)^{-1}. \quad (1.8b)$$

Функция $\varphi(x)$ имеет только одну неподвижную точку (НТ) на отрезке $[\mu_1, \mu_2]$, если она является сжимающей (сжатой), т. е. для всех $x \in [\mu_1, \mu_2]$, должны выполняться следующие условия [14, 15]:

$$\mu_1 \leq \varphi(x) \leq \mu_2, \quad (1.9)$$

$$|\varphi(x_2) - \varphi(x_1)| \leq \alpha |x_2 - x_1|, \quad 0 < \alpha < 1, \quad x_1, x_2 \in [\mu_1, \mu_2]. \quad (1.10)$$

Следовательно, условия (1.9), (1.10) являются достаточными для унимодальности функции $\varphi(x)$.

2. Основные теоретические результаты

Условие (1.10) выполняется для всех $x \in [\mu_1, \mu_2]$, если

$$|\varphi'_x(x)| < 1. \quad (2.1)$$

Для удобства введём обозначение

$$\psi_i(x) = \alpha_i \zeta_i(x), \quad (2.2)$$

тогда формула (1.8b) принимает вид

$$\varphi(x) = \left(\sum_{i=1}^2 \mu_i \psi_i(x) \right) \left(\sum_{i=1}^2 \psi_i(x) \right)^{-1}. \quad (2.3)$$

Продифференцировав функцию $\varphi(x)$ и проведя элементарные алгебраические операции, получаем:

$$\varphi'_x(x) = (\mu_2 - \mu_1) (\psi'_2 \psi_1 - \psi'_1 \psi_2) \left(\sum_{i=1}^2 \psi_i \right)^{-2}. \quad (2.4)$$

После возведения в квадрат последнего сомножителя в правой части выражения (2.4) и деления числителя и знаменателя на $\psi_1 \psi_2$ имеем:

$$\varphi'_x(x) = \frac{(\mu_2 - \mu_1)(\psi'_2\psi_2^{-1} - \psi'_1\psi_1^{-1})}{\psi_1\psi_2^{-1} + \psi_2\psi_1^{-1} + 2}. \quad (2.5)$$

Так как производная $\psi'_{ix}(x)$ выражается формулой

$$\psi'_{ix}(x) = \sigma_i^{-2}(\mu_i - x)\psi_i(x), \quad i=1, 2, \quad (2.6)$$

то

$$\psi'_2\psi_2^{-1} = \sigma_i^{-2}(\mu_i - x), \quad i=1, 2. \quad (2.7)$$

Подставив выражения (2.7) в формулу (2.5) получим

$$\varphi'_x(x) = \frac{(\mu_2 - \mu_1)[x(\sigma_2^2 - \sigma_1^2) + \mu_2\sigma_1^2 - \mu_1\sigma_2^2]}{\sigma_1^2\sigma_2^2(\psi_1\psi_2^{-1} + \psi_2\psi_1^{-1} + 2)}. \quad (2.8)$$

Найдём нижнюю и верхнюю границы производной $\varphi'_x(x)$ для $x \in [\mu_1, \mu_2]$. Знак $\varphi'_x(x)$ определяется знаком выражения в квадратных скобках формулы (2.8), которое обозначим через $y(x)$.

$$y(x) = x(\sigma_2^2 - \sigma_1^2) + \mu_2\sigma_1^2 - \mu_1\sigma_2^2. \quad (2.9)$$

При исследовании рассмотрим отдельно два случая:

$$1) \sigma_1 > \sigma_2, \quad (2.10)$$

$$2) \sigma_1 < \sigma_2. \quad (2.11)$$

Так как $\mu_1 \leq x \leq \mu_2$, то при выполнении неравенства (2.10) имеем

$$\begin{aligned} \mu_2(\sigma_2^2 - \sigma_1^2) < x(\sigma_2^2 - \sigma_1^2) < \mu_1(\sigma_2^2 - \sigma_1^2), \\ (\mu_2 - \mu_1)\sigma_2^2 < y(x) < (\mu_2 - \mu_1)\sigma_1^2. \end{aligned} \quad (2.12)$$

Для любого действительного числа $a > 0$, $a \neq 1$, имеет место неравенство [18]:

$$a + a^{-1} > 2. \quad (2.13)$$

Тогда на основании выражений (2.8), (2.9), (2.12) и (2.13) для всех $x \in [\mu_1, \mu_2]$ имеем:

$$\frac{(\mu_2 - \mu_1)^2}{\sigma_1^2\tau(x)} < \varphi'_x(x) < \frac{(\mu_2 - \mu_1)^2}{4\sigma_2^2}, \quad (2.14)$$

$$\tau(x) = \psi_1 \psi_2^{-1} + \psi_1^{-1} \psi_2 + 2. \quad (2.15)$$

Из неравенств (2.14) и (2.15) для $x \in [\mu_1, \mu_2]$ следует, что $\varphi'_x(x) > 0$ и что $\varphi'_x(x) < 1$, если

$$(\mu_2 - \mu_1)^2 \sigma_2^{-2} \leq 4, \quad \sigma_2 = \min(\sigma_1, \sigma_2). \quad (2.16)$$

Во втором случае, при выполнении неравенства (2.11) аналогично рассуждая, приходим к выводу: для всех $x \in [\mu_1, \mu_2]$ $\varphi'_x(x) > 0$ и $\varphi'_x(x) < 1$, если

$$(\mu_2 - \mu_1)^2 \sigma_1^{-2} \leq 4, \quad \sigma_1 = \min(\sigma_1, \sigma_2). \quad (2.17)$$

Итак, доказана теорема.

Теорема 2. При всех $x \in [\mu_1, \mu_2]$ производная $\varphi'_x(x) > 0$ и если

$$(\mu_2 - \mu_1)^2 \sigma_{\min}^{-2} \leq 4, \quad \sigma_{\min} = \min(\sigma_1, \sigma_2). \quad (2.18)$$

то $\varphi'_x(x) < 1$.

Для доказательства (1.9) в силу теоремы 2 достаточно убедиться, что

$$\varphi(\mu_1) > \mu_1, \quad \varphi(\mu_2) < \mu_2. \quad (2.19)$$

Из выражений (1.8b), (1.1b) и (1.6) имеем неравенства

$$\varphi(\mu_1) - \mu_1 = \alpha_2 \zeta_2(\mu_1) (\mu_2 - \mu_1) (\alpha_1 + \alpha_2 \zeta_2(\mu_1))^{-1} > 0,$$

$$\varphi(\mu_2) - \mu_2 = \alpha_1 \zeta_1(\mu_2) (\mu_1 - \mu_2) (\alpha_2 + \alpha_1 \zeta_1(\mu_2))^{-1} < 0,$$

равносильные неравенствам в (2.19).

Следовательно, доказана теорема.

Теорема 3. Функция $\varphi(x)$ в (1.8a) является сжимающей на отрезке $[\mu_1, \mu_2]$, если имеют место неравенства (2.18) и (2.19).

Так как уравнения (1.5) и (1.8a) равносильны, то в силу теорем 1 и 3 имеем следствие.

Теорема 4. Плотность вероятности смеси $f(x)$ унимодальна, если выполняются условия (2.18).

Если неравенство в (2.18) не выполняется, то исследуемая смесь может быть как унимодальной, так и бимодальной. Установим условия унимодальности смеси в этой ситуации.

Из доказанного выше для всех $x \in [\mu_1, \mu_2]$ имеем:

$$0 < \varphi'_x(x) < \rho_{\max}^2 (\tau(x))^{-1}, \quad (2.21)$$

$$\rho_{\max}^2 = (\mu_2 - \mu_1)^2 \sigma_{\min}^{-2}, \quad \sigma_{\min} = \min(\sigma_1, \sigma_2), \quad (2.22)$$

$\tau(x)$ имеет выражение (2.15).

Уточним верхнюю границу для $\tau(x)^{-1}$. Первое слагаемое правой части формулы (2.15) обозначим через $\tau_1(x)$, а второе — через $\tau_2(x)$. Тогда на основании выражений (2.2), (1.1b), (1.4), (1.6) для всех $x \in [\mu_1, \mu_2]$ имеем:

$$\begin{aligned} \tau_1(x) &= \psi_1 \psi_2^{-1} = \beta e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} e^{\frac{(x-\mu_2)^2}{2\sigma_2^2}} > \beta e^{-\frac{(\mu_2-\mu_1)^2}{2\sigma_1^2}} = \beta e^{-\frac{\rho_1^2}{2}}, \\ \tau_2(x) &= \psi_1^{-1} \psi_2 = \beta^{-1} e^{\frac{(x-\mu_1)^2}{2\sigma_1^2}} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} > \beta^{-1} e^{-\frac{(\mu_1-\mu_2)^2}{2\sigma_2^2}} = \beta^{-1} e^{-\frac{\rho_2^2}{2}}, \end{aligned}$$

где

$$\beta = (\pi_1 \sigma_2^3) (\pi_2 \sigma_1^3)^{-1}, \quad (2.23)$$

$$\rho_1^2 = (\mu_2 - \mu_1)^2 \sigma_1^2, \quad \rho_2^2 = (\mu_2 - \mu_1)^2 \sigma_2^2. \quad (2.24)$$

Тогда

$$\begin{aligned} \tau(x) &> \beta e^{-\frac{\rho_1^2}{2}} + \beta^{-1} e^{-\frac{\rho_2^2}{2}} + 2, \\ \tau^{-1}(x) &< \left(\beta e^{-\frac{\rho_1^2}{2}} + \beta^{-1} e^{-\frac{\rho_2^2}{2}} + 2 \right)^{-1}. \end{aligned} \quad (2.25)$$

Из выражений (2.21) и (2.25) следует, что $\varphi'_x(x) < 1$, если

$$\rho_{\max}^2 - 2 < \beta e^{-\frac{\rho_1^2}{2}} + \beta^{-1} e^{-\frac{\rho_2^2}{2}}. \quad (2.26)$$

Из наших рассуждений следует теорема.

Теорема 5. Функция $\varphi(x)$ в (1.18а) является сжимающей на отрезке $[\mu_1, \mu_2]$, если имеют место неравенство (2.26), в котором ρ_{\max}^2 , β , ρ_1^2 , ρ_2^2 выражаются формулами (2.22), (2.23) и (2.24) соответственно.

Так как уравнения (1.5) и (1.8а) равносильны, то в силу теорем 1 и 5 имеем следствие.

Теорема 6. Плотность вероятности $f(x)$ унимодальна, если имеет место неравенство (2.26), в котором параметры ρ_{\max}^2 , β , ρ_1^2 , ρ_2^2 выражаются формулами (2.22), (2.23) и (2.24) соответственно.

Следствие 6.1. Плотность вероятности $f(x)$ унимодальна, если $\rho_{\max}^2 - 2$ меньше одного из слагаемых в правой части (2.26).

Заключение

Два достаточных условия унимодальности исследуемой смеси (2.18) и (2.26) получены на основе свойств сжимающей функции $\varphi(x)$ в уравнении (1.8а). Если ни одно из этих условий не выполняется, то смесь может быть как унимодальной, так и бимодальной. И задача состоит в отыскании уравнения границы унимодальности и бимодальности семейства таких смесей. Уравнение такой границы получено в частном случае для $\sigma_1 = \sigma_2$ в [11].

Библиографический список:

1. Harris N. and Smith S. A. B. The sib-sib age of on set correlation among individuals suffering from a hereditary syndrome produced by more than one gene. // Annals of Eugenics. London, 1949. V. 14. Part 4. P. 309-318.
2. Eisenberger J. Genesis of Bimodal Distributions. // Technometrics. 1964. № 6. P. 357-364.

3. Волошин Г. Я., Бурлаков И. А., Косенкова С. Т. Статистические методы решения задач распознавания, основанные на аппроксимационном подходе. Ч. 1-2. – Владивосток: изд. Дальневост. отд-ния РАН, 1992. – 271 с.

4. Aprausheva N. N., Gorchach I. A., Zhelnin A. A., Sorokin S. V. An experiment on Automated Statistical Recognition of Clouds. // J. Computational Mathematics and Mathematical Physics. 1998. V. 38. № 10. P. 1715-1719.

5. Carreira-Perpiñán N. A. Mode-finding for mixture of Gaussian distributions. // IEEE Trans. On Pattern Analys. and Mach. Intell. 2000. 22. № 11. P. 1318-1323.

6. Kozhevnikova J., Shveikina V. and Domingues E. Modelling fluctuation of Caspian See levels using a mixed probability distribution. // J. of Flood Risk. 2012. № 5. P. 3-13.

7. Дороднов А. А. Ортонормированная система Гаусса. / Сборник аспирантских работ. Точные науки. – Казань: КГУ, 1969.

8. Henry Teicher. Identifiability of finite mixtures. // The annals of mathematical statistics. 1963. V. 34. P. 1265-1269.

9. Патрик Э. Основы теории распознавания образов. – М.: Советское радио, 1980.

10. Aprausheva N. N., and Sorokin S. V. On the Unimodality of a Simple Gaussian Mixture. // J. Computational Mathematics and Mathematical Physics, 2004. V. 44. №. 5. P. 785-793.

11. Aprausheva N. N., Sorokin S. V. The exact equation of a boundary of unimodal and bimodal domains of the simplest Gaussian Mixture. // 8th Open German-Russian Workshop. Pattern Recognition and Image Understanding. Nizhy Novgorod, The Russian Federation, 2011.

12. Aprausheva N. N., Sorokin S. V. On the Uni- and Bimodality of a Two-Component Gaussian Mixture. // ISSN 1054-6618. Pattern Recognition and Image Analysis, 2008, V. 18. № 4. P. 577-579.

13. Konstantilos A. C. Unimodality conditions for Gaussian sums // IEEE Transaction on Automatic Control, 1980, V. AC-25. № 4. P. 438-439.

14. Колмогоров А. Н., Фомин С. В. Элементы теории функций и функционального анализа. – М.: Наука, 1972.

15. Алимов Ш. А. Принципы сжатых отображений // Новое в жизни, науке, технике. Сер. Математика, кибернетика. – М.: Знание, 1983. № 5.

16. Behboodian J. On the Modes of a Mixture of Two Normal Distributions. // Technometrics. 1970. V. 12. № 1.

17. Крамер Г. Математические методы статистики. – М.: Мир, 1975.

18. Беккенбах Э., Беллман Р. Введение в неравенства. – М.: Мир, 1965. – 165 с.

Информация об авторах

Data on authors

Апраушева Нина Николаевна – канд. физ.-мат. наук, старший научный сотрудник отдела интеллектуальных систем «ВЦ РАН».

Тел.: (8499)135-40-98

E-mail: plat@ccas.ru

Aprausheva Nina Nikolaevna – Cand. Sc. (Physics, Mathematics) senior staff scientist on Department of Intelligent Systems «CC RAS».

Tel.: (8499)135-40-98

E-mail: plat@ccas.ru

ОБОБЩЕНИЕ ОПЕРАЦИИ УМНОЖЕНИЯ МАТРИЦ

GENERALIZATION OF THE OPERATION OF MATRIX MULTIPLICATION

Вводится семейство единообразно определяемых мультипликативных матричных операций, включающее в себя, в частности, обычные матрично-векторные операции умножения, операцию умножения матрицы/вектора на скаляр и тензорное произведение матриц/векторов. Рассматриваются критерий ассоциативности, операция транспонирования, некоторые подалгебры.

A family of uniformly defined multiplicative matrix operations is introduced, including, in particular, common operations of multiplication of matrices and vectors, multiplication of matrix/vector by scalar and tensor product of matrices/vectors. Criterion of associativity, transposition operation, certain sub algebras are considered.

Ключевые слова: матрично-векторное умножение, тензорное произведение, ассоциативность, подалгебра.

Keywords: matrix and vector multiplication, tensor product, associativity, sub algebra.

1. Мультииндексная индексация матричных элементов

В настоящей работе принимается технически удобное соглашение при индексации элемента M_{ν}^{μ} $m \times n$ -матрицы M полагать диапазоны

изменения индексов $0 \leq \mu < m$, $0 \leq \nu < n$ вместо обычных $1 \leq \mu \leq m$, $1 \leq \nu \leq n$. Скаляры, векторы-столбцы и векторы-строки рассматриваются как $m \times n$ -матрицы с $m = n = 1$, $n = 1$ и $m = 1$ соответственно.

Пусть $m = m_0 m_1 \dots m_{p-1}$, $n = n_0 n_1 \dots n_{q-1}$ с целыми положительными сомножителями. Тогда, очевидно, при условиях $0 \leq \mu_\pi < m_\pi$ ($0 \leq \pi < p$) и $0 \leq \mu < m$ формула

$$\mu = \mu_0 + m_0(\mu_1 + m_1(\dots(\mu_{p-1} + m_{p-1} \cdot 0)\dots))$$

ставит во взаимно-однозначное соответствие индекс μ и мультииндекс $(\mu_0, \mu_1, \dots, \mu_{p-1})$, $\mu \leftrightarrow (\mu_0, \mu_1, \dots, \mu_{p-1})$. Можно также сказать, что имеет место представление

$$\mu = (\mu_{p-1}, \dots, \mu_1, \mu_0) \text{base}(m_{p-1}, \dots, m_1, m_0)$$

в системе счисления с переменным основанием счисления: младшая цифра μ_0 по основанию m_0 , следующая цифра μ_1 по основанию m_1 , ..., старшая цифра μ_{p-1} по основанию m_{p-1} . Аналогично, $\nu \leftrightarrow (\nu_0, \nu_1, \dots, \nu_{q-1})$, если $0 \leq \nu_\kappa < n_\kappa$ ($0 \leq \kappa < q$), $0 \leq \nu < n$, и

$$\nu = \nu_0 + n_0(\nu_1 + n_1(\dots(\nu_{q-1} + n_{q-1} \cdot 0)\dots)),$$

что можно интерпретировать как представление

$$\nu = (\nu_{q-1}, \dots, \nu_1, \nu_0) \text{base}(n_{q-1}, \dots, n_1, n_0).$$

Таким образом, наряду с обычной индексацией возможна следующая мультииндексная индексация матричных элементов:

$$M_{\nu}^{\mu} = M_{(\nu_{q-1}, \dots, \nu_1, \nu_0) \text{base}(n_{q-1}, \dots, n_1, n_0)}^{(\mu_{p-1}, \dots, \mu_1, \mu_0) \text{base}(m_{p-1}, \dots, m_1, m_0)}.$$

Если $p = 1$, то $m = m_0$, соответствие $\mu \leftrightarrow (\mu_0, \mu_1, \dots, \mu_{p-1})$ принимает вид $\mu \leftrightarrow (\mu_0)$ так что $\mu = \mu_0$, $(\mu_0) \text{base}(m_0) = \mu_0 = \mu$. Аналогично упрощаются соответствие $\nu \leftrightarrow (\nu_0, \nu_1, \dots, \nu_{q-1})$ и представление ν в системе счисления с переменным основанием счисления при $q = 1$. Следовательно,

при $p = q = 1$ мультииндексная индексация сводится к простому переименованию μ в μ_0 , ν в ν_0 . В следующем разделе используется случай $p = q = 2$.

2. Определение обобщенной мультипликативной операции

Определение 1. Пусть даны $\nu_A \times h_A$ -матрица A , $\nu_B \times h_B$ -матрица B и целые положительные числа a, d . *Обобщенное произведение*

$$C = A \circ B = A \overset{a,d}{*} B$$

определено тогда и только тогда, когда одновременно h_A и ν_B делятся на a , ν_A и h_B делятся на d , так что числа $I_A = \nu_A / d$, $J_A = h_A / a$, $I_B = \nu_B / a$, $J_B = h_B / d$ – целые положительные. Имеются следующие четыре варианта определения матрицы C :

$$C_{(j_A, j_B) \text{ base}(J_A, J_B)}^{(i_A, i_B) \text{ base}(I_A, I_B)} = \sum_{0 \leq \alpha < a, 0 \leq \delta < d} A_{(j_A, \alpha) \text{ base}(J_A, a)}^{(i_A, \delta) \text{ base}(I_A, d)} B_{(\delta, j_B) \text{ base}(d, J_B)}^{(\alpha, i_B) \text{ base}(a, I_B)}, \quad (1)$$

$$C_{(j_A, j_B) \text{ base}(J_A, J_B)}^{(i_B, i_A) \text{ base}(I_B, I_A)} = \sum_{0 \leq \alpha < a, 0 \leq \delta < d} A_{(j_A, \alpha) \text{ base}(J_A, a)}^{(\delta, i_A) \text{ base}(d, I_A)} B_{(\delta, j_B) \text{ base}(d, J_B)}^{(i_B, \alpha) \text{ base}(I_B, a)}, \quad (2)$$

$$C_{(j_B, j_A) \text{ base}(J_B, J_A)}^{(i_A, i_B) \text{ base}(I_A, I_B)} = \sum_{0 \leq \alpha < a, 0 \leq \delta < d} A_{(\alpha, j_A) \text{ base}(a, J_A)}^{(i_A, \delta) \text{ base}(I_A, d)} B_{(j_B, \delta) \text{ base}(J_B, d)}^{(\alpha, i_B) \text{ base}(a, I_B)}, \quad (3)$$

$$C_{(j_B, j_A) \text{ base}(J_B, J_A)}^{(i_B, i_A) \text{ base}(I_B, I_A)} = \sum_{0 \leq \alpha < a, 0 \leq \delta < d} A_{(\alpha, j_A) \text{ base}(a, J_A)}^{(\delta, i_A) \text{ base}(d, I_A)} B_{(j_B, \delta) \text{ base}(J_B, d)}^{(i_B, \alpha) \text{ base}(I_B, a)}. \quad (4)$$

Из левых частей формул (1-4) видно, что результирующая матрица C имеет $\nu_C = I_A I_B = \nu_A \nu_B / (ad)$ строк и $h_C = J_A J_B = h_A h_B / (ad)$ столбцов.

Выбор обозначений « a » и « d » имеет простую наглядную мотивацию. В правых частях формул (1-4) воображаемая линия, соединяющая вхождения буквы a , повышается: $A_{\dots a \dots} B_{\dots a \dots}$, а аналогичная линия для вхождений буквы d понижается: $A_{\dots d \dots} B_{\dots d \dots}$. Преимуществом пары a, d (ascending/descending) перед иной приемлемой парой u, d

(up,upwards/down,downwards) стало наличие пары соответствующих букв a, d греческих букв α, δ .

Априорные предположения о природе элементов матриц A, B, C по возможности минимальны. Заведомо необходимо потребовать, чтобы все встречающиеся произведения элементов матриц существовали, чтобы встречающиеся произведения трех и более элементов матриц обладали свойством ассоциативности, и чтобы фигурирующая в правых частях (1-4) операция сложения обладала всеми привычными аксиоматическими свойствами. В качестве достаточно нетривиального содержательного примера можно указать на возможность применения Определения 1 к блочным матрицам, когда элементами матриц будут соответствующие блоки.

3. Важные частные случаи

Теорема 1. Пусть $h_A = v_B = a, d = 1$. Тогда все четыре формулы (1-4) дают $A \circ B = AB$, где в правой части обычное матричное умножение.

Теорема 2. Пусть $a = d = 1$, и пусть матрица A и/или матрица B есть скаляр, т.е. $v_A = h_A = 1$ и/или $v_B = h_B = 1$. Тогда все четыре формулы (1-4) дают $A \circ B = AB$, где в правой части обычное умножение скаляра и матрицы или двух скаляров.

Замечание. Дополним результаты этих теорем до исчерпывающего перечня таких наборов параметров a, d, v_A, h_A, v_B и h_B , при которых все четыре варианта (1-4) совпадают:

1) если одновременно $h_A = v_B = a$ и $h_B = v_A = d$, то $A \circ B$ есть в точности скаляр $\text{tr}(AB)$;

2) если $h_B = v_A = d$ и $a = 1$, то для получения $A \circ B$ нужно в обычном матричном произведении BA заменить все произведения элементов вида $B_j^i A_k^j$ на произведения в обратном порядке $A_k^j B_j^i$;

3) если $a = d = h_A = v_B = 1$, то $A \circ B$ есть обычное произведение AB матрицы-столбца A на матрицу-строку B с получением $v_A \times h_B$ -матрицы;

4) если $a = d = v_A = h_B = 1$, то для получения $A \circ B$ нужно в обычном произведении матрицы-столбца B на матрицу-строку A заменить все произведения вида $B_0^i A_k^0$ произведениями в обратном порядке $A_k^0 B_0^i$.

Теорема 3. Пусть $a = d = 1$. В случае формул (1) и (4) $A \circ B$ есть тензорное произведение $A \otimes B$, определяемое как

$$\begin{bmatrix} A_0^0 B & A_1^0 B & \cdots \\ A_0^1 B & A_1^1 B & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \text{ и } \begin{bmatrix} AB_0^0 & AB_1^0 & \cdots \\ AB_0^1 & AB_1^1 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

соответственно. В случае (2) для получения $A \circ B$ нужно первое из указанных тензорных произведений подвергнуть надлежащей перестановке строк, либо второе из этих тензорных произведений подвергнуть надлежащей перестановке столбцов. Случай (3) аналогичен случаю (2) с той разницей, что перестановки строк и столбцов меняются ролями.

4. Критерий ассоциативности

Как видно из предыдущего раздела, введенная в Определении 1 билинейная операция достаточно хорошо подходит на роль обобщенного умножения матриц. В настоящем разделе устанавливается критерий ассоциативности этой операции, то есть свойства вида $A \circ_1 (B \circ_2 C) = (A \circ_1 B) \circ_2 C$, позволяющего писать просто $A \circ_1 B \circ_2 C$.

Здесь и ниже манипуляция над символом « \circ », означаящим « $\overset{a,d}{*}$ », подразумевает, что эта манипуляция производится над a и над d одновременно, например $\circ_\sigma = \overset{a_\sigma, d_\sigma}{*}$ и т. п. Если встречаются несколько

операций обобщенного умножения, то подразумевается один и тот же выбор из вариантов (1-4) в Определении 1.

Теорема 4. Пусть целые положительные $a_1, d_1, a_2, d_2, v_A, h_A, v_B, h_B, v_C, h_C$ таковы, что определены произведения $A \circ_1 B$ и $B \circ_2 C$ для матриц A, B, C размеров $v_A \times h_A, v_B \times h_B, v_C \times h_C$ соответственно. Для того чтобы произведения $A \circ_1 (B \circ_2 C)$ и $(A \circ_1 B) \circ_2 C$ были определены и совпадали для любых матриц A, B, C указанных размеров, необходимо и достаточно, чтобы размеры v_B и h_B делились на $a_1 d_2$ и $a_2 d_1$ соответственно.

Отметим независимость от варианта (1-4) из Определения 1.

Оказалось, что наличие ассоциативности определяется матрицей B (точнее, ее размерами) и параметрами примыкающих слева и справа к B символов \circ_1 и \circ_2 . Имеет смысл оформить эту тройку как некое целое.

Определение 2. Конструкцию вида $\circ_1 B \circ_2$ назовем *мультипликативным звеном*, если v_B и h_B делятся на $a_1 d_2$ и $d_1 a_2$ соответственно.

5. Алгебра мультипликативных звеньев

Теорема 5. Если $\circ_1 A \circ_2$ и $\circ_2 B \circ_3$ суть мультипликативные звенья, то корректно определение матрицы $C = A \circ_2 B$, причем $\circ_1 C \circ_3$ также есть мультипликативное звено.

Определение 3. Произведение мультипликативных звеньев $\circ_1 A \circ_2$ и $\circ_3 B \circ_4$ будем считать определенным, если $\circ_2 = \circ_3$, а результатом произведения будем считать мультипликативное звено $\circ_1 C \circ_4$, где $C = A \circ_2 B$.

Из Теоремы 4 следует, что это произведение ассоциативно.

Теорема 6. Если a, b, c – мультипликативные звенья, и произведения ab, bc определены, то определены и равны между собой произведения $(ab)c$ и $a(bc)$.

Определение 4. Результат *транспонирования* мультипликативного звена $\circ_1 A \circ_2$ определяется как $\circ_2 A^T \circ_1$ (что, очевидно, также является мультипликативным звеном).

Теорема 7. Операция транспонирования мультипликативного звена имеет следующие свойства.

1. Всегда $(a^T)^T = a$.
2. Если произведение ab определено, то определено также произведение $b^T a^T$, причем $(ab)^T = b^T a^T$.

6. Подалгебры алгебры мультипликативных звеньев

Следующий результат показывает, что в алгебре мультипликативных звеньев содержится подалгебра, изоморфная обычной алгебре матриц.

Теорема 8. Обычная алгебра матриц естественным образом вкладывается в алгебру мультипликативных звеньев, если каждую $m \times n$ -матрицу M отождествить с мультипликативным звеном $\begin{smallmatrix} m,1 & n,1 \\ * & M & * \end{smallmatrix}$.

Согласно Определению 2 для параметров мультипликативного звена $\circ_1 A \circ_2$ справедливы неравенства $v_A \geq a_1 d_2, h_A \geq d_1 a_2$. Выделим особо случай, когда в обоих неравенствах достигается равенство, то есть размеры матрицы – минимальные возможные для данных \circ_1, \circ_2 .

Определение 5. Мультипликативное звено $\circ_1 A \circ_2$ назовем *s-минимальным* (от «size»), если $v_A = a_1 d_2, h_A = d_1 a_2$.

Теорема 9. Пусть мультипликативные звенья a, b *s-минимальны*.

1. Произведение ab (если оно определено) также *s-минимально*.
2. Транспонированное звено a^T также *s-минимально*.

Таким образом, s -минимальные звенья образуют подалгебру в алгебре мультипликативных звеньев.

Результат Теоремы 8 указывает на важность таких мультипликативных звеньев $\circ_1 A \circ_2$, для которых $d_1 = d_2 = 1$, то есть параметры d_1, d_2 принимают минимально возможное значение 1.

Определение 6. Мультипликативное звено $\circ_1 A \circ_2$ назовем d -минимальным, если $d_1 = d_2 = 1$.

Теорема 10. Пусть мультипликативные звенья a, b d -минимальны.

1. Произведение ab (если оно определено) также d -минимально.
2. Транспонированное звено a^T также d -минимально.

Определение 6 и Теорема 10 представляют частный случай некоторого общего способа выделять подалгебры в алгебре мультипликативных звеньев. Способ состоит в наложении одинакового условия как на \circ_1 , так и на \circ_2 (в Определении 6 это « $d_1 = 1$ » и « $d_2 = 1$ »). Имеет место следующий результат.

Теорема 11. Пусть $u(\circ)$ есть формулировка некоторого условия на целые положительные a, d , а U есть множество таких мультипликативных звеньев $\circ_1 A \circ_2$, что одновременно $u(\circ_1)$ и $u(\circ_2)$.

1. Если $a, b \in U$ и определено ab , то $ab \in U$.
2. Если $a \in U$, то и $a^T \in U$.

Пусть, например, $\circ_1 = \circ$ и $\circ_2 = \circ$ (\circ фиксировано).

Теорема 12. Мультипликативные звенья $\circ A \circ$ с фиксированным \circ образуют подалгебру.

Информация об авторах

Data on authors

Белянков Анатолий Яковлевич – канд. физ.-мат. наук, старший научный сотрудник отдела Прикладных проблем оптимизации ВЦ РАН.

Тел.: (499) 135-42-50

E-mail: belyankov@ccas.ru

Belyankov Anatoly Yakovlevich – Cand. Sc. (Phys.-Math.), senior scientific associate on Applied optimization problems division at CC RAS.

Tel.: (499) 135-42-50

E-mail: belyankov@ccas.ru

**О КОНЦЕПЦИИ ДЕКЛАРАТИВНО-ПОВЕДЕНЧЕСКОГО
ПРОГРАММИРОВАНИЯ В ИМИТАЦИОННОМ МОДЕЛИРОВАНИИ
СЛОЖНЫХ РАСПРЕДЕЛЕННЫХ СИСТЕМ**

**ON THE «DECLARATIVE BEHAVIOR» PROGRAMMING CONCEPT
FOR DISTRIBUTED SIMULATION OF COMPLEX SYSTEMS**

Обсуждаются вопросы моделирования таких сложных систем, про которые хорошо известно, из каких компонент они состоят, что эти компоненты умеют делать, по каким правилам взаимодействуют между собой. Проблемой моделирования, притом весьма непростой, является воспроизведение поведения и оценка возможностей такой системы в целом. Рассматриваются вопросы эффективности применения различных парадигм программирования для решения задачи синтеза многокомпонентной системы. Предлагается новая парадигма программирования, названная декларативно-поведенческой, эффективная для решения описанного класса задач.

The problem is discussed of modeling such complex systems, where it is well known from which components they are composed, what these components are able to do, and by what rules interact with each other. A new programming paradigm called “declarative behavior” is proposed to solve the synthesis problem for the multicomponent complex systems.

Ключевые слова: имитационное моделирование, сложные системы, парадигмы программирования, поведение системы, объектно-ориентированное программирование, декларативное программирование.

Keywords: simulation, complex systems, programming paradigms, behavior of system, object-oriented programming, declarative programming.

О каком классе систем и моделей пойдет речь

В данной работе мы будем говорить о моделировании сложных систем, понимая под сложной системой такую, которая состоит из компонент, также являющихся сложными системами. Такое понимание сложной системы отражено, например, в одноименной статье Н. П. Бусленко в БСЭ: «Сложная система – составной объект, части которого можно рассматривать как системы, закономерно объединённые в единое целое в соответствии с определенными принципами или связанные между собой заданными отношениями».

Со времен античности и до наших дней существует два основных подхода к моделированию сложных систем:

1. Один из них, называвшийся в разное время и в различных ситуациях «феноменологическим», «волновым», «системно-динамическим», состоит в том, что у изучаемой системы выделяются измеримые характеристики, описывающие эту систему в целом. Постулируется достаточность выделенных характеристик для описания системы (гипотеза о замкнутости) и ищутся связи между этими характеристиками. В рамках данного подхода работали Гераклит и милетские философы, Гюйгенс в споре с Ньютоном о природе света, создатели классической термодинамики.

2. Второй подход, называвшийся «агентным», «объектным», «атомистическим», «корпускулярным», предполагает выводить свойства сложной системы из свойств и способов взаимодействия «атомов», – неких простейших объектов, эту систему составляющих. Этот подход развивали античные атомисты Левкипп и Демокрит, в новое время – Ньютон в

упоминавшемся выше споре с Гюйгенсом о природе света, создатели статистической физики.

В настоящее время общепринятой точкой зрения по данному вопросу является признание равноценности и взаимной дополняемости «корпускулярного» и «волнового» описания реальности. Некоторые явления удобнее описывать «феноменологически», другие – «атомистически», а наиболее полное знание мы имеем в тех областях, где оба эти описания проникают друг в друга, как, например, истолкование классических термодинамических потенциалов методами статистической физики.

В данной работе будет рассматриваться исключительно «атомистический» подход. Не потому, что автор считает его «важнее» системно-динамического, а потому, что он оказывается примерно в половине случаев удобнее для моделирования реальных систем, и, следовательно, вполне заслуживает отдельного внимательного изучения (впрочем, как и системно-динамический).

Развивать атомистический подход к созданию сложных систем также можно двумя способами:

1. Дедуктивный способ. Предположим, что нужно создать сложную систему с заданной функциональностью (например, операционную систему или компилятор), а в выборе «атомов», реализующих различные функциональности мы свободны – разработка ведется с чистого листа. Тогда естественно поступить по-платоновски – проектировать систему сверху вниз, облакая основные идеи ее построения в соответствующие им формы, и воплощать эти формы в программный код. Идеалисты утверждают, что именно так и был создан этот мир. У программистов же для осуществления дедуктивного проектирования системы есть мощное средство – объектно-ориентированное программирование (ООП), чье

наследование с возможностью переопределения методов, позволяет построить иерархию классов вполне в духе Платона.

2. Индуктивный способ. Предположим, что мы не создаем новые миры, а моделируем уже созданные. Тогда «атомы» системы жестко заданы предметной областью. Они определены и не могут быть иными, их свойства и способы взаимодействия также хорошо известны. Задачей является воспроизведение и изучение поведения всей сложной системы, состоящей из этих «атомов». Заметим, что в данном случае обычно иерархическая структура классов ООП мало чем может помочь – нужно уметь работать с теми «атомами», которые есть в системе, а они далеко не всегда представляют собой строгую иерархию. Подробно эта проблема будет освещена далее.

Заметим, что в жизни дедуктивный и индуктивный способы построения сложных систем связаны своего рода диалектикой. Допустим, мы далеко продвинулись по дедуктивному пути, например, построили .NET – иерархическую систему из десятков тысяч классов. Теперь она всегда с нами, все классы отлажены, следовательно, вполне логично использовать их в своей работе – т. е., применять индуктивный способ, пользуясь кирпичиками .NET. Если же мы долгое время развивали индуктивный способ – со временем оказывается, что мы располагаем достаточно хаотичной коллекцией «атомов» с частично дублирующей друг друга функциональностью. Возникает естественное желание навести в этом богатстве определенный порядок – произвести дедуктивный реинжиниринг системы.

В данной работе будет рассматриваться исключительно индуктивный способ построения системы. Не потому, что он чем-то лучше дедуктивного, а потому, что класс задач мощностью примерно 25% (половина от половины) от всего на свете, заслуживает специального рассмотрения. Тем более что такое мощное средство как ООП, как можно

будет убедиться далее, работает для него гораздо хуже, чем для дедуктивного способа.

В качестве примеров корпускулярных индуктивных моделей сложных систем приведем моделирование боевых действий в ходе штабных игр с уровнем детализации полк – дивизия – армия – фронт; изучение возможностей стратегической оборонной инициативы (СОИ) – исследование, в котором участвовал автор в конце 80-х; моделирование работы холдинга, объединяющего ряд промышленных и финансовых предприятий.

О методе имитационного моделирования

Под имитационным моделированием, применительно к очерченному выше классу задач, будем понимать воспроизведение поведения сложной корпускулярной индуктивной системы в рамках тех или иных сценариев имитационных экспериментов, задающих соответствующие им диапазоны внешних переменных модели.

Любого рода аналитика, как например, исследование устойчивости траектории системы, оптимизация каких-либо функционалов, управляемость, игровые взаимодействия агентов, поддержка принятия решений и т. д., этим методом не отрицаются, а наоборот, базируются на нем, являясь различного рода надстройками над воспроизведением поведения системы. Однако разговор о таких надстройках останется за пределами данной работы.

В данной работе мы ограничимся лишь задачей воспроизведения поведения сложной системы – задачей самой по себе весьма нужной, интересной и нетривиальной.

О двух видах знания, а также о декларативном и императивном программировании

Философу времен заката Рима Аницию Бэоцию принадлежит утверждение: «Гораздо важнее знать, что делается, чем делать то, что

знаешь». Можно соглашаться или оспаривать данное утверждение, однако самым существенным в нем нам кажется выделение двух видов знания:

1. Про некоторые вещи мы точно (конечно, в пределах точности наших моделей) знаем, каковы они. Например, свое имя, паспортные данные и банковские реквизиты. Или каковы характеристики и способы взаимодействия «атомов» корпускулярной модели. Или какова должна быть страничка Web-сайта, которую мы описываем на языке HTML.

2. Про некоторые вещи мы не знаем, каковы они, а знаем (или только считаем, что знаем) лишь как об этом узнавать. Например, каково решение достаточно сложной системы дифференциальных уравнений – существуют различные алгоритмы решения таких систем, для некоторых из них известны строгие обоснования их применимости в определенных случаях. Или, например, как работает и на что способна предложенная Р. Рейганом СОИ – можно построить ее имитационную модель и поиграть на ней в звездные войны. Здесь проблема с обоснованием применимости модели, по-видимому, будет сложнее.

Следует отметить, что граница между этими видами знаний достаточно подвижна. Например, легко можно забыть свои паспортные данные и банковские реквизиты. Однако если вместе с ними не утрачены основные навыки бытовой культуры, то известно, какие стандартные действия нужно предпринять, чтобы восстановить эти знания. С другой стороны, многие скажут, что знают, что такое π , e или $\sqrt{2}$, хотя в очень строгом смысле, мы всего лишь знаем, как вычислять эти числа с некоторой (вовсе не какой угодно!) степенью точности. Аналогично, для специалистов в области вычислительной математики численное решение некоторых классов систем дифференциальных уравнений, особенно если под рукой имеются соответствующие инструментальные средства, является почти такой же рутинной, как вычисление $\sqrt{2}$, и в этом смысле можно

сказать, что такие специалисты «знают» решения упомянутых систем уравнений.

С приведенными выше двумя типами знания достаточно естественно связываются две известные и очень важные для нас парадигмы программирования – декларативная и императивная.

Вообще-то различных парадигм программирования гораздо больше. Например, в Википедии их можно насчитать около трех десятков. В данной работе будут упомянуты лишь те из них, которые имеют отношение к исследуемой теме – построению корпускулярной индуктивной модели сложной системы.

Под парадигмой программирования, вслед за Памелой Зейв (Pamela Zave), автор склонен понимать набор представлений о некотором классе программных систем, допускающих реализацию с помощью этой парадигмы «way of thinking about computer systems» [7]. Таким образом, парадигма есть не язык или система программирования, а некий порядок в голове программиста, позволяющий ему даже на ассемблере писать структурно, или объектно, или функционально, или же в рамках какой-то еще парадигмы. Тем не менее, многие современные языки программирования ориентированы на ту или иную парадигму программирования, и, следовательно, предоставляют специальные средства и удобства для ее реализации. Например, C++, Java, C# реализуют ООП (разновидность императивного программирования); LISP, F#, РЕФАЛ – функциональное программирование (разновидность декларативного); PROLOG – логическое.

При декларативном программировании описывается то, каким должен быть известный заранее желаемый результат, например, статическая страничка HTML, или документ в системе LaTeX, или образ DVD-диска со всей сложной системой его меню и видеоматериалов в системе авторинга Scenarist. При этом выбор последовательности

действий, приводящей к описанному в системе декларативного программирования результату, как правило, оставляется на усмотрение соответствующего компилятора или интерпретатора.

Под императивным программированием будем понимать самый распространенный подход к написанию программ, где программа есть последовательность инструкций-приказов, которые должен выполнить компьютер. Таким образом, применяя императивное программирование, мы описываем последовательность действий, достаточную (на наш взгляд) для получения результата проекта. Результат при этом не предполагается известным заранее, наоборот, скорее целью проекта императивного программирования как раз и является получение этого результата.

Мы видим, что декларативная парадигма программирования достаточно естественна для описания наших знаний о системе первого типа, т. е. того, что мы знаем наверняка. Императивное же программирование хорошо для описания знаний второго типа – алгоритмов узнавания того, что мы не знаем непосредственно, но хотим и умеем (или считаем, что умеем) узнавать.

Заметим, что как декларативное, так и императивное программирование дают достаточно простора для ошибок. Об этом знает всякий, кому приходилось программировать, хотя бы даже на HTML. Однако отлаживать декларативную программу не в пример легче – очень помогает знание о том, как все должно быть. Императивные же программы отлаживать очень трудно. Результат работы программы не известен заранее. В случае достаточно сложной программы, даже правильно работающей, этот результат может оказаться весьма неожиданным для исследователя (например, в задачах оптимального управления, вместо обывательских представлений о плавном восхождении от хорошего к еще лучшему, могут возникать скользящие режимы управления, весьма далекие от подобных представлений). Так что если даже императивная программа

не вылетела и не зациклилась, а благополучно что-то сосчитала – всегда остается сомнение: полученный ею результат действительно верен, или же программа где-то врет? Для решения этой проблемы разрабатываются специальные системы тестирования, иногда сопоставимые по затратам с разработкой основной программы. Заметим еще, что предполагаемое «знание о том, как узнавать» в императивных программах вполне может подвести – оказаться ложным – здесь также весьма обширное поле для разного рода иллюзий и ошибок.

Скажем еще несколько слов о функциональном программировании – разновидности декларативного. Здесь процесс вычисления трактуется как вычисление значений функций в математическом понимании последних. Математическая функция отличается от функции-подпрограммы императивного программирования тем, что осуществляет отображение своей области определения в область допустимых значений. Функция-подпрограмма устроена сложнее. Она также может оказаться отображением множества своих входящих параметров в множество возвращаемых, однако вполне может и не оказаться. Например, у нее могут существовать различные внутренние состояния, связанные со статическими переменными или со значениями записей базы данных, с которой она работает. И в зависимости от этих состояний, функция-подпрограмма при одном и том же входе может выдавать различные выходы. Функциональную парадигму легко осуществить средствами большинства современных универсальных языков программирования, ограничившись константами вместо переменных, и следя за тем, чтобы функции-подпрограммы были отображениями в математическом смысле. Функциональные программы хороши тем, что по сравнению с императивными, легче отлаживаются и тестируются: никаких побочных эффектов, никаких состояний – выход зависит лишь от входа, ошибка достаточно легко локализуется.

Из всего сказанного в этом параграфе следует вывод: если что-то в системе допускает декларативное описание, надо это ценить и стараться всячески использовать – отладка будет гораздо легче. Аналогично, если есть возможность сократить императивное программирование, ни в коем случае нельзя ее упускать – это существенно упростит последующую жизнь. Если есть возможность написать подпрограмму в функциональной парадигме, обязательно нужно этим воспользоваться – во время отладки и тестирования усилия окупятся сторицей.

Почему так трудно программировать модель сложной системы?

Потому, что она и в самом деле очень сложная!

Попробуем перечислить основные сложности:

1. Сложная структура системы и сложные связи между ее компонентами;
2. Сложная организация данных;
3. Сложное поведение компонент системы:
 - а) Сложная логика поведения;
 - б) Сложная функциональность действий.

Здесь мы выделили несколько типов сложности, а в модели сложной системы все они еще и тесно переплетены между собой. Психологи утверждают, что средний человек уверенно способен оперировать не более чем 4-6 объектами одновременно. Если объектов становится больше, человек начинает ошибаться. Быть может, природная одаренность в данной области или же опыт и тренировка, способны повысить этот порог с 4-6 даже до 15-20 объектов. Беда только в том, что в сложных системах приходится оперировать с сотнями и тысячами объектов.

Особенно сложным является императивное программирование, о чем уже говорилось выше. Кроме уже упомянутых сложностей выбора алгоритма, действительно приводящего к желаемому результату, отладки и тестирования, именно императивные программы обычно сочетают в себе

сложную логику со сложной функциональностью и сложной организацией данных.

Возникает вопрос, что же тогда делать, как найти технологию программирования, которая поможет преодолеть перечисленные сложности? На самом деле такие технологии существуют и достаточно давно применяются в области автоматизации проектирования.

Бей врага по частям, или похвала декомпозиции

Проблема синтеза сложных систем из заданных компонент достаточно успешно решается в различных системах автоматизации проектирования (САПР). Основная идея САПР – модульность разработки. Система проектируется не из самых мелких своих составляющих, а из модулей (блоков) самого высокого уровня, которые могут состоять в свою очередь из модулей более низкого уровня, и, наконец, модули самого низкого уровня состоят из самых мелких «атомов» системы. Как только тот или иной модуль отлажен, про его внутреннее устройство можно забыть. Для работы с ним на более высоком уровне, важны лишь его интерфейс и функциональность. Такой подход позволяет создавать, например, процессоры, содержащие десятки миллионов транзисторов, что было бы немыслимо при проектировании их на уровне отдельных транзисторов.

Теория геометрической декомпозиции произвольных математических объектов построена в цикле работ Ю. Н. Павловского (например, [17]). Это абстрактная математическая теория, основанная на бурбаковском формализме [13]. Здесь мы приведем лишь некоторые из ее основных положений, касающихся темы данной работы, притом на упрощенном, «бытовом» языке, отсылая интересующихся строгим изложением предмета к оригинальным работам (например, [17]).

Для нас будет важно, что существуют два основных типа декомпозиции математических объектов:

1. F-декомпозиция – декомпозиция факторизации. Неформально ее можно проиллюстрировать известным аристотелевским силлогизмом: «Сократ – человек, следовательно, Сократ смертен». Здесь объект разбивается на фактор-классы по некоторому отношению эквивалентности, и каждый член фактор-класса наследует от него, по крайней мере, то свойство, в силу которого он в нем оказался. В примере Аристотеля это может быть свойство смертности, наследуемое Сократом от класса «человек».

2. P-декомпозиция – декомпозиция подобъектов. Неформально ее иллюстрирует известное стихотворение С. Маршака:

«Дама сдавала в багаж
Диван, чемодан, саквояж,
Картину, корзину, картонку
И маленькую собачонку...»

Здесь совершенно разнородные объекты со всеми своими атрибутами попадают в один и тот же багаж, и тем самым приобретают некое новое общее свойство принадлежности этому багажу.

Оказывается, что любая декомпозиция математических объектов есть некоторое сочетание F- и P-декомпозиций, причем эти два вида декомпозиции находятся между собой в своеобразном отношении двойственности [17].

Для нас здесь будет важным то, что было названо выше дедуктивным построением сложной системы, хорошо описывается ее F-декомпозицией; то же, что называлось индуктивным построением, особенно интересующим нас в данной работе, наоборот, естественно описывать P-декомпозицией.

Исходя из изначально имеющейся P-декомпозиции сложной системы, хотелось бы построить ее синтез в духе приведенных выше методов САПР. В основе такого синтеза многокомпонентной системы лежит идея,

высказанная еще в работах Н. П. Бусленко – дать каждой из компонент, про которые нам все известно, максимально проявить себя, учитывая при этом и все межкомпонентные связи [14]. Посмотрим, как исторически реализовывалась эта идея.

Объектно-ориентированное программирование (ООП)

Обычно считается, что объектный анализ возник в последней трети XX века в результате развития с одной стороны идей САПР, а с другой – идей атомистического имитационного моделирования (Симула-67).

На самом деле, основные проблемы типизации математических объектов рассматривались ранее в работах Н. Бурбаки (например, [13]). Так класс объектов классического объектного анализа есть частный случай бурбаковского понятия «рода структуры» ([13], [17]).

Объектно-ориентированная парадигма является в настоящее время основным инструментом реализации проектов императивного программирования. Одним из самых первых объектно-ориентированных языков можно считать Симулу-67. Структура сложной системы, состоящей из многих экземпляров различных типов компонент, хорошо описывается иерархией классов объектно-ориентированного программирования. У ООП масса известных всем достоинств, которые и сделали этот подход бесспорным лидером в программировании последних 30 лет. Однако применительно к нашей задаче есть у ООП и существенные недостатки.

Главный из них (конечно, применительно к классу задач, о которых идет речь в данной работе) – отсутствие у объекта поведения. И это несмотря на то, что одному из основателей ООП, Гради Бучу, принадлежит следующее известное высказывание: «Объект – осязаемая сущность, которая четко проявляет свое поведение».

На самом деле, у объекта есть характеристики, есть масса всяческих умений – это его методы – а поведения, в смысле умения давать ответы на стандартные запросы внутренней и внешней среды, у него нет. Конечно,

ООП в соединении с соответствующим программным обеспечением промежуточного уровня, позволяет объектам даже в распределенных системах взаимодействовать между собою, наблюдать чужие характеристики, пользоваться чужими методами, однако все это делается в некотором смысле «вручную».

В этом смысле коллектив программистов, приступающий к созданию сложного проекта и вооруженный набором библиотек самых полезных классов объектов, похож на человека, которому выдали несколько компьютеров, каждый из которых оснащен большим набором полезных прикладных программ, но из системных программ имеется только загрузчик, и поставили задачу сделать из этого распределенную систему. Понятно, что современный программист возмутился бы страшно: «Так не бывает! Где моя сетевая операционная система? Где, наконец, ПО промежуточного уровня?». Однако при создании сложной системы средствами ООП дело обстоит именно так – к сложности содержательных вычислений предметной области добавляется еще и сложность организации поведения объектов, а также их связей и взаимодействий между собою.

Отсутствие у объекта поведения ведет к серьезному нарушению принципа модульности, который позволяет в САПР преодолевать многие сложности проектирования больших систем. Написав и отладив класс, невозможно навсегда забыть о его устройстве. Построив самую стройную и логичную иерархию классов, все равно приходится переходить к описанию поведения системы в целом. И вот тут-то оказывается, что почти все методы почти всех классов так или иначе могут быть связаны друг с другом, вызывать друг друга или следовать друг за другом, при определенных условиях. И как-то декомпозировать этот клубок взаимосвязей представляется весьма затруднительным. Во всяком случае, классический ООП не предлагает здесь каких-либо готовых решений.

Второй серьезный недостаток ООП, конечно, снова применительно к очерченному в начале работы классу задач – механизм наследования. С точки зрения геометрической теории декомпозиции ([17]), наследование, принятое в ООП, можно было бы назвать F-наследованием. Иерархия классов строится сверху вниз, вполне в платоновском духе, во многих объектных языках, например, в С#, любой класс является потомком исходного универсального класса «объект». Такой способ наследования замечательно подходит для упоминавшихся выше дедуктивных построений, например для создания с чистого листа нового сложного программного обеспечения. Однако для точного воспроизведения (имитационного моделирования) того, что уже существует или могло бы быть создано в предметной области из готовых компонент, такое наследование подходит плохо: в жизни часто в одном багаже могут оказаться совершенно не похожие друг на друга собачонки, диваны, картины и чемоданы! Здесь необходимо другое наследование, которое с точки зрения геометрической теории декомпозиции естественно было бы назвать Р-наследованием.

Что такое поведение системы?

На наш взгляд, поведение сложной системы есть функциональный аналог операционной системы в области системного программного обеспечения компьютера, и функциональный аналог бытовой культуры в социальных системах – способность давать стандартные ответы на стандартные запросы внутренней и окружающей среды. В имитационном моделировании сложных систем очень важно уметь моделировать поведение компонент. Как уже говорилось выше, именно из их поведения естественно синтезировать поведение системы в целом.

В истории развития информатики известно несколько подходов к объектному программированию, где объекты обладают поведением. Подходы эти зародились в среде исследователей, занимавшихся

проблемами искусственного интеллекта, и известны как модель акторов [3] и агентное программирование [5, 6]. Однако хотя нельзя не согласиться с главным посылом этих подходов – важностью моделирования поведения агентов системы – детали описания этого поведения в [3, 5, 6], на наш взгляд, слишком окрашены спецификой проблематики искусственного интеллекта, т. е., недостаточно универсальны. Так у Шохема [5, 6] поведение – это поведение ментальное, о состоянии которого рассуждают в категориях «убеждения», «обязанности», «способности» и т.д. Акторная модель Хьюитта [3] интересна своей ориентированностью на параллельные вычисления, однако асинхронный обмен акторов сигналами-сообщениями, а также постоянное порождение в системе новых акторов, представляется нам избыточными конструкциями, затрудняющими реализацию подобных систем. Подробно этот вопрос обсуждался в [12].

Проблема здесь в том, что согласившись на наличие поведения у компонент системы, нужно учиться описывать и реализовывать это поведение. Это поведение с одной стороны достаточно сложно, так как это поведение хотя и элементарной, но сложной системы. Заранее оно не может быть известно, поскольку может зависеть от внутренних и внешних по отношению к рассматриваемой компоненте условий, на которые она должна уметь реагировать должным образом. Следовательно, такие поведенческие моменты должны вычисляться, и, казалось бы, алгоритмы их вычисления должны быть описаны на императивных языках программирования. С другой стороны, кое-что о поведении компоненты мы всегда знаем заранее, так как она в силу исходных предположений работы имеет в предметной области моделирования достаточно хорошо изученный прообраз. Поэтому обычно мы заранее можем перечислить весь набор ее «умений» – элементарных действий, а также сказать какое из этих элементарных действий может перейти в какое, и под действием каких обстоятельств – просто уж так устроен прообраз этой компоненты в

предметной области моделирования. Это знание, коррелирующее с пониманием устройства в предметной области, как прообраза отдельной компоненты, так и всей многокомпонентной модели, как раз наличествует заранее, еще до построения модели, и не зависит от хода имитационных экспериментов. Поэтому такое знание об устройстве модели сложной системы вполне может быть выражено на декларативном языке программирования.

Такой подход позволяет получить «ортогональные» (подобным образом «ортогональны», например, описания стиля и наполнения в Word- или HTML-документах) описания модели сложной системы: на декларативном языке описывается то, как устроена система и правила поведения ее составляющих. На императивном языке описываются отдельные законченные элементарные алгоритмы, не взаимодействующие ни с чем в модели, кроме собственных входных и выходных параметров.

При этом, как будет показано далее, эти элементарные алгоритмы, хотя и описанные на языке императивного программирования, по своей сути вполне могут быть математическими отображениями, т. е. функциональными программами.

Предлагаемое разделение описаний на декларативное и императивное (на самом деле функциональное, т. е. тоже декларативное) оказывается весьма технологичным: декларативную и императивную части можно отлаживать независимо друг от друга. При этом самая сложная императивная составляющая программы распадается на ряд независимых друг от друга законченных элементарных алгоритмов. При этом она редуцируется до функционального программирования, настолько, что чаще всего возможности ООП оказываются слабо востребованными: алгоритмически цельную программу с фиксированным отображением входных параметров в выходные, чаще всего нетрудно реализовать хоть на FORTRANе. И это существенно облегчает отладку системы. Вся объектная

ориентированность, идущая от предметной области моделирования, оказывается в декларативном описании модели.

Одной из первых сред программирования, воплотивших описанную выше концепцию разделения описания сложной системы на декларативную и императивную составляющие, была разработанная в ВЦ АН СССР инструментальная система имитационного моделирования MISS (Multilingual Instrumental Simulation System). Концепция программирования, лежащая в ее основе, описана в работах [8, 9], а полное описание среды моделирования на уровне руководства пользователя в [10]. В системе MISS была реализована в среде MS-DOS система программирования на специальном декларативном языке описания сложных систем, интегрированная с базой данных, системой поддержки выполнения модели и системой презентации результатов моделирования. В качестве императивных языков программирования, на которых писались вычислительные алгоритмы были разрешены две версии языка MODULA-2, а также языки С и С++ в борландовской реализации. До этого в основном моделирующее сообщество создавало языки моделирования императивного типа.

Впоследствии идея разделения описания сложной системы на декларативную и императивную части применялась неоднократно. Так в спецификации архитектуры верхнего уровня (High Level Architecture – HLA) [4] – средстве создания сложных распределенных моделей. Устройство системы, ее компонент и связей между ними описывается специальными шаблонами. В коммерчески успешной отечественной инструментальной системе имитационного моделирования AnyLogic [16] в качестве декларативного языка описания сложной системы применяется интерактивная графическая система, где на экране размещаются пиктограммы компонент будущей системы и тут же проводятся связи между ними. В качестве императивного языка программирования

используется родной для системы AnyLogic язык Java. Наконец, появился и даже стал международным стандартом, язык моделирования объектных систем UML [15]. Однако автору он не симпатичен в силу своей громоздкости, «рыхлости», и порой, за счет этого, неоднозначности. В качестве результатов компиляции транслятор UML может выдавать заготовки модулей для императивных языков программирования.

В настоящее время в ВЦ РАН продолжается развитие концепции разделения описания сложной системы на декларативную и императивную составляющие. Работает макет инструментальной системы распределенного моделирования. Между прочим, определенное затруднение вызывает название описываемой концепции – такие названия, как агент, актер и даже компонента, уже заняты, и в программистском сообществе под ними понимаются вполне определенные и отличные от описанного выше подходы. Сами разработчики концепции в работах [1, 2, 8-11] называли свой подход объектным или объектно-событийным, что тоже не вполне верно, так как не отражает главных его моментов – наличия поведения у компоненты и декомпозицию описания сложной системы на декларативную и императивную составляющие.

В настоящей работе мы будем называть предлагаемую парадигму ДПП – декларативно-поведенческое программирование, или декларативно-поведенческий подход, или декларативно-поведенческая парадигма.

Декларативно-поведенческий подход

Опишем теперь кратко предлагаемую парадигму программирования моделей сложных систем. Подробное описание можно найти в [12]. В основе концепции лежит ключевое понятие компоненты. Компонента – это в некотором смысле «элементарная» сложная система. Основой конструкции компоненты будет объект объектного анализа, в том смысле, что описываемая ниже компонента есть некий тип или класс моделей, а заполняться данными и запускаться на счет будут экземпляры этого класса.

Однако главное отличие от объекта в том, что компонента – это объект с поведением. Второе важное отличие – иной механизм наследования, который в терминах геометрической теории декомпозиции [17], можно назвать P-наследованием, отличающийся от механизма наследования, принятого в объектно-ориентированном программировании, который с точки зрения упомянутой теории естественно назвать F-наследованием.

Опишем «устройство» компоненты.

Характеристики. Компонента, как и объект, имеет характеристики. Эти характеристики мы будем разбивать на внутренние и внешние. Внутренние характеристики – это то, что компонента моделирует, внешние – это то, что она знает о внешнем мире.

Процессы. Функциональность компоненты удобно структурировать следующим образом: считается, что компонента реализует один или несколько параллельно выполняющихся процессов. Процесс состоит в последовательном чередовании элементов – алгоритмически элементарных методов. Если какой-либо процесс какую-то часть модельного времени не выполняется, удобно считать, что в это время он выполняет «пустой» элемент, не меняющий никаких характеристик компоненты.

Элементы. Элементарные, алгоритмически однородные методы, реализующие функциональности компоненты (т. е. то, что компонента умеет делать: получение на основании значений некоторых внутренних и внешних характеристик компоненты, новых значений некоторых ее внутренних характеристик). Каждый элемент отображает некоторое подмножество внутренних и внешних переменных компоненты в начале шага моделирования и продолжительность шага моделирования в подмножество внутренних переменных модели в конце этого шага. Поэтому все элементы могут быть реализованы функциями в смысле парадигмы функционального программирования, что, конечно же, немаловажно.

По отношению к модельному времени некоторые элементы выполняются мгновенно, это сосредоточенные или быстрые элементы. Быстрыми элементами можно моделировать разрывные (дискретные) характеристики системы.

Выполнение других элементов занимает определенное время. Если при этом элемент для любого промежутка времени $\Delta\tau$, не превосходящего стандартный шаг моделирования, выдает некий осмысленный результат, такой элемент называется распределенным или медленным. Распределенные элементы – естественное средство вычисления непрерывных характеристик модели, например, описываемых системами дифференциальных уравнений.

Может оказаться и так, что выполнение элемента занимает определенное модельное время, но результат его действия наступает лишь в конце, после полного выполнения элемента, т. е. никаких промежуточных результатов за время меньше полного времени выполнения нет. Такие элементы называются условно-распределенными. Вообще говоря, условно-распределенные элементы можно не рассматривать как отдельный класс, а моделировать парой: пустой распределенный элемент, за которым идет сосредоточенный, выдающий результат.

Частью предлагаемой концепции является жесткая дисциплина работы методов с фазовыми переменными модели: каждый метод имеет право изменять только «свои» переменные. Эта дисциплина основана на принятии предположения о детерминированности и однозначности имитационных вычислений. В рамках предлагаемой концепции, конфликт доступа возникающий, когда методы A и B вычисляют одну и ту же характеристику $X = X_A$ и $X = X_B$, может быть разрешен, например, введением метода C , который получая на входе в качестве параметров X_A и X_B , вычисляет на их основании искомую характеристику X , устраняя тем самым не только конфликт доступа к ней, но и, очевидно имевшую

место, неоднозначность вычисления упомянутой характеристики. Принятая дисциплина доступа к характеристикам позволяет вызывать параллельно те методы, которые в модельном времени выполняются одновременно.

Наконец, последнее, что следует заметить относительно элементов. Как и всякий метод, каждый элемент имеет входные и возвращаемые параметры. Концепция моделирования предполагает возможность распределенного вычисления элементов, т. е. элемент может быть найден разработчиком компоненты на просторах Интернета, поэтому его параметры таковы, какими их сделал его автор, и скорее всего, никак не согласованы с характеристиками компоненты, которые придумал ее разработчик. Поэтому, при описании компоненты обязательно должно быть уделено внимание описанию коммутаций входных параметров элементов с внутренними и внешними характеристиками компоненты и выходных параметров элементов с ее внутренними характеристиками.

События. Содержательно, события – это то, что нельзя пропустить при моделировании динамики системы – точки синхронизации различных ее функциональностей, представляемых процессами. Моменты модельного времени, когда получены такие значения характеристик модели и внешних переменных, на которые обязаны отреагировать некоторые процессы компоненты.

Формально, событие – функция значений внутренних и внешних переменных в начале шага моделирования. С точки зрения организации имитационных вычислений, событие – это метод, входными параметрами которого является подмножество внутренних и внешних характеристик компоненты, а выходной параметр один – прогнозируемое время до наступления этого события. Если это прогнозируемое время равно нулю, значит, событие уже наступило. События управляют чередованием элементов в процессе.

Для каждой упорядоченной пары элементов процесса $\{A, B\}$, если между ними возможен переход, то ему обязан соответствовать метод-событие $E_{\{A, B\}}$, прогнозирующий время этого перехода. Возможны также события вида $E_{\{A, A\}}$, прерывающее выполнение элемента A , например, если его еще не нужно заканчивать, но он вычислил характеристики компоненты, которые могут повлечь смену элементов других процессов. Процесс перехода должен быть однозначным. Одновременное наступление событий $E_{\{A, B\}}$ и $E_{\{A, C\}}$ говорит лишь о том, что разработчик модели при ее проектировании упустил из своего рассмотрения этот случай, которому, быть может, должен соответствовать переход $\{A, D\}$.

Возможно тем, кто знаком с архитектурой компьютера или с событийно-ориентированными языками программирования такое определение событий покажется странным, и вызовет вопросы типа, а где же здесь что-нибудь подобное обработчикам событий? Дело в том, что в моделировании (если, конечно не брать моделирование полунатурное, когда к компьютерному комплексу подключают реальное изделие «в железе») чаще всего наступление события должна определить сама модель. Даже если это событие неожиданное, например, случайное, – все равно, в нужный момент модель вполне детерминировано должна запустить генератор случайных чисел, чтобы выяснить, не произошло ли оно. А подобный порядок действий вполне адекватно ложится на изложенную выше концепцию событий.

Выполнение компоненты. Компонента – элементарная, но, тем не менее, полноправная модель сложной системы. Поэтому ее можно запустить на выполнение имитационного эксперимента. Конечно, если имеются начальные данные и способ нахождения внешних характеристик компоненты в моменты событий.

Правила запуска компоненты на выполнение следующие. Во-первых, задается стандартный шаг моделирования Δt . Во-вторых, считается, что в начале шага моделирования известны текущие элементы всех процессов и все внутренние и внешние характеристики модели. Далее,

1. Вычисляются события связанные с текущими элементами процессов. Если есть наступившие события, проверяется, нет ли переходов к быстрым (сосредоточенным) элементам, если они есть – выполняются быстрые элементы (они становятся текущими), затем возврат к началу п.1; если нет переходов к быстрым элементам – совершаются переходы к новым медленным (распределенным) элементам, затем возврат к началу п.1.

2. Если нет наступивших событий – из всех прогнозов событий выбирается ближайший $\Delta \tau$.

3. Если стандартный шаг моделирования не превосходит прогнозируемого времени до ближайшего события, $\Delta t \leq \Delta \tau$ – моделируем текущие распределенные элементы со стандартным шагом Δt . В противном случае – моделируем их с шагом времени до ближайшего спрогнозированного события $\Delta \tau$.

4. Возвращаемся к началу п.1.

Отметим, что коль скоро имеются вполне формальные правила выполнения компоненты, то с разработчика системы снимается задача организации ее выполнения (в отличие, например, от ООП). Однажды описав и отладив компоненту, можно забыть о ее устройстве и проблемах организации связанного с ней вычислительного процесса, – выполняться она будет автоматически, специальной программой, в соответствии с приведенными выше правилами.

Далее следует заметить, что раз мы требовали от элементов однозначности имитационных вычислений и сумели этого добиться – на исполнение все одновременно выполняющиеся в модельном времени

элементы можно запускать асинхронно, т.е. загружать ими имеющиеся ядра процессора или же доступные распределенные компьютеры. Если же этого почему-то добиться не удалось – у системы поддержки выполнения модели есть полная информация о том, кто и что меняет (собственно это она должна будет по выполнении элементов обновить соответствующие данные в базе) – и она обязана выдать соответствующую диагностику ошибки времени выполнения.

В связи с описанными правилами выполнения компоненты могут возникнуть вопросы: не может ли выполнение быстрых элементов в п. 1 а также уменьшение шага времени в п. 3 привести к зацикливанию программы за счет возникновения точек накопления системных событий. Подробно этот вопрос анализируется в [12]. К сожалению, полную гарантию можно дать лишь для непрерывных систем, описываемых дифференциальными уравнениями с липшицевой правой частью. Для более широкого класса так называемых лапласовских моделей [12] можно доказать (неконструктивно) существование нециклящегося набора событий.

Комплексы. Компоненты могут объединяться в комплекс, при этом (необязательно) может оказаться, что некоторые компоненты явно моделируют внешние переменные некоторых других компонент. Для того, чтобы полностью описать комплекс, достаточно указать:

1. Какие компоненты и в каком количестве экземпляров в него входят.
2. Коммутацию компонент внутри комплекса, если она имеет место, т. е., какие внутренние переменные каких компонент являются какими внешними переменными и каких именно компонент комплекса.

При объединении компонент в комплекс, следует иметь в виду, что однозначность вычислительного процесса может быть потеряна (между прочим, эта ситуация очень похожа на проблему множественного

наследования в ООП). Так может произойти, если по каким-то причинам, несколько компонент вычисляют одну и ту же характеристику моделируемого явления. Например, с десятков фирм, входящих в комплекс как компоненты, предлагают прогноз погоды на завтра. В таком случае можно ввести в комплекс новую компоненту, которая в качестве внешних переменных получает весь многозначный набор значений упомянутой характеристики, а в качестве внутренней переменной каким-то образом вычисляет единственное ее значение.

Комплекс как компонента, Р-наследование. Комплекс, состоящий из многих компонент, вовне может проявляться в качестве единой компоненты.

Введем следующую операцию объединения компонент комплекса:

1. Внутренними переменными комплекса объявляется объединение внутренних переменных всех его компонент. При этом считается, что все неоднозначности, возникшие вследствие объединения характеристик различных компонент с одинаковой семантикой, уже разрешены. Например, так, как было указано выше – введением компонент, получающих на вход неоднозначный набор характеристик, и выдающих на выходе каким-либо образом выбранное единственное значение характеристики.

2. Процессами комплекса объявляется объединение всех процессов его компонент.

3. Методами комплекса объявляется объединение всех методов его компонент.

4. Событиями комплекса объявляется объединение всех событий его компонент.

5. Внешними переменными комплекса объявляется объединение всех внешних переменных его компонент, из которого исключаются все те переменные, которые моделируются явно какими-либо компонентами.

Операция объединения превращает комплекс в компоненту. Этот факт позволяет строить модель как фрактальную конструкцию, сложность которой (и соответственно подробность моделирования) ограничивается лишь желанием разработчика.

Отметим, что объединение компонент в комплекс с последующей операцией превращения комплекса в компоненту, есть некий аналог наследования в ООП, а именно, такое объединение вполне может быть названо Р-наследованием (двойственным F-наследованию классического ООП), в соответствии с идеями геометрической теории декомпозиции [17]. Еще отметим, что если в F-наследовании ООП множественное наследование представляет собой определенную проблему (настолько, что в некоторых языках, например, в С#, оно запрещено), то Р-наследование прежде всего интересно именно множественное. Выше было указано, как следует разрешать неоднозначности множественного Р-наследования.

Декларативный язык описания компонент и комплексов (ЯОКК)

Для описания состава и устройства сложной системы разработан специальный декларативный язык ЯОКК (язык описания компонент и комплексов). Он является развитием в сторону упрощения языка MISS [10] и описан в [11,12]. На языке ЯОКК с помощью его различных описателей можно описывать типы данных, методы-элементы, методы-события, компоненты, комплексы.

Описатель компоненты, при условии того, что описатели всех входящих в него типов данных, методов-элементов и методов-событий были успешно откомпилированы, компилируется в определенную структуру базы данных, связанную с этой компонентой. После заполнения в этой структуре полей, связанных с начальными значениями внутренних и внешних переменных компоненты, эту компоненту можно запустить на выполнение. Для этого существует специальная универсальная системная программа поддержки выполнения моделей, способная организовать

вычислительный процесс любой модели, представленной в базе данных моделей известной структурой таблиц, в соответствии с описанными выше правилами. Вычислительный процесс организован так, что происходящие одновременно в модельном времени элементы, могут быть вызываемы программой поддержки выполнения модели параллельно.

Описатель комплекса, при условии того, что все описатели входящих в него компонент были успешно откомпилированы, может быть откомпилирован в синтаксически верный описатель компоненты на ЯОКК, соответствующий объединенной компоненте-комплексу.

С примером использования языка ЯОКК для описания учебной имитационной модели «Пешеходы и муха» можно познакомиться в Интернете по адресу: <http://simul.ccas.ru/Distr/fly/fly.htm>.

Заключение

Декларативно-поведенческий подход (ДПП) позволяет до конца осуществить принцип модульности при создании моделей сложных корпускулярно-индуктивных моделей, примерно так, как это происходит в САПР. Однажды описав поведение компоненты, далее можно не вспоминать о нем – оно будет везде (где присутствуют экземпляры этой компоненты) и всегда автоматически воспроизведено программой поддержки выполнения модели. Однажды спроектированную и отлаженную компоненту можно включать в различные охватывающие комплексы, не заботясь при этом, как ее поведение впишется в поведение охватывающего комплекса. Такое вписывание поведения происходит автоматически, конечно в соответствии с принятыми в данной концепции «протоколами и интерфейсами», которыми здесь служит коммутация в рамках комплекса внутренних характеристик одних компонент с внешними характеристиками других, с учетом разрешения неоднозначностей множественного Р-наследования.

Декларативно-поведенческое программирование позволяет предельно декомпозировать построение сложной системы и отлаживать все декларативные описатели независимо друг от друга. Императивное программирование, во-первых, локализуется в программирование алгоритмически законченных методов-элементов и методов-событий. Во-вторых, оно заменяется функциональным программированием, поскольку методы-элементы и методы-события являются отображениями в математическом смысле, по крайней мере, в детерминированных (нестохастических) задачах.

Все перечисленные в начале работы сложности построения сложных систем оказываются строго отделены друг от друга: сложность структуры модели декларативно описывается иерархией комплексов и компонент; сложность и логика поведения компоненты – в декларативном описании правил чередования элементов процессов в зависимости от наступления тех или иных возможных событий; наконец, вся функциональность системы программируется (желательно в функциональной парадигме программирования) в методах-элементах и методах-событиях. Такая степень декомпозиции оказывается очень технологичной при разработке модели коллективом разработчиков: каждый разработчик или группа разработчиков может сконцентрировать максимум внимания на проектировании и отладке именно своего модуля, и тратить минимум сил на его взаимодействие со всей остальной системой.

Несомненно, что наряду с перечисленными выше достоинствами, ДПП имеет и свои недостатки. В первую очередь, также как и в САПР, за удобства модульности приходится платить увеличением накладных расходов, в число которых входит, в том числе и овладение концепцией ДПП. Например, всякий, кто познакомился с примером учебной имитационной модели (<http://simul.ccas.ru/Distr/fly/fly.htm>), несомненно отметит, что программист средней руки за полдня запросто напишет такую

модель от начала и до конца на каком-нибудь C# или VisualBasic, при этом не забывая себе голову всякими там ДПП или ЯОКК. В то же время автор вынужден признать, что даже он с помощью ДПП справился с задачей за несколько большее время, и конечно же, затратил на все эти дела больше усилий, чем, если бы от начала и до конца писал все на С.

В оправдание метода можно лишь сказать, что пушки отливают для великих битв, а не для стрельбы по воробьям и мухам. Начиная с некоторой степени сложности модели, накладные расходы начинают окупаться, а если модель и впрямь по-настоящему сложна – окупаются сторицей.

Отметим, что применение декларативного программирования в имитационном моделировании не ограничивается описанием устройства сложной системы. Если система и в самом деле достаточно сложна, всегда возникает вопрос рациональной организации ее данных, т. е. проектирования и описания базы данных. Общепринятым средством для этого является язык SQL, который применительно к задаче описания баз данных является декларативным, хотя может быть и императивным, если речь пойдет о выборках и обновлении данных.

Еще одна область имитационного моделирования, ожидающая плодотворного применения декларативных описаний – область подготовки презентаций результатов моделирования. Здесь автору не известны языки описания подобных презентаций. Тем не менее, необходимость в них явно имеется. Самое простое, но, тем не менее, весьма полезное, что можно было бы здесь сделать – это по таблице – выборке из базы данных строить диаграммы, так как это делают мастера диаграмм в MS Excel или в OpenOffice Calc. Второе – использование популярных в последнее время геоинформационных систем – отображение пиктограммами на интересующей карте динамики перемещения моделируемых объектов. Третье – использование простейшей анимации в стиле Flash – морфинг

форм и перемещение объектов с возможным изменением размеров. Более серьезная анимация пусть пока остается прерогативой компьютерных игр.

Основной вывод – модель сложной системы сложна не только сложностью своих имитационных вычислений (хотя, несомненно, эти вычисления могут быть весьма сложны) – на долю императивных вычислений практических ничего не остается, как мы видели, – их можно свести к вычислениям функциональным. Не менее сложны также организация данных модели, организация устройства модели (состав, связи, поведение компонент), а также организация отображения результатов моделирования. Последние три задачи обычно допускают декларативное описание.

Работа выполнена при поддержке РФФИ, грант № 10-07-00176-а.

Библиографический список:

1. Brodsky Yury I. Simulation Software //System Analysis and Modeling of Integrated World Systems – Volume 1, Oxford: EOLSS Publishers Co. Ltd., 2009, P. 287-298.
2. Brodsky Yury I., Tokarev Vladislav V. Fundamentals of simulation for complex systems. //System Analysis and Modeling of Integrated World Systems – Volume 1, Oxford: EOLSS Publishers Co. Ltd., 2009, P. 235-250.
3. Hewitt Carl Viewing Control Structures as Patterns of Passing Messages //Journal of Artificial Intelligence. June 1977.
4. Kuhl F., Weatherly R., Dahmann J. Creating Computer Simulation Systems: An Introduction to the High Level Architecture NY: Prentice Hall PTR, 1999. – 212 p.
5. Shoham Y. Agent-oriented programming //Artificial Intelligence, vol. 60, 1993, P. 51-92.
6. Shoham Y. MULTIAGENT SYSTEMS: Algorithmic, Game-Theoretic, and Logical Foundations Cambridge: Cambridge University Press, 2010, 532 p.

7. Zave, P. A compositional approach to multiparadigm programming. IEEE Software, 6(5): 15—25, September 1989.
8. Бродский Ю. И., Лебедев В. Ю., Огарышев В. Ф., Павловский Ю. Н., Савин Г. И. Общие проблемы моделирования сложных организационно-технических систем // Вопросы кибернетики. Проблемы математического моделирования и экспертные системы. – М.: Научный совет АН СССР по комплексной проблеме «Кибернетика», 1990. – С. 42-48.
9. Бродский Ю. И., Лебедев В. Ю. Инструментальная система для построения имитационных моделей хорошо структурированных организационно-технических комплексов // Вопросы кибернетики. Проблемы математического моделирования и экспертные системы. – М.: Научный совет АН СССР по комплексной проблеме «Кибернетика», 1990. – С. 49-64.
10. Бродский Ю. И., Лебедев В. Ю. Инструментальная система имитации MISS М.: ВЦ АН СССР, 1991. – 180 с.
11. Бродский Ю. И., Павловский Ю. Н. Разработка инструментальной системы распределенного имитационного моделирования. // Информационные технологии и вычислительные системы. – №4. – 2009. – С. 9-21.
12. Бродский Ю. И. Распределенное имитационное моделирование сложных систем. – М.: ВЦ РАН, 2010. – 156 с.
13. Бурбаки Н. Теория множеств. – М.: Мир. 1965. – 456 с.
14. Бусленко Н. П. Моделирование сложных систем. – М.: Наука, 1978. – 400 с.
15. Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя – 2-е изд. – М., СПб.: ДМК Пресс, Питер, 2004. – 432 с.
16. Осоргин А. Е. AnyLogic 6. Лабораторный практикум. – Самара: ПГК, 2011. – 100 с.

17. Павловский Ю. Н. Геометрическая теория декомпозиции и некоторые ее приложения. – М.: ВЦ РАН, 2011. – 93 с.

Информация об авторах

Data on authors

Бродский Юрий Игоревич, ведущий научный сотрудник ВЦ РАН, к.ф.-м.н., доцент, автор более 100 научных работ, в том числе 10 монографий. Область научных интересов – распределенное имитационное моделирование сложных систем, инструментальные средства моделирования. В течение последних лет читал лекции по математическому и компьютерному моделированию в МГТУ им. Н. Э. Баумана, МПГУ, МГУ им. М.В. Ломоносова.

Тел.: +7(926)416-49-03

E-mail: yury_brodsky@mail.ru

Brodsky Yury Igorevich, Leading Research Scholar at the Dorodnicyn Computing Center of the Russian Academy of Sciences. Associate Professor at several Moscow Universities, where delivered lectures on Mathematical Modeling and Computer Simulation. Current research interests center on Distributed Simulation of Complex Systems.

Tel.: +7(926)416-49-03

E-mail: yury_brodsky@mail.ru

ГРАФЫ РАБОТ ДЛЯ НЕКОТОРЫХ ВЫЧИСЛИМЫХ МОДЕЛЕЙ С РЕКУРСИЯМИ

WORKFLOW GRAPHS FOR SOME COMPUTABLE MODELS WITH RECURSIONS

Обсуждаются представления графами работ сценариев расчета характерных вычислимых компьютером математических моделей с рекурсиями. Представления ориентированы на расчеты, выполняемые в сети Internet с помощью разработанной сетевой библиотеки вычислимых моделей.

The representations by workflow graphs of calculation scenarios of typical computable models with recursions are discussed. The representations are focused on the calculations which are carried out in Internet network by means of developed network library of computable models.

Ключевые слова: распределенные вычисления, графы работ, представление моделей.

Keywords: distributed calculations, workflow graphs, representation of models.

Введение

В связи с нарастающим многообразием и сложностью знаний становится актуальным создание сетевых библиотек моделей. В [1-4] изложены концепция и архитектура сетевой библиотеки вычислимых математических моделей «Нива», результаты которых получают вычислениями на удаленном компьютере. К таким моделям можно отнести, например, имитационные модели.

Задача сетевой библиотеки – сохранять модели и предоставлять легкодоступные услуги использования моделей, прежде всего сложных, широкому кругу сетевых пользователей, как правило, не обладающих полным знанием модели. К таковым обычно относятся специалисты предметной области, для которых создаются модели.

В сетевой библиотеке модели представляются сценариями их расчета и, возможно, иного использования. Сценарий для расчета модели на удаленном компьютере пользователь самостоятельно конструирует на своем компьютере в виде графа работ модельных компонент. Это понуждает применять наиболее простые и безопасные представления моделей. Поэтому концепция сетевой библиотеки ориентирована на графическое представление пользователю структуры модели и сокрытие ее второстепенных деталей. Далее удаленный компьютер называется сервером. Сервер рассчитывает сценарий, и результаты пересылает пользователю.

Графы работ выбраны для представления моделей из-за их наглядности и универсальности. Они позволяют наглядно представлять многообразие сценариев расчета моделей с помощью композиций модельных компонент и организовать их хранение в базе данных сетевой библиотеки на сервере. Задавая различные входные данные, а также конструируя разные графы работ модельных компонент, пользователь может варьировать расчетную модель в предусмотренных пределах. Отметим, что модельной компонентой может быть исполняемый код любой степени сложности и назначения, причем даже не связанный с математическими вычислениями, например, интерпретатор или вычислитель символьных выражений.

Библиотека не передает пользователю код модели. С согласия автора библиотека может предоставлять пользователю текстовое описание модели. Это позволяет сохранять авторские права. Сетевая библиотека ориентирована на пользователей, а не авторов моделей.

Представление задач графами работ хорошо известно и используется в программировании, [5]. Обсуждаемые здесь графы работ выглядят иногда

слишком громоздкими для визуального конструирования, а их вычислительные возможности кажутся скромными. Однако заметим, что сетевая библиотека располагает мощными средствами поддержки работы с графами. Обмены данными и переходы между компонентами графа реализованы также весьма эффективно. Поэтому представляет интерес исследование вида графов работ, которые могут быть использованы для представления характерных вычислительных задач моделирования.

1. Графы работ и компоненты

Компонентой $M=\{F, D\}$ вычислимой модели называется исполняемый компьютером программный код, который реализует некоторую функцию F без параметров и исполняет ее над набором D специфических данных, называемых далее слотами. Компоненты, имеющие одни и те же F и D , но отличающиеся значениями данных из D , назовем МодОбъектами. Композицию МодОбъектов, представленных узлами в графе работ, назовем сценарием использования модели. Совокупность сценариев использования одной модели будем называть ГиперМоделью.

Каждому МодОбъекту приписывается содержательное имя вида $[m] [d]$, где m – имя для обозначения функциональности F компоненты $M=\{F, D\}$, а $[d]$ – имя набора значений слотов D этой компоненты. МодОбъекты вместе с данными их слотов сохраняются в базе данных сервера. Имена $[m] [d]$ используются для поиска МодОбъектов в базе данных и в графе работ. На графе можно не указывать $[d]$, если компонента имеет только один МодОбъект.

Сценарий визуально (мышкой) конструируется пользователем в виде графа работ. Сценарий рассчитывается удаленным сервером после ввода пользователем своих данных в компоненты и проверки сервером правильности графа. Результаты расчета сервер посылает пользователю на его компьютер.

Таким образом, модели в сетевой библиотеке представляются семействами допустимых сценариев – графов работ модельных компонент. Каждое семейство допустимых графов имеет формальное описание, хранимое в

библиотеке, в котором допустимые компоненты указываются именами вида [m] [d].

Под работой понимается исполнение компонентой $\{F, D\}$ своей функции F . Работы выполняются в последовательности, указанной стрелками графа работ.

Декомпозиция модели в граф работ поддерживается механизмом обмена данными между модельными компонентами с помощью слотов. Слотом здесь называется специальная переменная компоненты, имеющая следующие свойства. Как переменная слот имеет тип и значение. Слот может представлять массив чисел, строку символов и т. д. Перед началом вычислений слоты инициализируются начальными значениями. Эти значения либо встроены в код компоненты, либо задаются пользователем. Рабочая функция F компоненты $M=\{F, D\}$ может использовать значение своего слота $M.X$ как обычное данное и может изменять его значение. Функция F не может непосредственно изменять слоты других компонент.

Суть в том, что слоты различных компонент могут обмениваться значениями, если они связаны. Такими обменами поддерживается равенство значений связанных слотов. Так, если слот $M1.X$ компоненты $M1$ связан со слотом $M2.Y$ (это обозначается как $M1.X = @M2.Y$), тогда изменение компонентой $M1$ слота $M1.X$ немедленно вызывает такое же изменение значения слота $M2.Y$ другой компоненты. Таким образом, выполняя вычислительную работу, компонента может использовать результаты вычислительной работы других компонент, переданные ей через связанные с ними слоты.

Описания допустимых конфигураций графов работ и связей между слотами компонент помещаются в библиотеку вместе с компонентами модели. Используя такие описания, исполняющая подсистема библиотеки устанавливает связи между слотами перед началом вычисления заданного сценария. При этом если связанная слотами с компонентой $M1$ компонента $M2$

не будет найдена, значение слота M1.X не будет изменяться окружением компоненты M1.

Связи слотов могут изображаться на графе работ только для иллюстрации.

2. Представление характерных задач графами работ

Представление дерева выражения

Представление арифметических выражений деревьями выражений широко используется в программировании.

Рис. 1.1-1.3 иллюстрируют на примере простейшего выражения $F(x)=f(x)*h(z)+g(y)$ преобразование дерева выражения в граф работ на рис. 1.3. Рис. 1.2 иллюстрирует вычисление $F(x)$ с помощью наборов стандартных ячеек памяти. При переходе к графу работ на рис. 1.3 набор ячеек изображается узлом. Обмены данными, показанные пунктирными стрелками на рис. 1.2, на графе работ (рис. 1.3) не отображаются. При обходе графа работ по стрелкам, начиная с узла start, узлы выполняют операции (работы), указанные рядом с узлами. Подробности см. в [2], [3].

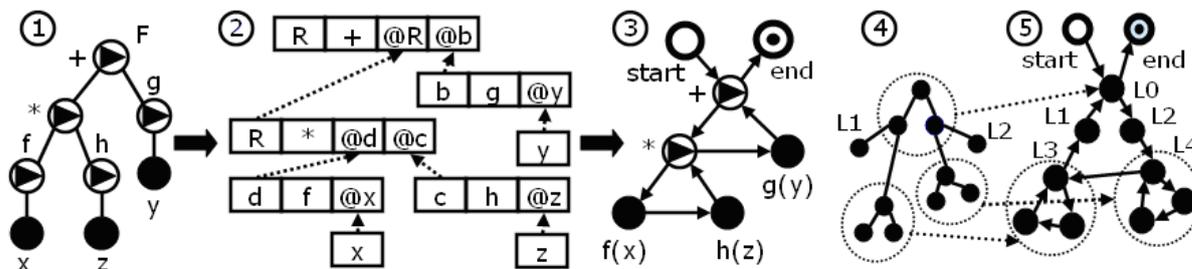


Рис. 1. Преобразование дерева выражения в граф работ

Разместив на сервере сетевой библиотеки множество элементарных компонент, выполняющих известные стандартные операции, можно позволить удаленному пользователю извлекать компоненты из библиотеки и так, мышкой, визуальнo конструировать графы работ для вычисления сложных выражений. В случае объемных графов используются деревья кольцевых подграфов работ, полученные агрегированием исходных графов работ. Для проверки корректности таких деревьев необходимы их описания, включающие описания

подграфов дерева и описания связей между слотами компонент подграфов для обмена данными. Описания хранятся в сетевой библиотеке. Рис. 1.4 и 1.5 иллюстрируют процедуру агрегирования объемных графов, [2], [4].

Представление дерева рекурсии

Рассмотрим представление рекурсий графами работ на примере простейшей рекуррентной последовательности

$$Y_n = 0.5 * (Y_{n-1} + A / Y_{n-1}), Y_1 = 1.5, n \geq 1, \text{сходящейся к } \sqrt{A}.$$

Выражение для Y_n является арифметическим выражением и для него можно привести очевидное дерево выражения (дерево рекурсии), аналогичное дереву на рис. 1.1. Описанной в предыдущем разделе процедурой это дерево можно преобразовать в графы работ, показанные на рис. 2. Здесь ромб изображает условную операцию if. Граф 2.2 отличается тем, что дерево выражения $Y_{n-1} + A / Y_{n-1}$ представлено кольцевым подграфом, а компонента if агрегирована с компонентой рекурсии [Recur][n]. Пунктирные стрелки изображают передачу связанными слотами компонент результатов вычислений. Чтобы избежать дублирующих вычислений, используются ссылочные компоненты, которые только копируют результат связанной компоненты. Простоты ради их имена на графах далее не указываются.

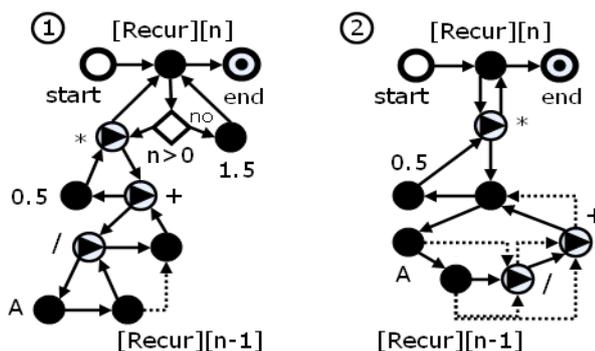


Рис. 2. Граф работ, представляющий рекуррентную последовательность Y_n

Рассмотрим рекурсивную функцию Func:

```
double Func(double A, int n)
```

```
{ return if n=1 then 1.5 else 0.5*(Func(A,n-1) + A/Func(A,n-1)); }
```

вычисляющую Y_n для действительного числа A и целого n . Напомним, как вычисляется Func. Чтобы вычислить Func (A,n), программа должна сначала вычислить Func ($A,n-1$). Если она не может этого сделать, например при $n>2$, она создает в памяти копию Func с параметрами ($A,n-1$), из которой вызывается Func ($A,n-2$). Если, вызвав Func ($A,n-2$), программа не может ее вычислить, опять создается ее копия, из которой вызывается Func ($A,n-3$). И так продолжается до тех пор, пока значение аргумента n в копии Func не окажется равным 1, и программа сможет вычислить Func ($A,1$). Теперь программа возвращается обратно к предыдущей копии и вычисляет предыдущую копию Func ($A,2$). Последовательно возвращаясь к предыдущей копии, она вычисляет результат Func(A,n).

Как и функция Func, компонента Recur создает копии подчиненного ей подграфа работ до тех пор, пока не сможет его вычислить. После этого вычисляются предыдущие копии подграфа в последовательности обратной к последовательности их создания копированием.

Представление задачи Коши, решаемой методом Рунге-Кутты

Рассмотрим часто решаемую задачу Коши для системы уравнений:

$$dx/dt = F(t, x) , x(0) = x_0 , x \in R^n , 0 \leq t \leq T . \quad (1)$$

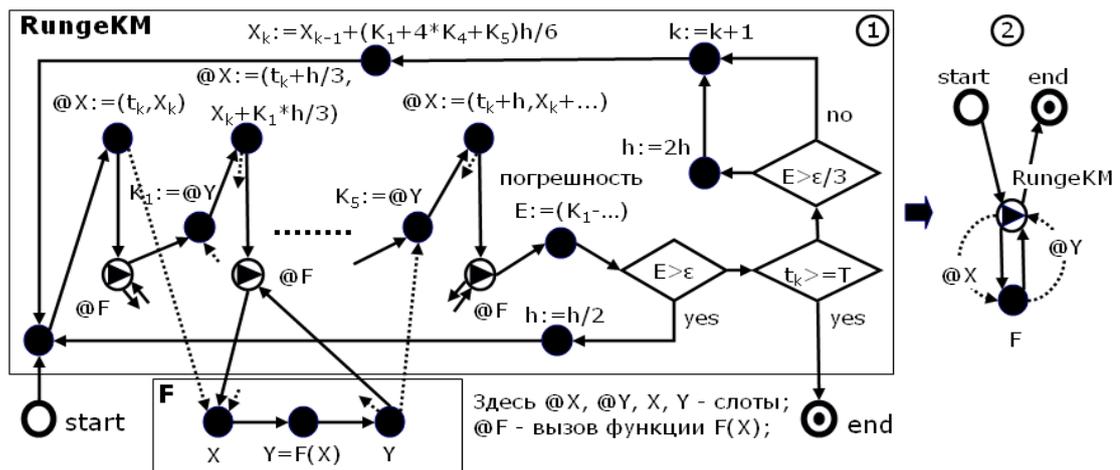


Рис. 3. Представление метода Рунге-Кутты-Мерсона графом работ

На рис. 3.1 в верхнем прямоугольнике RungeKM изображена блок-схема алгоритма Рунге-Кутты-Мерсона, а в нижнем прямоугольнике F – блок-схема

расчета правой части $Y = F(X)$, $X = (t, x)$, уравнения (1). Эти блок-схемы соответствуют компонентам RungeKM и F графа работ на рис. 3.2. Вызовы функции $F(X)$, выполняемые блоком RungeKM, соответствуют переходам по стрелке к компоненте F графа работ, исполнению работы F и возврату по стрелке обратно в компоненту RungeKM. Граф «работает» пока не достигнута заданная точность ε . Подробнее см. в [2].

Нетрудно видеть, что граф работ на рис. 3.2 представляет множество других задач, решаемых итерациями, например, задачу нахождения корня уравнения $F(x)=0$ или задачу поиска экстремума функции $F(x)$. Также компонента F может быть представлена последовательностью компонент или последовательностью деревьев [2].

Конструирование систем уравнений методом уточнения

Опишем весьма полезное свойство графов работ – метод уточнения. Поясним его на примере системы уравнений

$$dY/dt = H, H \equiv H(t, Y, @A, @B); \quad (2)$$

$$dZ/dt = G, G \equiv G(t, Z, @C, @D). \quad (3)$$

относительно неизвестных Y и Z , где $@A, @B, @C, @D$ – слоты в компонентах H и G , рассчитывающих правые части уравнений (2) и (3) соответственно. Пусть S – компонента, решающая системы вида (2) и (3), например, методом Рунге-Кутты-Мерсона, как было описано выше. Пусть при инициализации перед началом вычислений слотам присваиваются некоторые начальные значения A_0, B_0, C_0, D_0 , заданные пользователем.

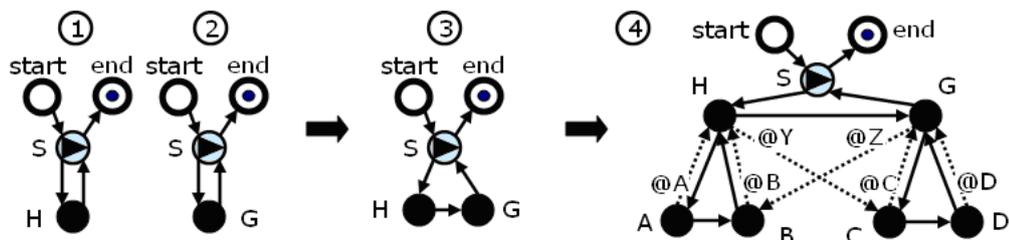


Рис. 4. Пример конструирования системы методом уточнения

Если при установлении связей между слотами перед началом вычислений, исполняющая подсистема библиотеки не обнаружит наличия в графе работ компонент А, В и С, D, подчиненных соответственно узлам Н и G, тогда значения слотов окажутся равными значениям, полученным при инициализации. В этом случае система (2)-(3) будет состоять из двух независимых подсистем (4)-(5) с постоянными $@ A = A_0$, $@ B = B_0$, $@ C = C_0$, $@ D = D_0$, представленных графом работ на рис. 4.3.

$$dY/dt = H(t, Y, A_0, B_0), \quad (4)$$

$$dZ/dt = G(t, Z, C_0, D_0). \quad (5)$$

Пусть теперь пользователь включил в граф работ, как показано на рис. 4.4, узлы А, В и С, D, вычисляющие, например, функции $A(Y)$, $B(Y, @Z)$, $C(@Y, Z)$, $D(Z)$ соответственно. Тогда после инициализации слотов начальными значениями исполняющая система библиотеки обнаружит подчиненные Н и G узлы А, В и С, D и установит связи между их слотами, как показано на рис. 4.4 пунктирными стрелками. В результате система уравнений окажется приведенной к виду (2)-(3) с параметрами $@ A = A(Y)$, $@ B = B(Y, Z)$, $@ C = C(Y, Z)$, $@ D = D(Z)$, т. е. к сопряженной системе уравнений

$$dY/dt = H(t, Y, A(Y), B(Y, Z)), \quad (6)$$

$$dZ/dt = G(t, Z, C(Y, Z), D(Z)). \quad (7)$$

Из этой системы также видно, что роли узлов А, D несколько отличаются от ролей узлов В, С. Узлы А и D не делают систему (2)-(3) сопряженной, а только изменяют функциональность каждого уравнения Н и G по отдельности. Напротив, узлы В и С, устанавливая обмен значениями переменных из разных уравнений, связывают их в сопряженную систему уравнений. Такое изменение функциональности узлов подчиненными узлами будем называть уточнением исходной модели (4)-(5). Уточнение исходной модели, выполняемое узлами А, D будем называть нисходящим уточнением, а уточнение путем сопряжения разных задач – уточнением сопряжения.

Нисходящее уточнение позволяет осмысленно организовать иерархию свойств модели. В такой иерархии верхнее положение занимают компоненты, реализующие самые общие свойства модели, допускающие нисходящее уточнение своих параметров и функциональности подчиненными компонентами второго уровня. Таких компонент может быть несколько, соединенных в графе последовательностью своих работ. В свою очередь компоненты второго уровня могут содержать параметры, нисходящим образом уточняемые подчиненными им компонентами третьего уровня. Заметим, что здесь просматриваются интересные аналогии с нисходящим уточнением функциональности в иерархиях объектов, широко используемым в объектно-ориентированных языках программирования для создания библиотек компонентов.

Уточнение сопряжением позволяет сопрягать разные задачи, имеющиеся в сетевой библиотеке. Интерес к этой возможности также обусловлен аналогией со ссылками на сопутствующие тексты, которые широко используются в HTML-текстах.

Нисходящее уточнение одной модели и сопряжение разных моделей можно отнести к фундаментальным свойствам обсуждаемых графов работ.

3. Метод прогонки

Решение системы уравнений

$$\begin{aligned} c_0 u_0 - b_0 u_1 &= f_0, \\ -a_i u_{i-1} + c_i u_i - b_i u_{i+1} &= f_i, \quad i = 1, 2, \dots, N-1; \\ -a_N u_{N-1} + c_N u_N &= f_N, \end{aligned} \quad (8)$$

при условиях

$$|c_i| \geq |a_i| + |b_i|, \quad i = 1, 2, \dots, N-1; \quad |c_0| \geq |b_0|, \quad |c_N| \geq |a_N| \quad (9)$$

находится как рекуррентная последовательность u_i

$$u_i = \alpha_{i+1} u_{i+1} + \beta_{i+1}; \quad i = N-1, N-2, \dots, 0; \quad u_N = \frac{f_N + a_N \beta_N}{c_N - a_N \alpha_N}, \quad (10)$$

где

$$\alpha_{i+1} = \frac{b_i}{c_i - a_i \alpha_i}, \beta_{i+1} = \frac{f_i + a_i \beta_i}{c_i - a_i \alpha_i}; \quad i = 1, 2, \dots, N-1; \quad (11)$$

$$\alpha_1 = c_0^{-1} b_0, \beta_1 = c_0^{-1} f_0$$

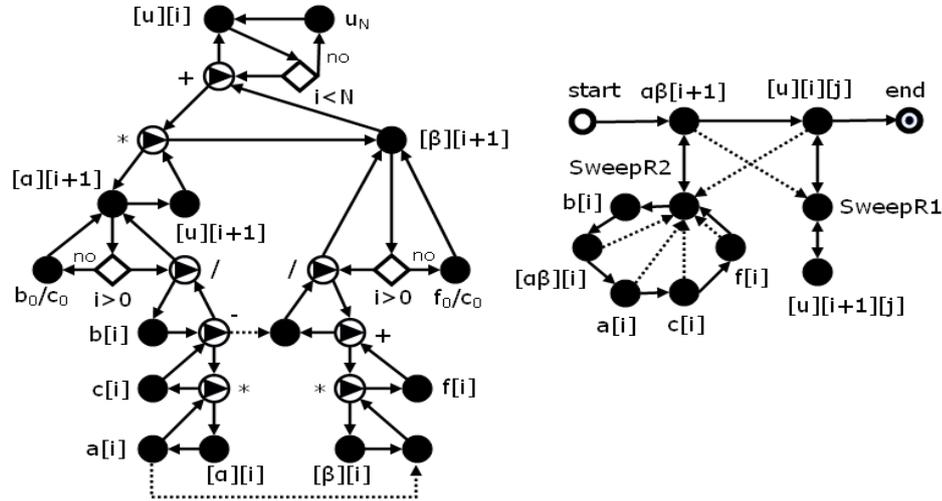


Рис. 5. Граф работ метода правой прогонки

Эти вычисления выполняет граф на рис. 5.1 и его агрегированный аналог на рис. 5.2. Компонента $\alpha\beta[i+1]$ помещает в массивы $\alpha[]$ и $\beta[]$ значения $\alpha[i+1]$ и $\beta[i+1]$, рассчитываемые компонентой SweepR2 согласно (11). Если компоненты $a[i]$, $b[i]$, $c[i]$, $f[i]$ уточняются подчиненными компонентами, как, например, на рис. 6, их значения вычисляются согласно методу уточнения; в противном случае $a[i] \dots f[i]$ равны постоянным, заданным пользователем при инициализации компонент. Если неизвестно значение $\alpha\beta[i]$, необходимое для вычисления $\alpha\beta[i+1]$, компонента $\alpha\beta[i]$ замещается копией подграфа, подчиненного $\alpha\beta[i+1]$, с параметром i вместо $i+1$ до тех пор, пока i не окажется равным 1. Тогда $\alpha\beta[1]$ вычисляет $\alpha[1] = c_0^{-1} b_0$, $\beta[1] = c_0^{-1} f_0$, и начинается вычисление предыдущих копий подграфа, т. е. прогонка по формулам (11).

После вычисления $\alpha[i]$ и $\beta[i]$ для $i = 1, \dots, N$ обратной прогонкой аналогично вычисляется последовательность $u[i]$. Компонента SweepR1

вычисляет значение $u_i = \alpha_{i+1}u_{i+1} + \beta_{i+1}$ и передает его компоненте $u[i+1]$, которая помещает его в массив $u[]$.

4. Двухфазная модель течения воды и воздуха в пористой среде

Рассмотрим пористую среду (почву), в порах которой протекают две «жидкости»: вода плотности ρ_w и воздух плотности ρ_a . Пусть вода занимает долю θ_w единичного объема среды, а воздух – долю θ_a . Обозначим через θ_p долю пор в объеме среды. Тогда $\theta_w + \theta_a = \theta_p$, плотность водной фазы равна $\theta_w\rho_w$, а плотность воздушной фазы – $\theta_a\rho_a$. Пусть в водной фазе растворены ионы вида $p = \{Na^+, Ca^{2+}, \dots\}$ с концентрациями C_p , кг-р/кг-Н₂O. Малый объем $\delta V_\alpha(t)$ α -фазы движется так, что сохраняется его масса, т. е. на траектории движения выполняются равенства

$$\frac{d}{dt} \int \theta_w \rho_w dV_w = - \int e_w dV_w, \quad \frac{d}{dt} \int \theta_a \rho_a dV_a = 0, \quad (12)$$

$$\frac{d}{dt} \int \theta_w \rho_w C_p dV_w = - \int \bar{j}_p d\bar{\sigma} + \int F_p dV_w.$$

Здесь j_p – диффузионный поток ионов вида p из объема $\delta V_w(t)$ через элемент его поверхности $d\bar{\sigma}$, F_p – функция источника ионов. Положив $\hat{\rho}_\alpha = \rho_\alpha / \rho_{w0}$, где ρ_{w0} – плотность ρ_w воды в нормальных условиях, получим дифф.-алгебраическую систему

$$\frac{\partial}{\partial t} (\theta_\alpha \hat{\rho}_\alpha) + \text{div}(\hat{\rho}_\alpha \bar{J}_\alpha) = f_\alpha, \quad \bar{J}_\alpha = -k_\alpha \nabla(\psi_\alpha - \hat{\rho}_\alpha \bar{d}z\bar{r}), \quad (13)$$

$$\alpha \in \{w, a\};$$

$$\theta_w \frac{\partial}{\partial t} C_p + \bar{J}_w \nabla C_p + \text{div}(D_p \nabla C_p) = \rho_w^{-1} F_p; \quad (14)$$

$$\theta_a + \theta_w = \theta_p, \quad \psi_a - \psi_w = \psi_c(\theta_w); \quad (15)$$

$$\hat{\rho}_\alpha = \hat{\rho}_\alpha(\psi_\alpha), \quad k_\alpha = k_\alpha(\theta_w), \quad \psi_w < 0, \quad \psi_c > 0, \quad \partial \psi_c / \partial \theta_w < 0;$$

Дифференцируя $\partial(\theta_\alpha \rho_\alpha) / \partial t$ по независимым переменным θ_w и ψ_w и усредняя по $[Z_{i-1/2}, Z_{i+1/2}]$, получим

$$A(\theta_w, \psi_w)_i \frac{d}{dt} \begin{bmatrix} \theta_w \\ \psi_w \end{bmatrix}_i + \tilde{h}_i^{-1} \begin{bmatrix} J_{w,i+1/2} - J_{w,i-1/2} \\ J_{a,i+1/2} - J_{a,i-1/2} \end{bmatrix} = \begin{bmatrix} f_w \\ f_a \end{bmatrix}_i, \text{ где} \quad (16)$$

$$J_{\alpha,i-1/2} = -(pk)_{\alpha,i-1/2} \left(\frac{\psi_{\alpha,i} - \psi_{\alpha,i-1}}{h_i} \psi_{\alpha} - \hat{\rho}_{\alpha,i-1/2} \right)$$

$$A(\theta_w, \psi_w)_i = \begin{bmatrix} \hat{\rho}_w & \theta_w \frac{\partial \hat{\rho}_w}{\partial \psi_w} \\ \theta_a \frac{\partial \hat{\rho}_a}{\partial \psi_a} \frac{\partial \psi_c}{\partial \theta_w} - \hat{\rho}_a & \theta_a \frac{\partial \hat{\rho}_a}{\partial \psi_a} \end{bmatrix}_i \quad (17)$$

ИЛИ

$$A_{w1,i} \dot{\theta}_{w,i} + A_{w2,i} \dot{\psi}_{w,i} + (-a_{w,i} \psi_{w,i-1} + c_{w,i} \psi_{w,i} - b_{w,i} \psi_{w,i+1}) + d_{w,i} = f_{w,i}; \quad (18)$$

$$A_{a1,i} \dot{\theta}_{w,i} + A_{a2,i} \dot{\psi}_{w,i} + (-a_{a,i} \psi_{w,i-1} + c_{a,i} \psi_{w,i} - b_{a,i} \psi_{w,i+1}) + d_{a,i} = f_{a,i}$$

где $A_{\alpha j,i}$ – j -я компонента строки α матрицы $A(\theta_w, \psi_w)$ из (10) в точке z_i . Здесь $\alpha = w$ соответствует первой, $\alpha = a$ – второй строке $A(\theta_w, \psi_w)$. Условиями сопряжения на границе слоев полагаем непрерывность потенциалов ψ_{α} и потоков J_{α} .

Положив в (16) $\dot{\psi}_{w,i} \approx \tau^{-1}(\psi_{w,i}^{j+1} - \psi_{w,i}^j)$, $\dot{\theta}_{w,i} \approx \tau^{-1}(\theta_{w,i}^{j+1} - \theta_{w,i}^j)$, получим систему (19)-(20), которую будем решать методом правой прогонки:

$$-\tilde{a}_{\psi,i} \psi_{w,i-1}^{j+1} + (\tilde{c}_{\psi,i} + |A_i| \tau^{-1}) \psi_{w,i}^{j+1} - \tilde{b}_{\psi,i} \psi_{w,i+1}^{j+1} = \tilde{F}_{\psi,i}, \quad (19)$$

$$\tilde{F}_{\psi,i} = |A_i| \tau^{-1} \psi_{w,i}^j + \tilde{f}_{\psi,i} - \tilde{d}_{\psi,i},$$

$$\theta_{w,i}^{j+1} = \theta_{w,i}^j + \tau |A_i|^{-1} \tilde{F}_{\theta,i}, \quad (20)$$

$$\tilde{F}_{\theta,i} = (\tilde{a}_{\theta,i} \psi_{w,i-1}^{j+1} - \tilde{c}_{\theta,i} \psi_{w,i}^{j+1} + \tilde{b}_{\theta,i} \psi_{w,i+1}^{j+1}) + \tilde{f}_{\theta,i} - \tilde{d}_{\theta,i}$$

где

$$\tilde{a}_{\psi,i} = a_{w,i} A_{a2,i} - a_{a,i} A_{w2,i}, \quad \tilde{b}_{\psi,i} = b_{w,i} A_{a2,i} - b_{a,i} A_{w2,i}, \quad \tilde{c}_{\psi,i} = \tilde{a}_{\psi,i} + \tilde{b}_{\psi,i}; \quad (21)$$

$$\begin{aligned} \tilde{d}_{\psi,i} &= d_{w,i}A_{a2,i} - d_{a,i}A_{w2,i}, \quad \tilde{f}_{\psi,i} = f_{w,i}A_{a2,i} - f_{a,i}A_{w2,i}; \\ \tilde{a}_{\theta,i} &= a_{a,i}A_{w1,i} - a_{w,i}A_{a1,i}, \quad \tilde{b}_{\theta,i} = b_{a,i}A_{w1,i} - b_{w,i}A_{a1,i}, \quad \tilde{c}_{\theta,i} = \tilde{a}_{\theta,i} + \tilde{b}_{\theta,i}; \end{aligned} \quad (22)$$

$$\begin{aligned} \tilde{d}_{\theta,i} &= d_{a,i}A_{w1,i} - d_{w,i}A_{a1,i}, \quad \tilde{f}_{\theta,i} = f_{a,i}A_{w1,i} - f_{w,i}A_{a1,i}; \\ a_{\alpha,i} &= \tilde{h}_i^{-1}h_i^{-1}(\rho k)_{\alpha,i-1/2}, \quad b_{\alpha,i} = \tilde{h}_i^{-1}h_{i+1}^{-1}(\rho k)_{\alpha,i+1/2}, \quad c_{\alpha,i} = a_{\alpha,i} + b_{\alpha,i}; \\ d_{\alpha,i} &= \tilde{h}_i^{-1}(\bar{k}_{\alpha,i+1/2}\hat{\rho}_{\alpha,i+1/2} - \bar{k}_{\alpha,i-1/2}\hat{\rho}_{\alpha,i-1/2}), \quad f_{\alpha,i} = f_{\alpha}(z_i). \end{aligned} \quad (23)$$

а $A_{w1,i}, A_{w2,i}, A_{a1,i}, A_{a2,i}$ – элементы матрицы $A(\theta_w, \psi_w)$ в точке z_i .

Систему (19)-(23) решает граф работ на рис. 6, полученный методом уточнения графа на рис. 5. Имена компонент графа отвечают обозначениям (19)-(23) и приводятся здесь без скобок. Имена ссылочных компонент опущены. Очевидно, можно упрощать граф, агрегируя его компоненты.

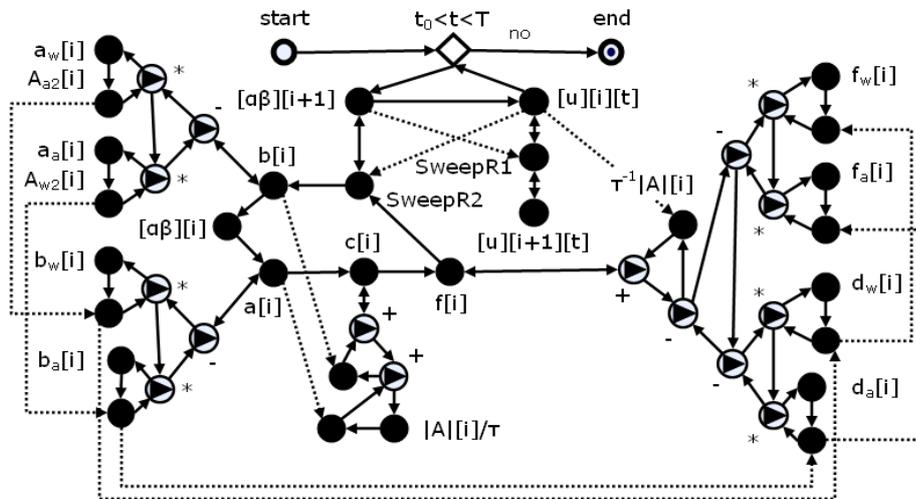


Рис. 6. Граф работ для решения двухфазной модели

5. Уравнение Ричардса

Рассмотрим частный случай (16), когда $\partial \hat{\rho}_w / \partial \psi_w = 0$, $\psi_a = Const$. Тогда $\rho_w = 1$, а θ_w становится зависимой от ψ_w в силу $\psi_a - \psi_w = \psi_c(\theta_w)$. Дифференцируя последнее по ψ_w , из (16) получим уравнение Ричардса в полудискретной форме (16)

$$r(\psi_w) \frac{d}{dt} \psi_{w,i} + \tilde{h}_i^{-1} (J_{w,i+1/2} - J_{w,i-1/2}) = f_{w,i}, \quad r(\psi_w) = -\frac{\partial \theta_w}{\partial \psi_c}, \quad (24)$$

Уравнение (24) решается следующим графом работ, полученным из графа на рис. 5 методом уточнения аналогично графу на рис. 6.

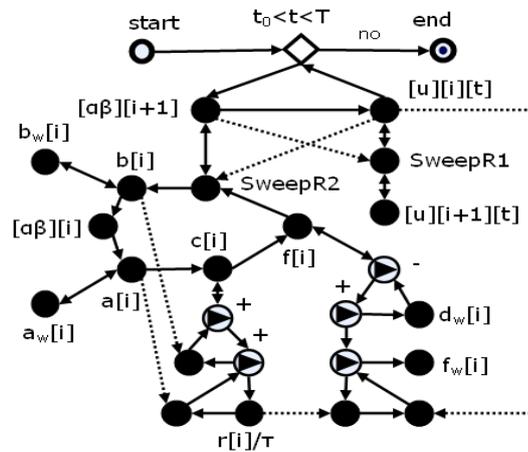


Рис. 7. Граф работ для решения уравнения Ричардса

Библиографический список:

1. Воротынцев А. В. Концепция сетевых информационно-вычислительных библиотек моделей. – М.: Вычислительный центр РАН, 2009. – 109 с.
2. Воротынцев А. В. О представлении вычислимых моделей графами работ // Труды V Всероссийской научной конференции «Математическое моделирование развивающейся экономики, экологии и биотехнологий – ЭКОМОД–2010», Киров, 5–11 июля 2010. – 2010. – С. 45–54
3. Воротынцев А. В. Представление вычислимых моделей иерархическими автоматами в сетевой библиотеке // Вестник БГУ. – 2011. – Вып. 9. – С. 87-88
4. Воротынцев А. В. Некоторые характерные графы работ для представления сценариев расчета математических моделей // Вестник БГУ. – 2012. – Вып. 9. – С. 147-153
5. Кознов Д. В. Основы визуального моделирования. – М.: Интернет-Университет информационных технологий. БИНОМ лаборатория знаний, 2008. – 246 с.

Информация об авторах

Data on authors

Воротынцев Александр Васильевич – канд. физ.-матем. наук, научный сотрудник ВЦ РАН

Тел.: (+7966) 135-3931

E-mail: avv_alexv@mail.ru

Vorotyntsev Alexander Vasilievich – Cand. Sc. (physical and mathematical sciences), research fellow of CCAS

Tel.: (+7966) 135-3931

E-mail: avv_alexv@mail.ru

**ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ ЛИНЕЙНЫМИ
СИСТЕМАМИ С НЕРЕГУЛЯРНЫМИ СМЕШАННЫМИ
ОГРАНИЧЕНИЯМИ И ОПРЕДЕЛЕНИЕ ГЕОМЕТРИИ
ОПТИМАЛЬНОЙ ТРАЕКТОРИИ ПРИ ПОМОЩИ
РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ**

**OPTIMAL CONTROL OF LINEAR SYSTEMS WITH
NONREGULAR CONTROL-STATE CONSTRAINTS AND
DETERMINATION OF THE GEOMETRY OF THE OPTIMAL
TRAJECTORY BY DISTRIBUTED CALCULATIONS**

Предложен двухэтапный метод решения задач оптимального управления со смешанными ограничениями. На первом этапе методами факторного анализа при помощи распределенных вычислений решается дискретная задача системы линейных алгебраических уравнений и несобственные задачи линейного программирования (ЛП). Далее формулируется гипотеза о геометрии оптимальной траектории. На втором этапе эта гипотеза проверяется аналитически по схеме Дубовицкого-Милютин. Дан пример использования этой схемы для решения модельной задачи оптимального управления долгом промышленного предприятия.

A two-step method for solving optimal control problems with control-state constraints is proposed. At first a discrete problem for system of linear algebraic equations and improper linear programming is solved by methods of factor analysis and distributed calculations. Then a hypothesis on geometry of optimal trajectory is formulated. In the second stage the hypothesis is checked

analytically by Dubovitsky–Milyutin’s scheme. An example of this scheme use for a model problem of optimal debt management for industrial firm is given.

Ключевые слова: оптимальное управление, двухэтапный метод, схема Дубовицкого–Миллютина, распределенные вычисления.

Key words: optimal control, two-step method, Dubovitsky–Milyutin’s scheme, distributed calculations.

Введение

В настоящей работе исследуются несовместные системы линейных алгебраических уравнений и несобственные задачи ЛП. Фундаментальные свойства совместности или несовместности линейных моделей, заданных смешанными системами уравнений и неравенств, связаны с теоремами об альтернативах – основой теорем существования решений для задач оптимизации, а также инструментом их эффективного численного решения благодаря методу штрафных функций и функций Лагранжа. Для решения многомерных задач добавим здесь распределенные вычисления.

Актуальность исследования несобственных математических объектов обозначена И. И. Ереминым; в частности, он указал, что в теории математических моделей прослеживается эволюция в сторону ослабления требований на исследуемый математический объект: единственность решения и устойчивость; единственность, неустойчивость (некорректность); неединственность и неустойчивость; несобственность; несобственность и плохая формализуемость; гибкое моделирование и т. д.

Несовместная (несобственная) модель не позволяет получить содержательную информацию об исследуемом процессе или явлении непосредственно. Наиболее общую форму коррекции модели называют матричной, она заключается в изменении коэффициентов левых и правых частей соответствующих уравнений и неравенств.

В методах матричной коррекции, рассмотренных А. А. Ватолиным, затем уточненных В. А. Гореликом и его учениками, остались нерешенными важные проблемы. Одна проблема связана с неединственностью решения задачи матричной коррекции, другая – с выбором показателя качества матричной коррекции, диктуемого прикладной задачей. Часто выбор связан со статистической гипотезой (евклидова норма – нормальное распределение ошибок, чебышевская – равномерное распределение, октаэдрическая – распределение со случайными выбросами). Все это влияет на методы исследования и решения задач матричной коррекции [1].

1. Факторный анализ в решении задач линейного программирования

Были изобретены различные способы уменьшения размерности данных в линейных моделях с потерей наименьшего количества информации: метод главных компонент, метод независимых компонент, многомерное шкалирование, а также многочисленные их нелинейные обобщения. Вычисление главных компонент сводится к вычислению собственных векторов и собственных значений ковариационной матрицы исходных данных.

Основная модель факторного анализа имеет вид [2]:

$$X = Lf + e, \quad X = (x_1, \dots, x_n)^T, \quad f = (f_1, \dots, f_k)^T, \quad e = (e_1, \dots) \quad (1.1)$$

Здесь x – вектор-столбец наблюдаемых переменных, f – вектор-столбец общих факторов, e – вектор-столбец специфических факторов, влияющих только на данную переменную.

Предполагается, что они не коррелированы и между собой, и с общими факторами; L – матрица факторных нагрузок.

Из (1.1) следует

$$R = L\Phi L^T + V, \quad (1.2)$$

где R – ковариационная (корреляционная) матрица наблюдаемых переменных; Φ – корреляционная матрица общих факторов (с учетом безразмерного характера общих факторов их дисперсии без ограничения общности могут быть приняты равными единице), V – ковариационная матрица специфических факторов, являющаяся диагональной.

В предположении о некоррелированности общих факторов между собой матрица Φ становится единичной; и мы получаем классическую модель факторного анализа.

2. Параметризация метода Якоби

Метод Якоби в окрестности решения имеет квадратичную сходимость. Метод сходится гораздо быстрее при специальной параметризации.

Вместо нахождения собственных значений матрицы $A + E$ рассмотрим матрицу, зависящую от параметра

$$B(\alpha) = (1 - \alpha)E + \alpha(A + E), \quad 0 \leq \alpha \leq 1.$$

Здесь E – единичная матрица, α – параметр задачи.

Процесс итерации начинается при $\alpha_0 = 0$. В этом случае $B(0) = E$. Далее полагаем $\alpha_1 = \alpha_0 + \Delta\alpha_k$ и используем в качестве первого приближения $B(0)$. Так как $\Delta\alpha$ мало, то скорость сходимости к решению будет квадратичной. В результате получим решение проблемы собственных значений для матрицы $B(\alpha_1)$:

$$B(\alpha_1) = (1 - \alpha_1)E + \alpha_1(A + E).$$

Этот процесс можно продолжить вплоть до $\alpha = 1$, однако в этом нет нужды. Достаточно вычислить собственные значения для $\alpha = \alpha_* < 1$ (например $\alpha_* = 0.1$). В этом случае собственные значения λ определяются по формуле

$$\Lambda = \frac{\Lambda_* - E}{\alpha_*},$$

где Λ_* – диагональная матрица собственных значений для $\alpha = \alpha_*$.

Полученное значение Λ можно использовать в качестве первого приближения для вычисления собственных значений матрицы A .

Среди других методов получения первого приближения можно указать итеративную декомпозицию матрицы Якоби и метод малого параметра. Пусть R – матрица наблюдений. Представим матрицу R в виде

$$R = aA + (1-a)B, \quad 0 \leq a \leq 1,$$

где B – трехдиагональная матрица, a – параметр. $A = R - B$. Здесь применяется метод продолжения по параметру. Существует много эффективных методов для приведения матрицы B к диагональному виду.

Далее приведем метод снижения размерности матрицы R при дискретизации дифференциального уравнения

$$t = v\tau, \quad \varepsilon \frac{dx}{d\tau} = f(x, u), \quad \tau \in [0, 1], \quad \varepsilon = \frac{1}{v}.$$

В результате найдено жесткое уравнение (сингулярно возмущенное).

3. Задача линейного программирования

Задача линейного программирования имеет вид

$$\begin{aligned} \max(C, X), \quad X \in \mathbb{R}^n, \quad C > 0, \quad C \in \mathbb{R}^n, \quad A_1 X = B_1, \\ X \geq 0, \quad A_1 > 0, \quad B_1 > 0, \quad A_1 \in \mathbb{R}^{m \times n}, \quad B_1 \in \mathbb{R}^m. \end{aligned} \quad (3.1)$$

Здесь B_1 и C – положительные вектора, A_1 – положительная матрица (положительны ее компоненты).

Задача в форме, зависящей от скалярного параметра t

$$\begin{aligned} \max t, \quad A_2 X = B_2, \quad (C, X) = t, \\ A_2 = \begin{pmatrix} C \\ A_1 \end{pmatrix}, \quad B_2 = \begin{pmatrix} t \\ B_1 \end{pmatrix}, \quad X \geq 0. \end{aligned} \quad (3.2)$$

Задача (3.2), вообще говоря, может не иметь решения. В этом случае рассматривается обобщенная задача ЛП

$$\max(t), \quad A_2^T A_2 Y = A_2^T B_2, \quad Y \geq 0. \quad (3.3)$$

Система (3.3) всегда совместна.

Вместо задачи (3.3) рассматривается следующая задача:

$$\begin{aligned} \max(t), \quad A_\delta Z_\delta = B, \quad Z_\delta \geq 0, \\ A_\delta = A + \delta E, \quad A_2^T A_2 = A, \quad B = A_2^T B_2. \end{aligned} \quad (3.4)$$

Известно, что $\lim_{\delta \rightarrow 0} Z_\delta = Z_0 \in \{Y\}$, где $\{Y\}$ – множество решений задачи (3.3).

Решение линейной системы (3.3) – линейная функция параметра t :

$$Z_\delta = C_1 + tC_2, \quad C_1, C_2 \in \mathbb{R}^n.$$

Условие $Z_\delta \geq 0$ выполняет неравенство

$$t_1 \leq t \leq t_2.$$

Если решение не ограничено сверху, то $t_2 = \infty$. Так что оценка решений сводится к решению некорректной линейной системы.

4. Решение системы линейных алгебраических уравнений

Рассматривается произвольная линейная система

$$AX = B, \quad (4.1)$$

где A – действительная матрица размером $m \times n$. Система (4.1) является совместной, если она имеет хотя бы одно решение.

Метод регуляризации Тихонова сводит (4.1) к решению уравнения

$$(A_0 + \delta I)Z = B_0, \quad \delta > 0, \quad A^T A = A_0, \quad A^T B = B_0, \quad A_1 = A \quad (4.2)$$

Вводится множество арифметических векторов размерности n

$$Z_1 = 10\dots 1^T, \quad Z_2 = 010\dots 0^T, \dots, \quad Z_n = 00\dots 1^T. \quad (4.3)$$

В результате применения оператора A_1 к системе (4.3) получаем систему, которая представляет собой матрицу наблюдений A_1

$$A_1 = (B_1 B_2 \dots B_n),$$

которая действительна и симметрична, причем ранг A_1 равен n для $\delta > 0$.

Применение факторного анализа (метода главных компонент) дает следующее решение:

$$B_j = \sum_{i=1}^p a_{ji} f_i, \quad p \leq n, \quad f_k = \sum_{i=1}^n \beta_{ki} B_i, \quad k=1, \dots, p, \quad j=1, \dots, n. \quad (4.4)$$

Здесь f_k являются факторами, a_{ij} и β_{ki} – коэффициентами факторного решения. Теоретически $p = n$. Любой вектор B_0 представим в виде линейной комбинации базисных векторов f_k , ортогональных с единичной нормой:

$$B_0 = \gamma_1 f_1 + \gamma_2 f_2 + \dots + \gamma_n f_n, \quad \gamma_k = (B_0, f_k). \quad (4.5)$$

Подставляя f_k из формулы (4.4) в выражение (4.5), получим

$$B_0 = b_1 B_1 + b_2 B_2 + \dots + b_n B_n.$$

Утверждение. Решение системы (4.4) является линейной комбинацией (4.5)

$$Z = b_1 Z_1 + b_2 Z_2 + \dots + b_n Z_n. \quad (4.6)$$

В случае сингулярной матрицы $p < n$ при $\delta = 0$, так как коэффициенты a_{ji} вычисляются по формуле

$$a_{ji} = b_{ji} \sqrt{\frac{\lambda_i}{\sqrt{b_{1i}^2 + b_{2i}^2 + \dots + b_{ni}^2}}}.$$

Теорема 1. Решение (4.6) является непрерывной функцией параметра δ . Если система (4.1) имеет единственное решение, то $p = n$. В противном случае мы имеем нормальное решение (4.6).

5. Минимизация долга промышленного предприятия

Модель описывает двухсекторную экономическую систему, первый сектор которой добывает ресурсы, а второй производит продукцию конечного потребления (включая фондообразующую) [3]. Погашение долга происходит за счет продажи продукции первого сектора. Минимизируемый функционал (долг)

$$J(x,u) = x_3(100), \quad 0 < t < T = 100.$$

Обозначения для количественных характеристик системы:

- $x_1(t), x_2(t)$ – суммарный объем инвестиций в секторы 1 и 2 за время $[0, t]$;
 $x_3(t)$ – долг в момент времени t ;
 $x_4(t), x_5(t)$ – суммарная амортизация фондов в 1 и 2 секторах за время $[0, t]$;
 $\alpha_i, i = 1, 2$ – коэффициент производительности труда в i -том секторе;
 $\beta_i, i = 1, 2$ – коэффициент амортизации фондов в i -том секторе;
 $u_1(t)$ – поток инвестиций в первый сектор;
 $u_2(t)$ – поток фондообразующей продукции в первый сектор;
 $u_3(t)$ – поток инвестиций во второй сектор;
 $u_4(t)$ – поток фондообразующей продукции во второй сектор;
 $u_5(t)$ – поток импорта потребительских товаров;
 $u_6(t)$ – объем выпуска продукции первым сектором (на экспорт);
 $u_7(t)$ – объем выпуска второго сектора (на конечное потребление);
 c_2 – минимальный объем непроемственного потребления;
 k – коэффициент для сопоставления внутренних и внешних цен;
 μ_2 – коэффициент учета необходимости обслуживания долга;

Система дифференциальных уравнений и ограничения задачи

$$\begin{aligned}
 \dot{x}_1 &= u_1 + u_2, & u_{i \min} &\leq u_i \leq u_{i \max}, i = \overline{1, 7}, \\
 \dot{x}_2 &= u_3 + u_4, & u_6 &\leq \alpha_1(x_1 - x_4), \\
 \dot{x}_3 &= \mu_3 x_3 + u_1 + u_3 + u_5 - k_1 u_6, & u_7 &\leq \alpha_2(x_2 - x_5), \\
 \dot{x}_4 &= \beta_1 u_6, & u_2 + u_4 &\leq u_7, \\
 \dot{x}_5 &= \beta_2 u_7, & u_2 + u_4 + c &\leq u_7 + u_5
 \end{aligned}$$

Значения коэффициентов задачи, начальных и конечных условий

$$\begin{aligned}
 \alpha_1 &= 0.03, & x_1(0) &= 25, & u_{i \min=0}, & t = \overline{1, 7}, \\
 \alpha_2 &= 0.03, & x_2(0) &= 15, & u_{1 \max} &= u_{3 \max} = 0.1, \\
 \beta_1 &= 0.18, & x_3(0) &= 400, & u_{2 \max} &= 0.5, \\
 \beta_2 &= 0.18, & x_4(0) &= 0, & u_{4 \max} &= u_{5 \max} = 0.3, \\
 k_1 &= 0.9, & x_5(0) &= 0, & u_{6 \max} &= u_{7 \max} = 2.0, \\
 \mu_3 &= 0.002, & x_1(T) - x_4(T) &= 25, & c &= 0.2. \\
 T &= 100, & x_2(T) - x_5(T) &= 15, & &
 \end{aligned}$$

Программа факторного анализа состоит из основной программы, программы ввода DATA и пяти подпрограмм: CORRE, EIGEN, TRACE, LOAD и VARMX, модифицированных для работы с задачами ЛП с большим числом переменных. Эта программа позволяет удобно описывать и решать дискретные аппроксимации задач ОУ при большом числе точек дискретизации. Решение задачи ЛП показано на рис. 1-4:

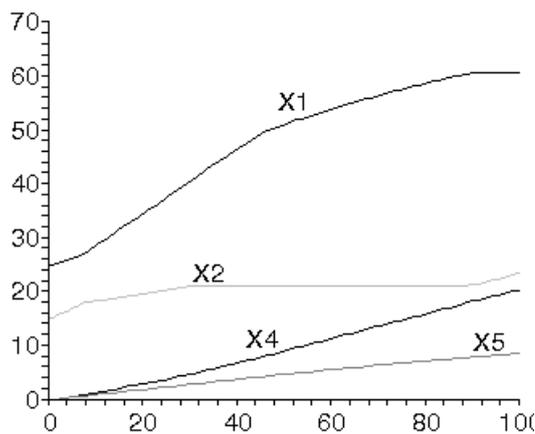


Рис. 1. Фазовые

переменные x_1, x_2, x_4, x_5

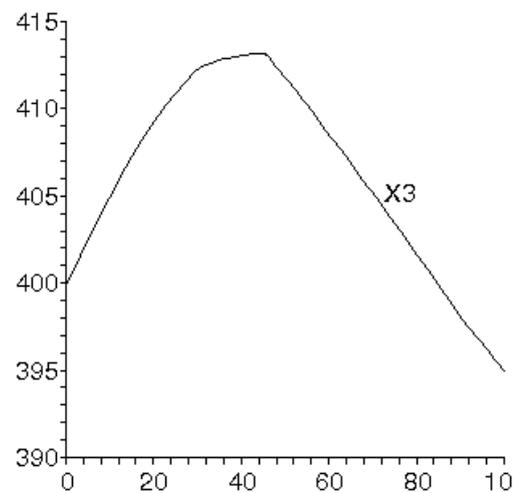


Рис. 2. Фазовая переменная x_3

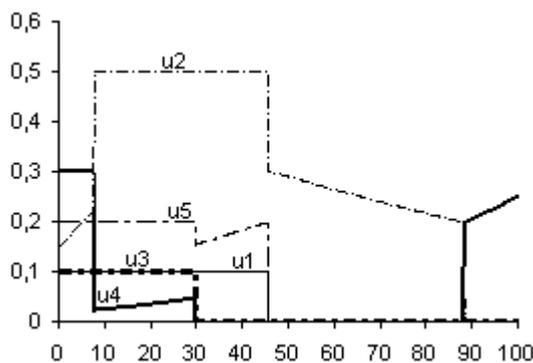


Рис. 3. Управления $u_1 - u_5$

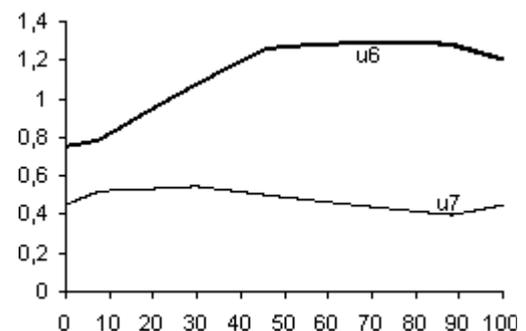


Рис. 4. Управления u_6, u_7

Нахождение численного решения при огромном числе точек дискретизации при последовательном вычислении занимает значительное время, которое растет с ростом числа точек. Расчет можно значительно ускорить за счет распределенных вычислений на кластерном суперкомпьютере. Распараллеливание расчета в цикле по точкам

дискретизации осуществлено с помощью технологии параллельных вычислений в интерфейсе передачи сообщений MPI [4]. Расчет осуществлялся на суперкомпьютерах МСЦ РАН и ВятГУ. Здесь представлен модельный пример, такое распараллеливание существенно для реальных практических задач.

Аналитическое решение для данной модели построено на основе принципа максимума [4-6]. Высказана гипотеза о том, что траектория системы состоит из пяти промежутков постоянства индексов активных ограничений. На каждом промежутке из активных ограничений выражались управления и аналитически решалась задача Коши, при этом момент времени переключения полагался параметром. Сопряженная система интегрировалась справа налево, с параметрами, являющимися моментами времен переключений. Значения времен переключений определялись из решения трансцендентных уравнений, связывающих фазовые переменные и сопряженные переменные задачи.

6. Нерегулярные точки

Для нахождения фазовых точек введем обозначения:

$$h_6(x, u) = u_6 - \alpha_1(x_1 - x_4) \leq 0, \quad u_6 = u,$$

$$h_7(x, u) = u_7 - \alpha_2(x_2 - x_5) \leq 0, \quad u_7 = v,$$

$$g_6(x, u) = u_6 - u_{6\max} \leq 0, \quad g_7(x, u) = u_7 - u_{7\max} \leq 0.$$

В этом случае получаем следующие соотношения:

$$\frac{d}{du}(g_6 - h_6) = 0 \Rightarrow x_1 - x_4 = 0 \Rightarrow \dot{x}_1 - \dot{x}_4 = 0,$$

$$\frac{d}{dv}(g_7 - h_7) = 0 \Rightarrow x_2 - x_5 = 0 \Rightarrow \dot{x}_2 - \dot{x}_5 = 0,$$

$$L = \sum_{i=1}^5 \dot{x}_i \psi_i + \lambda_6 h_6 + \lambda_7 h_7 + h_6 \frac{d\mu_6}{dt} + h_7 \frac{d\mu_7}{dt},$$

$$\dot{\psi}_i = -\frac{\partial L}{\partial x_i}.$$

Здесь $\frac{d\mu_6}{dt}, \frac{d\mu_7}{dt}$ – обобщенные функции.

7. Каноническая задача Дубовицкого–Милютина

Приведем схему Дубовицкого–Милютина для канонической задачи с гладкой зависимостью от времени. Каноническая задача имеет вид:

$$\begin{cases} \min J(p), & p = (x(t_0), x(t_1), t_0, t_1), \\ \dot{x} = f(x, u, t), & K(p) = 0, \quad \varphi(p) \leq 0, \\ g(x, u, t) = 0, \\ \Phi(x, u, t) \leq 0, \\ u = (u_1, u_2), & u_2 \in R, \end{cases}$$

Где $x \in E^n$, $u_i \in E^{k_i}$, $i = 1, 2$, $\dim f = n$, $\dim g = r_1$, множество $R \subset E^{k_2}$ – произвольное, вектор-функции φ и Φ имеют любые размерности $\dim \Phi = r_2$, $\dim \varphi = r_3$.

Предположения:

1) Функции $f(x, u, t)$, $K(p)$, $g(x, u, t)$ вместе со своими частными производными по x , u_1 , p непрерывны относительно x , u , t , p в некоторой окрестности поверхности $K = 0$, $g = 0$;

2) $\text{rang } g_{u_i} = \dim g \leq k_i$ для всех точек поверхности $g = 0$;

3) Функции J , φ , Φ локально выпуклы по x , u_1 , p , t . Размерности вектор-функций φ и Φ произвольны;

Минимум ищется в классе всех измеримых ограниченных функций $x(t)$, $u(t)$.

Приведем несколько необходимых нам определений.

Локально выпуклые функции

Непрерывная функция $f(z, y)$, определенная в области G конечномерного пространства z, y называется локально выпуклой по y , если:

1) $\left. \frac{d}{d\varepsilon} f(z, y + \varepsilon \bar{y}) \right|_{\varepsilon=0} = f'(z, y, \bar{y})$ существует при всех $z, y, \bar{y} \in G$ и является

сублинейной функцией \bar{y} ;

2) $f'(z, y, \bar{y})$ – полунепрерывная сверху функция z, y , т. е.

$$\overline{\lim_{\substack{z \rightarrow z_0 \\ y \rightarrow y_0}} f'(z, y, \bar{y})} \leq f'(z_0, y_0, \bar{y});$$

3) На единичной сфере $|\bar{y}|=1$ имеет место

$$|f(z, y + \varepsilon \bar{y}) - f(z, y) - \varepsilon f'(z, y, \bar{y})| = O(\varepsilon).$$

Производная по направлению $f'(z, y, \bar{y})$ называется вариацией нелинейной функции $f(z, y)$.

Сублинейность по \bar{y} означает однородность, т. е.

$$f'(z, y, \alpha \bar{y}) = \alpha f'(z, y, \bar{y}), \quad \alpha \geq 0.$$

Вектор функция $\varphi(z, y)$ локально выпукла по y , если ее составляющие локально выпуклы по y .

Класс локально выпуклых по y функций пространства z, y содержит все непрерывные функции, выпуклые по y .

Регулярные ограничения

Решение канонической задачи Дубовицкого-Милютинна существенным образом зависит от структуры фазовых ограничений. Более того, общий запас фазовых точек определяется изначально структурой смешанных ограничений. Дадим определения разных видов структуры ограничений.

$$g(x, u, t) = 0, \quad \Phi(x, u, t) \leq 0 \quad \text{и} \quad u_2 \in \bar{R};$$

$$V(x, t) = u \mid u = (u_1, u_2), \quad u_2 \in \bar{R}, \quad g = 0, \Phi \leq 0.$$

Смешанные ограничения $g(x, u, t) = 0, \quad \Phi(x, u, t) \leq 0$ называются регулярными, если для каждой точки x, u, t , где $u(t) \in V(x(t), t)$ выполнено одно из условий: 1) множество активных индексов $j(x, u, t)$ для смешанных ограничений типа неравенств пусто; 2) если множество $j(x, u, t)$ не пусто, то

существует вектор \bar{u}_1 , такой что $(\Phi_{u_k}, \bar{u}_1) < 0$ для всех $k \in j(x, u, t)$, $(g_{u_1}, \bar{u}_1) = 0$, где производная по локально выпуклой функции Φ_{u_k} есть квазиградиент.

Точки x, u, t , где $u(t) \in V(x(t), t)$, для которых выполняется одно из условий регулярности, называются точками регулярности для смешанных ограничений $g(x, u, t) = 0, \Phi(x, u, t) \leq 0$.

Класс регулярных ограничений существенно шире класса независимых ограничений, которые действуют в задаче Блисса–Больца.

Фазовые ограничения

Каждая точка $x, u, t, u(t) \in V(x(t), t)$, не являющаяся точкой регулярности смешанных ограничений, называется фазовой. Можно дать также аналитическое определение фазовой точки.

Точка $x, u, t, u(t) \in V(x(t), t)$ – фазовая, если существуют векторы α, β , $\|\alpha\| = \sum_i \alpha_i = 1, \alpha_i \geq 0$, такие, что

$$(\Phi_{u_i}(x, u, t), \alpha) - (g_{u_i}(x, u, t), \beta) = 0, \quad \alpha_i \Phi_i = 0.$$

Если ограничения g, Φ не зависят от управления u_1 , то все точки $x, u, t, u(t) \in V(x(t), t)$ – фазовые.

Если на множестве активных индексов $j(x, u, t)$ градиенты g_{u_i}, Φ_{u_i} линейно независимы, то все точки $x, u, t, u(t) \in V(x(t), t)$ – регулярные.

Замыкание по мере для функции

Пусть на числовой прямой задана ограниченная измеримая функция $g \in E^n$ и измеримое множество E , причем $\text{mes } E > 0$. Тогда точка принадлежит замыканию по мере функции $g, t \in \overline{g(E)}$, если для любых окрестностей V_t, V_g точек t, g

$$\text{mes } V_t \cap g^{-1}(V_g) \cap E > 0.$$

Иными словами, замыканием по мере функции $g(t)$ называется замыкание по мере ее графика.

Приведем интегральный принцип максимума в регулярном случае.

Определим функцию Понтрягина, функцию Гамильтона и краевую функцию Лагранжа:

$$\Pi(x, u, \psi^x, \psi^t, t) = (\psi^x, f) + \psi^t,$$

$$H(x, u, \psi^x, \psi^t, t, m(t), n(t)) = \Pi + (m(t), g) - (n(t), \Phi),$$

$$l(p) = \alpha_0 J(p) + (\beta_0, \varphi(p)) + (C, K(p)),$$

где $\dim \beta = \dim \varphi$.

Пусть $x_0(t)$, $u_0(t)$, t_0 , t_1 — экстремаль задачи. Тогда существуют константа $\alpha_0 \geq 0$, вектор-константы $\beta_0 \geq 0$, C и ограниченные измеримые вектор-функции $m(t)$, $n(t)$ такие, что

$$\beta_{0_i} \varphi_i(p_0) = 0, \quad i = 1, \dots, r_3$$

$$\max_{u \in V(x, t)} \Pi(x_0(t), u, \psi^x, \psi^t, t) = \Pi(x_0(t), u_0(t), \psi^x, \psi^t, t)$$

$$\dot{\psi}^x = -H_x(x_0(t), u_0(t), \psi^x(t), \psi^t(t), t, m(t), n(t)),$$

$$\dot{\psi}^t = -H_t(x_0(t), u_0(t), \psi^x(t), \psi^t(t), t, m(t), n(t)),$$

$$\psi^x(t_0) = l_a(p_0), \quad \psi^x(t_1) = -l_b(p_0),$$

$$\psi^t(t_0) = l_{t_0}(p_0), \quad \psi^t(t_1) = -l_{t_1}(p_0).$$

Множители Лагранжа $m(t)$, $n(t)$ определяются из условий Блесса, условия дополняющей нежесткости и уравнений связи

$$H_{u_i}(x_0, u_0, \psi^x, \psi^t, t, m(t), n(t)) = 0$$

$$n_j(t) \Phi_j(x_0, u_0, t) = 0, \quad n_j \geq 0, \quad j \leq r_2$$

$$g(x, u, t) = 0, \quad \Phi_j(x, u, t) \leq 0, \quad j \leq r_2.$$

Выполняется условие нормировки

$$\alpha_0 + \|\beta_0\| + \|C\| + \|m(t)\| + \|n(t)\| > 0,$$

где $\|\beta_0\| = \sum_{i=1}^{r_3} |\beta_{0_i}|$.

Для формулировки интегрального принципа максимума π_0 нам потребуется понятия фазового скачка.

Фазовый скачок

Фазовым скачком точки x, u, t называется вектор

$$\gamma = \Phi_{x,t}(x, u, t)\alpha - (g_{x,t}(x, u, t), \beta).$$

Совокупность всех фазовых скачков точки x, u, t обозначается через $\Gamma(x, u, t)$.

Интегральный принцип максимума π_0

Пусть $x_0(t), u_0(t), t_0, t_1$ – экстремаль задачи (4.6.1). Тогда существуют функции ограниченной вариации

$$\psi^x \in BV^n, \quad \psi^t \in BV^1,$$

функции

$$m(t) \in L_1^1, \quad n(t) \in L_1^2, \\ \Phi_{x,u,t}(x_0(t), u_0(t), t) \in L_\infty^{(n+k_1+1)}$$

такие, что почти всюду на $t \in [t_0, t_1]$

$$n_j(t)\Phi_j(x_0, u_0, t) = 0, \quad n_j \geq 0, \quad j \leq r_2$$

и выполнены условия

- 1) $\Pi(x_0(t), u, \psi^x, \psi^t, t) \leq 0$ для всех $u, t, u \in U_0(t), t \in [t_0, t_1]$;
- 2) $\Pi(x_0(t), u_0(t), \psi^x, \psi^t, t) = 0$ почти всюду на $t \in [t_0, t_1]$;
- 3) $H_{u_i}(x_0(t), u_0(t), \psi^x(t), \psi^t(t), t, m(t), n(t)) = 0$ почти всюду на $t \in [t_0, t_1]$.

Компоненты принципа максимума связаны следующими соотношениями. Существуют конечная мера $\mu, d\mu \geq 0$, функция $\gamma(t)$, измеримая по мере $d\mu$, для которой почти всюду в смысле меры $d\mu$ имеет место $\gamma(t) \in \tilde{\Gamma}(x_0(t), \overline{u_0(t)}, t)$, константа $\alpha_0 \geq 0$, вектор-константы $\beta_0 \geq 0, C, \beta_{0,i} \varphi_i(p_0) = 0, i = 1, \dots, r_3$ такие, что

$$1) -d\psi^{x,i}(t) = H_{x,i} dt - \gamma(t)d\mu, \quad \psi_{np}^{x,i}(t_0) = l_{x_0,t_0}(p_0) + \gamma(t_0)\mu(t_0), \quad \psi_{ng}^{x,i}(t_0) = -l_{x_1,t_1}(p_0) - \gamma(t_1)\mu(t_1);$$

2) в каждой точке разрыва меры μ

$$\psi_{ng}^{x,i}(t) = \psi^{x,i} - \mu_{ng}(t)\gamma_{ng}(t),$$

$$\begin{aligned}\psi_{np}^{x,t}(t) &= \psi^{x,t} - \mu_{np}(t)\gamma_{np}(t), \\ \mu_{лв,np}(t) &\geq 0, \quad \mu_{лв}(t) + \mu_{np}(t) = \mu(t), \\ \gamma_{лв,np}(t) &\in \tilde{\Gamma}(x_0(t), u_0(t)).\end{aligned}$$

3) принцип максимума π_0 нормирован неравенством

$$\alpha_0 + \|\beta_0\| + \|C\| + \int_{t_0}^{t_1} \|n(t)\| dt + \|\mu\| > 0$$

Работа выполнена при финансовой поддержке РФФИ. Коды проектов: №10-08-00624, №11-07-00201, №12-01-00916.

Библиографический список:

1. Горелик В. А., Ерохин В. И., Оптимальная матричная коррекция несовместных систем линейных алгебраических уравнений и несобственных задач линейного программирования // Моделирование, декомпозиция и оптимизация сложных динамических процессов. – М.: ВЦ РАН, 2004. – С. 35–63.
2. Харман Г. Современный факторный анализ. – М.: Статистика, 1972. – 420 с.
3. Умнов А. Е., Умнов Е. А., Чекарев Д. А. Параметрический двухуровневый метод решения задач оптимального управления со смешанными ограничениями // Моделирование процессов управления. М.: МФТИ, 2004. – С. 68–73.
4. Оленев Н. Н. Основы параллельного программирования в системе МРІ. М.: ВЦ РАН. 2005. – 80 с.
5. Дубовицкий А. Я., Милютин А. А. Необходимые условия экстремума в некоторых линейных задачах со смешанными ограничениями / Вероятностные процессы и управление. – М.: Наука, 1978. – С. 42–74.

6. Дикусар В. В., Кошьяка М., Фигура А. Задачи оптимального управления при наличии ограничений общего вида. – М.: МФТИ, 2001. – 185 с.

7. Дикусар В. В., Шомполова О. И. Применение факторного анализа для решения задач линейного программирования // Тр. ИСА РАН. – 2010. – Т. 53(3). – Вып. 14. – Стр. 142–153.

Информация об авторах

Data on authors

Дикусар Василий Васильевич, д.ф.-м.н., профессор, главный научный сотрудник отдела прикладных проблем оптимизации ФГБУН ВЦ РАН

Тел.(499) 135-00-80

E-mail a: dikussar@yandex.ru

Dikusar Vasily, professor, chief researcher at the Department of Applied Problems of Optimization FGBUN CCAS

Tel: (499) 135-00-80

E-mail: dikussar@yandex.ru

Оленёв Николай Николаевич, к.ф.-м.н., доцент, старший научный сотрудник ВЦ РАН, доцент кафедры математического моделирования в экономике ФГБОУ ВПО «ВятГУ».

Тел.(499) 783-33-28

Электронная почта: nolenev@mail.ru

Olenev Nicholas, docent, senior researcher at CCAS, Associate Prof. on Math Modeling in Economics FGEI HPE «VyatSU».

Tel: (499) 783-33-28

E-mail: nolenev@mail.ru

**ПРЕДСКАЗАНИЕ БИОЛОГИЧЕСКОГО СОСТОЯНИЯ МОЛЕКУЛ
ИСХОДЯ ИЗ ИХ ХИМИЧЕСКИХ СВОЙСТВ**

**PREDICTION OF A BIOLOGICAL RESPONSE OF MOLECULES
FROM THEIR CHEMICAL PROPERTIES**

Многомерные данные, при использовании значительно большего количества признаков относительно меньшего числа наблюдений, порождают хорошо известную проблему переобучения. В связи с этим, задача построения системного автоматического метода для оценки качества произведенного решения приобретает особую актуальность. Опираясь на механизм однородного ансамблирования (известный также как бэггинг), мы предлагаем конструировать паспорт решения как единую высококачественную валидационную траекторию применительно ко всем тренировочным данным. Исходя из предположения, что паспорта зеркально отображают тестовые решения, мы можем использовать их для рассмотрения многих важных задач, включая оптимизацию смесей и ансамблей, вычисление сдвиговых погрешностей и любое другое тестирование. В работе приведены результаты, основанные на данных, которые использовались при проведении одного из наиболее популярных международных соревнований по анализу данных на платформе Kaggle.

Ensembles are often capable of greater prediction accuracy than any of their individual members. As a consequence of the diversity between individual base-learners, an ensemble will not suffer from overfitting. In this regard, development of the systematic and automatic approach for the evaluation of the

solution is particularly important. Based on the mechanism of homogeneous ensembling (known, also, as bagging), we can construct a passport of the solution as unified validation trajectory against all available training data. Assuming that passports mimic closely the corresponding test solutions, we can use them for the consideration of many tasks including optimizations of blends and ensembles, calculation of the bias and any other tests as required. The reported results were obtained online during one of the most popular International data mining competition on the Kaggle platform in 2012.

Ключевые слова: ансамбль и элементарный классификатор, скользящий контроль, отбор признаков, вторичные признаки, бустинг, случайный лес, дерево решений.

Kew words: ensemble and base learner, cross-validation, feature selection, secondary features, boosting, random forests, decision tree

Введение

Ансамбли (включая методы голосования и усреднения) – это алгоритмы обучения, которые основаны на множестве элементарных классификаторов (base-learners) [1]. Как правило, ансамбли элементарных решений вычисляются по методу выборочного среднего. Хорошо известен тот факт, что ансамбли зачастую являются более точными, чем составляющие их элементарные классификаторы [2]. Например, ансамбль решающих деревьев, или так называемый «случайный лес» (random forest) [3], может быть использован в качестве иллюстрации эффективного классификатора.

В классической постановке любой элементарный классификатор (или, так называемый, слабый ученик) базируется на случайно отобранном подмножестве наблюдений и случайно отобранных признаках. Согласно построению, мы можем рассмотреть эл. классификаторы как независимые случайные величины с математическим ожиданием θ и стандартным

отклонением σ , где задача состоит в оценке θ . Тогда, арифметическое среднее эл. классификаторов (ансамбль) будет иметь базовые параметры $\{\theta, \frac{\sigma}{\sqrt{m}}\}$, где m есть размер ансамбля или количество элементарные классификаторов. Последнее означает, что вероятность среднего квадратичного уклонения ансамбля от θ стремится к нулю с ростом m .

Однако ситуация на практике может оказаться значительно сложнее. Кроме того, оценивающий критерий совсем не обязательно совпадает со среднеквадратичным отклонением. Отметим также, что вычисление однородного ансамбля представляет собой достаточно трудоемкий процесс, так как размер ансамбля m должен быть достаточно большим (как правило, $m \geq 100$) для того, чтобы обеспечить высокий уровень качества оценки. Поэтому представляется целесообразным вычислять качество ансамбля (паспорт) параллельно с вычислением самого ансамбля.

Мы можем дополнить совершенно естественным образом процесс вычисления однородного ансамбля формированием оценочной траектории применительно ко всем наблюдениям из тренировочной базы данных. Эта траектория может быть рассмотрена как сильная оценка, поскольку, по определению, она была сформирована как ансамбль слабых оценок. В разделе 3 приведено детальное описание предлагаемого метода.

Работа структурирована следующим образом: в разделе 2 мы предлагаем структурный подход, основанный на сравнении дистанций, для отбора наиболее влиятельных признаков с целью уменьшения негативного эффекта от перетренировки; в разделе 4 мы рассматриваем использование паспортов отдельных решений (базовых учеников) для конструирования неоднородного ансамбля; в разделе представлены методы для формирования новых признаков; наконец, раздел 6 подводит итог данной работы.

1. Соревнование Boehringer и используемые данные

Международное соревнование по анализу данных Boehringer было организовано на платформе Kaggle¹ и привлекло 703 команды в качестве активных участников (третье соревнование по популярности за всю историю Kaggle). Соревнование продолжалось 91 день в период с 16 марта по 15 июня 2012 года. Наш официальный результат (лучший из пяти специально отобранных решений) был 8-м. Однако наш абсолютно лучший результат значительно превосходит результат победителя (в ходе соревнования наша команда подготовила и официально представила 177 различных решений).

Таблица 1

Список наших лучших результатов (см. рис.1)

№	Дата	Public	Private	Комментарий
1	07 Jun 2012	0.42536	0.37093	N79 in Public
2	04 Jun 2012	0.41060	0.37163	
3	04 Jun 2012	0.41622	0.37207	
4	03 Jun 2012	0.41110	0.37262	
5	08 Jun 2012	0.41074	0.37288	
6	06 Jun 2012	0.40957	0.37288	
7	07 Jun 2012	0.43239	0.37314	
8	05 Jun 2012	0.41060	0.37316	лучше результата победителя
9	01 Jun 2012	0.41627	0.37459	
10	12 Jun 2012	0.40595	0.37483	
11	03 Jun 2012	0.41208	0.37519	
12	08 Jun 2012	0.41108	0.37554	
13	05 Jun 2012	0.41188	0.37568	
14	15 Jun 2012	0.40139	0.37597	N8 in Private out of 5 selected
15	01 Jun 2012	0.41442	0.37637	
22	13 Jun 2012	0.39831	0.37807	N4 in Public

¹ <http://www.kaggle.com>

Данные для соревнования были предоставлены международной фармацевтической компанией Boehringer. Тренировочная (значения целевой функции известны) и тестовая части данных включают 3751 и 2501 наблюдений соответственно. Под каждым наблюдением подразумевается молекула, которая охарактеризована при помощи 1776 признаков. Следует отметить, что интерпретация конкретных признаков (аналоги генов в случае микрочипов) не предоставлена.

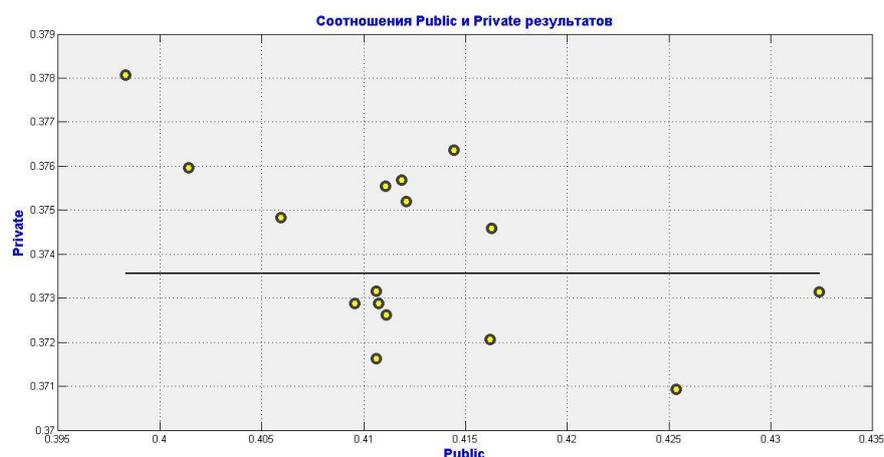


Рис.1. Иллюстрация соотношения Private/Public данных из Таблицы 1, где горизонтальная линия соответствует результату победителя в соревновании Boehringer.

Данные для тестирования были поделены на две части в соотношении 1/3, где меньшая часть использовалась для предварительного тестирования (Public) через LeaderBoard и большая часть использовалась для подведения заключительных результатов (Private).

Для оценки качества представленных решений использовался следующий критерий:

$$LogLoss = -\frac{1}{N} \sum_{i=1}^N (y_i \log(\hat{f}_i) + (1 - y_i) \log(1 - \hat{f}_i)),$$

где N – число наблюдений, y и \hat{f} – целевая переменная (бинарная) и её оценка.

Замечание 1. Согласно определению *LogLoss* предпочтение будет отдано той оценке, которая лучшим образом аппроксимирует вероятность интересующего нас события.

2. Структура дистанционных критериев для отбора признаков

Удачный отбор признаков может значительно снизить негативный эффект от переобучения. Для отбора признаков мы использовали специальные критерии, основанные на сравнении дистанций между «своим» и «чужим» центрами:

$$r_i = \frac{1}{N} \left(\sum_{j \in I_0} \frac{d_1(i, j) + \gamma}{d_0(i, j) + \gamma} + \sum_{j \in I_1} \frac{d_0(i, j) + \gamma}{d_1(i, j) + \gamma} \right), \quad (1)$$

где i и j – индексы соответствующие признаку и наблюдению, $\gamma > 0$ – регулирующий параметр,

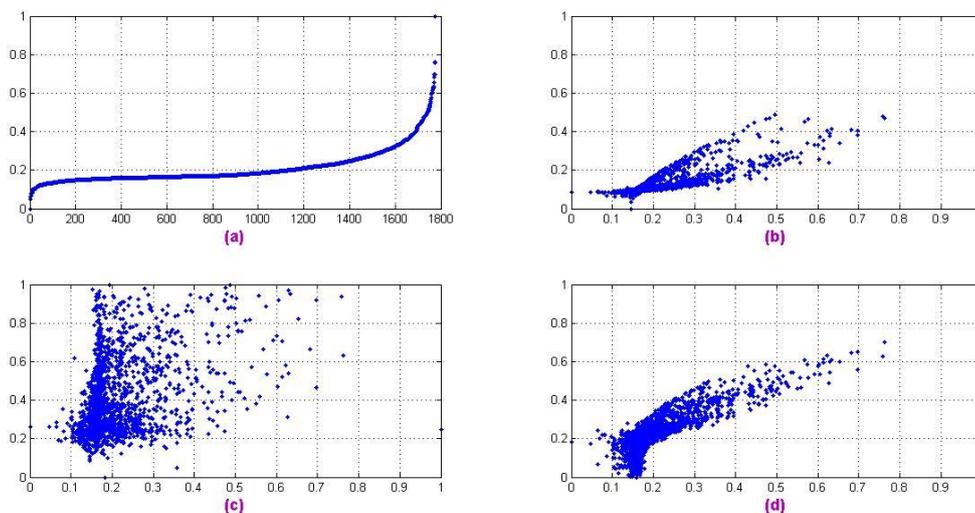


Рис.2. (a) Рейтинги 1776 признаков упорядоченные в возрастающей последовательности; отношения (b) r и R ; (c) r и Wilcoxon; (d) r и Fisher Discriminant (определения критериев Wilcoxon и Fisher Discriminant даны в [4])

$$d_k(i, j) = |x_{ij} - x_k(i)|, x_k(i) = \frac{1}{N_k} \sum_{j \in I_k} x_{ij}, N_k = \#I_k, k \in \{0, 1\},$$

где x_{ij} – признак, $x_k(i)$ – центр соответствующий метке k .

Используя идею (1) в качестве структурного подхода, мы можем определить новый критерий при использовании различных расстояний. Например, мы можем применить экспоненциальную функцию

$$R_i = \frac{1}{N} \left(\sum_{j \in I_0} \exp(\alpha(d_1(i, j) - d_0(i, j))) + \sum_{j \in I_1} \exp(\alpha(d_0(i, j) - d_1(i, j))) \right), \quad (2)$$

где $\alpha = 100$ – регулирующий параметр.

Таблица 2

Список наиболее важных признаков, где показатель важности был вычислен согласно (1) при использовании регулирующего параметра $\gamma=0,03$ (значения были умножены на 1000 и округлены до ближайшего целого)

№	Индекс	Важность
1	27	2028
2	659	1376
3	469	1346
4	83	1269
5	182	1252
6	1281	1249
7	217	1244
8	747	1243
9	209	1228
10	595	1220

Замечание 2. Рис.2 иллюстрирует сходство критерия (1) и Fisher Discriminant, который также основан на сравнении расстояний (аппроксимационный тип). Wilcoxon критерий принадлежит другому классу разделителей. Соответственно, Рис.2 (с) иллюстрирует значительные расхождения при отборе признаков.

3. Однородное ансамблирование и вычисление валидационных паспортов для решающих функций

В прямом соответствии с принципами однородного ансамблирования, представляется целесообразным рассмотреть вычисление решающей функции как среднее большого количества базовых решающих функций (эл. классификаторов), где построение каждого эл. классификатора опирается на случайную подвыборку и подмножество случайно отобранных признаков. Отметим, что отбор признаков может быть осуществлен при использовании дистанционных критериев (1) или (2).

Предположим, что для построения отдельного эл. классификатора мы использовали 70% имеющихся данных. Оставшиеся 30% данных могут быть использованы для оценочного контроля. В соответствии с принципом скользящего контроля, мы предлагаем исследовать стабильность оценочных результатов рассматривая последовательность случайных разбиений.

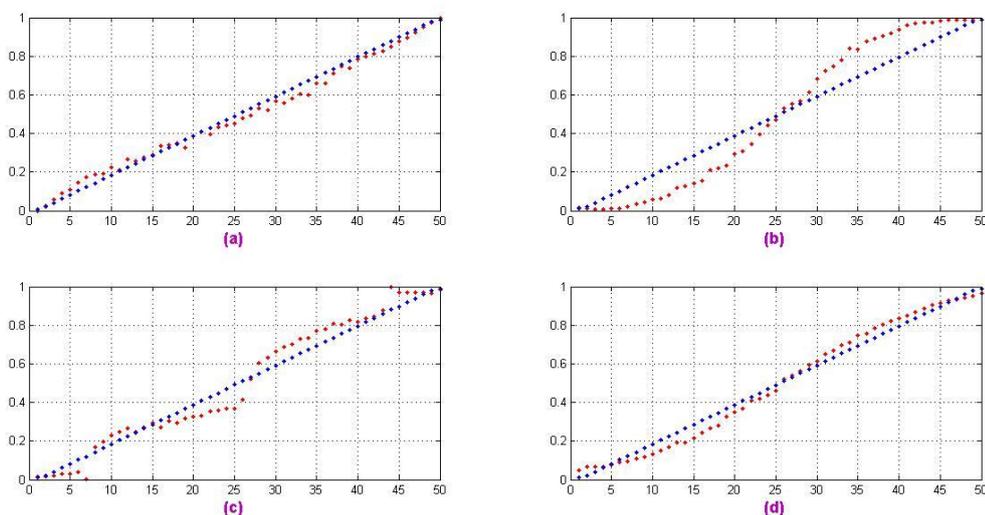


Рис.3. Сдвиговые смещения в случае эл. классификаторов (a) GBM, (b) kridge, (c) SVC и (d) NN

где GBM соответствует одноименному пакету в R, остальные функции доступны в Matlab (CLOP package). Синие точки соответствуют

усредненным предсказаниям, красные точки соответствуют реальным вероятностям.

Основная новизна и идея нашего метода состоит в следующем. Мы предлагаем накапливать оценочные (или валидационные) результаты для всех наблюдений из обучающей выборки параллельно вычислению однородного ансамбля. Усредненную оценочную траекторию мы будем называть паспортом, соответствующем однородному ансамблю.

В ходе соревнования Voehringer мы использовали до 1500 случайных разбиений в соотношении 1/3, где большая часть использовалась для тренировки, а меньшая для формирования паспорта и для промежуточного тестирования. Результаты промежуточного тестирования в терминах LogLoss лежали в интервале от 0.3899 до 0.48 (GBM использовался в качестве эл. классификатора). Результаты, соответствующие паспортам, приведены в Таблице 3.

Таблица 3

Тестовые результаты паспортов при использовании различных моделей

Модель	Результат
GBM	0.4274
RF	0.4302
kridge	0.4594
SVC	0.483
NN	0.4938

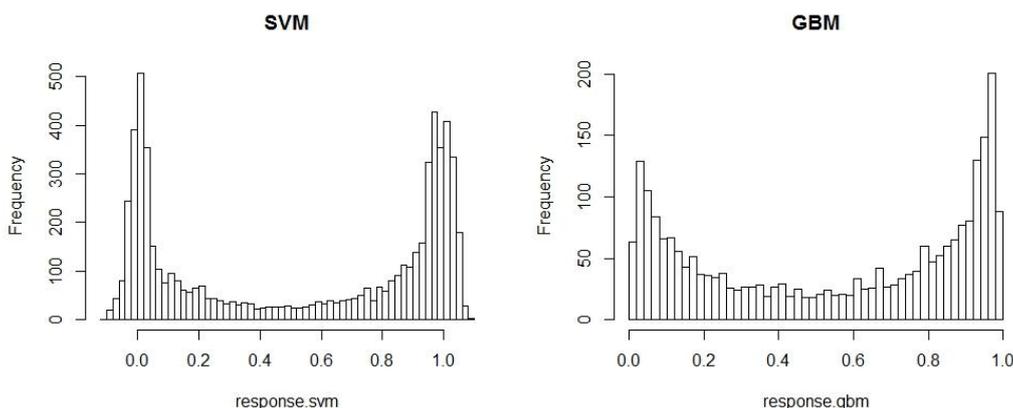


Рис.4. Гистограммы решений SVM и GBM

Отметим также, что паспорта могут быть использованы для различных целей. Например, рис.3 иллюстрирует сдвиговые смещения (отклонения прогнозируемых и реальных вероятностей) для различных моделей. Согласно нашему опыту, корректировка согласно сдвиговым смещениям приводит к существенному улучшению качества аппроксимации оценки вероятности интересующего нас события. Кроме того, в следующем разделе предложен метод неоднородного ансамблирования, который опирается на паспорта отдельных решений.

4. Кусочно-линейные комбинации различных решений основанные на соответствующих паспортах и их обобщения (неоднородное ансамблирование)

В качестве примера, предположим, что мы имеем два решения, полученных при использовании различных методов 1) GBM (gradient boosting machine, метод градиентного бустинга) [5] и 2) SVM (support vector machine, метод опорных векторов) [6]. Как можно заметить (см. рис. 4), первый алгоритм является более «осторожным» и

количество значений на концах интервала $(0,1)$ существенно меньше, чем во втором случае.

Опираясь на результаты скользящего контроля, мы можем сформировать новое решение (или оценку) как линейную комбинацию исходных (базовых) решений. Однако такой подход может оказаться довольно грубым, поскольку соотношение качественных характеристик составляющих решений не является однородным. Поэтому представляется целесообразным разбить все двумерное пространство решений на отдельные части и затем оптимизировать коэффициенты специально для каждой части.

4.1. Кусочное ансамблирование (обобщение и структура нашего лучшего решения)

Таблица 4

Структура ансамблированного решения соответствующего

$$\text{LogLoss}=0.37093$$

	$response.svm \leq 0.1$	$0.1 < response.svm < 0.9$	$response.svm \geq 0.9$
$response.gbm \leq 0.1$	min	min	0.55
$0.1 < response.gbm < 0.9$	0.1	mean	0.9
$response.gbm \geq 0.9$	0.75	max	max

Вообще говоря, ансамблированное решение совсем не обязательно вычисляется как линейная комбинация исходных решений. Например, в Таблице 4 дано определение нелинейного ансамбля с девятью (3×3) кусочными подмножествами, который произвел лучший известный результат в соревновании Boehringер.

Генерация новых признаков.

В разделе 2 мы рассмотрели критерий для фильтрации несущественных признаков с тем, чтобы уменьшить негативный эффект от переобучения. С другой стороны, представляется целесообразным рассмотреть создание новых признаков как функции данных признаков. Во многих случаях [7], новые признаки могут существенно улучшить качество классификатора.

5.1. Генерация признака «активность соседа»

В этом разделе мы представим метод для формирования вторичных признаков, который был мотивирован идеей кластерного анализа (обучение без учителя). Рассмотрим множество всех бинарных признаков (признаков с возможными значениями 0 или 1) и рассчитаем матрицу попарных расстояний между наблюдениями из обучающего и тестового подмножеств. В качестве меры расстояния мы использовали так называемую «манхэттенскую» меру.

Заметим, что манхэттенская мера для бинарных признаков равна количеству различных признаков для каждой пары наблюдений. Для каждого наблюдения из обучающего подмножества найдем ближайший элемент согласно рассчитанной матрице расстояний и определим значение целевой переменной для этого элемента («активность соседа»). Аналогичным образом, для каждого наблюдения из тестового подмножества найдем ближайшего соседа из обучающего подмножества. Таким образом, новый бинарный признак определен как для обучающего, так и для тестового подмножеств.

Замечание 3. Предлагаемый способ генерации нового признака не случаен. Мы руководствовались следующим простым соображением: для данной базы данных (напомним, что каждый элемент представляет собой молекулу) вероятнее всего бинарные признаки описывают присутствие

того или иного химического элемента в составе молекулы. Этот факт позволяет сделать предположение о том, что молекулы с похожим составом в большинстве случаев обладают одинаковыми активностями. Стоит также отметить, что коэффициент корреляции Пирсона между признаком «активность соседа» и активностью на обучающем подмножестве составил около 0.50.

Замечание 4. Описанный метод дает возможность для формирования целого множества новых признаков. Для этого достаточно модифицировать набор признаков, используемых для расчета расстояний, меру или количество рассматриваемых соседей. Например, в качестве нового признака можно взять среднюю активность пяти ближайших соседей согласно евклидовому расстоянию, рассчитанному по всем признакам.

5.2. Генерация производных активностей

5.2.1. Генерация производных активностей для бинарных признаков

Обозначим $x_{ij} \in \{0; 1\}$ – значение бинарного признака i для наблюдения j обучающего подмножества, y_j – активность элемента j .

Определим новый вектор $x_i^{(2)}$ для обучающего подмножества согласно

следующей формуле $x_{ij}^{(2)} = XOR(x_{ij}, y_j)$, а именно:

$x_{ij} = 0, y_j = 0, x_{ij}^{(2)} = 1;$
$x_{ij} = 0, y_j = 1, x_{ij}^{(2)} = 0;$
$x_{ij} = 1, y_j = 0, x_{ij}^{(2)} = 0;$
$x_{ij} = 1, y_j = 1, x_{ij}^{(2)} = 1.$

По определению, рассчитанный вектор $x_i^{(2)}$ назовем производной активностью (индикатор совпадения признака и целевой переменной). Отметим, что точная реализация определения производной активности не представляется возможным, поскольку значение целевой переменной определено только на обучающем подмножестве. В случае тестового подмножества, где значения активностей y неизвестно, мы использовали прогноз \hat{y} для построения нужных нам производных активностей $x^{(2)}$.

5.2.2. Генерация производных активностей для небинарных признаков.

Как и в предыдущем под-разделе, предположим, что $x_{ij} \in D_i$ – значение небинарного признака i для наблюдения j обучающего/тренировочного подмножества, D_i – множество возможных значений признака i (если i – непрерывный признак, то в качестве D_i рассматривается промежуток числовой прямой), y_j – активность элемента j обучающего подмножества. Выбрав некоторое пороговое значение t_i , определим два правила для расчета производной активности:

Правило 1:

$$\begin{aligned} x_{ij} < t_i, y_j = 0, x_{ij}^{(2)} &= 1; \\ x_{ij} < t_i, y_j = 1, x_{ij}^{(2)} &= 0; \\ x_{ij} \geq t_i, y_j = 0, x_{ij}^{(2)} &= 0; \\ x_{ij} \geq t_i, y_j = 1, x_{ij}^{(2)} &= 1. \end{aligned}$$

Правило 2:

$$\begin{aligned} x_{ij} = t_i, y_j = 0, x_{ij}^{(2)} &= 1; \\ x_{ij} = t_i, y_j = 1, x_{ij}^{(2)} &= 0; \\ x_{ij} \neq t_i, y_j = 0, x_{ij}^{(2)} &= 0; \\ x_{ij} \neq t_i, y_j = 1, x_{ij}^{(2)} &= 1. \end{aligned}$$

Примеры генерации производных активностей:

1) производная активность для бинарного признака *Pr27*

	$Pr27 = 0$	$Pr27 = 1$
Активность = 0	1	0
Активность = 1	0	1

2) производная активность для небинарного признака *Pr659*

	$Pr659 < 0.20$	$Pr659 \geq 0.20$
Активность = 0	1	0
Активность = 1	0	1

3) производная активность для небинарного признака *Pr182*

	$Pr182 = 0.0943$	$Pr182 \neq 0.0943$
Активность = 0	1	0
Активность = 1	0	1

Заключительные замечания

Характерной особенностью соревнования Boehringer явился тот факт, что финальные результаты оказались значительно лучше в сравнении с предварительными результатами (см. Таблицу 1). Последнее означает, что отбор данных для предварительного тестирования был осуществлен не случайным образом (более сложные наблюдения были отобраны для валидации метода через LeaderBoard). Как следствие, финальное ранжирование участников существенно отклонилось от предварительного: $\{1,2,3,4\} \Rightarrow \{3,29,25,8\}$, где наш предварительный результат был четвертым, а финальный - восьмым. Кроме того, заметим, что в сравнении с вторым и третьим участниками (согласно предварительному тестированию) наш результат оказался довольно высоким, что означает устойчивость и стабильность нашего метода. Отметим также, что

результат победителя в соревновании Boehringер 0.37356 существенно уступает нашему абсолютно лучшему результату (см. Таблицу 1 и Рис.1).

Библиографический список:

1. Djukova E. V., Zhuravlev Yu. I., and Sotnezov R. M. Construction of an ensemble of logical correctors on the basis of elementary classifiers. *Pattern Recognition and Image Analysis*. 2011. 21(4). P. 599-605.

2. Biau G., Devroye L., and Lugosi G. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*. 2007. 9. P. 2015-2033.

3. Breiman L. Random forests. *Machine Learning*. 2001. 45(1). P. 5-32.

4. Nikulin V., Huang T.-H., and McLachlan G.J. Classification of high-dimensional microarray data with a two steps procedure via a Wilcoxon criterion and multilayer perceptron. *International Journal of Computational Intelligence and Applications*. 2011. 10(1), P. 1-14.

5. Friedman J.H. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*. 2001. 29. P. 1189-1232.

6. Burges C. J. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*. 1998. 2(2), P. 121–167.

7. Nikulin V. On the Homogeneous Ensembling with Balanced Random Sets and Boosting. *JRS 2012, Chengdu, China. LNAI Springer 7413, J.T.Yao et.al. (Eds.)*. 2012. P. 180-189.

Информация об авторах

Data on authors

Ефимов Дмитрий Александрович – канд. ф.-м. наук, ассистент кафедры математического анализа механико-математического факультета МГУ им.М. В.Ломоносова

Тел. +7(495)740-84-20

E-mail: diefimov@gmail.com

Efimov Dmitry – Dr. of Mathematics & Physics, Assistant on Calculus Department of Lomonossov University, Moscow

Tel. +7(495)740-84-20

E-mail: diefimov@gmail.com

Никулин Владимир Николаевич – канд. ф.-м. наук, доцент, докторант кафедры математических методов в экономике ФГБОУ ВПО «ВятГУ»

Тел.: +7(8332)64-48-16

E-mail: vnikulin.uq@gmail.com

Nikulin Vladimir – Dr. of Mathematics & Physics, Associate Prof. on Mathematical Modeling in Economics FGEI HPE «VyatSU»

Тел.: +7(8332)64-48-16

E-mail: vnikulin.uq@gmail.com

Козлов М. В., Костюк Ф. В., Сорокин С. В., Тюленев А. В.

Kozlov M. V., Kostyuk F. V., Sorokin S. V., Tyulenev A. V.

**РЕШЕНИЕ ЗАДАЧИ КОММИВОЯЖЕРА МЕТОДОМ
ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С
ПОСЛЕДОВАТЕЛЬНЫМ ИСКЛЮЧЕНИЕМ ПОДЦИКЛОВ:
ОПИСАНИЕ И АЛГОРИТМИЧЕСКАЯ РЕАЛИЗАЦИЯ**

**SOLVING TRAVELLING SALESMAN PROBLEM BY INTEGER
LINEAR PROGRAMMING WITH CUMULATIVE SUBTOUR
ELIMINATION: DESCRIPTION AND IMPLEMENTATION**

Предлагается новый метод решения задачи коммивояжера. Метод сводит проблему к решению последовательности задач целочисленного линейного программирования с нарастающим исключением подциклов. Приводится алгоритм решения, доказывається конечность метода и оптимальность полученного решения. Предлагается его модификация, обеспечивающая нахождение всех оптимальных решений, либо доказательство единственности оптимального решения. Исследуются различные стратегии исключения подциклов.

A new method for travelling salesman problem is presented. The method transforms the problem to a sequence of integer linear programming problems with cyclic paths cumulative elimination. The algorithmic implementation and substantiation is presented. Various strategies of cycle elimination and their influence on the solving effectiveness are discussed.

Ключевые слова: задача коммивояжера, комбинаторная оптимизация, численные методы, графы, целочисленное линейное программирование.

Keywords: travelling salesman problem, combinatorial optimization, numerical methods, graphs, integer linear programming.

Введение

Задача коммивояжера (ЗК) является ярким примером классической оптимизационной задачи. Она была сформулирована строго математически в 30-х годах прошлого века, но корнями восходит к 1832 году, когда в наставлении коммивояжеру, изданном на немецком языке, предлагается оптимизировать маршрут передвижения между 47 городами Германии [1]. Вариант распознавания этой задачи относится к классу NP-полных и поэтому не может быть решен эффективным (т. е. полиномиальным) алгоритмом, в предположении, что классы P и NP не совпадают [2, 3]. ЗК имеет многочисленные практические приложения. В течение последних 80 лет она привлекала пристальное внимание исследователей. Для ее решения были разработаны многочисленные методы, ни один из которых не может претендовать на универсальность, эффективно решая одни частные задачи, но имея крайне низкую эффективность на других [4, 5].

Задача коммивояжера состоит в следующем. Имеется n городов. Между некоторыми парами городов заданы расстояния. Надо найти кратчайший простой циклический маршрут, проходящий через все города и заходящий в каждый город ровно один раз. Длиной маршрута считается сумма расстояний между городами вдоль него. Приведенная формулировка описывает симметричную задачу коммивояжера (СЗК). Несимметричная задача коммивояжера (НСЗК) отличается от симметричной тем, что расстояние между парой городов А и Б в прямом направлении (из А в Б) не всегда равно расстоянию в обратном направлении (из Б в А). В этом случае для каждой пары городов задаются два расстояния.

В формулировке ЗК говорится о расстоянии между городами, но в качестве «расстояния» можно использовать стоимость проезда, время в пути и другие величины. ЗК является несимметричной при наличии дорог с односторонним движением, реконструкции участков дорог, изменяющих их пропускную способность в реконструируемом направлении, наличии воздушных коридоров, присутствии устойчивых ветров по маршруту движения воздушного судна и в других случаях наличия ограничений.

Не требуется, чтобы расстояния были неотрицательными, либо чтобы они удовлетворяли неравенству треугольника. Если оба эти условия выполнены, ЗК называется метрической и обладает рядом замечательных свойств.

ЗК может быть сформулирована в терминах теории графов [6, 7, 8, 9]. Задан взвешенный граф $G = \{V, E, C\}$, где V – множество вершин, E – множество дуг и C – веса дуг. Дуги могут быть неориентированными (для симметричной задачи) или ориентированными (для несимметричной задачи). В первом случае они называются ребрами. Требуется найти на графе G простой циклический маршрут, проходящий через все вершины графа (такие маршруты называются гамильтоновыми циклами), минимального веса. Мы будем использовать термин «дуга» и в случае неориентированного, и в случае ориентированного графа. И только тогда, когда необходимо будет подчеркнуть, что дуга неориентированная, будем использовать термин «ребро». Граф не содержит петель, между каждой парой вершин имеется не более одного ребра (в неориентированном графе) и не более одной дуги в каждом направлении (в орграфе).

Для решения ЗК были разработаны многочисленные точные и приближенные методы. Некоторые из этих методов были впоследствии обобщены для решения других задач комбинаторной оптимизации. Например, метод ветвей и границ, впервые примененный для решения ЗК в

диссертации В. Истмана [10], был через два года развит Лэнд и Дойг для других задач [11] и был выкристаллизован, получив ныне знакомый нам понятийный аппарат, в 1963 году [12]. К наиболее эффективным точным методам помимо метода ветвей и границ относятся методы отсекающей гиперплоскости, ветвлений и отсечений, отсечений Гомори, динамического программирования. В первых четырех случаях на итерациях метода приходится решать задачи линейного программирования (ЛП) или целочисленного линейного программирования (ЦЛП). Исключительное значения в таких случаях приобретает выбор уравнений или неравенств, генерируемых в процессе счета. Среди наиболее распространенных ограничений выделяются ограничения на степень вершин, ограничения на подциклы, ограничения на подмножества, комб (ограничения типа расчески, comb condition), неравенства 2-соответствия (2-matching inequalities), неравенства клики, а также неравенства пути (path inequalities), которые обобщают комб.

Удачный подбор соответствующих ограничений значительно ускоряет процесс решения. С ростом числа ограничений уменьшается допустимая область, на которой происходит решение промежуточных задач, и получается решение более близкое к оптимальному решению исходной задачи. Однако чем больше ограничений, тем сложнее для решения промежуточные задачи в связи ростом размера, исчерпанием оперативной памяти, системными ограничениями алгоритмов и существенным увеличением времени их работы. Поэтому многочисленные публикации содержат различные методики и эвристические подходы выбора ограничений, процедуры их исключения из рассмотрения и перемещения в специальный пул ограничений, где они могут периодически пересматриваться. Блестящим примером использования такого подхода стали прорывные работы, выполненные Падбергом и Ринальди [13,14].

Для решения полученных ЦЛП разрабатывались оригинальные алгоритмы, учитывающие специфику ограничений и функционала. Среди исследований, в которых использовались стандартные пакеты решения ЦЛП, необходимо отметить работы [15, 16].

Принципиальная сложность ЗК привела к необходимости использовать эвристические методы для решения задач большой размерности [17, 18]. Так, максимальный размер ЗК, для которой в настоящий момент получено точное решение, составляет 85900 городов [19]. Максимальный размер ЗК, решенный наиболее эффективным на сегодняшний момент эвристическим алгоритмом LKH, реализованным профессором Роскилдского университета Келдом Хелсгауном [20, 21], составляет 1904711 городов (с доказанной точностью 0.058%). Важным элементом решения ЗК большой размерности является предварительная обработка исходных данных. Она заключается в исключении из рассмотрения дуг, заведомо не входящих в оптимальное решение, декомпозиции исходной задачи, особенно эффективной при ее сильной кластеризованности или расположении нескольких вершин на коротком отрезке прямой линии (гирлянда), формировании неравенств, существенно уменьшающих область допустимых решений без исключения из рассмотрения оптимального гамильтонова цикла. Примеры таких неравенств, а также некоторые важные подходы к ЗК с точки зрения дискретного программирования даны в монографиях [22, 23].

В первой части работы дано описание нового метода решения ЗК.

Во второй части приводится алгоритмическая реализация метода.

Третья часть содержит описание программной реализации.

Обоснование и тестовые испытания будут представлены в отдельной статье.

1. Описание метода

Исходными данными ЗК являются количество городов и расстояния между парами городов. Для несимметричной задачи для некоторых пар городов А и Б расстояние в прямом направлении (из А в Б) отличается от расстояния в обратном направлении (из Б в А).

Введем обозначения, которые понадобятся для описания метода:

n – количество городов, перенумерованных натуральными числами от 1 до n . В теории графов городам соответствуют вершины, а в теории сетей – узлы.

c_{ij} – расстояние от города i до города j , $i=1..n, j=1..n, i \neq j$, c_{ij} и c_{ji} могут отличаться. Величины c_{ij} образуют квадратную матрицу C , где i – это номер строки, а j – номер столбца. Без ограничения общности, когда это будет удобно, будем полагать $c_{ii} = 0$.

Рассмотрим сначала НСЗК, в которой c_{ij} может быть не равно c_{ji} для некоторых пар (i, j) . Без ограничения общности, будем считать, что c_{ij} определено для любых i и j , $i=1..n, j=1..n, i \neq j$. Например, для тех пар (i, j) , для которых c_{ij} не определено, можно положить их равными сумме всех c_{ij} , определенных в задаче.

Введем $n(n-1)$ булевых переменных $x_{ij} \in \{0;1\}$ $i=1..n, j=1..n, i \neq j$. Их физический смысл состоит в том, что $x_{ij} = 1$, если коммивояжер переезжает из города i в город j , и $x_{ij} = 0$ в противном случае. Для каждой пары городов i и j есть две переменные - x_{ij} и x_{ji} , так как дуги из i в j и из j в i – это разные дуги. Все множество этих переменных будем обозначать $X = \{x_{ij}, i=1..n, j=1..n, i \neq j\}$.

Метод сводит проблему к решению последовательности задач булевого линейного программирования (БЛП) с нарастающим исключением циклов. При этом каждая последующая задача отличается от предыдущей добавлением не менее одного нового ограничения, сужающего допустимое множество. На первом шаге решается следующая задача.

Для целевой функции

$$L = \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n c_{ij} x_{ij} \quad (1.1)$$

найти минимум по переменным x_{ij}

$$L_* = \min_x L(x) \quad (1.2)$$

при ограничениях

$$\sum_{\substack{j=1 \\ j \neq i}}^n x_{ij} = 1, \quad i = \overline{1, n} \quad (1.3)$$

$$\sum_{\substack{i=1 \\ i \neq j}}^n x_{ij} = 1, \quad j = \overline{1, n} \quad (1.4)$$

$$x_{ij} \in \{0; 1\}, \quad i = \overline{1, n}, \quad j = \overline{1, n}, \quad i \neq j. \quad (1.5)$$

Система уравнений (1.3) гарантирует, что из каждого города будет выходить ровно одна дуга. Система уравнений (1.4) гарантирует, что в каждый город будет входить ровно одна дуга.

Задача (1.1)-(1.5) является классической задачей о назначениях, и для нее существуют эффективные методы решения. В результате решения этой задачи БЛП будут получены значения переменных x_{ij} , ровно n из которых будут равны 1, а остальные – 0. Вследствие ограничений (1.3)-(1.4), из каждого города будет выходить ровно одна дуга, и в каждый город будет входить ровно одна дуга. В результате получится решение, состоящее из одного или нескольких простых вершинно-непересекающихся циклов. Чтобы выделить эти циклы в явном виде, надо выбрать в полученном решении любую переменную, равную единице. Пусть это будет переменная $x_{i_1 i_2}$. То, что она равна единице, означает, что используется дуга из города i_1 в город i_2 . Теперь среди переменных $x_{i_2 1} \dots x_{i_2 n}$ найдем ту, которая равна единице (такая переменная ровно одна в силу ограничений (1.3)). Пусть это переменная $x_{i_2 i_3}$. Она соответствует дуге из города i_2 в

город i_3 . Продолжим этот процесс, пока не попадем снова в город i_1 . Мы получили простой цикл. Чтобы найти остальные циклы, надо в решении выбрать новую переменную, равную 1, которая не была использована в уже найденном цикле, и повторить процесс. Так повторяется до тех пор, пока для каждой единичной переменной из решения не будет определен цикл, которому она принадлежит. При этом число звеньев цикла равно либо n , либо k , $2 \leq k \leq n-2$.

Если цикл один, то он является решением исходной ЗК, и процесс решения заканчивается. Если таких циклов несколько (будем называть их подциклами), то к ограничениям, использованным на предыдущем шаге, добавится еще одно или несколько дополнительных ограничений, и решается новая задача БЛП. Для каждого подцикла к задаче, полученной на очередном шаге, добавляется одно или два новых ограничения. Допускается, что для некоторого подцикла не будет добавлено ни одно ограничение, при условии, однако, что на каждом шаге будет добавлено хотя бы одно новое ограничение, соответствующее какому-либо подциклу.

Используются три типа ограничений. Пусть подцикл длины k , $2 \leq k \leq n-2$, проходит через города с номерами i_1, i_2, \dots, i_k в указанном порядке. Обозначим через S множество вершин данного подцикла.

Ограничение С1а (неравенство подцикла):

$$\sum_{j=1}^{k-1} x_{i_j i_{j+1}} + x_{i_k i_1} \leq k - 1 \quad (1.6)$$

Смысл этого ограничения в том, что из k дуг, составляющих этот подцикл, в решении задачи могут присутствовать не более $k-1$, то есть не все одновременно. Добавление такого ограничения исключает возможность появления соответствующего подцикла в решениях на следующих шагах, так как в этом подцикле используются все k дуг, и сумма в левой части неравенства (1.6) будет равна k , что нарушит ограничение.

Ограничение С2а (прямое неравенство подмножества):

$$\sum_{\substack{i \in S \\ j \notin S}} x_{ij} \geq 1 \quad (1.7)$$

Это ограничение делает недопустимыми те решения, в которых каждый подцикл проходит по вершинам только либо из множества S , либо из дополнения этого множества. Иными словами, в решении должен присутствовать хотя бы один цикл, «соединяющий» множество вершин S и $V \setminus S$.

Ограничение С3а (обратное неравенство подмножества):

$$\sum_{\substack{i \in S \\ j \in S \\ j \neq i}} x_{ij} \leq k - 1 \quad (1.8)$$

Это ограничение, как будет доказано ниже, математически эквивалентно Ограничению С2а в условиях (1.3). Причина, по которой мы не ограничиваемся Ограничением С2а, а вводим Ограничение С3а, типична для комбинаторной оптимизации. Несмотря на математическое тождество, они не эквивалентны алгоритмически. Для коротких подциклов Ограничение С3а содержит намного меньше переменных. Это приводит к большей разреженности матрицы ограничений, что ускоряет процесс счета одними алгоритмами, или дает возможность применять другие алгоритмы, специально разработанные для разреженных матриц и, потому, более эффективные для них. Любое из ограничений С2а и С3а не слабее С1а. В самом деле, сумма, стоящая в левой части неравенства (1.8), включает в себя все слагаемые суммы, стоящей в левой части (1.6), что в условиях неотрицательности переменных x_{ij} обеспечивает выполнение (1.6) при выполненном (1.8). Однако любое из ограничений С2а и С3а содержит не меньше переменных, чем С1а.

Таким образом, для каждого подцикла целесообразно формировать не более одного ограничения. Заметим, что ни одно из ограничений С1а,

C2a, C3a не исключает из области допустимых решений ни одного гамильтонова цикла.

Выбор того, какое из ограничений применяется для данного подцикла, является важным параметром метода, обеспечивающим его гибкость в настройке к различным приложениям. Как было отмечено ранее, нет необходимости формировать ограничение для каждого подцикла. Однако на каждой итерации хотя бы для одного подцикла ограничение должно быть сформировано. Именно это, как будет доказано ниже, обеспечивает корректность работы метода и получение точного решения за конечное число итераций.

Все ограничения, добавленные на предыдущих шагах, сохраняются. Исключения допустимы при определенных условиях.

После добавления ограничений получается новая задача БЛП, которая решается на следующем шаге алгоритма. Новое полученное решение снова содержит один или несколько подциклов. Если циклов несколько, то к ограничениям задачи добавляются новые ограничения типа (1.6)-(1.8), и процесс повторяется. Оптимальное значение целевой функции задачи БЛП не убывает на итерациях метода и является оценкой снизу для ЗК.

Алгоритм заканчивает работу, когда решение состоит из единственного цикла, который является оптимальным гамильтоновым циклом – решением ЗК. Значение целевой функции является оптимумом ЗК.

Метод позволяет ответить на вопрос о единственности оптимального решения и перечислить все оптимальные решения. Для этого после нахождения первого оптимального решения надо поочередно исключить найденные гамильтоновы циклы добавлением Ограничения C1a до тех пор, пока значение целевой функции совпадает с оптимумом ЗК. Все гамильтоновы циклы, которые встретились во время дополнительного

процесса, кроме последней итерации, являются оптимальными решениями. Оптимальных решений может быть чрезвычайно много, вплоть до $(n-1)!$.

2. Алгоритмическая реализация

Описанный алгоритм обладает значительной гибкостью и определяет целый класс алгоритмов решения ЗК. Приведем формулировку алгоритмов, реализующих метод. Все приведенные ниже алгоритмы решают задачу БЛП на каждой итерации и завершают работу при выполнении одного из следующих условий:

Условие У1. Задача БЛП не имеет решения. Это означает, что в исходной ЗК нет ни одного допустимого гамильтонова цикла и, следовательно, ЗК не имеет решения.

Условие У2. Задача БЛП имеет решение, соответствующее единственному простому циклу. Этот цикл является оптимальным решением ЗК.

Алгоритм А1g. Положить начальное множество ограничений CS равным (1.3)-(1.5), и на каждом шаге выполнить следующие процедуры:

Решить задачу минимизации функционала (1.1) при ограничениях CS и выделить подциклы, соответствующие решению.

При выполнении Условия У1 либо Условия У2 завершить работу.

Для каждого подцикла добавить одно из ограничений $C1a, C2a, C3a$ в CS .

Данный алгоритм реализует основную идею метода в чистом виде, когда для каждого подцикла формируется одно ограничение. Очевидное развитие идеи метода состоит в более тщательном анализе подциклов с тем, чтобы учитывать подциклы в списке ограничений не немедленно, а тогда, когда их учет необходим для получения решения.

Алгоритм А2g. Положить начальное множество ограничений CS равным (1.3)-(1.5), и на каждом шаге выполнить следующие процедуры:

Решить задачу минимизации функционала (1.1) при ограничениях CS и выделить подциклы, соответствующие решению.

При выполнении Условия У1 либо Условия У2 завершить работу.

Для каждого подцикла добавить одно из ограничений С1а, С2а, С3а в CS, либо не добавлять. При этом хотя бы для одного подцикла ограничение должно быть добавлено.

Алгоритмы А1g и А2g являются общими, реализуя основные возможности метода, но не конкретизируя тип формируемого ограничения и даже подциклы, для которых эти ограничения формируются. Конкретизацией этих параметров можно получить частные алгоритмы, как описано ниже.

Алгоритм А3. Положить начальное множество ограничений CS равным (1.3)-(1.5), и на каждом шаге выполнить следующие процедуры:

Решить задачу минимизации функционала (1.1) при ограничениях CS и выделить подциклы, соответствующие решению.

При выполнении Условия У1 либо Условия У2 завершить работу.

Для каждого подцикла добавить в CS Ограничение С1а.

Алгоритм А4. Положить начальное множество ограничений CS равным (1.3)-(1.5), и на каждом шаге выполнить следующие процедуры:

Решить задачу минимизации функционала (1.1) при ограничениях CS и выделить подциклы, соответствующие решению.

При выполнении Условия У1 либо Условия У2 завершить работу.

Добавить в CS Ограничение С3а для самого длинного подцикла и Ограничение С1а для остальных подциклов, если их число превышает 1.

Алгоритм А5. Положить начальное множество ограничений CS равным (1.3)-(1.5), и на каждом шаге выполнить следующие процедуры:

Решить задачу минимизации функционала (1.1) при ограничениях CS и выделить подциклы, соответствующие решению.

При выполнении Условия У1 либо Условия У2 завершить работу.

Для каждого подцикла длиной не более $n/2$ добавить в CS Ограничение C3a, причем если два подцикла имеют длину $n/2$, то для 2-го подцикла добавлять ограничение не нужно.

Алгоритмы A3, A4 и A5 являются частными случаями алгоритма A1g. Следующий алгоритм является частным случаем алгоритма A2g.

Алгоритм A6. Положить начальное множество ограничений CS равным (1.3)-(1.5), множество подмножеств вершин VS пустым, и на каждом шаге выполнить следующие процедуры:

Решить задачу минимизации функционала (1.1) при ограничениях CS и выделить подциклы, соответствующие решению.

При выполнении Условия У1 либо Условия У2 завершить работу.

Добавить в CS Ограничение C3a для подциклов, множество вершин которых принадлежат VS, исключить из VS эти множества, а из CS – Ограничения C1a, соответствующее подциклам, проходящим по тем же множествам вершин. Для других подциклов добавить Ограничение C1a в CS, а множество его вершин – в VS.

Если ЗК имеет более одного оптимального решения, то метод находит одно из оптимальных решений. Задача нахождения всех оптимальных решений ЗК либо доказательства единственности решения, может быть решена небольшой модификацией предложенных алгоритмов,

Алгоритм A7w. (Нахождение всех оптимальных решений ЗК). Применить любой из ранее изложенных Алгоритмов A1g, A2g, A3, A4, A5, A6. Включить найденный оптимальный гамильтонов цикл в множество HS. Добавить ограничение C1a, соответствующее этому циклу, к множеству ограничений CS. Запомнить значение длины оптимального цикла L^* , и на каждом шаге выполнить следующие процедуры:

Решить задачу минимизации функционала (1.1) при ограничениях CS и выделить подциклы, соответствующие решению.

При выполнении Условия У1, либо если значение целевой функции

больше L_* , завершить работу. Если выполнено Условие У2, включить найденный гамильтонов цикл в множество HS, а ограничение C1a, соответствующее этому циклу, – в множество ограничений CS. В противном случае, для каждого подцикла добавить одно из ограничений C1a, C2a, C3a в CS.

Учитывая, что в СЗК число переменных в два раза меньше, чем в НСЗК, и используя аналоги формул (1.1)-(1.8) для симметричного случая, мы сохраняем полную применимость приведенных алгоритмов для СЗК.

3. Программная реализация метода

Для реализации описанных алгоритмов был разработан комплекс программ на языках C и VisualBasic. В состав комплекса входят программы преобразования исходных данных между несколькими форматами, формирования файла в формате MPS, являющемся стандартом де-факто для модулей решения задач ЛП, программы для добавления ограничений к задаче ЛП в формате MPS, программы для обработки результатов решения задачи ЛП и программы для получения вспомогательной информации.

Для решения задачи БЛП использовался свободно распространяемый пакет ЛП GLP в виде EXE-модуля, входными данными которого служит MPS-файл, а выходными – текстовый. Данный модуль был любезно предоставлен ведущим научным сотрудником ЦЭМИ Малковым У.Х. [25], которому авторы выражают искреннюю благодарность.

При программной реализации алгоритма А3 для несимметричной задачи для уменьшения количества шагов и времени решения одновременно с ограничением (1.6) добавляется аналогичное ограничение для запрещения обратного цикла, то есть, цикла, проходящего через те же вершины в том же порядке, но в обратном направлении, а именно:

$$\sum_{j=1}^{k-1} x_{i_{j+1}i_j} + x_{i_k} \leq k-1. \quad (3.1)$$

Добавление этого ограничения не сужает множество решений ЗК.

Библиографический список:

1. Ein alter Commis-Voyageur. Der Handlungsreisende: wie er sein soll und was er zu tun hat, um Aufträge zu erhalten und eines glücklichen Erfolgs in seinen Geschäften gewiss zu sein. V. Fr. Voigt, Ilmenau:1832.
2. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация: алгоритмы и сложность. – М.: Изд-во Мир, 1984. – 510 с.
3. Меламед И. И., Сергеев С. И., Сигал И. Х. Задача коммивояжера. Вопросы теории // АиТ. – 1989. – №9. – С.3-33
4. Гэри В., Джонсон Д. Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982.
5. Меламед И. И., Сергеев С. И., Сигал И. Х. Задача коммивояжера. Точные методы // АиТ. – 1989. – №10. – С.3-29
6. Зыков А. А. Основы теории графов. М.: Наука, 1987
7. Оре О. Теория графов. М.: Наука, 1980
8. Хараре Ф. Теория графов. М.: Едиториал УРСС, 2003
9. Кристофидес Н. Теория графов. Алгоритмический подход, М.: Мир, 1978.
10. Eastman, W. L. Linear Programming with Pattern Constraints. Ph.D.Thesis. Department of Economics, Harvard University, Cambridge, Massachusetts, USA. 1958.
11. Land, A. H., Doig A. G. An automatic method of solving discrete programming problems // Econometrica. 1960. № 28. P. 497–520.
12. Little, J. D. C., Murty K.G., Sweeney D.W., Karel C. An algorithm for the traveling salesman problem // Operations Research. 1963. № 11. P. 972–989.
13. Padberg, M., G. Rinaldi. Optimization of a 532-city symmetric

traveling salesman problem by branch and cut // *Operations Research Letters*. 1987. № 6, p.1–7.

14. Padberg, M., G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems // *SIAM Review*. 1991. № 33. P. 60–100.

15. Crowder, H., M.W. Padberg. Solving large-scale symmetric travelling salesman problems to optimality // *Management Science*. 1980. № 26, p.495–509.

16. Groetschel, M., O. Holland. Solution of large-scale symmetric travelling salesman problems // *Mathematical Programming*. 1991. № 51. p.141–202.

17. Меламед И. И., Сергеев С. И., Сигал И. Х. Задача коммивояжера. Приближенные алгоритмы // *АиТ*. – 1989. – №11. – С.3-26

18. Хачатуров В. Р. и др. Комбинаторные методы и алгоритмы решения задач дискретной оптимизации большой размерности. – М.:Наука, 2000. – 360 с.

19. Cook W., Espinoza D., Goycoolea M. Computing with dominoparity inequalities for the TSP// *INFORMS Journal on Computing*. 2007. Vol. 19. №.3. P.356-365.

20. Helsgaun, K. An effective implementation of the Lin-Kernighan traveling salesman heuristic // *European Journal of Operational Research*. 2000. №126. P.106–130.

21. Helsgaun K. An Effective Implementation of K-opt Moves for the Lin-Kernighan TSP Heuristic // *DATALOGISKE SKRIFTER* (Writings on Computer Science). Roskilde University. 2006. No. 109.

22. Корбут А. А., Финкельштейн Ю. Ю. Дискретное программирование. – М.: Наука, 1969.

23. Сигал И. Х., Иванова А. П. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы. – М.: Физматлит, 2007. – 304 с.

24. Reinelt, G. TSPLIB—A traveling salesman problem library // ORSA Journal on Computing. 1991. № 3, P.376–384.

25. Гольштейн Е. Г., Малков У. Х., Соколов Н. А. Результаты экспериментального сравнения метода уровней и принципа разложения Данцига-Вульфа. // Экономика и мат. методы. – 2003. – Т.39. – Вып.2.

Информация об авторах

Data on authors

Козлов Михаил Васильевич – инженер-исследователь, ВЦ РАН

Тел.: (499)135-0170

E-mail: kz2pit@rambler.ru

Kozlov Mikhail Vasilievich – research engineer, CC RAS

Tel.: (499)135-0170

E-mail: kz2pit@rambler.ru

Костюк Феликс Владиславович – инженер-исследователь, ВЦ РАН

Тел.: (499)135-0170

E-mail: fil@ccas.ru

Kostyuk Felix Vladislavovich –research engineer, CC RAS

Tel.: (499)135-0170

E-mail: fil@ccas.ru

Сорокин Станислав Викторович – ведущий программист, ВЦ РАН

Тел.: (499) 135-1498

E-mail: www2011@ccas.ru

Sorokin Stanislav Victorovich – leading software designer, CC RAS

Tel.: (499) 135-1498

E-mail: www2011@ccas.ru

Тюленев Александр Васильевич – к.э.н., математик, ВЦ РАН

Тел.: (499) 135-4250

E-mail: tyulavas@yandex.ru

Tyulenev Alexander Vasilievich – PhD. in economics, mathematician, CC
RAS

Tel.: (499) 135-4250

E-mail: tyulavas@yandex.ru

**РЕШЕНИЕ ЗАДАЧИ КОММИВОЯЖЕРА МЕТОДОМ
ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С
ПОСЛЕДОВАТЕЛЬНЫМ ИСКЛЮЧЕНИЕМ ПОДЦИКЛОВ:
ОБОСНОВАНИЕ, ТЕСТОВЫЕ ИСПЫТАНИЯ, ПРИМЕНЕНИЕ**

**SOLVING TRAVELLING SALESMAN PROBLEM BY INTEGER
LINEAR PROGRAMMING WITH CUMULATIVE SUBTOUR
ELIMINATION: SUBSTANTIATION, TESTING, APPLICATION**

Дается обоснование нового метода решения задачи коммивояжера. Метод сводит проблему к решению последовательности задач целочисленного линейного программирования с нарастающим исключением подциклов. Доказывается конечность метода и оптимальность полученного решения. Доказывается нахождение всех оптимальных решений, либо доказательство единственности оптимального решения. Приводятся результаты использования метода для решения ряда тестовых и практических задач. Исследуется поведение метода на случайно сгенерированных метрических задачах. Исследуются различные стратегии исключения подциклов.

A substantiation of a new method for travelling salesman problem is presented. The method transforms the problem to a sequence of integer linear programming problems with cyclic paths cumulative elimination. All optimal Hamilton cycles are found by the method. An application of the method to some classical instances of traveling salesman problems, practical tasks and stochastic metric tasks is given. Various strategies of cycle elimination and their influence

on the solving effectiveness are discussed.

Ключевые слова: задача коммивояжера, комбинаторная оптимизация, численные методы, графы, целочисленное линейное программирование.

Keywords: travelling salesman problem, combinatorial optimization, numerical methods, graphs, integer linear programming.

Введение

Описание нового метода решения задачи коммивояжера (ЗК) приведено в [1]. Там же содержится описание алгоритмической и программной реализации метода, вводятся некоторые понятия и сокращения. При ссылке на формулы из [1] будем использовать следующие обозначение, например, (I-1.6) для формулы (1.6) из [1].

В первой части настоящей работы дается обоснование метода. Доказывается его сходимость за конечное число итераций. Дается оценка количества итераций и их сложности.

Во второй части приводятся примеры использования метода для решения тестовых ЗК, описанных в [2], а также двух новых задач.

В третьей части изучается эффективность метода для разного типа задач. Рассматриваются различные стратегии исключения циклов и оценка их влияния на эффективность решения. Обсуждаются дальнейшие пути развития метода. Приводятся практические приложения ЗК.

1. Обоснование метода

Докажем, что метод решает ЗК и выполняется за конечное время. Будем проводить доказательство для алгоритмов решения НСЗК, указывая, где это необходимо, отличия для случая симметричной задачи. Нам потребуются некоторые понятия и утверждения.

Определение 1. Назовем остовным набором циклов (ОНЦ) графа G

один или несколько вершинно-непересекающихся простых циклов графа G таких, что каждая вершина графа инцидентна хотя бы одному ребру какого-либо из циклов.

Лемма 1. Любое допустимое решение системы ограничений (I-1.3)-(I-1.5) представляет собой остовный набор циклов графа G .

Лемма 2. Существует взаимно-однозначное соответствие между допустимыми решениями задачи (I-1.1)-(I-1.8) и остовными наборами циклов.

Заметим также, что в каждом решении задачи (I-1.1)-(I-1.8) ровно n переменных равны единице, а остальные переменные – нулю. Это видно из того, что при сложении всех равенств (I-1.3) слева получится сумма всех переменных, а справа – n . Тот же результат получится при сложении равенств (I-1.4). Важное свойство ограничений (I-1.7) и (I-1.8) дает следующая

Лемма 3. При выполнении (I-1.3), неравенства (I-1.7) и (I-1.8) эквивалентны.

Доказательство. В самом деле,

$$\sum_{\substack{i \in S \\ j \in S \\ i \neq j}} x_{ij} + \sum_{\substack{i \in S \\ j \notin S}} x_{ij} \equiv \sum_{\substack{i \in S \\ j \in V \\ i \neq j}} x_{ij} \equiv \sum_{i \in S} \sum_{\substack{j=1 \\ j \neq i}}^n x_{ij} \stackrel{(I-1.3)}{\equiv} |S| = k \quad (1.1)$$

Учитывая, что, в силу (I-1.5), все x_{ij} неотрицательны, первое слагаемое в левой части может быть меньше либо равно $k-1$ тогда и только тогда, когда второе больше либо равно 1.

Лемма 4 Любой гамильтонов цикл удовлетворяет всем ограничениям (I-1.3)-(I-1.8).

Поскольку эти и только эти ограничения используются на итерациях алгоритмов, верно следующее

Утверждение 1. Любой гамильтонов цикл является допустимым решением задачи булевого ЛП на каждом шаге Алгоритмов A1g, A2g.

Это Утверждение позволяет нам доказать

Теорему 1. Алгоритмы A1g, A2g решают задачу коммивояжера и выполняются за конечное время.

Доказательство. Алгоритмы заканчивают работу либо в случае, когда решением задачи булевого ЛП является единственный цикл, либо когда эта задача неразрешима.

Если Алгоритмы заканчивают работу, когда решением задачи БЛП на итерации является единственный цикл, то он и является решением ЗК – оптимальным гамильтоновым циклом, а значение целевой функции равно длине этого цикла. В самом деле, переменные задачи булевого ЛП принимают значения 0 или 1. Значение целевой функции будет равно сумме тех x_{ij} , индексы которых соответствуют дугам, входящим в найденный цикл. Но эта же сумма равна длине цикла в ЗК.

Покажем, что этот цикл является оптимальным решением ЗК. Предположим противное: полученный цикл не является оптимальным, то есть существует гамильтонов цикл, длина которого меньше. Но этот последний гамильтонов цикл является допустимым решением задачи булевого ЛП в силу Утверждения 1, и, значит, у задачи булевого ЛП есть решение со значением целевой функции, меньшим найденного. Мы пришли к противоречию. Если алгоритмы заканчивают работу в связи с тем, что задачу булевого ЛП (I-1.1)-(I-1.8) решить не удалось, то допустимое множество задачи булевого ЛП пусто, а это, в силу Утверждения 1, указывает, что гамильтоновых циклов в исходном графе нет, и данная ЗК не имеет решения.

Покажем, что общее количество итераций конечно. Количество простых циклов на полном ориентированном графе без кратных дуг, содержащем n вершин, равно

$$Rc(n) = \frac{n(n-1)}{2} + \frac{n(n-1)(n-2)}{3} + \dots + \frac{n!}{n} \quad (1.2)$$

Каждое слагаемое в этой сумме соответствует количеству циклов определенной длины. Первое слагаемое – это количество простых циклов длины 2, последнее слагаемое – количество циклов длины n . Каждое слагаемое, а также число слагаемых, конечны – значит, конечна и сумма. В случае неориентированного графа в сумме отсутствует первое слагаемое, и каждое из остальных слагаемых вдвое меньше.

Докажем, что каждое ограничение вида (I-1.6) запрещает один из подциклов. Оно было построено для запрета цикла, проходящего через вершины с номерами i_1, i_2, \dots, i_k . Этот цикл уже никогда не появится в решении, так как в этом цикле используются все k дуг и сумма в ограничении (I-1.6) будет равна k , что противоречит этому ограничению. Таким образом, каждое ограничение (I-1.6) уменьшает количество подциклов, которые могут появиться в решении задачи булевого ЛП на одной из итераций Алгоритмов (допустимых подциклов), как минимум на один.

Количество собственных подмножеств вершин графа с n вершинами равно

$$Rv(n)=2^n-1 \quad (1.3)$$

Покажем, что каждое ограничение (I-1.7) запрещает «изолировать» соответствующее подмножество вершин S . Более строго, решение, допустимое в силу (I-1.3)-(I-1.5) и (I-1.7) обязательно «соединяет» S и его дополнение $V \setminus S$, т.е. имеет в своем составе дугу, одна вершина которой лежит в S , а другая – в $V \setminus S$. Предположим противное. Если в решении нет ни одной дуги из вершины из S в вершину из $V \setminus S$, тогда все переменные x_{ij} , соответствующие этим дугам, равны нулю. Значит, равна нулю и их сумма, что противоречит (I-1.7). По Лемме 3, ограничение (I-1.8) эквивалентно (I-1.7) в предположении (I-1.3), поэтому указанное утверждение равно верно для ограничения (I-1.8). Отметим, что (I-1.7), как и (I-1.8) запрещает,

в частности, любой простой цикл, проходящий через все вершины соответствующего подмножества S и только через них.

Поскольку общее количество подциклов и подмножеств вершин графа конечно и ограничено величиной

$$R(n) = Rv(n) + Rc(n), \quad (1.4)$$

алгоритм закончит работу за конечное число шагов.

На каждом шаге алгоритма решается задача булевого ЛП. Количество переменных равно $n(n-1)$ для несимметричной задачи и $n(n-1)/2$ для симметричной. Поскольку каждая переменная может принимать только два значения, количество возможных решений не может превысить $2^{n(n-1)}$. Для проверки соответствия решения системе ограничений нужно проверить не более $2n$ ограничений (I-1.3)-(I-1.4), не более $Rc(n)$ ограничений вида (I-1.6), не более $Rv(n)$ ограничений типа (I-1.7) или (I-1.8). Проверка каждого ограничения требует не более $n+1$ операции для ограничения типа (I-1.6), $[n/2]*n/2$ операций для ограничения типа (I-1.7) и не более $[n/2]*([n/2]+1)$ операций для ограничений типа (I-1.8) (в два раза меньше для симметричной задачи). Значит, задача булевого ЛП будет решена за конечное время даже в случае применения метода полного перебора.

Итак, мы доказали, что Алгоритмы $A1g$, $A2g$ решают ЗК, выполняются за конечное время, и значение целевой функции на последнем шаге равно длине оптимального цикла ЗК. Теорема доказана.

Следствие. Алгоритмы $A3$, $A4$, $A5$, $A6$ решают задачу коммивояжера и выполняются за конечное время.

Теорема 2. Алгоритм $A7w$ решает задачу поиска всех оптимальных гамильтоновых циклов задачи коммивояжера за конечное время.

Доказательство. На начальном этапе Алгоритма $A7w$ используются Алгоритмы $A1g$, $A2g$, $A3$, $A4$, $A5$, $A6$, следовательно, по Теореме 1 и ее

Следствию, ЗК решается, и находится один из оптимальных гамильтоновых циклов за конечное время. Все гамильтоновы циклы, полученные в алгоритме A7w, имеют длину, равную оптимальной, и, следовательно, также являются оптимальными. Когда Алгоритм A7w получает очередной гамильтонов цикл, к системе ограничений добавляется новое ограничение вида (I-1.6), которое для данного цикла нарушается. Таким образом, однажды встретившийся гамильтонов цикл не может встретиться повторно.

Докажем, что каждый оптимальный гамильтонов цикл будет получен. Для этого убедимся, что ограничение (I-1.6), построенное для некоторого гамильтонова цикла, не запрещает ни один другой гамильтонов цикл. Рассмотрим решение задачи булевого ЛП (I-1.1)-(I-1.8) с добавленным ограничением (I-1.6) для данного гамильтонова цикла. Это добавленное ограничение может быть нарушено только в случае, если все переменные, стоящие в левой части неравенства равны единице. Но эти переменные соответствуют дугам именно данного гамильтонова цикла и не могут одновременно быть равны единице ни для какого другого гамильтонова цикла. Таким образом, ограничение (I-1.6), построенное для данного гамильтонова цикла не может запретить какой-либо другой гамильтонов цикл.

Алгоритм A7w на каждом шаге запрещает как минимум один цикл. Поскольку общее число циклов, в том числе гамильтоновых, конечно, то алгоритм закончит работу за конечное число шагов. Учитывая, что каждая итерация алгоритма выполняется за конечное время, мы доказали, что алгоритм A7w за конечное время строит все оптимальные решения ЗК.

2. Тестовые испытания

Испытание метода проводилось в три этапа. На первом этапе были решены задачи из библиотеки TSPLIB, первая версия которой была

составлена в 1991 году профессором Гейдельбергского университета Г.Райнельтом[2], и которая является стандартом де-факто для ЗК. Из 120 задач были взяты и решены двенадцать, включая симметричные метрические задачи gr24, gr48, gr96, gr120, gr202, lin318 и несимметричные задачи br17, p43, ftv33, ftv55, ftv70, ftv170. Для проведения испытаний использовались ПК со следующими характеристиками: Pentium4 3GHz, 1Gb RAM, Windows7 Max SP1 32bit (ПК1), Pentium4 3GHz, 1Gb RAM, Windows XP Pro v2002 SP2 32bit (ПК2), AMD Opteron 246 2GHz, 3Gb RAM, Windows XP Pro x64 v2003 SP1 (ПК3). Решение проводилось частными алгоритмами A3-A5 для оценки их сравнительной эффективности. Информация о времени решения задачи и количестве итераций сведена в Таблице 1. Там же даны краткие сведения о типе и размере тестовой задачи.

Решение некоторых задач выполнялось в две стадии. Сначала задачи булевого ЛП, возникающие в ходе работы алгоритма, решались с ограничением времени. Полученное решение могло быть при этом не оптимальным. Затем, начиная с некоторой итерации, алгоритм продолжал работу без ограничения времени на шаге до получения окончательного решения. На втором этапе метод был применен для решения двух новых практических ЗК, описанных ниже.



Рис. 1. Оптимальное решение задачи ESP47

В задаче ESP47 ищется кратчайший гамильтонов цикл между 47 городами континентальной европейской части Испании, являющимися столицами автономных областей и провинций. Расстояние между городами задаются не по птичьему полету, а по существующим автодорогам по состоянию на 2009 год с точностью до одного километра. Задача является симметричной, метрической. Граф задачи является полным. Результат решения задачи ESP47 на картографической подложке показан на Рисунке 1. Отметим, что маршрут, проложенный по дорогам, проходит через Севилью, Мадрид и Барселону более одного раза, поскольку являясь важными транспортными узлами соответствующих автономных областей, эти города лежат на кратчайших путях между центрами прилегающих провинций. Это не противоречит постановке ЗК, так как дуга между двумя городами не считается проходящей через,

например, Мадрид, даже если действительный путь проходит через Мадрид.

В задаче MR109 используются 109 городов Московской области по состоянию на полночь 01.07.12, когда были изменены границы Москвы и прилегающих областей. Использовались декартовы координаты городов на карте Московской области. Расстояния вычислялись по птичьему полету. Задача является симметричной, метрической. Граф задачи – полный. Результаты решения задач ESP47 и MR109 также представлены в таблице 1.

Таблица 1

Решение тестовых задач

Задача			Алгоритм А3		Алгоритм А4		Алгоритм А5	
Имя	Тип	Раз.	Время	Шаги	Время	Шаги	Время	Шаги
gr96	СЗК	96	45 час.	75	4 мин.	14	20 сек	10
gr120	СЗК	120	15 час.	13	6 час.	11	28 мин.	6
gr202	СЗК	202	-	-	53 часа	54	1 час	14
lin318	СЗК	318	-	-	-	-	27 час.	15
br17	НСЗК	17	39 час.	312	3 сек	8	3 сек	5
p43	НСЗК	43	-	-	-	-	80 мин.	16
ftv55	НСЗК	55	10 мин.	52	30 сек	9	6 сек	10
ftv70	НСЗК	70	3 мин.	17	2 мин.	10	9 сек	8
ftv170	НСЗК	170	2 часа	15	4 часа	18	8 мин.	13
ESP47	СЗК	47	2 сек	6	2 сек	6	2 сек	6
MR109	СЗК	109	25 сек	5	36 сек	5	19 сек	5

Для решения задачи MR109 потребовалось 5 итераций. На рисунке 2 изображено решение MR109 на картографической подложке. Решения, полученные на каждой итерации метода, изображены на рисунке 3.

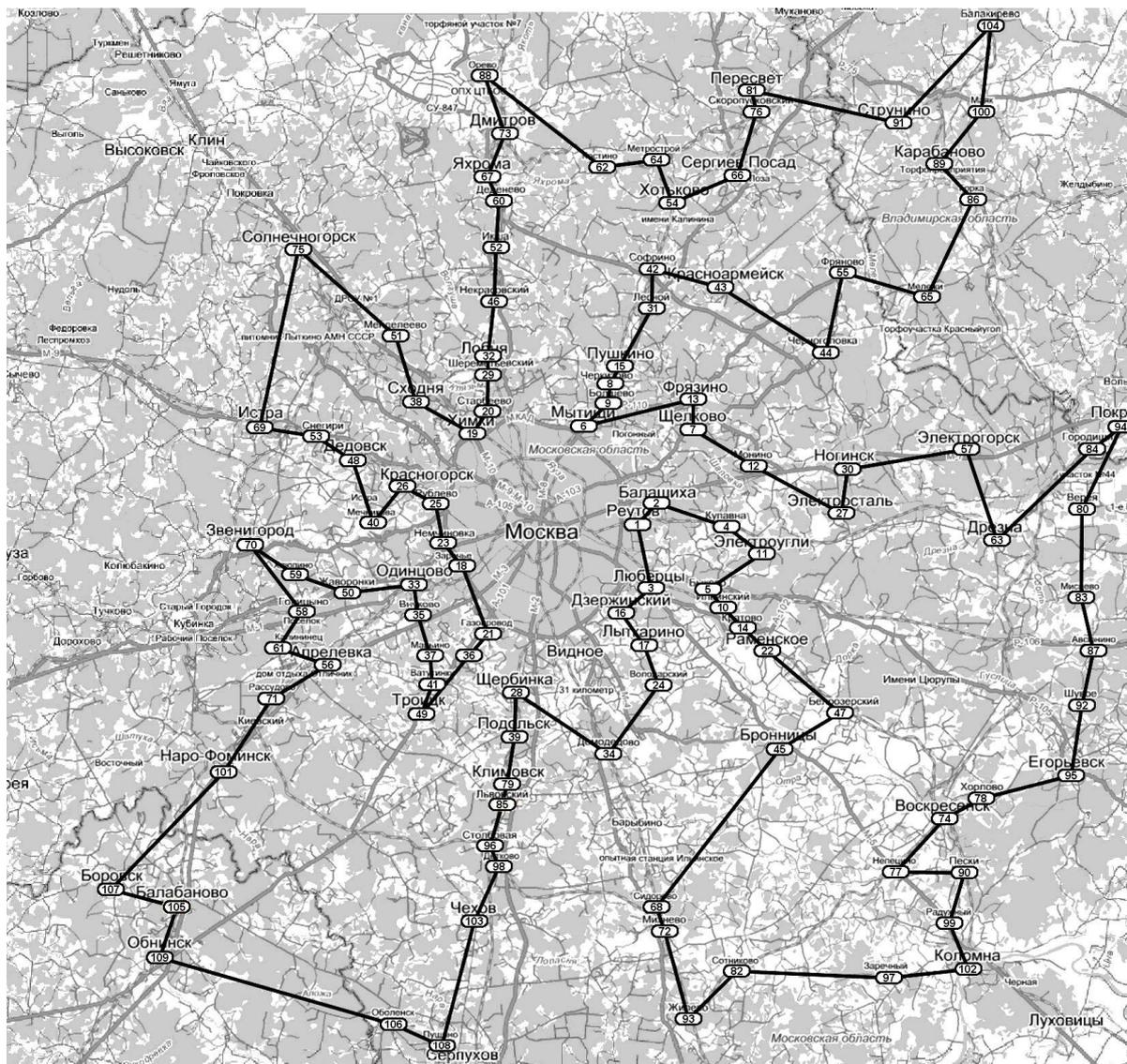


Рис. 2. Оптимальное решение задачи MR109

Несимметричная задача br17 демонстрирует недостатки алгоритма A3. В этой задаче есть кластер вершин, расстояние между которыми мало по сравнению с расстояниями до вершин, не входящих в кластер. В оптимальном решении задачи булевого ЛП на каждой итерации присутствует подцикл, проходящий только через вершины кластера. Алгоритм A3 запрещает этот подцикл, но не запрещает другие подциклы на тех же вершинах. В результате на следующем шаге получается другой подцикл на тех же вершинах. Это приводит к очень большому числу шагов алгоритма A3. Алгоритмы A4 и A5 запрещают сразу все подциклы на вершинах этой группы и, преодолевая эту проблему, легко справляются с задачей.

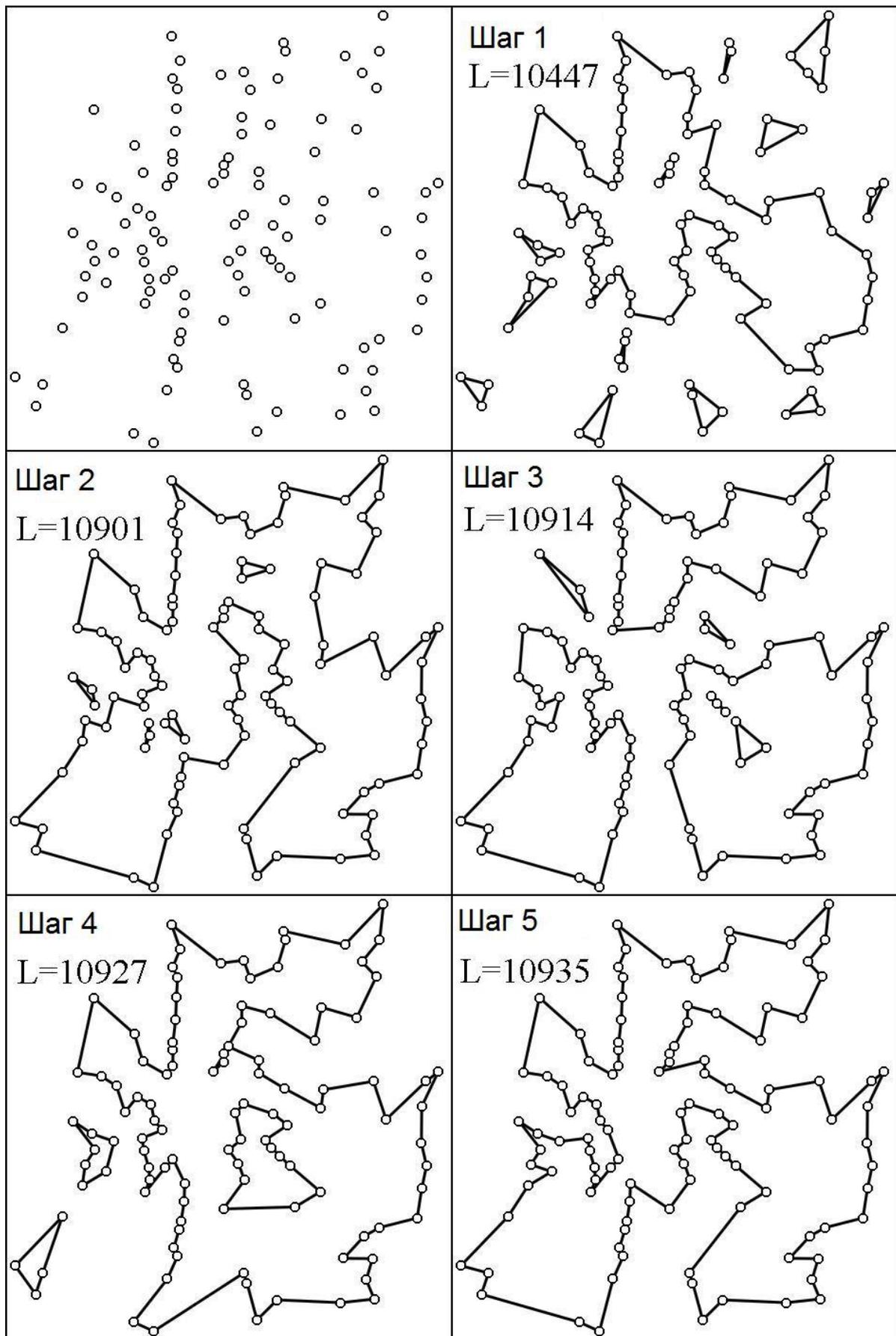


Рис. 3. Пошаговый процесс получения решения задачи MR109

В то же время, задача $ftv170$ показывает, что алгоритм А3 может оказаться эффективнее, чем А4. Это связано с тем, что ограничения в алгоритме А4 содержат больше переменных, и задачу булевого ЛП труднее решать.

На третьем этапе была изучена зависимость времени решения ЗК от размера задачи. Для алгоритма А5 были проведены исследования времени решения для случайного расположения вершин на плоскости (карта). Каждая вершина случайным образом размещалась в квадрате 1000×1000 , то есть координаты x и y каждой вершины были независимыми случайными величинами, равномерно распределенными на отрезке $[0; 1000]$. Между всеми парами вершин вычислялись декартовы расстояния, которые округлялись до целого. Количество вершин варьировалось от 80 до 140 с шагом 10. Для каждого количества вершин генерировалось от 100 до 200 случайных карт.

По результатам вычислительного эксперимента для каждого количества вершин определялись оценки медианы распределения времени решения (50%-го квантиля), 80%-го квантиля и 90%-го квантиля. Каждый квантиль показывает, для какого процента задач время решения меньше указанного. Эти оценки помещены в таблицу 2. Все величины указаны в секундах. Погрешность оценок составляет 10-15%. В последнем столбце приведено максимальное время решения.

Таблица 2

Квантили распределения времени решения метрической ЗК

Число вершин	Медиана	80% кв.	90% кв.	Максим.
80	8	16	29	208
90	12	40	80	894
100	22	72	175	11761
110	27	85	177	3547
120	64	173	403	1783
130	112	484	1429	15939
140	323	1519	3209	-

Максимальное время очень неустойчиво, имеет большой разброс и указывается для демонстрации этого разброса и того, насколько максимальное время может превышать даже 90%-й квантиль (в 7-67 раз). Это говорит о том, что у распределения очень длинный хвост.

В исследованном диапазоне количеств вершин оценка медианы растет примерно экспоненциально и увеличивается в 1,7 раза на каждые 10 вершин. Конечно, нельзя распространять эти результаты за пределы исследованного диапазона.

Количество итераций алгоритма слабо коррелирует с временем решения, не оказывая на него существенного влияния. Обычно, количество итераций находится в пределах от 5 до 20.

3. Особенности метода, области применения и пути развития

Количество шагов в алгоритмах А3-А5 с ростом числа вершин n может расти экспоненциально. Задача булевого ЛП, которая решается на каждом шаге алгоритма, также может потребовать экспоненциального по n количества операций. Применение этих алгоритмов к практическим задачам показало, что сложность задачи БЛП с ростом n растет намного быстрее, чем количество шагов алгоритма. При этом на практике, при $n=100$ количество шагов алгоритма и время выполнения одного шага оказываются небольшими.

Были выявлены ситуации, когда алгоритмы могут оказаться неэффективными. Алгоритм А3, в котором используются только ограничения С1а, может потребовать большое количество шагов, если имеется подмножество вершин (кластер вершин), расстояние между которыми намного меньше расстояния от них до вершин, не входящих в это подмножество. Чем больше вершин входит в это подмножество, тем больше шагов может потребоваться, вплоть до $(k-1)!$, где k – мощность подмножества. Ограничения С2а и С3а запрещают любой подцикл,

проходящий по всем вершинам соответствующего подмножества и только по ним, поэтому в данной ситуации они более эффективны.

Для задачи с несколькими сотнями вершин пакету GLP может потребоваться более 1 Гб оперативной памяти.

ЗК имеет многочисленные области применения. Среди них: минимизация облучения при обходе опасных зон атомной станции, схема маршрутов школьных автобусов, доставка пиццы, сбор предметов на прямоугольном складе, оптимизация сверления и управление плоттером при нанесении фотомаски при производстве печатных плат, планирование наблюдений орбитальным телескопом, оптимальная дифрактометрия в рентгеновской кристаллографии, задача о построении линейной структуры генома, планирование кольцевых сетей телекоммуникаций, проблема сбора геофизических сейсмических данных, оптимальный туристический маршрут либо маршрут через тематический парк, оптимизация инспекционных проверок, оптимизация контроля цепей при тестировании микрочипов, оптимальные кольцевые системы поддержания пластового давления на нефтяных месторождениях. Проведенные испытания показали возможность использования метода для таких задач.

Расширение типов неравенств, используемых на итерациях метода, включая комб, 2-соответствия и другие, которые формируют грани многогранника ЗК, могут, при их удачном подборе, заметно сократить количество итераций.

Важным потенциалом повышения эффективности метода является предварительная обработка данных задачи как до начала счета, так и в процессе решения на очередных итерациях, с целью существенно уменьшить размерность задачи.

Задача булевого ЛП, решаемая на каждой итерации метода, является NP-полной. Заслуживает рассмотрения возможность уменьшить трудоемкость каждой итерации, заменив полное решение задачи булевого

ЛП на приближенное решение. Это можно осуществить, ограничив время решения задачи БЛП на итерации, и варьируя это время для достижения разумного компромисса между качеством приближения и сложностью решения. Для любого допустимого целочисленного решения задачи (I-1.1)-(I-1.8) – не обязательно оптимального – сохраняется ряд важных свойств. А именно, вновь добавляемые ограничения сужают область допустимых решений, при этом ни один гамильтонов цикл из этой области не исключается. Другие важные свойства, в частности, неубывание функционала на итерациях метода и то, что полученный на любой итерации гамильтонов цикл является оптимальным решением для ЗК, перестают выполняться. Поэтому на последней итерации метода необходимо полное решение задачи БЛП. Проведенные испытания показали, что этот подход оправдывает себя для некоторых задач, несмотря на увеличение числа итераций.

Проведенные тесты выявили тенденцию к уменьшению количества шагов в алгоритме А5 по сравнению с алгоритмом А4, а у того по сравнению с А3. Это объясняется тем, что в алгоритме А3 используются только Ограничения С1а, а в других алгоритмах - еще и Ограничения С3а. Первое ограничение запрещают один подцикл, в то время как последнее - сразу несколько подциклов. В результате количество шагов оказывается меньше. В тестовых задачах Ограничения С3а обрабатывались модулем GLP быстрее, чем аналогичные им Ограничения С2а. Но могут существовать задачи, в которых соотношение противоположное.

Другой путь развития метода состоит в отказе от использования библиотечного блока решения задачи БЛП и замене его на блок собственной сборки или дизайна, учитывающий специфику ограничений задачи БЛП, формируемой на каждой итерации и, особенно, их кумулятивность (сохранение на более поздних итерациях всех ограничений, сформированных на более ранних) и относительную

стабильность (как правило, количество добавляемых в множество ограничений элементов существенно меньше мощности этого множества).

Обобщение предложенного метода для решения других задач комбинаторной оптимизации кажется весьма перспективным и рассматривается как предмет дальнейших исследований авторским коллективом.

Библиографический список:

1. Козлов М. В., Костюк Ф. В., Сорокин С. В., Тюленев А. В. Решение задачи коммивояжера методом целочисленного линейного программирования с последовательным исключением подциклов: описание и алгоритмическая реализация // Advanced Science. – 2012. – №2.

2. Reinelt, G. TSPLIB – A traveling salesman problem library // ORSA Journal on Computing. 1991. № 3, P.376–384.

Информация об авторах

Data on authors

Козлов Михаил Васильевич – инженер-исследователь, ВЦ РАН

Тел.: (499)135-0170

E-mail: kz2pit@rambler.ru

Kozlov Mikhail Vasilievich – research engineer, CC RAS

Tel.: (499)135-0170

E-mail: kz2pit@rambler.ru

Костюк Феликс Владиславович – инженер-исследователь, ВЦ РАН

Тел.: (499)135-0170

E-mail: fil@ccas.ru

Kostyuk Felix Vladislavovich – research engineer, CC RAS

Tel.: (499)135-0170

E-mail: fil@ccas.ru

Сорокин Станислав Викторович – ведущий программист, ВЦ РАН

Тел.: (499) 135-14-98

E-mail: www2011@ccas.ru

Sorokin Stanislav Victorovich – leading software designer, CC RAS

Tel.: (499) 135-14-98

E-mail: www2011@ccas.ru

Тюленев Александр Васильевич – к.э.н., математик, ВЦ РАН

Тел.: (499) 135-4250

E-mail: tyulavas@yandex.ru

Tyulenev Alexander Vasilievich – PhD. in economics, mathematician, CC
RAS

Tel.: (499) 135-4250

E-mail: tyulavas@yandex.ru

ПАРАДОКСЫ СИСТЕМНОЙ РЕГРЕССИИ

PARADOXES SYSTEMS REGRESSION

Показана противоречивость современных подходов к моделированию и практическому использованию систем эконометрических уравнений. Сделан анализ некоторых причин парадоксов. Сделана попытка поиска полезности таких моделей.

Shows the contradictory nature of modern approaches to modelling and practical use of the systems of econometric equations. Made the analysis of some the causes of the paradoxes. The attempt is made to search of use of systems models.

Ключевые слова: эконометрика, регрессия, системы эконометрических уравнений, проблемы идентифицируемости.

Keywords: econometric, regression, systems of econometric equations, the problems of identifications.

Введение

Несмотря на развитие новых направлений в современной эконометрике, некоторые её традиционные разделы, по-видимому, просто устарели. Особенно это относится к системной регрессии. В этом разделе не только не видно развития (в смысле новых моделей и методов), но и, как нам кажется, имеются просто ошибочные подходы. Непонятны, например, такие аспекты:

- зачем вообще нужны системы и как их можно практически использовать?
- почему для идентификации систем всегда нужны специфические методы?

– почему исходная информация по наблюдаемым данным всегда избыточна для идентификации (поиска числовых значений параметров) систем и всегда необходимо исключить из некоторых уравнений системы часть переменных?

В первой части настоящей работы рассмотрены основные противоречия в сложившихся подходах к регрессионным задачам.

Во второй части проанализированы причины некоторых противоречий.

В заключении приведены основные выводы по изложенной проблеме.

1. Некоторые сложившиеся парадоксы в задачах регрессии

Начнём с парной регрессии. В математической статистике выборочные уравнения парной регрессии обычно записывают системно в виде

$$\begin{cases} \overline{y/x} - \bar{y} = b_{yx} (x - \bar{x}) \\ \overline{x/y} - \bar{x} = b_{xy} (y - \bar{y}) \end{cases} \quad (1.1)$$

Параметры уравнений могут быть найдены по методу наименьших квадратов (МНК) и считаются состоятельными.

В эконометрике «забывают» про «симметричное» уравнение, а оставшееся записывают с остатками ε

$$y = a + bx + \varepsilon \quad (1.2)$$

Пусть, например, x – выпуск, y – издержки и по данным примера из [1] с помощью МНК получены уравнения

$$\begin{cases} y = -5.79 + 36.84x + \delta_1 \\ x = 0.21 + 0.027y + \delta_2 \end{cases} \quad (1.3)$$

Парадокс 1: в (1.3) видим, что переменные x и y связаны друг с другом, а, следовательно, и с остатками. Поэтому (по теореме Гаусса-Маркова) МНК не позволит получить состоятельные оценки параметров. Но в парной

регрессии всегда есть «симметричное» уравнение (во множественной их много). Значит ли это, что МНК не применим никогда?

Парадокс 2: числовые значения параметров регрессии в моделях (1.1) и (1.3) при одних и тех же данных совпадают, но в записи (1.1) они состоятельны [2], а в (1.3) – нет.

Парадокс 3: модель (1.3), где обе переменные могут оказаться как слева, так и справа от знака равенства, относят системе одновременных уравнений (СОУ) или же к структурной форме (СФ) записи модели. Прогнозной же принято считать [1] приведённую форму (ПФ), которая применительно к (1.3) вырождается в такую

$$\begin{cases} y_1 = 3.14 + \varepsilon_1 \\ y_2 = 110 + \varepsilon_2 \end{cases} \quad (1.4)$$

Формы (1.3) и (1.4) придётся признать *абсолютно* идентифицируемыми, т. к. подстановка «прогнозов» из ПФ (1.4) в СФ (1.3) даст *абсолютно* те же числовые результаты. Однако формально модель (1.3) вообще не идентифицируема, т. к. не удовлетворяет общепринятым условиям, изложенным, например, в [1, 2, 3]. Кроме того, несмотря на одинаковость прогнозов, характеристики надёжности моделей разные.

Дальнейшие парадоксы обусловлены в основном тем, что обычный МНК для систем игнорируется. В «обход» придуманы методы, связанные с априорным редуцированием параметров возможной модели (без этого якобы возникает проблема идентифицируемости). В результате даже в учебных примерах, например в [1, 2], получились абсурдные результаты, явно оторванные от исходных данных. Оторванность в том, что общая сумма квадратов (зависящая только от исходных данных) получается меньшей, чем факторная плюс остаточная (обе зависят от модели). Так в [1] остаточная сумма может превышать общую в сотни раз [4]. В примере [3] разница не так значительна, но «закон сохранения» всё же не выполняется.

2. Основные причины сложившихся противоречий

По нашему мнению, при рассмотрении уравнений регрессии (парной и множественной) системно, например как в (1.3), нельзя априорно предположить зависимость остатков и регрессоров. А именно этим объясняется неприменимость обычного МНК к системам в нередуцированном виде. Для редуцированных систем применяют такие «необычные» методы, как косвенный двух- и трёхшаговый МНК и др. Необычность в том, что первым делом модель приходится портить для того, чтобы обеспечить так называемую идентифицируемость СФ и ПФ. При этом уравнения ПФ в общем-то независимы, а переменные слева от знака равенства в отличие от СФ в виде СОУ не могут оказаться справа. В ПФ переменные вроде бы чётко разделяются на эндогенные (слева) и предопределённые (справа), а в СФ принято считать, справа от знака равенства могут быть и эндогенные переменные. По сути, в этом мы уподобляемся барону Мюнхгаузену, ибо хотим одну и ту же переменную представить и как фактор, и как результат. Барон взял себя (фактор) за волосы и вытащил себя (результат) из болота. Т. к. реально это вряд ли возможно, то мы вынуждены считать, что переменные справа и слева от знака равенства, даже имея одинаковую смысловую нагрузку, (например, «затраты») являются *разными* с точки зрения своей роли в модели. Любая переменная справа от знака равенства не может быть эндогенной (оцениваться внутри данной модели в текущий момент). На этапе идентификации значения этих переменных мы берём из выборки. Пусть целью построения модели является прогнозирование значений эндогенных переменных. Очевидно, что прогнозное значение такой переменной (например, имеющей смысл «затраты») не совпадёт со значениями переменной «затраты», входящей в другие уравнения справа. Впрочем, встаёт вопрос, а откуда же взять эти значения? Относительно просто вопрос решается в рекурсивных системах: там, на этапе прогнозирования, одно из уравнений волевым решением назначается независимым от эндогенных переменных, следующее – может содержать только одну эндогенную переменную, значение которой определяется из

предыдущего уравнения и т. д. Сразу и не заметно, что это – то же вытягивание за волосы.

Если же признать, что переменные справа *все* экзогенны, то уже нельзя априорно заподозрить зависимость их и остатков, они должны обладать «презумпцией» независимости. В изложении [4] случайная компонента в уравнении регрессии обусловлена неконтролируемыми факторами (видимо, эту случайность должен учитывать закон распределения), а также ошибками измерений наблюдаемых величин. По нашему мнению, факт стохастичности регрессоров не означает их связь с остатками, да и дисперсия результирующей переменной при этом не изменяется. Пусть, например, по оси абсцисс на рис. 1 отложены наблюдаемые значения регрессора, по оси ординат – результата, связь между переменными неслучайна (остатки нулевые). Тогда как бы случайно ни распределился в выборке регрессор, результат неизбежно окажется на линии регрессии (с нулевой дисперсией).

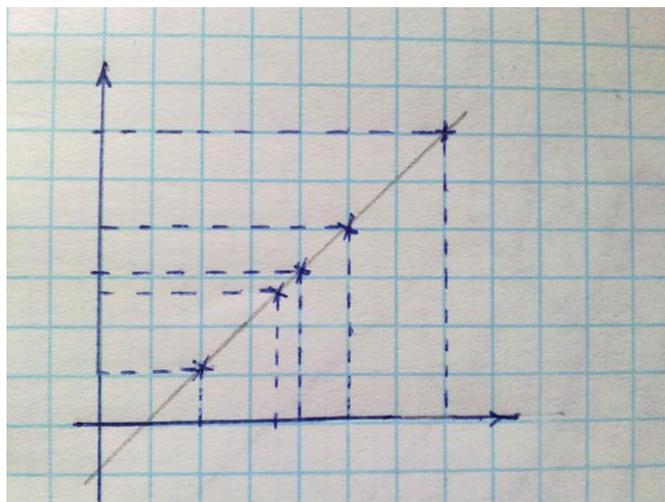


Рис. 1

Если остатки не нулевые (рис. 2), то на двумерном графике трудно сделать вывод об их зависимости с регрессором.

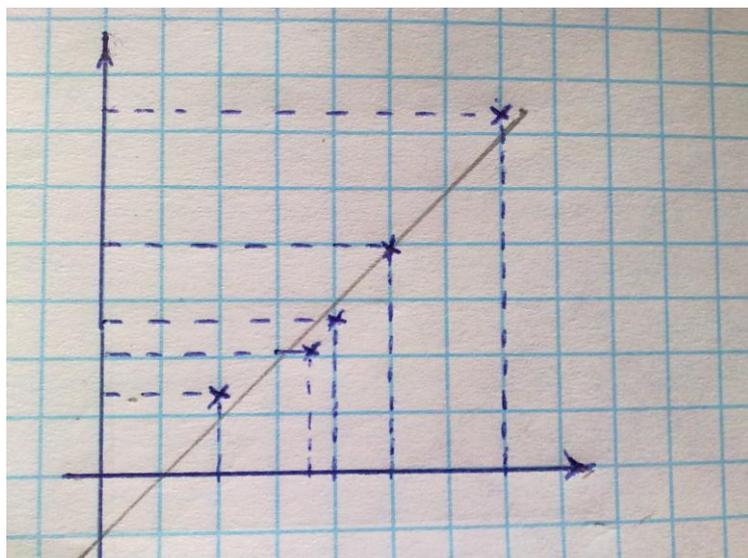


Рис. 2

Однако в случае гомоскедастичности остатков (рис. 3) на трёхмерном графике (вертикальная ось – плотность распределения остатков) видно, что остатки распределены независимо от регрессора.

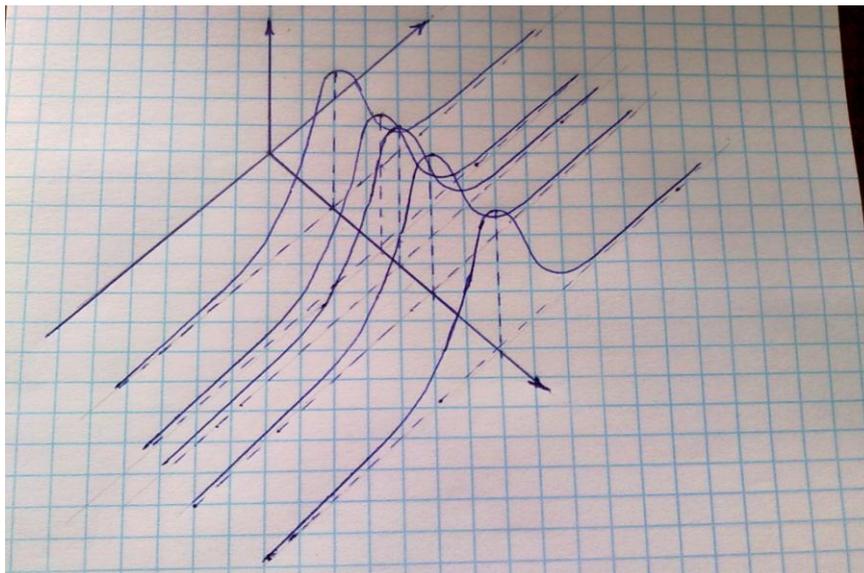


Рис. 3

Если же имеем ошибки измерения, то они-то почему должны быть взаимозависимыми с остатками? Здесь, конечно, возможны специфические задачи, когда ошибки связаны, например, с опросом лиц по их доходам (появится смещение). Многие задачи, связанные с производством, потреблением, продажами и др., используют строго документированную статистическую информацию. Здесь об ошибках измерения говорить не приходится. В силу изложенного, в большинстве случаев в системной

регрессии нет необходимости прибегать к редуцированным формам моделей и соответствующим им специальным методам, т. к. нет препятствий по применению МНК.

3. Заключение

Вопрос о том, где взять значения экзогенных переменных для прогнозирования, остался пока открытым. Ясно, что внутри одной модели (пусть и системной) одна и та же переменная не может быть и эндогенной, и экзогенной. Но ответ, как нам кажется, банален. Как и в обычной множественной регрессии, значения переменных для прогноза нужно брать извне. В частности, для этого могут быть использованы другие модели, где предварительные оценки переменных для прогнозов могут быть определены, например, методом экспертных оценок, как это предложено в [5]. Другой подход рассмотрен в [6], где системные переменные зависят не только друг от друга, но также и от времени. При этом предварительные прогнозы осуществляются из соответствующих временных рядов, а результирующие – с учётом системных взаимосвязей между переменными.

Библиографический список:

1. Эконометрика / Под ред. И. И. Елисеевой. – М.: Финансы и статистика, 2007. – 576 с.
2. Эконометрика / Под ред. В. С. Мхитаряна. – М.: Проспект, 2009. – 384 с.
3. Айвазян С. А. Основы эконометрики / Прикладная статистика. Основы эконометрика. – Т. 2. – М.: ЮНИТИ-ДАНА, 2001. – 432 с.
4. Айвазян С. А., Мхитарян В. С. Теория вероятностей и прикладная статистика / Прикладная статистика. Основы эконометрика. – т. 1. – М.: ЮНИТИ-ДАНА, 2001. – 656 с.
5. Кошкин Ю. Л. Системность в эконометрическом моделировании. V Всероссийская научная конференция ЭКОМОД-2010 / Сб. трудов. – Киров: ГОУ ВПО ВятГУ, 2010. – С. 89-97

6. Кошкин Ю. Л. / Структурно-временные модели (СВМ) [Электронный ресурс] / Ю. Л. Кошкин, М. А. Ушакова // Общество, наука, инновации (НТК-2011): ежегод. открыт. всерос. науч.-технич. конф., 18-29 апр. 2011.-/электрон. опт. Диск (CD-ROM). (Социально-экономический факультет. Секция «Математические методы и модели в экономике». Статья № 4)

Информация об авторах

Data on authors

Кошкин Юрий Леонидович – к.т.н., доцент ВятГУ

Тел.: (8332)51-17-17

E-mail: Koshyuri@gmail.com

Koshkin Yuri – kand. of Teh. Science, docent VyatSU

Tel.: (8332)51-17-17

E-mail: Koshyuri@gmail.com

МНОГОКРИТЕРИАЛЬНАЯ СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

MULTI-CRITERIA DECISION SUPPORT SYSTEM

В работе представлена разработанная система многокритериального выбора, которая реализует работу двух методов нахождения оптимального многокритериального решения, проведён анализ экспериментов, полученных разными методами (метод линейной свертки и метод главного критерия).

This report describes a system of multi-choice, which sells the work of two methods for finding optimal multi-criteria solutions, the analysis of the experiments obtained by different methods (method of linear convolution and main criterion method).

Ключевые слова: система многокритериального выбора, критерии, альтернативы, система Quick choise.

Keywords: system of multi-choice, criteria, alternative, Quick choise system.

Процессы принятия решения лежат в основе любой целенаправленной деятельности. При создании новой техники они составляют важный этап в проектировании новых машин и механизмов, устройств, приборов, комплексов, в разработке технологии их производства и эксплуатации. В экономике они предшествуют созданию производственных и хозяйственных организаций, обеспечивают их оптимальное функционирование и взаимодействие. В научных исследованиях позволяют выделить важнейшие научные проблемы, найти способы их изучения и т. д.

Оптимальные решения позволяют достигать цели при минимальных затратах трудовых, материальных и сырьевых ресурсов. К сожалению, при принятии решений приходится иметь дело с такими большими объемами информации о различных вариантах решений, что их анализ становится задачей, непосильной для лица принимающего решение (ЛПР).

На сегодняшний день не существует такой системы многокритериального выбора вариантов, которая позволяла бы учитывать все запросы современного ЛПР с точки зрения максимального соответствия действительности. В настоящее время практически нет открытых публикаций по системам принятия решений в условиях многокритериальности, и они малоизвестны.

Постановка задачи.

Рассматривается следующая математическая модель задачи принятия решений (ПР):

- X – множество альтернатив;
- Y – множество исходов;
- $f_i: Y \rightarrow R, i = 1..m$ — множество показателей качества (критериев);
- $\varphi: X \rightarrow Y$ – детерминистская функция, отображающая множество альтернатив во множество исходов.

(Здесь R – множество вещественных чисел).

Таким образом, каждому решению $x \in X$ соответствует единственный элемент $y \in Y$, где $y = \varphi(x)$ «качество» или «полезность» исхода y , а тем самым и соответствующего решения x оценивается несколькими (m) числами в соответствии с зависимостями f_i . С помощью суперпозиции

$$J_i(x) = f_i(\varphi(x)), i = 1..m$$

имеем возможность непосредственно оценивать качество самого решения x .

В результате получаем многокритериальную модель принятия решений, или задачу многокритериальной оптимизации вида

$$J_i(x) \rightarrow \max_{x \in X}, i = 1..m, X \subset R^n$$

Предполагается, что все альтернативы параметризованы и каждой из них соответствует точка $x \in R^n, x = (x_1 \dots x_n)$

Для решения задачи многокритериального выбора существует несколько подходов, но какой из них даёт лучшие результаты неизвестно, поэтому необходимо провести эксперимент и исследовать их. Для реализации системы многокритериального выбора вариантов были выбраны и рассмотрены два метода выбора оптимальной альтернативы: метод главного критерия и метод линейной свертки.

Метод главного критерия [1, 5, 6]. В методе главного критерия в качестве целевой функции выбирается один из критериев f_i , например f_1 , наиболее полно с точки зрения исследователя отражающий цель ПР. Остальные требования к результату, описываемые критериями $f_2 \dots f_n$, учитываются с помощью введения необходимых дополнительных ограничений. Таким образом, вместо задачи многокритериального выбора решается другая, уже однокритериальная задача.

Формально получается более простая задача поиска максимума функционала f_1 на новом допустимом множестве. Добавляются ограничения вида $f_i(x) > t_i$, показывающие согласие не добиваться максимальных значений для функционалов $f_2 \dots f_n$ сохраняя требование их ограниченности снизу на приемлемых уровнях. Важно понимать, что переход от задачи многокритериального выбора к новой задаче вовсе не есть переход от одной эквивалентной задачи к другой. Произошло существенное изменение исходной постановки задачи, которое в каждой конкретной ситуации требует отдельного обоснования. Применение этого метода на интуитивном уровне обычно наталкивается на трудности, связанные с возможным наличием нескольких «главных» критериев, находящихся в противоречии друг с другом. Кроме того, не всегда ясен алгоритм выбора нижних границ t_i . Их

необоснованное задание может привести, в частности, к пустому множеству альтернатив.

Метод линейной свертки [1, 4, 6]. Это наиболее часто применяемый метод «скаляризации» (свёртки) задачи многокритериального выбора, позволяющий сопоставить конкретное число каждой альтернативе. Он основан на линейном объединении всех частных целевых функционалов в один

$$A_i = \sum_{j=1}^m k_j a_{ij} \quad i = 1..n,$$

где A_i - вес i -ой альтернативы;

k_j – вес j -го критерия;

a_{ij} – значение i -ой альтернативы по j -му критерию.

Чем большее значение мы придаем j -му критерию, тем больший вклад он даёт в сумме. При наличии разнохарактерных частных критериев достаточно сложно указать окончательный набор коэффициентов k_j , исходя из неформальных соображений, связанных, как правило, с результатами экспертного анализа.

Рассмотренные методы получения оптимального решения в многокритериальной задаче выбора вариантов существенно отличаются друг от друга. Метод главного критерия сводит задачу многокритериального выбора к однокритериальной задаче, а метод линейной свёртки учитывает все критерии. Для того чтобы определить, какой из подходов более полно решает поставленную задачу методы были реализованы, после чего был проведен эксперимент, показывающий поведение методов в конфликтной ситуации.

Разработанная в Вятском государственном университете автоматизированная система многокритериального выбора вариантов использует в качестве аналога известную систему выбора альтернатив Quick Choice[1].

В разработанной системе многокритериального выбора вариантов предусмотрены следующие возможности:

– задание критериев и их значений (значения непрерывных критериев могут принимать рациональные значения и находиться в диапазоне от $1,7 \cdot 10^{-308}$ до $1,7 \cdot 10^{308}$ (тип переменных double), а значения перечислимых критериев – строки неограниченной длины (тип переменных string);

– задание альтернатив и их значений;

– поиск оптимальной альтернативы (альтернатив) по заданным пользователем критериям с учётом их важности для ЛПР.

Система многокритериального выбора вариантов разработана в среде Microsoft Visual Studio 2008 на языке C++.

В процессе функционирования программы передаются данные о критериях и альтернативах между окнами приложения. Для этого были разработаны два независимых класса: класс критериев и класс альтернатив. Информация о критериях и альтернативах передаётся как массивы экземпляров классов, что значительно упрощает работу с критериями и обработку множества альтернатив.

При нахождении оптимального решения методом главного критерия из множества альтернатив выбираются альтернативы с наилучшим значением главного критерия, если таких альтернатив несколько, то берётся следующий по значимости критерий, по которому снова выбираются лучшие альтернативы по данному критерию и т.д., пока не будет найдена оптимальная альтернатива.

При нахождении оптимального решения методом взвешенных критериев, каждой альтернативе сопоставляется её вес. Альтернативы с наибольшим весом выводятся как результат работы программы.

Для тестирования работы программы и сравнения результатов по двум алгоритмам рассмотрена задача принятия решения о выборе квартиры из множества имеющихся альтернатив. Результаты экспериментов, представленные на рис.1, показывают, что метод главного критерия предлагает пользователю только один оптимальный вариант (квартира под

№4), в то время как метод линейной свёртки предлагает две альтернативы на выбор (квартиры под №1 и №2).

QuickChoice

Файл Критерии Альтернативы Справка

Результаты метода главного критерия

	Название	Общая площадь	Кол комнат	Цена	Ремонт	Площадь кухни	Район	О
	4	63,4	3	5000000	Евро	13	Радужный	Де
	2	56,6	2	2700000	Евро	10	Ленинский	Де
	3	42,8	1	1500000	Черновая	12	Центр	Де
	1	56,6	2	2700000	Евро	10	Центр	Де
▶*								

Результаты метода линейной свёртки

	Название	Общая площадь	Кол комнат	Цена	Ремонт	Площадь кухни	Район	О
▶	1	56,6	2	2700000	Евро	10	Центр	Де
	2	56,6	2	2700000	Евро	10	Ленинский	Де
	3	42,8	1	1500000	Черновая	12	Центр	Де
	4	63,4	3	5000000	Евро	13	Радужный	Де
*								

Рис. 1. Результаты решения многокритериальной задачи выбора квартиры методом главного критерия и методом линейной свертки

В случае принятия решения методом линейной свёртки пользователь может либо сам выбрать один из вариантов, либо изменить ограничения по некоторым критериям и повторить эксперимент.

Решение многокритериальной задачи методом линейной свёртки оказывается в данном случае более близким к действительности, чем результат, сформированный методом главного критерия. Это свидетельствует о том, что метод главного критерия целесообразно применять, когда имеется только один главный критерий, существенно отличающийся от всех остальных критериев[6].

При необходимости учёта всех критериев целесообразнее решать задачу методом линейной.

Заключение

Разработанная система многокритериального выбора вариантов предлагает пользователю выбрать один из двух методов нахождения оптимального решения, что позволяет ЛПР в зависимости от цели оптимизации выбрать наилучшую альтернативу. Программа не привязана к конкретной предметной области, что позволяет охватить широкий спектр задач выбора оптимального варианта решений.

Библиографический список:

1. Черноруцкий И. Г. Методы принятия решений. СПб: БХВ-Петербург, 2005.
2. Батищев Д. И., Шапошников Д. Е. Многокритериальный выбор с учетом индивидуальных предпочтений. – Нижний Новгород: ИПФ РАН, 1994.
3. Методы многокритериального выбора вариантов [Интернет-ресурс]. – <http://bigor.bmstu.ru/?cnt/?doc=МО/base.cou>
4. Электронное научно-техническое издание «Наука и образование» [Интернет-ресурс]. – <http://technomag.edu.ru/doc/141623.html>.
5. Планирование маркетинговой деятельности в условиях полной определенности, неопределенности ресурсов [Интернет-ресурс]. – <http://www.informancer.ru/uncategorized/planirovanie-marketingovoj-deyatelnosti-v-usloviyax-polnoj-opredelennosti-neopredelennosti-i-riska/>
6. Многокритериальное принятие решений [Электронный ресурс]. – <http://www.levvu.narod.ru/Papers/Multicrit.pdf>

Информация об авторах

Data on authors

Лопаткина Мария Андреевна – студентка ФГБОУ ВПО «ВятГУ»

Тел.: 89195123441

E-mail: ms777_07@mail.ru

Lopatkina Maria Andreevna – Student FGEI HPE «VyatSU»

Тел.: 89195123441

E-mail: ms777_07@mail.ru

Зобнин Константин Васильевич – студент ФГБОУ ВПО «ВятГУ»

Тел.: 89195123445

E-mail: dpblhbl4@mail.ru

Zobnin Konstantin Vasilievich – Student FGEI HPE «VyatSU»

Тел.: 89195123441

E-mail: dpblhbl4@mail.ru

Ростовцев Владимир Сергеевич – к.т.н., доцент кафедры ЭВМ ФГБОУ
ВПО «ВятГУ»

Тел.: 89229077847

E-mail: rostov_kirov@mail.ru

Rostovcev Vladimir Sergeevich – Cand. Sc. (Engineering), Associate Prof. on
Electric Power Systems FGEI HPE «VyatSU»

Tel.: 89229077847

E-mail: rostov_kirov@mail.ru

ИССЛЕДОВАНИЕ РАВНОВЕСИЯ В ЗАДАЧЕ ЗАЕМЩИКА И КРЕДИТОРА

EQUILIBRIUM IN BORROWER AND LENDER PROBLEM

В работе построена и исследуется модель взаимодействия заемщика и кредитора в условиях асимметричной информации. Исследованы случаи совершенной конкуренции и игры двух игроков. Доказано, что в случае совершенной конкуренции не существует равновесия, в котором спрос равен предложению, но есть возможность рационирования. В игре двух игроков равновесие существует только в случае асимметричной информации.

A model of borrower and lender interaction under asymmetric information is developed and investigated. The cases of perfect competition and game of two players are considered. It is proved that in perfect competition market there is no equilibrium in which demand equals supply, but there is a possibility of rationing. In two-player game equilibrium exists only in when the information is asymmetric.

Ключевые слова: равновесие, кредитный рынок, заемщик и кредитор, рационирование кредита.

Key words: equilibrium, credit market, borrower and lender, credit rationing.

Введение

В большинстве моделей кредитного рынка рассматривается два типа игроков: заемщик – предприятие, нуждающиеся в капитале для осуществления проекта и кредитор. В некоторых моделях, например [1] и [2] доход является

случайной величиной, и тип заемщика определяется параметром функции распределения доходов (в дальнейшем будем обозначать его θ , причем с ростом θ увеличивается дисперсия дохода). В других (например [3]) различие состоит в индивидуальных характеристиках, а именно в различных потерях при объявлении дефолта.

В случае симметричной информации – если кредитор знает тип заемщика – задача поиска равновесия разбивается на множество более простых задач. В случае ассиметричной информации кредитор вынужден применять одну и ту же стратегию ко всем типам игроков. При этом наблюдается эффект отрицательного отбора. Он заключается в том, что фирма берет кредит в том и только в том случае, если ее параметр θ больше θ_0 , причем θ_0 увеличивается с ростом ставки кредита. Так как математическое ожидание доходов банка является убывающей функцией θ (доказательство представлено в [1]), это не выгодно для кредитора. Один из методов борьбы с отрицательным отбором – это рacionamento кредита (credit rationing) ([3], [1]), представляющее собой ограничение предложения кредита со стороны банка. В равновесии с рационированием увеличивается как прибыль безрисковых заемщиков, так и доход кредитора.

Важным различием рассмотренных моделей является используемое в них определение понятия равновесия. В большинстве статей рассматривается либо конкурентное равновесие, либо монополия. В данной работе задача поиска равновесия будет ставиться в терминах теории игр. Построена модель, основанная на [3], [1] и [4]. Основным усложнением по сравнению с указанными моделями, является учет зависимости прибыли заемщика от величины кредита. Были поставлены следующие задачи:

- 1) Исследование равновесия в случае совершенной конкуренции;
- 2) Построение игры в стратегической форме и байесовской игры для задачи с ассиметричной информацией, нахождение и исследование равновесий;

3) Исследование модели совершенной конкуренции на возможность рационализации.

Построение модели

В момент 0 заемщик получает от кредитора сумму L . В момент 1 он получает прибыль P , являющуюся случайной величиной и отдает задолженность в размере RL (R – ставка кредита, будем называть пару (L, R) контрактом). Если $P < RL$, заемщик объявляет дефолт и отдает всю прибыль.

Основные предположения:

- 1) Все игроки риск-нейтральны;
- 2) Расходы кредитора на обеспечение кредита равны Π . Экономическая интерпретация I – ставка межбанковского кредита;
- 3) Каждый заемщик имеет параметр риска $\theta \in [0, \theta_{\max}]$ $1/2 \leq \theta_{\max} < 1$, причем его распределение (доля заемщиков со значением параметра на $(\theta; \theta + d\theta)$) $g(\theta)d\theta$ и это общеизвестная информация. $g(\theta)$ непрерывно дифференцируема на отрезке $[0, \theta_{\max}]$ и $g'(\theta) \leq 0$;

4) Доход заемщика $P(L, \theta) = U(L)(1 + X(\theta))$, где X – случайная величина, принимающая значения $\{\theta/(1-\theta), -1\}$, $p(0) = \theta$. Легко увидеть, что математическое ожидание X равно 0. На функцию U накладываются ограничения:

4.1) U дважды дифференцируема, $U' > 0, U'' < 0$;

4.2) $U/L > I$, $\lim_{L \rightarrow \infty} U(L)/L = I$ где I – безрисковая ставка кредита,

входящая в функцию прибыли кредитора;

4.3) $U'(0) > I/(1 - \theta_{\max})$. Смысл этого условия заключается в том, что

существует контракт, приносящий прибыль всем заемщикам и кредитору.

Равновесие в модели совершенной конкуренции.

В этой части мы докажем отсутствие объединяющего (pooling) равновесия – такого, в котором банк предлагает одинаковые условия для всех типов заемщиков. Этот результат был получен в [6] для рынка страхования.

Ожидаемая прибыль заемщика

$$P(L, R, \theta) = U(L) - (1 - \theta)RL \quad (1)$$

Кривая спроса задается условием, что при каждом фиксированном R заемщик максимизирует свою полезность

$$\begin{aligned} L_\theta(R) &= \arg \max(U(L) - (1 - \theta)RL) \\ R(1 - \theta) &= U'(L_\theta), \text{ при } U'(0) / (1 - \theta) > R > I / (1 - \theta) \\ L_\theta &= 0, \text{ при } R \leq U'(0) / (1 - \theta) \\ L_\theta &= \infty, \text{ иначе} \end{aligned} \quad (2)$$

При известном типе заемщика ожидаемая прибыль банка

$$(R - I)L * Ind(U(L) > (1 - \theta)RL) \quad (3)$$

где $Ind(A)$ – индикаторная функция множества A . Эта формула учитывает отрицательный отбор – при достаточно высоких ставках заемщики с низкой степенью риска не будут брать кредит. Так как банк не знает θ заемщика, то он руководствуется математическим ожиданием (аналогично модели [1]).

$$E(Q) = RL \frac{\int_{\theta^*}^{\theta_{\max}} (1 - \theta)g(\theta)d\theta}{\int_{\theta^*}^{\theta_{\max}} g(\theta)d\theta} - IL \stackrel{\text{def}}{=} \frac{\lambda(R, L)RL}{\mu(R, L)} - sIL \quad (4)$$

где $\theta^* = \min(\max((1 - U(L) / RL), 0), \theta_{\max})$ – минимальная степень риска заемщика, при которой тот согласится брать кредит на условиях (L, R)

Кривая предложения определяется условием нулевой прибыли:

$$R\lambda(R, L) - \mu(R, L)I = 0 \quad (5)$$

Обозначим $\lambda_0 = 1 - E(\theta)$. Так как $\lim_{L \rightarrow 0} U(L)/L = U'(0)$ и в силу непрерывности U и условия 4.3 на функцию U при малых L выполняется $\theta^*(L, I\mu_0/\lambda_0) = \min(1 - U(L)\lambda_0/LI\mu_0, 0) = 0$. Следовательно, при малых L кривая предложения имеет вид $R = I/\lambda_0$

Утверждение 1.1

Кривая предложения имеет асимптоту $R = \frac{I}{1 - \theta_{\max}}$

Доказательство. При $R > \frac{I}{(1 - \theta_{\max})}$ выполняется $\theta^* < \theta_{\max}$ и

$$R\lambda(R, L) - \mu(R, L)I > \int_{\theta^*}^{\theta_{\max}} \left(\frac{I}{(1 - \theta_{\max})} (1 - \theta) - I \right) g(\theta) d\theta > 0 \quad (6)$$

При любом фиксированном $R < \frac{I}{1 - \theta_{\max}}$

$$\lim_{L \rightarrow \infty} R\lambda(R, L) - \mu(R, L)I = \int_{\frac{1 - I}{R}}^{\theta_{\max}} (R(1 - \theta) - I) g(\theta) d\theta < 0 \quad (7)$$

Утверждение 1.2

При совершенной конкуренции не существует равновесия без рационирования.

Доказательство. Совокупный спрос равен:

$$L(R) = \int_0^{\theta_{\max}} L_{\theta}(R) g(\theta) d\theta \quad (8)$$

Свойство совокупной кривой спроса:

$$L(I / (1 - \theta_{\max})) = \infty \quad (9)$$

Из вида кривых следует, что равновесия не существует:

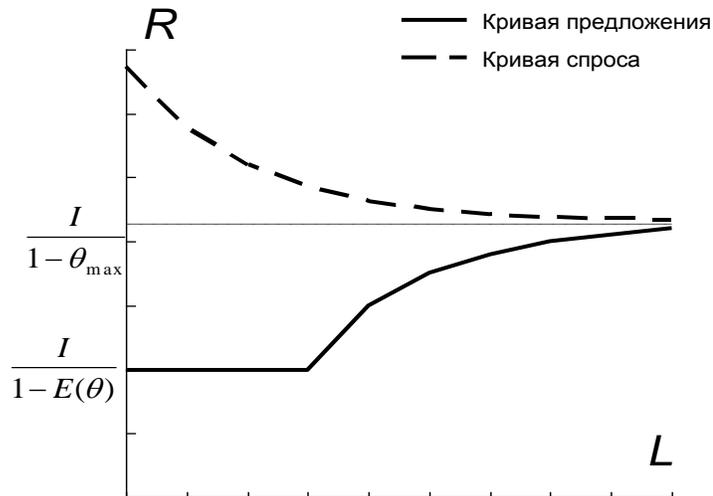


Рис. 1 Отсутствие равновесия при совершенной конкуренции

Рационирование кредита в условиях совершенной конкуренции

Рационирование кредита – ограничение банками предложения кредита.

Для описания равновесия с рационированием формализуем игру - построим игру по Байесу (определение согласно [5]). В нашей игре бесконечное количества кредиторов (все одного типа), и заемщиков (разного типа). Стратегией кредитора является выбор R и L_{\max} – максимального размера кредита, стратегией заемщика – выбор одного из предложенных R и размера кредита, не превышающего максимальный установленный кредитором. Очевидно, что в равновесии каждый заемщик максимизирует свою прибыль. Прибыль кредитора равна 0, так как существует совершенная конкуренция.

Утверждение 2.1

В равновесии все кредиторы предлагают контракт с одинаковым значением R и для всех типов заемщика:

$$\max_{L \leq L_{\max}} (U(L) - (1 - \theta)RL) = U(L_{\max}) - (1 - \theta)RL_{\max} \quad (10)$$

Доказательство. Пусть кредитор 1 предлагает контракт (L_{\max}, R_1) , кредитор 2 - (L_{\max}, R_2) , $R_1 < R_2$, при этом прибыль каждого равна 0. Для каждого из

кредиторов находится тип заемщика, предпочитающий его конкуренту.

Обозначим

$$D(\theta) = \max_{L \leq L_{\max}} (U(L) - (1 - \theta)R_1L) - \max_{L \leq L_{\max}} (U(L) - (1 - \theta)R_2L) \quad (11)$$

Очевидно, это непрерывная функция θ , следовательно, существует $\bar{\theta}$, при котором она равна нулю, то есть для такого заемщика безразличен выбор между предложениями кредиторов. Докажем, что $\max_{L \leq L_{\max}} (U(L) - (1 - \bar{\theta})R_1L)$ достигается при

$L = L_{\max}$. Если это не так, то

$$\begin{aligned} \max_{L \leq L_{\max}} (U(L) - (1 - \bar{\theta})R_1L) &= \max_L (U(L) - (1 - \bar{\theta})R_1L) > \\ &> \max_L (U(L) - (1 - \bar{\theta})R_2L) \geq \max_{L \leq L_{\max}} (U(L) - (1 - \bar{\theta})R_2L) \end{aligned} \quad (12)$$

То есть если кредитор уменьшит L_{\max} на ε , то

$$\begin{aligned} \max_{L \leq L_{\max} - \varepsilon} (U(L) - (1 - \bar{\theta})R_1L) &< U(L_{\max}) - (1 - \bar{\theta})R_1(L_{\max}) = \\ &= \max_{L \leq L_{\max}} (U(L) - (1 - \bar{\theta})R_1L) = \max_{L \leq L_{\max}} (U(L) - (1 - \bar{\theta})R_2L) \end{aligned} \quad (13)$$

Заемщики типа $\bar{\theta}$ перейдут к конкуренту (кредитор 2). Так как прибыль кредитора 1 от заемщика типа θ равна $(1 - \theta)RL - IL$ и суммарная прибыль равна 0, то прибыль от $\bar{\theta}$ (наиболее рискованного заемщика) отрицательна. Поменяв $L_{\max} \rightarrow L_{\max} - \varepsilon$ кредитор 1 увеличит прибыль, то есть при разных R не может быть равновесия. При одинаковых R кредитор путем уменьшения L_{\max} может отсеять рискованных заемщиков и увеличить прибыль (если $\max_{L \leq L_{\max}} (U(L) - (1 - \theta)RL) = U(L_{\max}) - (1 - \theta)RL_{\max}$). Чтобы сделать это невозможным, необходимо выполнение условия (10).

Будем искать равновесие так же, как в [3] – точку, в которой кривая безразличия (кривая, вдоль которой его прибыль постоянна) наименее рискованного заемщика касается кривой предложения.

Утверждение 2.2

Кривая безразличия заемщика с $\theta = 0$ касается кривой предложения в точке $(L_1, \lambda_0 I), U'(L_1) = I / \lambda_0$ (λ_0 определено в предыдущем пункте).

Доказательство. Так как $L_1 < L^*$, где L_1 решение $U'(L_1) = I / \lambda_0$, L^* решение $U(L^*) / L^* = I / \lambda_0$, кривая спроса заемщика $\theta = 0$ пересекает кривую предложения в точке $(L_1, \lambda_0 I), U'(L_1) = I / \lambda_0$. Эта точка лежит на участке кривой предложения $R = I / \lambda_0$. Как показано в [3] в точках кривой спроса заемщика $\theta = 0$, как и любого другого, кривая безразличия имеет нулевую производную, следовательно она касается прямой $R = I / \lambda_0$.

Утверждение 2.3

Точка $(L_1, I / \lambda_0), U'(L_1) = I / \lambda_0$ является равновесием.

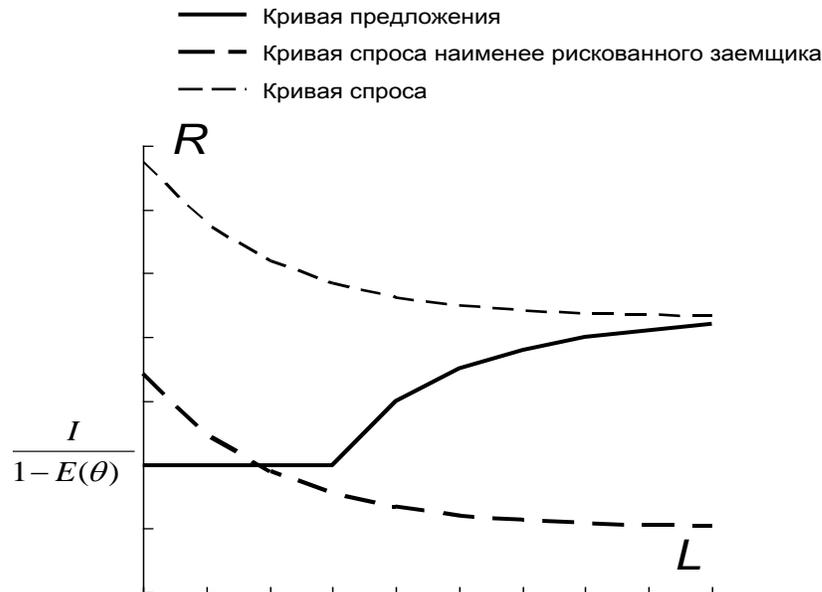


Рис.2. Рационарирование кредита в условиях совершенной конкуренции

Доказательство (нестрогое). Пусть это не так: существует контракт (L_{\max}, R_1) , предложив который (в то время как остальные предлагают $(L_1, I / \lambda_0)$) банк получает положительную прибыль. Можно изменить этот контракт так, что все заемщики будут выбирать размер контракта L_{\max} и прибыль банка останется положительной (доказательство эквивалентно доказательству утверждения 2.1).

Пусть θ_1 – минимальная степень риска заемщика, при котором он предпочитает контракт (L_{\max}, R_1) . Кривая безразличия этого заемщика, проходящая через $(L_1, I/\lambda_0)$:

$$R = \frac{U(L) - U(L_1) + (1 - \theta_1)IL_1 / \lambda_0}{(1 - \theta_1)L} \quad (14)$$

Заемщик примет контракт тогда и только тогда, когда предлагаемая точка лежит ниже кривой безразличия. Если это выполнено для некоторого θ^* , то это выполнено и для любого $\theta > \theta^*$ (кривая безразличия лежит ниже). Суммарная прибыль кредитора:

$$\begin{aligned} & \int_{\theta_1}^{\theta_{\max}} (L(1 - \theta) \frac{U(L) - U(L_1) + (1 - \theta_1)IL_1 / \lambda_0}{(1 - \theta_1)L} - IL) g(\theta) d\theta < \\ & < \int_{\theta_1}^{\theta_{\max}} ((1 - \theta) \frac{(L - L_1)I / \lambda_0}{(1 - \theta_1)} + (1 - \theta)IL_1 / \lambda_0 - IL) g(\theta) d\theta = \left| \frac{\int_{\theta_1}^{\theta_{\max}} \theta g(\theta) d\theta}{\int_{\theta_1}^{\theta_{\max}} g(\theta) d\theta} \right| \quad (15) \\ & = \frac{I}{\lambda_0} \int_{\theta_1}^{\theta_{\max}} \frac{(E(\theta) + \theta_1 - \bar{\theta} - E(\theta)\theta_1)L - \theta_1(1 - \bar{\theta})L_1}{(1 - \theta_1)} g(\theta) d\theta < 0 \end{aligned}$$

при достаточно малых L , так как

$$\frac{E(\theta) + \theta_1 - \bar{\theta} - E(\theta)\theta_1}{\theta_1(1 - \bar{\theta})} \leq \left| g'(\theta) \leq 0 \Rightarrow \bar{\theta} \leq \frac{\theta_{\max} + \theta_1}{2} \right| < 1/2 \quad (16)$$

Строгое доказательство равновесия требует дополнительных условий на функцию $U(L)$.

Таким образом, в системе возможно равновесие, в котором спрос больше предложения, причем это единственное равновесие.

Игра одного заемщика и одного кредитора

Равновесие по Нэшу в игре в стратегической форме

Найдем равновесие по Нэшу в условиях симметричной информации. Стратегией заемщика является выбор L , кредитора R . Затем игроки могут отказаться от контракта.

Утверждение 3.1

В построенной игре равновесие возможно только при $L=0$.

Доказательство. Исход является равновесным тогда и только тогда, когда заемщик максимизирует свою прибыль при фиксированном R , кредитор – при фиксированном L .

Кривая спроса определяется условием:

$$L_{\theta}(R) = \arg \max(U(L) - (1 - \theta)RL) \Leftrightarrow R = \frac{U'(L)}{(1 - \theta)} \quad (17)$$

Кривая предложения (отличная от построенной в предыдущем пункте):

$$R = \arg \max((1 - \theta)RL - IL) | U(L) - (1 - \theta)RL > 0 \Leftrightarrow R = \frac{U(L)}{L(1 - \theta)} \quad (18)$$

Кривые пересекаются только в точке $(0, U'(0))$ – существует только нулевое равновесие.

Равновесие Байеса-Нэша

В игре 2 игрока – заемщик и кредитор. Множество типов заемщика – отрезок $[0, \theta_{\max}]$. Кредитор может иметь только один тип. Выигрыш игрока – его ожидаемый доход. Представление кредитора о типе заемщика строится исходя из функции распределения $g(\theta)$.

Стратегией заемщика является выбор L , кредитора R . При этом после оглашения результатов оставим за игроками право отказаться от контракта. Исход является равновесным тогда и только тогда, когда заемщик максимизирует свою прибыль при фиксированном R , кредитор – при фиксированном L . Кривая спроса

совпадает с кривой для игры с симметричной информацией, кривая предложения отличается:

$$R = \arg \max(\lambda(R, L)RL - \mu(R, L)IL) \quad (19)$$

Для кредитора условие максимизации может выполняться либо при обращении в 0 производной выражения (19)

$$\frac{\partial(E(Q)/L)}{\partial R} = \lambda(R, L) - g(\theta^*)(U(L)/RL)^2 + IU(L)/(R^2L)g(\theta^*) \quad (20)$$

Где $\theta^* = \min(\max((1 - \frac{U(L)}{RL}), 0), \theta_{\max})$, либо в точке разрыва производной ($\theta^* = 1 - \frac{U(L)}{RL} = 0$).

Утверждение 4.1 (свойства кривой предложения)

1) Рассмотрим случай малых L. Кривая $\theta^* = 0$ задается уравнением

$R = \frac{U(L)}{L}$. Докажем, что при малых L точка $R = \frac{U(L)}{L}$ является точкой максимума.

$$\begin{aligned} \frac{\partial(E(Q)/L)}{\partial R} &\leq 1 - E(\theta) - \frac{1}{\theta_{\max}}(1 - I/R) = 1 - E(\theta) - \frac{1}{\theta_{\max}}(1 - \frac{IL}{U(L)}) \\ &\leq \left| \begin{matrix} \text{м.к.} \\ U'(0) > I/(1 - \theta_{\max}) \end{matrix} \right| \leq 1 - E(\theta) - \frac{1}{\theta_{\max}}(1 - (1 - \theta_{\max})) = -E(\theta) < 0 \end{aligned} \quad (21)$$

Следовательно при $R > \frac{U(L)}{L}$ функция (19) не может иметь максимума, при

этом, так как левая производная в точке $R = \frac{U(L)}{L}$ положительна и равна L, и

максимум достигается в точке $R = \frac{U(L)}{L}$

2) Рассмотрим случай больших L. Обозначим $\theta^*(L, R^*) \stackrel{def}{=} \theta^*(L)$, где R^* – решение (18)

Докажем, что $\theta^*(L) \xrightarrow{L \rightarrow \infty} \theta_{\max}$. Предположим, что это не так. Тогда

$$\liminf_{L \rightarrow \infty} \theta^*(L) = \theta_2 < \theta_{\max} \quad (22)$$

Существует такая последовательность $L_n \rightarrow \infty$, что $\theta(L_k) \rightarrow \theta_2$ и

$$E(Q)/L = \int_{\theta^*}^{\theta_{\max}} (R(1-\theta) - I)g(\theta)d\theta \xrightarrow{L \rightarrow \infty} \int_{\theta_2}^{\theta_{\max}} \left(\frac{I}{1-\theta_2}\right)(1-\theta) - I)g(\theta)d\theta \quad (23)$$

Получили противоречие, так как отрицательное значение не может быть максимумом. Мы доказали, что $\theta^*(L) \xrightarrow{L \rightarrow \infty} \theta_{\max}$

Утверждение 4.2

В игре с асимметричной информацией существует равновесие.

Доказательство. Мы доказали, что точка $(0, U'(0))$ лежит на кривой предложения, а точка $(0, U'(0)/(1-\theta))$ – на кривой спроса (доказано в предыдущем разделе). При $L \rightarrow \infty$ на кривой предложения:

$$R \geq \frac{U(L)}{L(1-\theta_{\max})} \quad (24)$$

А на кривой спроса

$$R = \frac{U'(L)}{(1-\theta)} < \frac{U(L)}{L(1-\theta_{\max})} \quad (25)$$

Следовательно, равновесие существует.

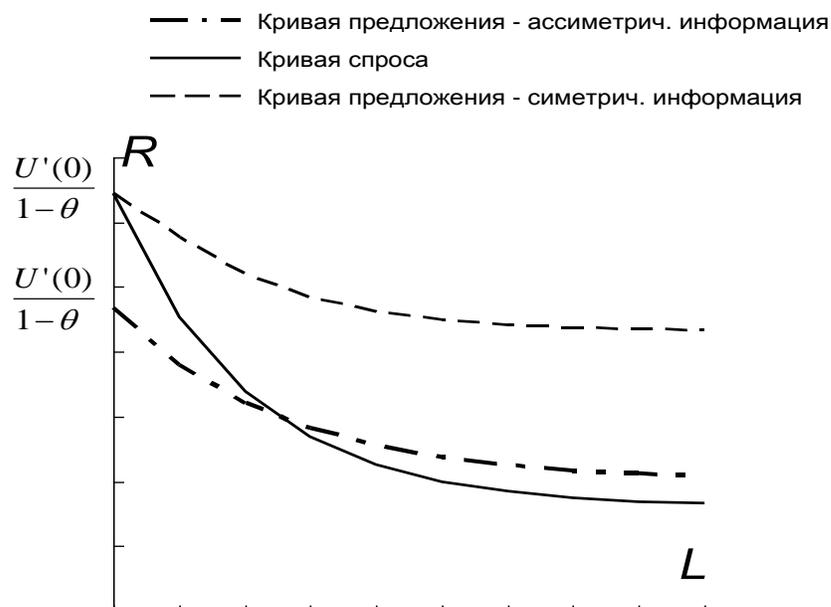


Рис.3. Равновесие в игре двух игроков

Заключение

Были рассмотрены 2 модели и получены следующие результаты:

1) В модели совершенной конкуренции без рационирования нет равновесия, однако в модели с возможностью рационирования есть равновесие:

$$R = \frac{I}{1 - E(\theta)}, U'(L) = R \quad (26)$$

2) В модели взаимодействия двух игроков есть равновесие в условиях асимметричной информации. В нем выполняется $R = \frac{U'(L)}{1 - \theta}$. В случае симметричной информации в равновесии размер кредита нулевой.

Библиографический список:

1. Stiglitz J. E., Weiss A. Credit Rationing in Markets with Imperfect Information, The American Economic Review, Volume 71, Issue 3(June,1981), 393-410
2. Besanko D. , Kanatas G. Credit Market Equilibrium with Bank Monitoring and Moral Hazard, The Review of Financial Studies 1993 Volume 6, number 1

3. Jaffe D. M. , Russel T. Imperfect information, Uncertainty, and Credit Rationing The quarterly Journal of Economics, Volume 90, Issue 4(Nov., 1976), 651-666.

4. Webb D. C. Imperfect information and credit market equilibrium , European Economic Review, Oct/Nov84, Vol. 26 Issue ½

5. Меньшиков И. С. Лекции по теории игр и экономическому моделированию. – М, М3 Пресс, 2006

6. Rothschild M. , Stiglitz J. Equilibrium in competitive insurance markets: an essay on the economics of imperfect information, Quarterly Journal of Economics. Nov76, Vol. 90 Issue 4, p629-649. 21p.

Информация об авторах

Data on authors

Макарова Мария Александровна – студент кафедры математического моделирования сложных процессов и систем МФТИ

Тел.: 8-915-183-24-86

E-mail: makarova.mar@gmail.com

Maria Makarova – student at the department of mathematical modeling of complex processes and systems MIPT

Tel.: 8-915-183-24-86

E-mail: makarova.mar@gmail.com

**ЛИНЕЙНАЯ ЗАВИСИМОСТЬ ПЛОТНОСТИ ПОЧВЫ И ВЫСОТЫ
РЕЛЬЕФА В ПОЛУПУСТЫНЕ СЕВЕРНОГО ПРИКАСПИЯ И
ПРОГНОЗИРОВАНИЕ ЕЁ СВОЙСТВ**

**LINEAR DEPENDENCE BETWEEN SOIL DENSITY AND A RELIEF
HEIGHT IN SEMI-DESERT OF NORTH CASPIAN AND
FORECASTING IT PROPERTIES**

По исходной выборке из 12 точек построена адекватная математическая модель зависимости плотности почвы и высоты микрорельефа в исследуемом районе. Модель состоит из трёх уравнений линейной регрессии, полученных отдельно по каждому классу (микровозвышений, западин, микросклонов). Алгоритм прогнозирования свойств почвы, построенный на принципах кластер-анализа, проиллюстрирован двумя примерами.

An adequate mathematical model of the dependence between a relief height and a soil density in the investigated region have been constructed on the initial sample of 12 points. The model consists in three linear regression equations, which were received in each cluster separately (rises, micro-rise-sides, pits). The forecasting soil property algorithm has been constructed by the cluster analysis principles and it has been illustrated by two examples.

Ключевые слова: высота рельефа, плотность почвы, уравнение регрессии, кластер-анализ, прогнозирование.

Keywords: relief height, soil density, regression equation, cluster analysis, forecasting.

1. География района

«Всякая почва есть продукт совокупной деятельности материнских горных пород, климата, растительности и рельефа местности».

В. В. Докучаев

Докучаев, основатель науки почвоведения, впервые сформулировал определение почвы [1]. Он предложил понятие о почве, «как о вполне самостоятельном естественно-историческом теле, которое является продуктом совокупной деятельности: грунта, климата, растений и животных, возраста страны и отчасти рельефа местности». Это положение легло в основу и географии почв. Стало очевидным, что распространение почв на Земле имеет вполне закономерный характер, зависящий от различного сочетания факторов почвообразования [1].

Почвы полупустыни Прикаспийской низменности издавна вызывали большой интерес географов-почвоведов. Особенно детально исследовались почвы Джаныбекского заповедника, расположенного в северо-западной части Прикаспийской низменности. Большую часть его территории занимает межпадинная равнина с хорошо развитым микрорельефом. В её поверхность вкраплены относительно крупные замкнутые понижения – большие падины и лиманы, а также мелкие замкнутые углубления – западины. С основной поверхностью западины соединены пологими микросклонами [2].

В исследуемом районе географы-почвоведы выделяют три вида микрорельефа: микровозвышения, западины и микросклоны, из которых первые образуют сплошную сеть и занимают свыше 50% всей площади. Колебания высот этих форм микрорельефа не превышают 30–40 см,

изредка достигая 50 см. Ландшафт Джаныбекского заповедника характерен для засушливых степей юга России и Украины.

Такой микрорельеф и вызываемое им перераспределение поверхностных вод являются причинами резко выраженной комплексности почвенно-растительного покрова и грунтовых вод, подробно описанной в [2]. В западинах со злаково-разнотравной растительностью развиты более плотные тёмноцветные почвы, а на микровозвышениях между западинами – менее плотные солончаковые солонцы с разреженным покровом чёрной полыни и прутняка. К тёмноцветным почвам относятся тёмнокаштановые и чернозёмовидные почвы. Третий, промежуточный тип – светлокаштановые почвы встречаются на микросклонах и реже в мелких западинах, их площадь составляет около 25% площади межпадинной равнины.

Ход процесса почвообразования в исследуемом районе описан в [2]. В первоначальный период, после выхода низменности из-под Каспийского моря, почвообразовательный процесс шёл по пути солончакообразования. На засоленных породах морских отложений образовались солончаки, богатые хлоридами и сульфатами. Второй период почвообразования характеризовался деятельностью существовавшей гидрогеографической сети, дренировавшей эту территорию. В этот период процесс изменялся в сторону рассоления. При затухании деятельности гидрогеографической сети замедлялся и процесс рассоления, он стал чередоваться с периодическим засолением. Процесс засоления проходил неравномерно по причине различной отдалённости от дренирующих потоков и существования первоначального микрорельефа.

Развитие почвообразовательного процесса и эволюция микрорельефа – два взаимно связанных процесса. Рассоление почв, вызываемое действиями воды, усиливает развитие микропонижений, что усиливает процесс выщелачивания и рассоления; процесс развития

западин идёт не только вглубь, но и вширь, постепенно захватывая склоны микроповышений.

2. Постановка задачи и методы её решения

Задача состоит в построении адекватной математической модели, описывающей связь между видом почвы и формой микрорельефа в исследуемом районе. Эта задача была рассмотрена в [3], в которой по малой выборке из 8 точек, $\Omega = \{ 1, 2, \dots, 8 \}$, взятых на различных формах микрорельефа, искомая связь даётся уравнением линейной регрессии,

$$\Omega: y = -126.83x + 202.13, \quad (1)$$

y – высота микрорельефа, x – средняя плотность почвенной толщи глубины 200 см.

Анализ профилей трёх основных видов почвы [2, 3], различающихся по химическим, физическим и механическим свойствам, привёл к заключению, что исходная выборка Ω не является однородной, а состоит из хорошо различимых классов: ω_1 – солончаковые солонцы на микровозвышениях, ω_2 – тёмноцветные почвы на западинах, ω_3 – светло-каштановые почвы на микросклонах и небольших западинах. Поэтому зависимость высоты микрорельефа и плотности почвы корректнее искать в каждом классе отдельно. В работе [4] искомая зависимость получена лишь по двум классам ввиду недостаточного объёма выборки Ω :

$$\omega_1: y = -204.67x + 304.48, \quad \omega_2^*: y = -745.00x + 1146.48, \quad (2)$$

класс ω_1 состоит из четырёх точек, класс ω_2^* включает две точки на западинах и одну точку на микросклоне. Уравнения в (2) точнее определяют искомую зависимость, чем уравнение (1): средние абсолютные ошибки высот, вычисленных по уравнениям (1), (2) равны соответственно $|\Delta y| = 3.3$, $|\Delta y_1| = 0.7$, $|\Delta y_2| = 1.1$, их средне-квадратичные отклонения равны: $\tilde{\sigma} = 3.8$, $\tilde{\sigma}_1 = 0.8$, $\tilde{\sigma}_2 = 1.2$.

3. Уточнённая математическая модель

Уточнение модели искомой зависимости состояло в том, что уравнения линейной регрессии строились отдельно по классам ω_1 , ω_2 , ω_3 при бóльшем объёме исходной выборки $\Omega_1 = \{ 1, 2, \dots, 12 \}$.

Данные этих наблюдений, взятые из работ [2, 3], представлены в табл. 1. Знаком (*) помечены номера точек, значения высот y которых проинтерполированы.

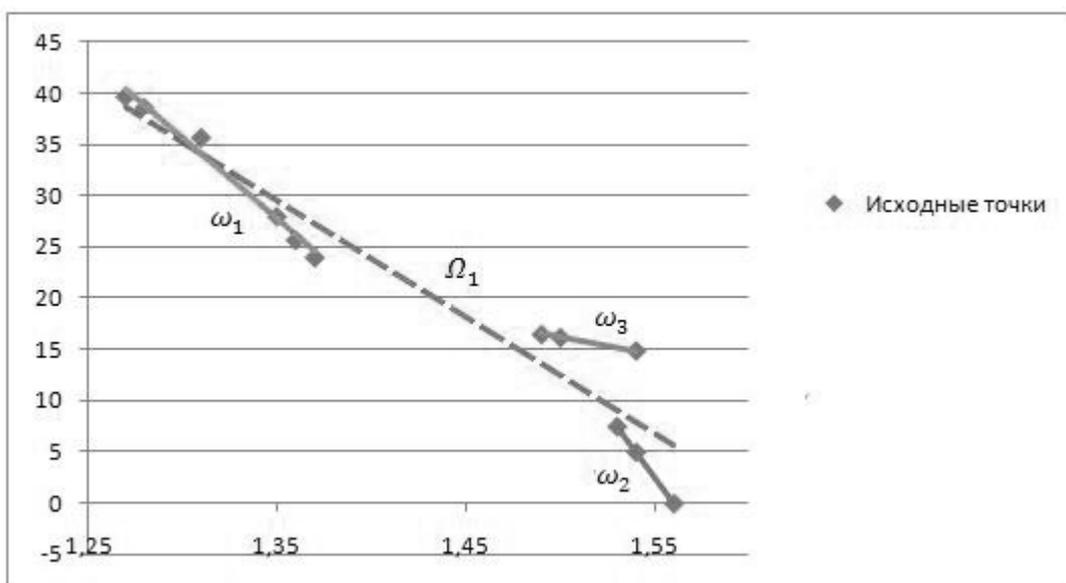


Рис. 1. Исходные точки и графики функций линейной регрессии, полученных по выборкам Ω_1 , ω_1 , ω_2 , ω_3 методом наименьших квадратов [5]

Уравнения линейной регрессии имеют вид:

$$\Omega_1: \bar{y}(x) = -113.9941x + 183.4250, \quad (3)$$

$$\omega_2: \tilde{y}(x) = -159.5255x + 243.0555, \quad (4)$$

$$\omega_2: \hat{y}(x) = -250.0000x + 390.0000, \quad (5)$$

$$\omega_3: y^*(x) = -32.1429x + 64.4024. \quad (6)$$

В табл. 1 помещены значения высот микрорельефа, вычисленные по уравнениям (3)-(6), а в табл. 2 – значения статистических параметров, характеризующих точность их вычислений.

Анализ данных таблиц 1, 2 позволяет утверждать, что уравнения линейной регрессии, полученные по каждому из трёх классов, существенно точнее определяют зависимость высоты микрорельефа и средней плотности почвы, чем аналогичное уравнение, построенное по объединённой выборке. Наиболее надёжным уравнением является уравнение (4) для класса микровозвышений ω_1 , полученное по шести точкам.

Таблица 1

№ точк и	Классы ω_i $i = 1, 2, 3$	Средняя плотность почвы в г/см ³	Фактичес кая высота в см	Высоты, вычисленные по уравнениям в см			
				(3), Ω_1	(4), ω_1	(5), ω_2	(6), ω_3
1	ω_1 – микровозвы шения на солончаков ых солонцах	1.27	39.7	38.6	40.5		
2*		1.28	38.6	37.5	38.9		
3		<u>1.31</u>	<u>35.7</u>	34.1	34.1		
4		1.35	28.0	29.5	27.7		
5		1.36	25.7	28.4	26.1		
6*		1.37	24.0	27.2	24.5		
7*	ω_2 – западины на тёмноцветн ых почвах	1.53	7.5	9.0		7.5	
8		<u>1.54</u>	<u>5.0</u>	7.9		5.0	
9		1.56	0.0	5.6		0.0	
10	ω_3 – микросклон ы со светло- каштановы ми почвами	1.49	16.5	13.6			16.51
11*		<u>1.50</u>	<u>16.2</u>	12.4			16.19
12		1.54	14.9	7.9			14.90

Используя неравенство Бьенэмэ-Чебышева [5],

$$P\{|y(x) - \tilde{y}(x)| < t\sigma\} > 1 - \frac{1}{t^2}, \quad t > 0, \quad (7)$$

и данные табл. 2, имеем доверительный коридор для высот класса ω_1 при $t = 4$, $\tilde{\sigma} = 0.87$,

$$P\{\tilde{y}(x) - 3.48 < y(x) < \tilde{y}(x) + 3.48\} > 0.9375.$$

Таблица 2

Выборки	Оценки точности вычислений			Коэффициент корреляции $\rho(x, y)$
	Средняя абс. ошибка $ \Delta y $	Максим. абс. ошибка $ \Delta y _{\max}$	Средне-квартичное отклонение $\tilde{\sigma}$	
Ω_1	2.91	7.03	3.55	-0.9630
ω_1	0.64	1.62	0.87	-0.9920
ω_2	$0.942 \cdot 10^{-10}$	$0.981 \cdot 10^{-10}$	$0.965 \cdot 10^{-10}$	-1.0000
ω_3	0.008	0.012	0.011	-0.9999
$\omega_2 + \omega_3$	2.75	7.70	3.95	-0.8178

Для уточнения уравнений регрессии в классах ω_2 , ω_3 необходимо увеличить число наблюдений в каждом из них хотя бы до 5. К сожалению, не было такой возможности.

4. Статистический анализ исходных данных

Проведённый статистический анализ исходных данных не противоречит географической классификации почвенно-растительных комплексов исследуемого района.

Для плотностей почв были проверены гипотезы о равенстве дисперсий классов по критерию Фишера [6] и равенстве их средних

значений. Использование F-критерия основано на том, что средние значения плотностей почв во всех точках распределены асимптотически нормально в силу центральной предельной теоремы [5]. В табл. 3 даны значения статистик, используемых в статистических выводах.

Таблица 3

Классы	Число точек n_i	Плотность почвы		
		Среднее значение $\tilde{\mu}_i$	Дисперсии $\tilde{\sigma}_i^2$	$\tilde{\sigma}_i^2 / n_i$
ω_1	6	1.32	0.00183	0.00030
ω_2	3	1.54	0.00025	0.00008
ω_3	3	1.51	0.00070	0.00023

Таблица 4

Гипотезы	\tilde{F}_{ij}	Число степеней свободы	$F_{кр}$	Статистический вывод
$H_{12}: \sigma_1^2 = \sigma_2^2$	7.308	6 3	27.911	Гипотеза H_{12} принимается
$H_{13}: \sigma_1^2 = \sigma_3^2$	2.61	6 3	27.911	Гипотеза H_{13} принимается
$H_{32}: \sigma_3^2 = \sigma_2^2$	2.80	3 3	29.457	Гипотеза H_{32} принимается

По данным табл. 3 были вычислены статистики $\tilde{F}_{12} = \tilde{\sigma}_1^2 / \tilde{\sigma}_2^2$, $\tilde{F}_{13} = \tilde{\sigma}_1^2 / \tilde{\sigma}_3^2$, $\tilde{F}_{32} = \tilde{\sigma}_3^2 / \tilde{\sigma}_2^2$. По таблице распределения F-критерия для уровня значимости $\alpha = 0.01$ нашли его критические значения $F_{кр}$ при соответствующих степенях свободы n_i, n_j [6]. Информация о проверке гипотез H_{ij} , $i \neq j$, $i, j = 1, 2, 3$, помещенная в табл. 4, показывает, что

различие дисперсий плотностей почв в классах статистически незначимо. Поэтому была проведена проверка гипотез о равенстве средних значений плотностей почв в классах с использованием выражения общего средне-квадратичного отклонения $\tilde{\sigma}_{ij}$ классов ω_i и ω_j , $i \neq j$, $i, j = 1, 2, 3$, [5],

$$\tilde{\sigma}_{ij} = \sqrt{\tilde{\sigma}_i^2/n_i + \tilde{\sigma}_j^2/n_j}, \quad i \neq j, \quad i, j = 1, 2, 3.$$

Если верна гипотеза $H'_{ij} : \mu_i = \mu_j$, то статистика $|\tilde{\mu}_i - \tilde{\mu}_j| \tilde{\sigma}_{ij}^{-1}$, $i \neq j$, $i, j = 1, 2, 3$, распределена асимптотически нормально со средним значением 0 и дисперсией 1 [5]. Гипотеза H'_{ij} отвергается на выбранном уровне значимости α , если $|\tilde{\mu}_i - \tilde{\mu}_j| > \lambda_\alpha \tilde{\sigma}_{ij}$, λ_α — α -процентное значение нормального отклонения. Гипотеза H'_{ij} принимается, если $|\tilde{\mu}_i - \tilde{\mu}_j| \leq \lambda_\alpha \tilde{\sigma}_{ij}$.

Из таблицы нормального распределения в [5] имеем: при $\alpha = 0.01$, $\lambda_\alpha = 2.5758$, при $\alpha = 0.05$, $\lambda_\alpha = 1.96$. Информация о проверке гипотез H'_{ij} , $i \neq j$, $i, j = 1, 2, 3$, при $\alpha = 0.01$ помещена в табл. 5, из которой следует, что при двустороннем 1%-м уровне значимости разность $|\tilde{\mu}_i - \tilde{\mu}_j|$ статистически значима для классов ω_1 и ω_2 , ω_1 и ω_3 , но статистически незначима для классов ω_2 и ω_3 .

Таблица 5

Гипотезы	$ \tilde{\mu}_i - \tilde{\mu}_j $	$\tilde{\sigma}_{ij}$	$\lambda_{0.01} \tilde{\sigma}_{ij}$	Статистический вывод
$H'_{12} : \mu_1 = \mu_2$	0.22	0.0194	0.05	Гипотеза H'_{12} отвергается
$H'_{13} : \mu_1 = \mu_3$	0.19	0.0230	0.06	Гипотеза H'_{13} отвергается
$H'_{23} : \mu_2 = \mu_3$	0.03	0.0176	0.04	Гипотеза H'_{23} принимается

Для более точного статистического вывода относительно различия средних в классах ω_2 и ω_3 следует значительно увеличить объёмы выборок. Чтобы убедиться, что классы ω_2 и ω_3 существенно различаются, было построено уравнение линейной регрессии по их объединённой выборке,

$$\omega_2 + \omega_3 : \bar{y} = -211.3208x + 332.6330. \quad (8)$$

Значения параметров, характеризующих точность вычисления высот по уравнению (8), помещены в последнюю строку табл. 2, из данных которой следует, что модель, описываемая уравнением (8), по точности намного хуже модели, описываемой уравнениями (5) и (6).

Из трёх рассмотренных математических моделей, описывающих зависимость высоты микрорельефа и средней плотности почвы в исследуемом районе, наиболее точной является модель в виде уравнений линейной регрессии, полученных отдельно по каждому из трёх выделенных классов.

5. Прогнозирование свойств почвы

Используя уравнение регрессии класса ω_{i_0} , $i_0 \in \{ 1, 2, 3 \}$ для любой новой точки $z = (x, y)$ этого класса можно получить оценку \tilde{x} её средней плотности почвы по известной высоте или получить оценку её высоты \tilde{y} по известной средней плотности почвы x . Оценивание неизвестной средней плотности почвы является приоритетной задачей, так как определение этой плотности намного сложнее измерения высоты.

Задача прогнозирования свойств почвы в конкретной точке $z_0 = (x, y_0)$, высота которой известна, включает в себя не только определение оценки её неизвестной средней плотности \tilde{x} , но и получение информации о плотности почвы каждого слоя высоты 5 см или 10 см по всей 2-х метровой толще и о её физико-химических свойствах.

Для решения поставленной задачи уравнения (4)-(6) целесообразно записать в виде:

$$\omega_1: \tilde{x} = -0.0063 y + 1.5236, \quad (9)$$

$$\omega_2: \tilde{x} = -0.0040 y + 1.5600, \quad (10)$$

$$\omega_3: \tilde{x} = -0.0311 y + 2.0036. \quad (11)$$

Кроме того, нам понадобятся такие характеристики классов, как: эталоны, типичные представители, средне-квадратичные отклонения в уравнениях регрессии (9)-(11).

Эталон класса ω_i — это вектор средних значений его признаков,

$$\bar{\mu}_i = (\bar{x}_i, \bar{y}_i), \quad \bar{x}_i = \frac{1}{n_i} \sum_{x_s \in \omega_i} x_s, \quad \bar{y}_i = \frac{1}{n_i} \sum_{y_s \in \omega_i} y_s, \quad i = 1, 2, 3,$$

n_i — число точек обучающей выборки класса ω_i .

Типичный представитель (ТП) класса ω_i — это ближайшая соседняя точка $z_{i0} = (x_{i0}, y_{i0})$ для эталона $\bar{\mu}_i$,

$$z_{i0} = \arg \min_{z_s \in \omega_i} r(\bar{\mu}_i, z_s), \quad i = 1, 2, 3,$$

$r(\bar{\mu}_i, z_s)$ — расстояние Евклида между точками $\bar{\mu}_i, z_s, z_s \in \omega_i$.

$$r(\bar{\mu}_i, z_s) = \sqrt{ (\bar{x}_i - x_s)^2 + (\bar{y}_i - y_s)^2 }. \quad (12)$$

Значения статистических характеристик классов помещены в табл. 6.

Таблица 6

Классы	Эталоны		Номер типичного представителя в табл. 1	Границы множеств значений y [α_i, β_i]	Средне-квадратичное отклонение $\tilde{\sigma}_x$
	\bar{x}_i	\bar{y}_i			
ω_1	1.32	31.95	3	[24, 39.7]	0.0051
ω_2	1.54	4.17	8	[0, 7.5]	$0.386 \cdot 10^{-12}$
ω_3	1.51	15.87	11	[14.9, 16.5]	$0.342 \cdot 10^{-3}$

Значения признаков (x, y) типичных представителей классов в табл. 1 подчеркнуты. По данным табл. 6 множества значений признака y (высоты) классов $[\alpha_i, \beta_i]$, $i=1, 2, 3$, не пересекаются. Следовательно, признак y является информативным. Признак называется информативным, если он содержит информацию о наличии классов.

Алгоритм прогнозирования свойств почвы в точке $z_0 = (x, y_0)$ с известной высотой y_0 состоит из 3-х частей:

1. Отнесение точки $z_0 = (x, y_0)$ к одному из имеющихся классов ω_i , $i=1, 2, 3$. Эту классификацию можно провести географически или математически. Во втором случае достаточно определить, в какое из множеств $[\alpha_i, \beta_i]$, $i=1, 2, 3$, попадает значение её высоты y_0 . Если $y_0 \in [\alpha_{i_0}, \beta_{i_0}]$, $i_0 \in \{1, 2, 3\}$, то точка $z_0 \in \omega_{i_0}$.

2. Вычисляется оценка \tilde{x} средней плотности почвы из уравнения регрессии класса ω_{i_0} , $i_0 \in \{1, 2, 3\}$, (одного из уравнений (9)-(11)).

3. Для точки $z_0 = (\tilde{x}, y_0)$ находится ближайшая соседняя точка $z_c \in \omega_{i_0}$ по правилу:

$$z_c = \arg \min_{z_s \in \omega_{i_0}} r(z_0, z_s), \quad (13)$$

$r(z_0, z_s)$ — расстояние Евклида в (12).

Полагаем, что свойства почвы в точке z_0 близки к свойствам почвы в точке z_c . Прогноз почвы в точке z_0 будет тем точнее, чем больше элементов в каждой обучающей выборке.

6. Результаты экспериментов

За испытуемую точку z_0 была взята точка № 4 из табл. 1, $z_0 = (x, 28)$. Так как $28 \in [24, 39.7]$ (табл. 6), то $z_0 \in \omega_1$. По обучающей

выборке класса ω_1 , состоящей из 5-и точек, построили уравнение регрессии

$$\tilde{x} = -0.0062y + 1.5217. \quad (14)$$

Из этого уравнения нашли оценку средней плотности почвы в точке z_0 , $\tilde{x} = 1.35$. Так как средне-квадратичное отклонение регрессии в (14) $\tilde{\sigma}_x = 0.005$, то на основании неравенства Бьенеме-Чебышева (7) при $t = 4$ имеем доверительный интервал для неизвестного значения x в точке z_0 ,

$$P\{ 1.33 < x < 1.37 \} > 0.9375.$$

Для точки $z_0 = (1.35, 28)$ определяем ближайшую соседнюю точку z_c , используя формулы (12), (13) и данные табл. 1,

$$\min_{z_s \in \omega_1} r(z_0, z_s) = r(z_0, z_5) = 2.3,$$

$$z_5 = (1.36, 25.7), \quad z_c = (1.36, 25.7).$$

У нас нет данных о физико-химических свойствах почвы в точках $z_0 = z_4$ и $z_c = z_5$. Но для этих точек в [3] имеются значения плотности каждого почвенного слоя высоты 5 см, Δx_i , $i = 1, 2, \dots, 40$ по всей 2-х метровой толще.

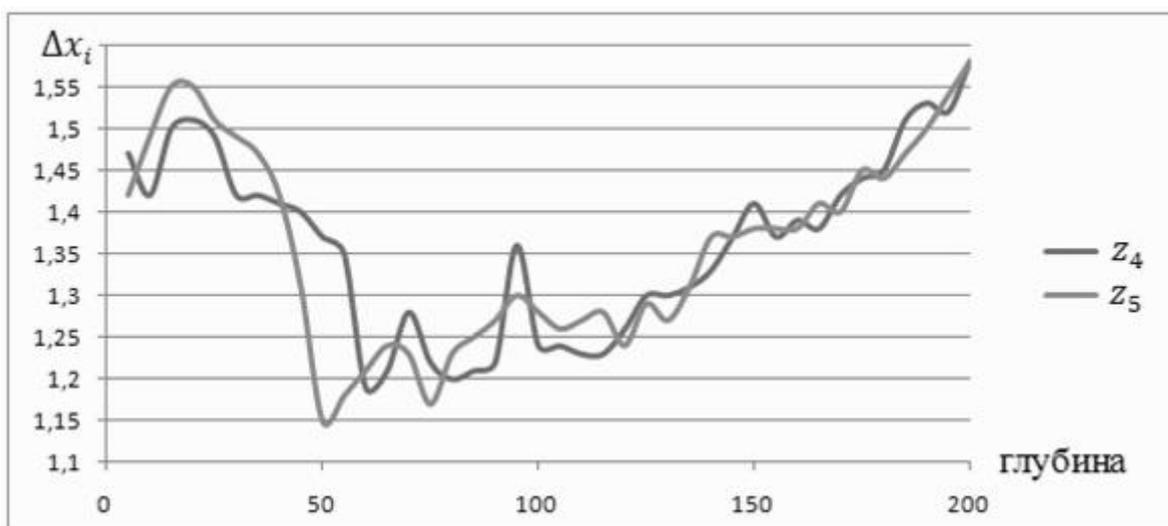


Рис. 2. Графики значений Δx_{4i} , Δx_{5i} , $i = 1, 2, \dots, 40$, на различной глубине их залегания

Второй эксперимент проводился аналогично. За испытуемую точку z_0 были взята точка № 2 из табл. 1, её высота $y_0=38.6$ см, $z_0=(x, 38.6)$. Так как $38.6 \in [24, 39.7]$ (табл. 6), то $z_0 \in \omega_1$. По обучающей выборке класса ω_1 , состоящей из 5-и точек, построили уравнение регрессии

$$\tilde{x} = -0.0061y + 1.5178. \quad (15)$$

Из этого уравнения нашли оценку средней плотности почвы в точке z_0 , $\tilde{x} = 1.28$. Так как средне-квадратичное отклонение регрессии в (15) $\tilde{\sigma}_x = 0.005$, то на основании неравенства Бьенеме-Чебышева в (7) при $t = 4$ имеем доверительный интервал для неизвестного значения x в точке z_0 ,

$$P\{ 1.26 < x < 1.30 \} > 0.9375.$$

Для точки $z_0=(1.28, 38.6)$ определяем ближайшую соседнюю точку z_c , используя формулы (12), (13) и данные табл. 1,

$$\min_{z_s \in \omega_1} r(z_0, z_s) = r(z_0, z_1) = 1.1,$$

$$z_1 = (1.27, 39.7), \quad z_c = (1.27, 39.7).$$

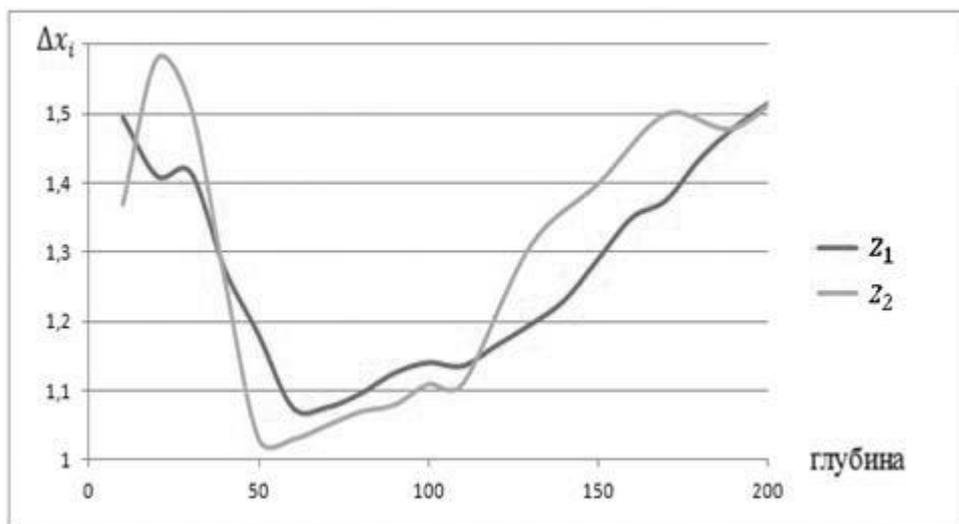


Рис. 3. Графики значений плотности каждого почвенного слоя высоты

На рис. 3 для точек z_1 и z_2 по данным из [2, 3] построены графики значений плотности каждого почвенного слоя высоты 10 см Δx_{1i} , Δx_{2i} , $i=1, 2, \dots, 20$, на различной глубине их залегания.

Полагаем, что физико-химические свойства почвы в точке z_2 наиболее близки к таким же свойствам почвы в точке z_1 . Наблюдаемое значительное различие значений Δx_{1i} и Δx_{2i} , например, на глубине 125 см – 180 см, возможно обусловлено большим временным промежутком в обработке этих скважин: скважина № 2 была пробурена в 50-е годы XX века, а скважина № 1 – в начале XXI века. Прогноз свойств почвы в исследуемой точке будет тем точнее, чем больше объём выборки каждого класса.

Заключение

По исходной выборке из 12 точек построена адекватная математическая модель зависимости плотности почвы и высоты микрорельефа в исследуемом районе. Модель состоит из трёх уравнений линейной регрессии, полученных отдельно по каждому классу (микровозвышениям, западинам, микросклонам). Алгоритм прогнозирования свойств почвы, построенный на принципах кластер-анализа, проиллюстрирован двумя примерами.

Выражаю глубокую благодарность Апраушевой Н. Н. за консультации и конструктивные замечания.

Библиографический список:

1. Добровольский Г. В. География почв: учебник. – М.: Изд-во Моск. ун-та: Наука, 2006. – 460 с.
2. Роде А. А., Польский М. Н. Почвы Джаныбекского стационара, их морфологическое строение, механический и химический состав и

физические свойства. // Почвы полупустыни северо-западного Прикаспия и их мелиорация. Тр. Почв. ин-та им. В.В.Докучаева. – М.: Изд-во АН СССР. – 1961. – Т. 56. – С. 3-214.

3. Абатуров Б. Д. Плотность почвы как фактор формирования микрорельефа в полупустыне Северного Прикаспия. // Почвоведение. – 2007. – №7. – С. 831-937.

4. Апрашева Н. Н., Сорокин С. В. О функциональной зависимости микрорельефа и плотности почвы в полупустыне северного Прикаспия. // Динамика неоднородных систем. Тр. Института системного анализа РАН. – 2010. – Т. 50(1).

5. Крамер Г. Математическая статистика. – М.: Мир, 1975.

6. Большев Л. Н., Смирнов Н. В. Таблицы математической статистики. – М.: Наука. Главная редакция физико-математической литературы, 1983.

Информация об авторах

Data on authors

Сорокин Станислав Викторович – ведущий программист отдела вычислительных систем и научной информации «ВЦ РАН».

Тел.: 8-499-135-14-98

E-mail www2011@ccas.ru

Sorokin Stanislav Viktorovich – leader programmer on Department of Computational Systems and Scientific Information «CC RAS ».

Тел.: 8-499-135-14-98

E-mail www2011@ccas.ru

**НОВЫЕ ПРИМЕНЕНИЯ АСИМПТОТИЧЕСКИХ МЕТОДОВ В
КРАЕВЫХ ЗАДАЧАХ МАТЕМАТИЧЕСКОЙ ФИЗИКИ**

**NEW APPROACH OF ASYMPTOTICAL METHODS FOR
BOUNDARY PROBLEMS OF MATHEMATICAL PHYSICS**

В данной работе описываются новые направления применения асимптотических методов в задачах математической физики и механики. В первую очередь это относится к приложениям теории возмущений к решению нелинейных сингулярно возмущенных задач в локальной области. Обсуждается применение в асимптотическом анализе метода Паде-аппроксимант для перестройки асимптотического разложения в дробно-рациональную или квазидробную функцию.

In this works, we present in some detail new trends in application of asymptotic to mathematical physics and mechanics problems. First we consider the various methods which give a possibility to extend a space of application of perturbation series and hence to omit local character.

In this work we also are going to emphasize the role of Pade-approximants (PA), which seems to play a more important role in today's non-linear mechanics and mathematical physics.

Ключевые слова: математическое моделирование, асимптотические методы, Паде-аппроксиманты, пограничный слой.

Keywords: mathematical modeling, asymptotic methods, Pade-approximants, boundary layer.

1. Методическое обоснование

В предметном пространстве асимптотические методы используются как инструмент для решения задач и находятся в подчиненном положении. В пространстве методов они самоценны и определяют целостную систему математического моделирования в предельных случаях. Складываясь в недрах механики, математической физики, где встречается множество процессов, имеющих *асимптотический* характер, они, с одной стороны, обогатились широким спектром приложений, приемов, с другой – избежали строгостей дефиниций, свойственных парадигме классической математики. Несмотря на строгое определение асимптотического ряда (разложения), данное А. Пуанкаре, роль и место асимптотических методов в классической математике оставались маргинальными отчасти потому, что арсенал этих методов велик и неформален и, как следствие, несовместим с принятыми методами классического математического анализа. Крупнейшие математики современности определяют специфику асимптотических методов в несвойственных терминах: «*асимптотология, асимптотический анализ – это искусство обращения с прикладными математическими системами в предельных случаях, ...искусство в том, чтобы знать, когда можно быть небрежным, а когда требуется точность*» [3, 9, 18]. Следуя Р. Г. Баранцеву, определим место асимптотических методов в методологическом пространстве методов математического моделирования. Исходя из критерия адекватности реальным объектам и моделируемым процессам, примем триадическую структуру дефиниции, содержащую аналитический (рациональный), качественный (эвристический) и субстанциональный (интуитивный) аспекты [3]. Отличие асимптотических методов от методов классической математики состоит в том, что уровень точности в них конкурирует с размерами области действия. Рассмотрим асимптотическое разложение функции $f(x)$ по последовательности функций $\varphi_n(x)$:

$$f(x) \sim \sum_{n=0}^N a_n \varphi_n(x), \quad N = 0, 1, 2, \dots \quad (1.1)$$

При $x \rightarrow 0$ величина

$$\Delta = \left| f(x) - \sum a_n \varphi_n(x) \right| \quad (1.2)$$

характеризует *точность*, x -*локальность*, N -*простоту*, образуя системную триаду. Каждая пара из этих параметров находится в отношении дополнительности, а третий – задает меру совмещения. В классической математике x фиксировано, $N \rightarrow \infty$, и исследуется *сходимость* Δ . В асимптотической математике N фиксировано, $x \rightarrow 0$, и говорится об эффективности *приближения* Δ . Совмещая в себе простоту эвристических представлений с точностью аналитических оценок, асимптотические методы не вырождаются в «золотую середину», а занимают особое место, замыкая системную триаду.

Принципиальной особенностью асимптотических методов является *локальность* получаемых с их помощью решений. При этом в сложных задачах зависимость от малого параметра редко бывает линейной. Источники нелинейности, как правило, локализованы. Места концентрации сильных нелинейных эффектов усиленно изучаются в теории разрушения, гидродинамике обтекаемых тел, гиперзвуковой аэродинамике и других областях прикладной математики. Неравномерность асимптотических разложений в окрестности таких мест значительно затрудняет оценку и анализ локально-нелинейных асимптотик.

Использование асимптотического ряда в качестве приближения в окрестностях особенностей всегда приводит к необходимости определения числа членов разложения, адекватно приближающих к искомому решению. Отсюда недостаток асимптотических методов или методов возмущений

определяется локальным свойством основанных на них решений. Поскольку методы асимптотического суммирования широко распространены, проблемы устранения локальности, продолжения локальных асимптотик крайне актуальны.

Существует много подходов к этим задачам [8]. Метод аналитического продолжения (например, преобразование Эйлера $\bar{\varepsilon} = \varepsilon(1 + \varepsilon)^{-1}$, или растяжения координат в локальной области) требует знания области единственности разложения искомой функции малого параметра ε . Эти методы полезно применять, когда известно большое количество членов ряда разложения. Тогда становится возможным выполнить, зная области единственности разложения, аналитическое продолжение, пользуясь, например диаграммой Домба-Сайкса [8, 13]. Для применения методов обобщенного суммирования [8] необходимо также знать значительное количество членов ряда разложения. На практике в этих рядах обычно известны 3-5 компонентов и именно из этого сегмента ряда приходится извлекать имеющуюся информацию.

Для этой цели может быть очень полезен метод Паде-аппроксимант [1, 2, 12-17]. Пусть задано разложение функции $F(\varepsilon)$ степенным рядом по малому параметру ε

$$F(\varepsilon) = \sum_{i=0}^{\infty} c_i \varepsilon^i \quad (1.3)$$

тогда дробно-рациональная функция

$$F_{mm}(\varepsilon) = \frac{\sum_{i=0}^m a_i \varepsilon^i}{\sum_{i=0}^n b_i \varepsilon^i} \quad (1.4)$$

называется Паде-аппроксимантой (РА), где коэффициенты a_i, b_i определяются из следующего условия: первые компоненты $(m+n)$ разложения дробно-рациональной функции (1.4) в ряд Маклорена совпадают с первыми компонентами $(m+n+1)$ ряда (1.3).

Таким образом, Паде-аппроксиманта выполняет мероморфное продолжение функции, заданной в виде степенного ряда, и по этой причине позволяет достичь успеха в случаях, когда аналитическое продолжение применить нельзя. Если РА сходится к заданной функции, то корни в его знаменателе стремятся к сингулярным точкам.

В математической литературе исследовались, главным образом, одноточечные дробно-рациональные аппроксимации типа Паде [16, 19]. Двухточечные аппроксимации Паде, соединяющие асимптотики в переходных слоях, использовались лишь в отдельных прикладных задачах [2, 13, 14]. Ранее в работах [4, 5] был предложен и эффективно применен на примерах функций Эйри и Блазиуса метод соединения внутренней и внешней асимптотик с помощью кусочно-монотонной интерполяции [6] и с условиями гладкости в переходной области [7]. Затем в работах [10, 11] данный метод был использован для немонотонной интерполяции, в качестве которой применяется Паде-аппроксиманта с условием тривиальности для кривизны в точке максимума. В настоящей работе систематизируется и методологически обосновывается процедура применения соединения асимптотических разложений в задачах математической физики и обтекания плоской пластины гиперзвуковым потоком реагирующего газа.

2. Применение одноточечных Паде-аппроксимант (РА) в начальных задачах математической физики

Рассмотрим уравнение Дуффинга [8], которое можно исследовать как модельное с помощью различных методов, позволяющих сравнивать их

эффективность. Применим к задаче метод возмущений, считая малым параметром амплитуду колебаний a .

Уравнение имеет вид

$$\ddot{u} + u + u^3 = 0 \quad (2.1)$$

с начальными условиями

$$u(0) = a \quad (2.2)$$

$$\dot{u}(0) = 0$$

Выражение для частоты колебаний следует из точного решения и имеет вид

$$\omega = \frac{\pi\sqrt{1+a^2}}{2\sqrt{2}} K(\theta), \quad \theta = \operatorname{arctg} \sqrt{\frac{a^2}{a^2+2}} \quad (2.3)$$

где $K(\theta) = \int_0^{\pi/2} \frac{d\psi}{\sqrt{1 - \frac{a^2}{2+a^2} \sin^2 \psi}}$ - эллиптический интеграл 1-го рода.

Асимптотическое разложение для ω имеет вид

$$\omega = 1 + \frac{3}{8}a^2 - \frac{21}{256}a^4 + \frac{81}{2048}a^6 - \frac{6549}{262144}a^8 + \dots \quad (2.4)$$

Если ограничиться первыми тремя членами разложения (2.4) (отбрасывая члены ряда $O(a^3)$ и выше), то Паде-аппроксимация будет иметь вид

$$F_{22}(a) = \omega_2 = \frac{32 + 19a^2}{37 + 7a^2} \quad (2.5)$$

Принимая во внимание компоненты ряда (2.4) до четвертого порядка, получим

$$F_{44}(a) = \omega_4 = \frac{1 + 1.13a^2 + 0.261a^4}{1 + 0.756a^2 + 0.0599a^4} \quad (2.6)$$

Сравнение полученных приближений (2.5) и (2.6) показывает их удовлетворительное соответствие с точным решением (2.3)

3. Применение двухточечных Паде-аппроксимант (ТРРА) в краевых задачах

Для решения краевых задач будем использовать гипотезу о существовании двух асимптотик, для двух предельных значений параметра. С физической точки зрения каждая нетривиальная асимптотика обычно имеет инверсию. Другими словами, если для $\varepsilon \rightarrow 0$ ($\varepsilon \rightarrow \infty$) существует асимптотика, то можно построить и асимптотику для $\varepsilon \rightarrow \infty$ ($\varepsilon \rightarrow 0$). Но теперь возникает сложнейшая проблема для асимптотических методов – построение решений, приемлемых для $0 \leq \varepsilon \leq \infty$. Для этого используются различные подходы и методики. Наиболее распространенным является метод сращиваемых асимптотических разложений (Mathing method). При этом оперируют понятиями *внутренней* и *внешней асимптотик*, действующих в областях $\varepsilon \rightarrow 0$ и $\varepsilon \rightarrow \infty$ соответственно. Однако для корректного применения метода сращивания необходимо знать точку сращивания или, по крайней мере, область перекрытия асимптотик. Точное описание всего переходного слоя $0 < \varepsilon < \infty$ существует лишь в тех случаях, когда имеются специальные функции типа функции Эйри, связывающие в один узел разное поведение решений по обе стороны слоя.

Для соединения неперекрывающихся асимптотик в последнее время интенсивно разрабатывается метод, опирающийся на двухточечные аппроксимации Паде (ТРРА-Two-Points Pade Approximants) [6, 7, 13, 18, 20, 21].

В работах [5, 7] таким путем построены температурные профили в пограничном слое газа. В последующих работах [10, 11] метод хорошо зарекомендовал себя при исследовании теплообмена в гиперзвуковом пограничном слое. В работах [6, 17] метод соединения асимптотик применяется к пограничному слою в реагирующем газе, а в работах [4, 15] – к слою Кнудсена в разреженном газе. Метод соединения использовался в исследовании устойчивости развивающегося слоя Экмана на вращающейся пластине [20].

Двухточечные Паде-аппроксиманты вводятся следующим образом: пусть существуют внутренняя асимптотика

$$\varphi^i(\varepsilon) = \sum_{i=0}^{\infty} a_i \varepsilon^i \quad \varepsilon \rightarrow 0 \quad (3.1)$$

и внешняя асимптотика

$$\varphi^e(\varepsilon) = \sum_{i=0}^{\infty} b_i \varepsilon^{-i} \quad \varepsilon \rightarrow \infty \quad (3.2)$$

Тогда функция

$$\Phi_{b/m}(\varepsilon) = \frac{\sum_{l=0}^b \alpha_l \varepsilon^l}{\sum_{l=0}^m \beta_l \varepsilon^l} \quad (3.3)$$

будет называться двухточечной Паде аппроксимантой асимптотических разложений (3.1), (3.2). Коэффициенты α_l, β_k определяются так, что первые p коэффициентов правильной части разложения функции (3.3) в ряд Лорана совпадают с коэффициентами (3.1), а $(m+n+2-p)$ коэффициентов главной части ряда Лорана совпадают с коэффициентами (3.2).

Рассмотрим в качестве примера применения ТРРА краевую задачу Эйри

$$\begin{aligned}
y'' - \lambda^2 xy &= g(x)y \\
y(0) &= 1 \\
y(\infty) &= 0, \lambda \gg 1
\end{aligned}
\tag{3.3}$$

Асимптотическое решение имеет вид [14]

$$y(x) = U(\xi) [1 + O(\lambda^{-1})], \quad \xi = x\lambda^{2/3} \tag{3.4}$$

где $U(\xi)$ – функция Эйри:, определяемая из

$$\begin{cases}
U'' - \xi U = 0 \\
U(0) = 1 \\
U(\infty) = 0
\end{cases}
\tag{3.5}$$

Переходный слой задается областью $x = c_* \lambda^{2/3}, c_* - const.$

Внутренняя область переходного слоя – $x \in [0, x_-(\lambda)], x_- = o(\lambda^{-2/3})$

Внутренняя асимптотика функции Эйри

$$U = 1 - a\xi + \frac{1}{6}\xi^3 + O(\xi^4), \xi \rightarrow 0, \tag{3.6}$$

$$a = 3^{1/3} \frac{\Gamma(2/3)}{\Gamma(1/3)} \approx 0.7290 .$$

Внешняя область переходного слоя – $x \in [x_+(\lambda), \infty), \lambda^{-2/3} = j(x_+).$

Внешняя асимптотика функции Эйри

$$\begin{aligned}
U &= b\xi^{-1/4} \exp\left(-\frac{2}{3}\xi^{3/2}\right) \times \\
&\left[1 - \frac{5}{48}\xi^{-3/2} + O(\xi^{-3})\right], \xi \rightarrow \infty
\end{aligned}
\tag{3.7}$$

Паде-аппроксиманта

$$U_a = \frac{1 - a\xi + \frac{2}{3}\xi^{2/3} - \frac{2}{3}a\xi^{5/2} + \frac{32}{5}a\xi^4}{1 + \frac{32}{5}\frac{a}{b}\xi^{17/4}} \times \exp\left(-\frac{2}{3}\xi^{3/2}\right) \quad (3.8)$$

Аппроксимация функции Эйри (3.8) сохраняет по три члена асимптотик (3.6) и (3.7) на обоих концах и обеспечивает хорошую точность внутри слоя. Относительная ошибка

$$\Delta U = \frac{|U_m - U_a|}{U_m} \cong 1.5\% \quad (3.9)$$

где U_m – точное решение уравнения (3.5).

Специфика краевой задачи учитывается в форме Паде аппроксиманты – умножая ее на добавочную функцию $\exp(-\frac{2}{3}\xi^{3/2})$, мы обеспечиваем «мягкий» выход на асимптотики при $0 < \xi < \infty$ вплоть до полного совпадения с ними в граничных точках области.

4. Теплообмен в пограничном слое реагирующей смеси газов

Рассмотрим установившееся ламинарное течение в сверхзвуковом пограничном слое бинарной смеси газов в условиях отсутствия массовых сил, термо- и бародиффузии и излучения. Будем предполагать, что течение смеси является «химически замороженным», то есть изменение концентрации компонент в пограничном слое происходит за счет их конвекции и диффузии. Тогда автомодельные уравнения при $\mu = T$ имеют вид [6, 17]

$$\begin{aligned} \varphi''' + \varphi\varphi'' &= 0; \quad T'' + \sigma\varphi T' + a\sigma \\ \varphi''^2 + 2\sigma T' \sum_{i=1}^2 J_i &= 0 \\ J_1' - \varphi c_1' &= 0; \quad J_i = -\frac{Le_i}{\sigma} c_i'; \quad \sum_{i=1}^2 c_i = 1 \end{aligned} \quad (4.1)$$

где:

$$\varphi(\zeta) = \frac{\Psi}{\sqrt{x}}; \zeta = \frac{\eta}{2\sqrt{x}};$$

$$\eta = \int_0^y \frac{dy}{T}; T = T(\zeta); a = \frac{1}{4} M^2 (\kappa - 1);$$

Ψ – функция тока, $T(\zeta)$ – безразмерная температура, J_i – безразмерная плотность диффузионного потока концентрации компоненты i , c_i – безразмерная концентрация компоненты i , σ – число Прандтля, Le – число Льюиса, M – число Маха, κ – показатель адиабаты.

Граничные условия на поверхности пластины

$$\varphi(0) = \varphi'(0) = 0; \quad T(0) = T_S; \quad c_1(0) = c_{1S};$$

$$\varphi(\infty) = 2; \quad T(\infty) = 1; \quad c_1(\infty) = c_{10}; \quad c_2(\infty) = c_{20} \quad (4.2)$$

Внутренняя асимптотика при $\zeta \rightarrow 0$

$$\varphi = a_2 \zeta^2 + O(\zeta^4); \quad T = T_S + T_1 \zeta - 2a\sigma a_2^2 \zeta^2 + O(\zeta) \quad (4.3)$$

$$c_1 = c_{1S} + c_{11} \zeta + O(\zeta^4)$$

Внешняя асимптотика при $\zeta \rightarrow \infty$

$$\ln(\varphi'') = -\zeta^2 + c\zeta + \ln A + o(1),$$

$$\ln(\varphi''') = -\zeta^2 + c\zeta + \ln A + o(1), \quad (4.4)$$

$$\ln(-c_1') = -2 \frac{\sigma}{Le_1} + \ln B_1 + o(1)$$

Решение краевой задачи (4.1)-(4.2) аппроксимируем, соединяя асимптотики (4.3) и (4.4) Паде-аппроксимантами

$$\varphi'_a(\zeta) = 2 \left(1 - \frac{1 + A\zeta^3}{1 + \alpha_1\zeta + \alpha_2\zeta^2 + 4\zeta^4} \exp(-\zeta^2 + c\zeta) \right) \quad (4.5)$$

$$T_a(\zeta) = \frac{\zeta_m - \zeta}{\beta_0 + \beta_1\zeta} \exp(-\sigma(\zeta^2 - c\zeta)) \quad (4.6)$$

$$c'_{1a}(\zeta) = c_{11} \frac{1 + \gamma_1\zeta + \gamma_2\zeta^2}{1 + \gamma_2\zeta} \exp(-2 \frac{\sigma}{Le_1} \zeta) \quad (4.7)$$

Следуя методике [26, 27], определим параметры Паде-аппроксимант (4.5)-(4.7), используя асимптотические разложения (4.3), (4.4)

$$\alpha_1 = a_2 + c; \quad \alpha_2 = a_2^2 + ca_2 - \frac{c^2}{2} - 1; \quad \beta_0 = \frac{\zeta_m}{T_1}; \quad \beta_1 = \frac{c\sigma\zeta_m}{T_1} - \frac{1}{T_1}$$

$$\gamma_1 = \frac{2\sigma(Le_1 - \sigma)}{Le_1 - 2\sigma}, \quad \gamma_2 = \frac{2\sigma^2}{Le_1^2(1 - 2\sigma)} \quad (4.8)$$

При $\zeta \rightarrow \infty$ три члена в асимптотике (4.4) обеспечиваются, если положить $\beta_1 = -\frac{1}{B}$. Граничные условия (4.2) удовлетворяются, принимая

$$\varphi_a(\zeta) = \int_0^\zeta \varphi'_a(\zeta) d\zeta; \quad (4.9)$$

$$T_a(\zeta) = T_s + \int_0^\zeta T'_a(\zeta) d\zeta;$$

$$c_{1a} = c_{s1} + \int_0^\zeta c'_{1a}(\zeta) d\zeta$$

Интегральные условия (4.9) дополним нормирующими соотношениями

$$1 = T_s + \int_0^\infty T'_a d\zeta; \quad c_{10} = c_{s1} + \int_0^\infty c'_{1a} d\zeta \quad (4.10)$$

и интегральными соотношениями, полученными на основании совпадения Паде-аппроксимант (4.6)-(4.7) с точностью до трех членов во внутренних разложениях (4.3), (4.4)

$$a_2 = \frac{1}{2} \int_0^{\infty} \varphi_a' (2 - \varphi_a') d\zeta; \quad c = \int_0^{\infty} (2 - \varphi_a') d\zeta \quad (4.11)$$

Используя внешнее разложение, получим интегральное соотношение для A : умножим первое уравнение системы (4.1) на $\exp(\zeta^2 - c\zeta)$, проинтегрируем от 0 до ∞ , вычисляя интеграл от первого члена уравнения по частям с учетом (4.4), получим

$$A = 2a_2 - \int_0^{\infty} (\varphi - 2\zeta + c) \varphi'' \exp(\zeta^2 - c\zeta) d\zeta \quad (4.12)$$

Аналогично, из второго уравнения системы (4.1) с учетом (4.4) получим интегральное соотношение для B

$$B = \sigma \int_0^{\infty} \left(T \left(\varphi - 2\zeta + c - \frac{Le_1 - Le_2}{\sigma} c_1' \right) + a\varphi'^2 \right) \exp(\sigma \zeta^2 - c\zeta) d\zeta - T_1. \quad (4.13)$$

Интегральное соотношение для B_1 получим, умножая третье уравнение (4.1) на величину $\exp\left(\frac{2\sigma}{Le_1} \zeta\right)$

и интегрируя полученное выражение от 0 до ∞ , с учетом (4.4) как это было сделано выше

$$B_1 = \int_0^{\infty} \left(\varphi - \frac{2\sigma}{Le_1} \right) c_1' \exp\left(\frac{2\sigma}{Le_1} \zeta\right) d\zeta. \quad (4.14)$$

Априори при больших значениях числа Маха известно, что распределение температуры имеет максимум вблизи стенки в точке ζ_m . Таким образом, из уравнений (4.1) следует очевидное условие

$$T''(\zeta_m) = -a\sigma\varphi''^2(\zeta_m). \quad (4.15)$$

Подставляя в (4.15) производную аппроксиманты из (4.8), получим условие для определения точки экстремума ζ_m . Так как через величину B распределение концентраций оказывает влияние на распределение температуры, а вычисление B показывает зависимость от верхнего предела интегрирования, то условие, полученное из (4.15) служит средством контроля при выборе верхнего предела интегрирования для B .

Полученные результаты. Обсуждение. Задача была решена для эффективной бинарной смеси газов $CO + O_2 + N_2 + CO_2$, причем ввиду близости молекулярных масс CO, O_2, N_2 эти компоненты объединялись в одну группу (сопровождаясь индексом 1), а компонента CO_2 представляла вторую группу (с индексом 2). В предположении химически замороженного течения предполагалось, что концентрация азота остается постоянной, а гомогенная реакция протекает по схеме Вулиса $2CO + O_2 = 2CO_2$. Расчеты проводились при числах Маха $M=5; 10$, числе Прандтля $\sigma = 0.72$, числах Льюиса $L=1,2; 1,3$.

Библиографический список:

1. Алексеева, Е. А. Соединение температурных асимптотик в пограничном слое. / Е. В. Алексеева, Р.Г. Баранцев, А.В. Шатров // Вестник СПбГУ. – 1996. – сер. 1, № 8. – С. 96-99
2. Андрианов, И. В. Обращение преобразований Лапласа с помощью двухточечных Паде-аппроксимант / И. В. Андрианов, Г. Д. Шапиро // Проблемы машиностроения. – 1990. – № 34. – С. 58-60
3. Баранцев, Р. Г. Дефиниция асимптотики и системные триады. / Р. Г. Баранцев // Асимптотические методы в теории систем. – Иркутск. – 1980. – С.70-81
4. Баранцев, Р. Г. Асимптотическое исследование теплообмена к горячей стенке в высокоскоростном потоке / Р. Г. Баранцев, Е. В. Майоров, И. В. Прохоров // Тепломассообмен-ММФ-92. – Минск, 1992. – Т.1. – С. 86-89

5. Баранцев, Р. Г. Соединение асимптотик в переходном слое / Р. Г. Баранцев, Д. А. Пашкевич // Асимптотические методы в задачах аэродинамики и проектирования летательных аппаратов. – Иркутск. – 1994. – С. 67-70
6. Баранцев, Р. Г. Теплоперенос в пограничном слое реагирующего газа / Р. Г. Баранцев, Д. А. Пашкевич, А. В. Шатров // Тепломассообмен-2000: Минский междунар. форум (MIF-2000). – Минск, 2000. – С. 185-188
7. Баранцев, Р. Г. Соединение асимптотик в пограничном слое с помощью Паде-аппроксимант / Р. Г. Баранцев, А. В. Шатров // 10 Зимняя школа по механике сплошных сред. Тезисы докладов. – Екатеринбург, 1995. – С. 24-25
8. Ван-Дайк, М. Методы возмущений в механике жидкости. / М. Ван-Дайк. – М.: Мир, 1967. – 287 с
9. Грэхем, Р. Конкретная математика. Основания информатики / Р. Грэхем, Д. Кнут, О. Поташник. – М.: Мир, 1998. – 703 с
10. Шатров, А. В. Использование Паде-аппроксимант при соединении асимптотических решений в гидродинамике / А. В. Шатров // Математика. Компьютер. Образование». – Вып. 6, Ч. 2. – М.: Прогресс-Традиция, 1999. – С. 305-312
11. Шатров А. В. Соединение внутренних и внешних асимптотик в переходных слоях вязкой жидкости и газа / А.В. Шатров // VIII всероссийский съезд по теоретической и прикладной механике. Аннотации докладов. (Пермь, 23-29 августа 2001 г.). – Екатеринбург: УрОРАН, 2001. – С. 602-603
12. Andrianov, I. V. Application of Pade Approximants in Perturbation Methods/ I. V. Andrianov // Advances in Mechs. – 14, 2., – 1991. – p.3-15
13. Andrianov, I. V. Two-Point Pade Approximants and Their Applications to Solving Mechanical Problems / I. V. Andrianov, Yu. V. Mikhlin, S. Tokarzewsk // J. Theor. And Appl. Mech. – 35, – 3. – 1997. – p. 577-606

14. Avrejcevicz, T Asymptotic Approach in Nonlinear Dynamics / T. Avrejcevicz, I. V. Andrianov, L. I. Manevich. – Berlin u.a.. – 1998. – 310 p.
15. Anolik, M. V. Combination of asymptotics in the Knudsen layer. II. Testing / M.V. Anolik, R.G. Barantsev // Rarefied Gas Dynamics, 21st Intern. Symp., Book of Abstracts. – Marseille. – 1998. – v.2. – p. 75-76
16. Baker, G. A. The Pade approximants / G.A. Baker, jr. & T.L. Gammel. // J. Math. Anal. And Appl.. – 2. – № 1. – 1961. – p. 21
17. Barantsev, R. G. Combination of asymptotics in the boundary Layer of a reacting gas mixture / R. G. Barantsev, D. A. Pashkevich, A. V. Shatrov // Proc. of the 5th conf. on dynamical systems theory and applications. – Łódź. – 1999. – p. 137-140
18. Kruskal, M. B. Asymptotology / M.B. Kruskal // Math. Models in Phys. Sci. N-J. – 1963. – p.17-48
19. Martin, P Two-point quasifractional approximant in physics. Truncation error / P. Martin, G. A. Baker, Jr. // J. Math. Phys.. – 1991. – 32:6. – p. 1476-1477
20. Shatrov, A. V. Combination of asymptotics in boundary problems of hydrodynamics / A. V. Shatrov // OFEA'2001. Abstracts of intern. Conf., June, 25-29. – 2001. – St.-Petersburg. – 2001. – p. 59-60
21. Shatrov, A. V. New trends in asymptotical methods of mechanics and applied mathematics / A. V. Shatrov. // Abstracts of Int. conf. Asymptotics and Differential Equations (AsDEq-2002). – Ufa. – 23-30 May, 2002

Информация об авторах

Data on authors

Шатров Анатолий Викторович – докт. физ.-мат. наук, заведующий кафедрой
ММЭ ФГБОУ ВПО «ВятГУ»

Тел.: 89127096301

E-mail: shatrov1@yandex.ru

Shatrov Anatoly – Prof., PhD of Mathematics & Physics, Head of Department
of Mathematical Modeling in Economics FGEI HPE «VyatSU»

Tel.: +79127096301

E-mail: avshatrov1@yandex.ru

Advanced science

Научный журнал

№ 2 (2)

«Вятский государственный университет» ПРИП ФГБОУ ВПО «ВятГУ» 610000,

Киров, ул. Московская, 36 Тел.: (8332) 64-23-56, <http://vyatsu.ru>