МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ
ИМЕНИ М. В. ЛОМОНОСОВА

Факультет
вычислительной математики
и кибернетики

Сборник статей молодых ученых факультета ВМК МГУ

Выпуск 9

МОСКВА — 2012

УДК 004+517+519+681
ББК 22
С23

Редакционный совет сборника:
С. А. ЛОЖКИН, А. Н. ДАРЬИН, Ю. С. КОРУХОВА,
А. В. ПОЗДНЕЕВ, И. Г. ШЕВЦОВА, А. А. ВОРОНЕНКО

Составители:
И. Г. ШЕВЦОВА, А. В. ПОЗДНЕЕВ

Технический редактор:
А. В. ПОЗДНЕЕВ

С23 **Сборник статей молодых ученых факультета ВМК МГУ** / Ред. совет: Ложкин С. А. и др. — М.: Издательский отдел факультета ВМК МГУ (лицензия ИД № 05899 от 24.09.2001), 2012. — Выпуск 9. — 255 с.

В настоящий сборник вошли статьи, написанные молодыми учеными факультета вычислительной математики и кибернетики Московского государственного университета имени М. В. Ломоносова в 2011–2012 гг.

УДК 004+517+519+681
ББК 22

© Составление и оформление. Шевцова И. Г., Позднеев А. В., 2012
© Совет молодых ученых факультета ВМК МГУ, 2012
© Издательский отдел факультета ВМК МГУ, 2012

ISBN 978-5-89407-482-5

Данный выпуск посвящается
100-летию со дня рождения
Б. В. Гнеденко —
выдающегося математика,
крупного специалиста
по теории вероятностей
и математической статистике

СОДЕРЖАНИЕ

Современные математические модели влияния на цену: приложение к задаче оптимального управления портфелем на рынке, движимом заявками
Н. А. Андреев 7

Усовершенствованный алгоритм нечеткого поиска и особенности его реализации в облачной архитектуре
А. И. Аристов 26

Задача оптимального управления для математической модели финансового кризиса
М. А. Губанова 46

Об одной смешанной задаче для уравнения теплопроводности, приводящей к спектральной задаче с граничными условиями третьего рода, одно из которых содержит спектральный параметр
Д. А. Гуляев 64

Об одном подходе к решению задачи поиска нот музыкальных произведений
Ю. С. Корухова, М. В. Мытрова 68

Оценка фона изображения для автоматического обнаружения разметки и дефектов дорожного покрытия
А. С. Лебедев 86

Управление эллипсоидальным контейнером
А. И. Месяц 102

Схемы оценки уверенности в экспертных системах
Н. А. Моросанова 122

Построение множеств достижимости некоторых управляемых систем пиксельным методом
А. О. Новикова 136

О числе множеств, k-свободных от нуля, в группах простого порядка
В. Г. Саргсян 154

Алгоритм оценки кривой доходности по нескольким группам облигаций с оптимальным выбором весов
И. С. Сорокин, В. А. Лапшин 173

Алгоритм восстановления глобального контекста
Д. Е. Стельмашенко 191

Гарантированное оценивание в задаче поиска на плоскости
В. А. Степанович 210

Библиотечная реализация Пролог-решателя
И. В. Струков 235

РЕФЕРАТЫ 251

УДК 519.866

Современные математические модели влияния на цену: приложение к задаче оптимального управления портфелем на рынке, движимом заявками

© 2012 г. Н. А. Андреев

nandreev@cs.msu.su

Кафедра системного анализа ВМК МГУ

1 Введение

Основной задачей при управлении портфелем является динамическое определение его структуры с целью максимизации прибыли и/или достижения определенного уровня риска. В связи с меняющимся состоянием рынка необходимо постоянное наблюдение за стоимостью портфеля и его перестройка в случае необходимости, что сводится к покупке или продаже некоторого количества активов. В современной финансовой математике при описании цен финансовых инструментов стандартным предположением является предпосылка о *совершенном рынке* (*perfect market*) — в любое время можно торговать любым активом в неограниченном объеме, без влияния на рыночную цену и комиссии. На практике гипотеза совершенного рынка не подтверждается в силу наличия транзакционных издержек и ограниченного доступного объема активов. Данная проблема может казаться несущественной для относительно активного рынка и небольших сделок. Совершение

Исследование осуществлено в рамках Программы фундаментальных исследований НИУ ВШЭ в 2011 году.

сделки достаточно большого объема приводит к значительному отклонению ее итоговой стоимости от рыночной. При необходимости совершения серии сделок возникает риск движения цены в неблагоприятную сторону, что приводит к необходимости учитывать как волатильность рынка, так и влияние уже совершенных сделок на будущую цену, которое особенно заметно при транзакциях большого объема.

Рассмотренным проблемам было уделено значительное внимание, особенно в последние годы, с появлением доступных высокочастотных данных о торгах внутри дня. Повышенный интерес к моделям динамики рынка связан с актуальностью задачи управления крупным портфелем, особенно на низколиквидном рынке или рынке с малым объемом торговли. Данная проблема актуальна в первую очередь для институциональных инвесторов, для которых незначительная реструктуризация портфеля влечет сделки больших объемов. В данной работе приводится обзор основных моделей влияния на цену для рынка, движимого заявками. Приводится классификация и взаимосвязь основных подходов, проводится анализ сделанных предположений и границ применимости моделей. Также рассматриваются смежные вопросы, касающиеся динамики книги заявок и приводится краткий обзор современных результатов.

2 Транзакционные издержки и ликвидность рынка

В данной работе рассматривается только динамика *рынка, движимого заявками* (*order-driven market*), который представляется в виде *книги лимитированных заявок* (*limit order book*) — множеств заявок на покупку (bid orders) и продажу (ask orders), доступных в данный момент. Каждая заявка в книге содержит информацию о своей цене и соответствующем объеме. При сделке объема V происходит исполнение лучших

(по цене) заявок суммарным объемом V, т.е. данные заявки изымаются из книги. В любой момент игроки могут выставлять и снимать свои заявки по усмотрению.

Отказ от предпосылки о совершенном рынке приводит к возникновению дополнительного риска рыночной ликвидности: реальная цена сделки на рынке будет отличаться от расчетного значения. Достаточно учесть, что на рынке не существует цены актива как таковой, существуют лишь цены заявок на покупку/продажу. Разумное предположение о том, что цена $P(t)$ находится между лучшими ценами обеих сторон: $P(t) \in (B(t), A(t))$, где $A(t)$ — лучшя цена продажи, $B(t)$ — лучшая цена покупки, — приводит к тому, что продажа актива, например, будет в лучшем случае осуществляться по цене $B(t) < P(t)$. Обычно в качестве цены берется $P(t) = \frac{A(t)+B(t)}{2}$. При этом, если объем сделки превышает объем лучшей заявки, то остаток исполнится по худшей цене и средняя цена сделки отклонится от рыночной еще больше. Допустим, продажа происходит в момент времени t_0. Сразу после исполнения лимитированных заявок лучшая цена покупки становится $B(t_0 + 0) < B(t)$. Прокси рыночной цены принимает меньшее значение $P(t_0 + 0) = \frac{A(t_0+0)+B(t_0+0)}{2} < P(t_0)$, тем не менее формальное падение цены не было вызвано какой-либо информацией об активе, а связано с устройством самого рынка. Поскольку мнение рынка о цене не изменилось, при необходимости совершить серию сделок инвестору выгодно выждать определенное время, за которое книга заявок восстановится и цена актива вернется на прежний уровень.

Рассмотренный пример показывает важность понятия ликвидности рынка и транзакционных издержек. Тем не менее не существует четкого определения *рыночной ликвидности* (*market liquidity*). Неформально можно сказать, что «ликвиден тот рынок, на котором участники совершают крупные сделки часто и с незначительным влиянием на цену» [14]. Изучение процесса торговли и организации рынков привело к созданию

теории рыночной микроструктуры и появлению целого класса микроструктурных моделей. К сожалению, на данный момент нет единого мнения о наборе характеристик, описывающих функционирование рынка, вследствие чего наблюдается их вольная интерпретация и методы измерения. В данной работе рассматривается одна из точек зрения на природу риска ликвидности, введенная Кайлом [8]. Согласно данной теории, изменение цены вызвано ассиметрией информации между участника рынка: объем и цена заявки воспринимаются остальными участниками как дополнительная информация об истинной цене актива, вследствие чего формируется новое представление рынка. Как следствие, в работе [8] возникает понятие *постоянного влияния на цену* (*permanent price impact*) — изменение в представлении рынка о рыночной цене p_t после прихода новой информации (о новых заявках или сделках). В частности, для случая дискретного времени

$$p_t = p_{t-1} + \lambda q_t,$$

где q_t — дисбаланс заявок на покупку/продажу в текущем аукционе. Параметр λ был назван «лямбдой Кайла» и является первой численной характеристикой ликвидности.

С целью измерения рыночной диквидности необходимо представление данного сложного понятия в виде набора простых характеристик. Общепринятым является подход Кайла, выделившего следующие аспекты (атрибуты) ликвидности:

1. *сжатость*(*tightness*) — затраты на открытие и закрытие позиции за короткий период времени. В данной работе под сжатостью будет подразумеваться *котируемый спрэд*(*quoted spread*), т.е. разность между лучшей ценой продажи и лучшей ценой покупки.
2. *глубина*(*depth*) — минимальный объем сделки, который ведет к сдвигу цен. Из определения понятно, что глубина — это объем лучшей заявки в книге лимитированных заявок.

3. *релаксация*(*resiliency*) — способность цен возвращаться в нормальное состояние после возмущения, не вызванного приходом новой информации (неинформативного шока). Под временем релаксации понимается время, за которое произошло возвращение.

Несложно видеть, что три характеристики отвечают за возможные издержки в рассмотренном выше примере с продажей. Абсолютно ликвидный рынок должен обладать нулевой сжатостью, бесконечной глубиной и нулевым временем релаксации. Третий аспект является самым нечетким в плане определения, так как формализуется через такие понятия как «нормальное состояние», «возмущение» и «неиформативный шок». Вследствие этого исследования данного аспекта начали появляться совсем недавно, при этом в достаточно вольной интерпретации.

3 Модели влияния на цену

Одной из основных задач при управлении портфелем является моделирование влияния конкретной сделки агента на цену актива. При этом принято разделять постоянный эффект (*permanent price impact*) и временный (*temporary price impact*). Постоянное изменение цены непосредственно связано с тем, как информация о сделке воспринимается рынком. В дальнейшем $S(t)$ обозначает фундаментальную рыночную цену актива (*fundamental price*), $P(t)$ — подразумеваемую рынком цену актива. Обычно в качестве $P(t)$ берут среднее между ценами лучших заявок, как было сделано выше. Важно понимать разницу между фундаментальной и подразумеваемой ценой: $S(t)$ — ненаблюдаемая величина, зависящая от фундаментальной информации об активе (в случае акций — это состояние компании); $P(t)$ — цена, определяемая рынком на основании торгов и отражающая мнение рынка об истинном значении $S(t)$. Обозначим за $g(\cdot)$ постоянное изменение подразумеваемой цены в

ходе торгов, за $\rho(\cdot)$ — временное изменение цены в результате конкретной сделки. Всюду далее подразумевается наличие одного крупного инвестора, который решает задачу оптимального управления портфелем. Только данный агент имеет возможность совершать сделки достаточно крупного размера, для которых необходимо учитывать влияние на цену. Современные модели рассматривают несколько подходов к формальному описанию динамики цены.

В работе Альмгрена и Крисса [3] рассматривается задача ликвидации/открытия позиции в дискретном времени с шагом Δt. Динамика подразумеваемой цены тогда описывается следующим уравнением:

$$P(t_k) = S(t_k) + \Delta t g(\xi_{t_k}),$$

где xi_{t_k} — средняя скорость торговли агента за прошедший период времени. Реальная цена сделки будет отличаться от $P(t_k)$ при достаточно большом объеме. Это формализуется в модели следующим образом: средняя цена сделки за единицу объема равна

$$\tilde{P}(t_k) = P(t_k) + \rho(\xi_{t_k}).$$

Таким образом, в модели присутствуют как постоянное, так и временное изменение цены. Влияние на цену зависит лишь от интенсивности торгов. Авторы не рассматривают задачу ликвидации позиции в общем виде, а лишь для линейных g и ρ, получая решение в аналитическом виде. Такое же упрощение встречается и в более ранней работе Бертсимаса и Ло [4], впервые применивших метод динамического программирования для данной задачи. Авторы предполагают, что цена после сделки объема v_{t_k} ($v_{t_k} > 0$ означает покупку, $v_{t_k} < 0$ — продажу) удовлетворяет уравнению

$$P(t_k) = S(t_k) + \Delta t v_{t_k}.$$

Модель Бертсимаса и Ло является, с одной строны, упрощением модели Альмгрена и Крисса в том смысле, что не учитывает

временное влияние на цену, что эквивалентно предположению о бесконечной глубине рынка. С другой стороны, постоянное влияние на цену в ней зависит от объема сделки, что эквивалентно предыдущему подходу лишь в случае равномерного разбиения по времени или линейности $g(\cdot)$. Ни одна из моделей не учитывает релаксацию цены, предполагая, что агент входит в рынок с интервалами, большими времени восстановления.

Более общая модель динамики для задачи открытия позиции описана в работе Обижаевой и Вана [10]. Авторы разделяют фундаментальную цену $S(t)$, подразумеваемую цену $P(t)$ и лучшую цену продажи $A(t)$, описывая динамику каждой из них в непрерывном времени. Выход агента на рынок происходит конечное число раз за фиксированный период (импульсное управление портфелем). Существенным упрощением является предположение плоской стационарной структуры книги заявок: пусть объем, доступный в момент t по ценам в интервале $[P, P + dP)$, равен $q(t, P)\, dP$; предполагается, что

$$q(t, P) = q = \mathrm{const}\,.$$

Динамика $S(t)$ не специфицируется и рассматривается в общем виде. Постоянное влияние на цену проявляется в динамике $P(t)$:

$$P(t) = S(t) + \lambda X(t),$$

где $X(t)$ — объем позиции агента в момент t. Таким образом, постоянное влияние описывается линейной функцией, зависящей от объема сделки. Временное изменение цены также линейно зависит от времени, что следует из предположения о блочной структуре. Важным преимуществом модели Обижаевой и Вана является учитывание релаксации лучшей цены. Обозначим как

$$D(t) = A(t) - P(t) - \frac{s}{2}$$

отклонение от справедливой цены продажи в момент t, где s — текущий спрэд. Модель предполагает неизменное значение спр-

эда, т.е. сжатость рынка не меняется со временем. Пусть в момент t_0 произошла сделка объема v_0. Согласно [10], до наступления следующей сделки отклонение описывается следующей формулой:

$$D(t) = v_0 \kappa e^{-\alpha(t-t_0)}, \tag{1}$$

где $v_0\kappa$ — начальное отклонение, соответствующее мгновенному изменению цены в результате сделки. Экспоненциальный закон сходимости следует из стационарности рынка, в частности, интенсивности поступления новых заявок.

Подход Обижаевой и Вана используется во многих современных работах при решении задачи оптимальной ликвидации/открытия позиции. Модель имеет несколько обобщений в двух направлениях: обобщение законов релаксации и введение сложной функции плотности заявок. В работе Альфонси, Шида и Шульца [2] закон убывания отклонения $D(t)$ в прежних обозначениях имеет вид

$$D(t) = v_0 \kappa e^{-\int\limits_{t_0}^{t_0+t} \alpha(\tau)\, d\tau}, \tag{2}$$

где $\alpha(t)$ — положительная измеримая функция на рассматриваемом интервале. Очевидно, что (1) является частным случаем (2) при $\alpha(t) \equiv \mathrm{const}$. Данная модель допускает общий закон релаксации, который не является стационарным. Недостатком метода является то, что скорость релаксации $\alpha(t)$ является динамической характеристикой самого рынка и не зависит от конкретного момента наступления шока.

Другая формализация релаксации введена Альфонси, Фрутом и Шидом в [1]. Рассматривается модель рынка в рамках подхода Обижаевой и Вана со сложной стационарной плотностью распределения заявок $q(P)$. Рассматривается непрерывная положительная функция плотности в терминах отклонения от справедливой цены, которая связана с отклонением от справедливой цены после сделки следующим образом:

$$\int\limits_{0}^{D(t_0+0)} q(D)\,dD = v_0, \tag{3}$$

если книга в момент t_0 находилась в невозмущенном состоянии ($D(t_0) = 0$). На рисунке 1 приводится иллюстрация к динамике книги до и после сделки. Таким образом можно рассматри-

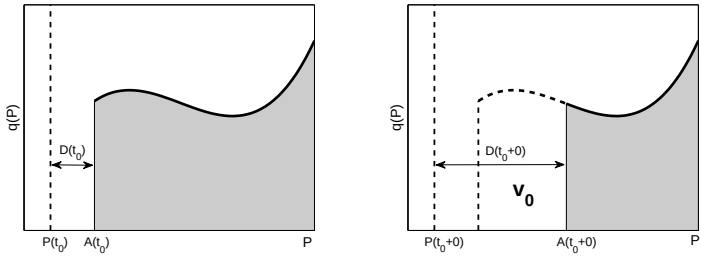


Рис. 1. Динамика книги заявок в момент перед наступлением сделки объема v_0 (слева) и сразу после (справа).

вать взаимнооднозначное отображение $D_0(v) : v \to D(t_0 + 0, v)$ Из уравнения (3) несложно видеть, что функция мгновенного изменения цены $D_0(\cdot, v)$ является линейной функцией от объема только при условии $q(D) \equiv \text{const}$. Таким образом, модель допускает нелинейное мгновенное влияние на цену. Предположение о стационарности функции плотности интерпретируется следующим образом: рынок стремится воссоздать некоторую установленную форму книги заявок. После прихода очередной сделки или снятия заявки часть объема удаляется из книги, что приводит к постепенному восполнению, пока не будет достигнуто исходное состояние. При фиксированной форме книги заявок обозначим недостающий объем в момент t как $E(t)$.

Тогда верно равенство

$$E(t) = \int\limits_0^{D(t)} q(D)\, dD.$$

Отображение $D(t) \to E(t)$ является взаимнооднозначным в силу положительности $q(D)$. [1] рассматривает две возможных формализации релаксации рынка:

1. восстановление цены происходит по экспоненциальному закону:
$$D(t+s) = D(t)e^{-\alpha_1 s}. \tag{4}$$
2. восстановление недостающего объема происходит по экспоненциальному закону:
$$E(t+s) = E(t)e^{-\alpha_2 s}. \tag{5}$$

В случае $q(D) \equiv \text{const}$ получаем $E(t) = qD(t)$, что означает эквивалентность подходов (4) и (5). Значит, подход к релаксации в [10] имеет два естественных и равноправных обобщения в случае сложной формы книги заявок. Более того, подходы эквивалентны лишь при этом условии.

Наиболее общей из рассмотренных работ является [11], описывающая влияние на цену и издержки для книги заявок с учетом дискретности цен и релаксации. Главным предположением является, как и ранее, стационарная структура книги заявок. Задача оптимального управления ставится принципиально другим образом: вместо максимизации прибыли ставится задача минимизации издержек при достижении целевого состояния позиции (приобретение/ликвидация). Новая постановка избавляет от необходимости моделировать цену актива и, в частности, цену лучшей заявки, оставаясь в терминах отклонения $D(t)$. Тем самым уменьшается количество сделанных предположений. Рассматривается одна сторона книги, далее будет

описываться сторона заявок на продажу. Форма книги описывается с помощью функции суммарного объема: для достаточно большого числа M вводится бесконечная σ-конечная мера μ и непрерывная слева функция

$$F(x) = \mu\left([0, x)\right).$$

Константа M имеет смысл максимально возможного отклонения от лучшей цены, бесконечность меры интерпретируется как возможность провести сделку сколь угодно большого объема, что является классическим предположением, которое можно считать формально верным, так как в подавляющем большинстве случаев единственная сделка не покрывает все имеющиеся заявки, если рынок относительно ликвиден. σ-конечность меры запрещает концентрацию бесконечного объема на конкретной цене. Функция $F(x)$ имеет смысл суммарного объема по ценам с отклонением $D \in [0, x)$. Данная формализация позволяет рассматривать реальную форму книги, в которой распределение заявок не бывает непрерывным. В непрерывном случае модель сводится к предыдущим работам:

$$\mu(A) = \int\limits_A q(x)\, dx,$$

где $q(\cdot)$ — производная Радона-Никодима.

Модель Шрева учитывает релаксацию в терминах динамики недостающего объема, но в более общем случае, нежели [1]:

$$\frac{d}{dt}E(t) = -h(E_t), \tag{6}$$

где $h(E)$ — строго-монотонно возрастающая локально липшицева функция, характеризующая скорость восполнения объема. Видим, что [1] есть частный случай (6) при $h(E) = \alpha_2 E$. Отклонение $D(t)$ цены лучшей заявки от справедливой цены связано с $E(t)$ следующим соотношением:

$$D(t) = \psi(E(t)),$$

$$\psi(y) = \sup\{x \geqslant 0 \mid F(x) < y\}, \; y > 0.$$

4 Обсуждение основных предположений

4.1 Временное влияние на цену

Вопрос о моделировании временного влияния активно обсуждается в современной литературе, но на данный момент не существует единого мнения на этот счет. Эффект временного влияния логично разделить на мгновенное влияние и релаксацию. Функцией мгновенного влияния назовем величину, на которую изменится лучшая цена сразу после исполнения сделки. Таким образом, рассматривается функция от объема сделки $f(v)$, где $v > 0$ соответствует покупке актива, $v < 0$ — продаже. Как было видно, большинство из рассмотренных работ предполагает линейную функцию влияния. Данное предположение неоднократно подвергалось критике и зачастую отвергается реальными данными. На рисунке 2 приведены графики $f(v)$ для разных моментов времени в течение одного торгового дня для акций ОАО «Лукойл» на бирже ММВБ. Видим, что поведение функции не обязательно линейно, могут наблюдаться выпуклая, вогнутая и выпукло-вогнутая формы. Объяснение и исследование класса возможых форм кривой зависит от модели динамики всей книги заявок. Микроструктурную модель, дающую обоснование приведенным формам кривой, можно найти, например, в [12].

4.2 Учет арбитражных возможностей

В сложным моделях, начиная с Обижаевой и Вана, рассматривалась лишь одна сторона книги заявок. Интерпретация данного предположения сводится к тому, что агент, управляющий позицией, имеет возможность либо покупать, либо продавать актив. Таким образом, модель применима лишь к частной задаче управления — открытию или ликвидации позиции.

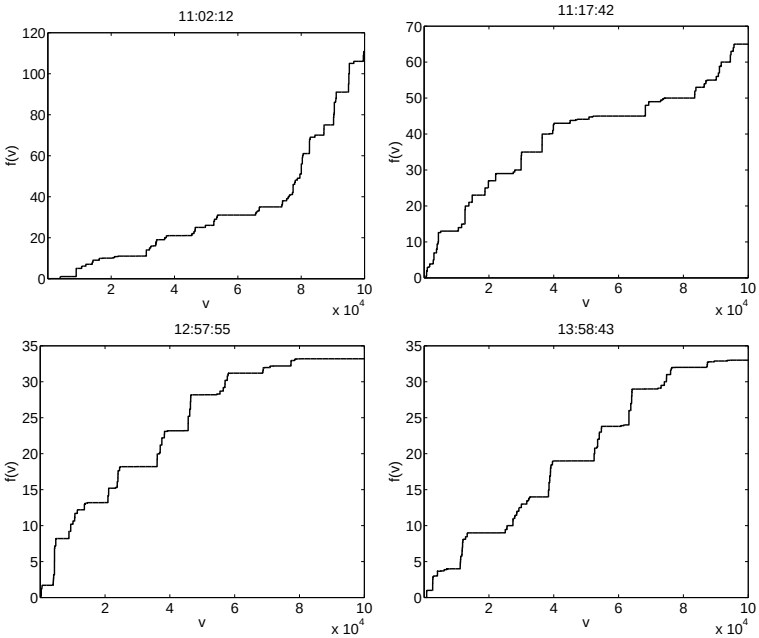


Рис. 2. Графики функции влияния $f(v)$ в зависимости от объема сделки для разных моментов времени в течение одного торгового дня.

Ограничение на стратегию агента может показаться слишком жестким, так как стратегия даже при открытии позиции может включать кратковременную продажу с целью извлечения дополнительной прибыли. С другой стороны, подобное ограничение обосновано по двум причинам:

1. продажа при необходимости покупки может рассматриваться как манипулирование рыночной ценой в личных интересах и введение рынок в заблуждение, что рассматривается регулятором как недопустимое поведение.
2. если задачей агента является не извлечение прибыли, а минимизация издержек при проведении операций, то совершение сделок противоположной направленности не со-

ответствует задаче агента.

Совместное рассмотрение двусторонней книги заявок и стратегий, допускающих покупку и продажу в рамках одной стратегии, накладывает дополнительные ограничения на динамику книги и функцию влияния $f(v)$. Связано это в первую очередь с принципом отсутствия арбитражных возможностей на рынке — фундаментальным принципом финансовой математики, говорящим о том, что невозможно извлечение безрисковой прибыли на конечном горизонте. Данное условие может быть формализовано различными способами и налагает дополнительные ограничения на функцию влияния. Не ограничивая общности, рассмотрим случай непрерывной торговли агента. Пусть x_t — объем позиции агента, при этом возможно открытие т. н. «короткой позиции», т. е. $x_t < 0$. В общем случае, модель динамики лучшей цены с релаксацией в непрерывном времени записывается в виде

$$D(t) = D(0) + \int\limits_0^t f(\dot{x}_s) G(t-s)\, ds, \tag{7}$$

к которому сводятся модели [1,3,4,10]. Издержки при торговле с использованием стратегии $X = x_t$, $t \in [0, T]$, равны

$$C(X) = \int\limits_0^T \dot{x}_t\, dt \int\limits_0^t f(\dot{x}_s) G(t-s)\, ds. \tag{8}$$

В работе [7] говорится, что рынок не должен допускать квазиарбитраж, т. е. не существует стратегии, не изменяющей объема позиции

$$X\colon \int\limits_0^T \dot{x}_t\, dt = 0,$$

но приносящей положительный доход. Показано, что в рамках введенной микроструктурной модели данное ограничение влечет $f(v) = \kappa v$. Гатерал [5] накладывает на рынок более слабое

условие, говорящее об отсутствии динамического арбитража: для любой стратегии, не изменяющей объем позиции, издержки неотрицательны:

$$X\colon \int\limits_0^T \dot{x}_t\,dt = 0 \quad \Rightarrow \quad C(X) \geqslant 0.$$

Условие является более слабым, нежели в [7], и является аналогом отсутствия статистического арбитража. В целом, оно говорит о том, что невозможно манипулирование ценой, т. е. влияние на цену при отсутствии фактической покупки или продажи актива. Таким образом, появляется дополнительное условие, устанавливающее связь между возможной функцией f и функцией затухания G, которую необходимо задавать априорно или калибровать на данных. В частности, при $G(t) \equiv 1$ (только постоянное влияние на цену) имеем результат, полученный также в [7]: $f(-v) = -f(v)$. Это означает, что заявки на обеих сторонах книги должны быть расположены симметрично, что чаще всего подтверждается данными.

4.3 Стационарность формы книги заявок

Модели [1, 2, 10, 11] построены для случая стационарной книги заявок. Необходимость данного предположения вызвана трудностью моделирования и прогнозирования структуры книги заявок. Итоговая форма является достаточно сложным математическим объектом — результатом совместной игры многих игроков, имеющих в общем случае индивидуальные адаптивные стратегии выставления лимитированных и рыночных заявок, а также обменивающихся информацией с различной частотой. Предположение о постоянстве формы, с одной стороны, позволяет свести динамику всего объекта к динамике одной характеристики (лучшей цены или недостающего объема). В то же время гипотеза оправдывает себя на небольшом горизонте для относительно ликвидных рынков. В этом случае

форма книги достаточно инерционна и не успевает существенно измениться. Тем самым накладывается естественное ограничение на применимость данного класса моделей — управление портфелем для короткого горизонта.

Создание динамической модели книги заявок является актуальной проблемой рыночной микроструктуры и активно обсуждается в литературе. На данный момент можно выделить два подхода к моделированию структуры книги. Один из них направлен на создание микроструктурной модели динамического равновесия рынка и определения стратегий игроков. Данный подход является фундаментальным и направлен на выявление процессов, задающих динамику рынка. Естественно, подобная модель опирается на многочисленные предположения, проверка которых на практике бывает затруднительна, обоснование предположений базируется в основном на наличии косвенных признаков. Пример подобной модели приведен, например, в [12].

Второе направление ставит своей задачей построение эмпирической модели, позволяющей с заданной точностью моделировать и предсказывать процессы на рынке. При этом основные предположения вытекают из анализа входных сигналов. Большинство подобных эмпирических моделей опирается на совместное моделирование интенсивностей прихода заявок определенного типа (например, приход лимитированной заявки на покупку/продажу, отмена заявки), а также объемов событий. Так, в [13] считается, что суммарный объем поступивших лимитированных и рыночных заявок в единицу времени постоянен во времени, вероятность отмены заявки считается стационарной и не зависит от времени. Хасбрук [6] предлагает моделировать совместное наступление событий k типов с помощью k-мерного пуассоновского процесса. Данный подход предполагает независимость протекающих процессов от предыстории, что может быть актуально только в случае отсутствия долгосрочной памяти в стратегиях игроков, например, на высоколиквид-

ном рынке, где информация о заявке мгновенно усваивается участниками. На практике зачастую наблюдаются лавинообразные всплески интенсивностей событий определенного вида. Для моделирования подобного рода эффектов Ларж [9] предложил использовать не пуассоновские потоки, а многомерный процесс Хокса (Hawkes process) $N(t)$, использующийся для моделирования хаотических лавинообразных явлений: $N(t)$ — k-мерный точечный процесс, интенсивность λ которого зависит от времени согласно следующей формуле:

$$\lambda(t;\theta) = \mu(t;\theta) + \int\limits_0^t W(t-s;\theta)\, dN(s),$$

где θ — вектор параметров модели, $\mu(t;\theta)\colon \mathbb{R}_+ \times \Theta \to \mathbb{R}_+^k$ — детерминированная функция, позволяющая учитывать среднюю интенсивность наступления событий в течение дня, $W(t;\theta)\colon \mathbb{R}_+ \times \Theta \to \mathbb{R}_+^{k\times k}$ — детерминированная функция затухания. Процесс Хокса является, таким образом, самовозбуждающимся — каждое наступившее событие повышает вероятность наступления событий в будущем. Данный подход имеет много преимуществ, но требует сложной процедуры оценки вида неизвестных функций и калибровки параметров, что возможно лишь на достаточно ликвидных рынках. В остальных случаях необходимо вводить априорные предположения о классе функций и зависимостях между событиями.

5 Заключение

Были рассмотрены основные модели влияния на цену и связанные с ними смежные вопросы. Проведен анализ сделанных предположений и изучена взаимосвязь рассмотренных подходов. На основании рассуждений можно утверждать, что наиболее общей статической моделью, допускающей реально наблюдаемую форму книги заявок и учитывающей релаксацию,

является модель Шрева [11]. Принципиально другой подход к моделированию релаксации приведен в [2], где скорость восстановления вводится как характеристика рынка, не зависящая от событий, приведших к шоку. Также дана классификация подходов к динамическому моделированию книги заявок и приведены некоторые эмпирические модели, наиболее распространенные на данный момент.

Список литературы

[1] Alfonsi A., Fruth A., Schied A. Optimal execution strategies in limit order books with general shape functions // Quantative Finance. — 2009. — Vol.10, 2. — pp.143 —157.

[2] Alfonsi A., Schied A., Schulz A. Constrained portfolio liquidation in a limit order book model. — 2009. — Working paper.

[3] Almgren R., Chriss N. Optimal execution of portfolio transactions // Journal of Risk. — 2001. — №1. — pp.5 —39.

[4] Bertsimas D., Lo A. W. Optimal control of execution costs // Journal of Financial Markets. — 1998. — №1. — pp. 1 —50.

[5] Gatheral J. No-Dynamic-Arbitrage and Market Impact // Quantitative Finance. — 2010. — №10. — pp. 749 —759.

[6] Hasbrouck J. Trading fast and slow: security market events in real time // Working paper, Stern School of Business, New York University. — 1999.

[7] Huberman G., Stanzl W. Price manipulation and quasiarbitrage // Econometrica. — 2004. — №72. — pp. 1247 —1275.

[8] Kyle A. Continuous auctions and insider trading // Econometrica. — 1985. — №53. — pp. 1315 —1336.

[9] Large J. Measuring the resiliency of an electronic limit order book // Journal of Financial Markets. — 2007. — Vol.10, 1. — pp.1 —25.

[10] Obizhaeva A., Wang J. Optimal trading strategy and supply/demand dynamics — 2005. — Available at SSRN: papers.ssrn.com/sol3/papers.cfm?abstract id=686168.

[11] Predoiu S., Shaikhet G., Shreve S. Optimal execution of a general one-sided limit-order book // SIAM Journal on Financial Mathematics. — 2011. — №2. — pp. 183 —212.

[12] Rosu I. A dynamic model of the limit order book // Review of Financial Studies. — 2009. — Vol.22, 11. — pp. 4601 —4641.

[13] Smith E., Farmer J. D., Gillemot L., Krishnamurthy S. Statistical theory of the continuous double auction // Quantitative Finance. — 2003. — №3(6). — pp. 481 —514.

[14] Bank for International Settlements, 1999, «Market Liquidity: Research Findings and Selected Policy Implications», Bank for International Settlements – Monetary and Economic Department, Committee on the Global Financial System, Basel, May.

УДК 004.424.62

Усовершенствованный алгоритм нечеткого поиска и особенности его реализации в облачной архитектуре

© 2012 г. А. И. Аристов

ai_aristov@mail.ru

Кафедра нелинейных динамических систем и процессов управления

1 Введение

В настоящее время получает распространение программное обеспечение, реализованное в облачной архитектуре. В основе этой архитектуры лежит принцип доступа к общим ресурсам через сеть, причем на серверах происходит не только хранение, но и обработка данных.

В частности, находит применение система управления взаимоотношениями с клиентами (CRM, Customer Relationship Management) Salesforce, имеющая облачную реализацию. Она предназначена для автоматизации работы с заказчиками, анализа результатов взаимодействия с ними и улучшения маркетинга. В ней организовано хранение данных на серверах, а также имеется стандартная функциональность, соответствующая распространенным процессам обработки данных. Пользователи работают с этой системой, в основном, через браузер.

Для расширения стандартной функциональности в Salesforce используются язык программирования Apex и разметка Visualforce.

Данная статья основана на опыте разработки одного алгоритма для Salesforce. Он реализует поиск в базе данных по нетривиальному критерию: «строка нечетким образом соответствует некоторому шаблону».

Здесь будет предложен такой алгоритм, затем будут обсуждаться особенности его реализации как в облачной архитектуре, так и в традиционной.

Возьмем за основу определение расстояния между строками (конечными последовательностями символов) из [1].

Определение 1. *Элементарными преобразованиями строки называются преобразования следующего вида: вставка одного символа, удаление одного символа, замена одного символа на другой.*

Определение 2. *Расстоянием между двумя строками S_1 и S_2 (расстоянием Левенштейна) называется минимальное количество элементарных преобразований, переводящих одну строку в другую. Будем обозначать его через $\rho_0(S_1,\ S_2)$.*

В связи с этим определением назовем статью [2], где рассматривались строки из символов «0» и «1», и впервые было таким способом введено расстояние.

Будем обозначать длину строки S через $|S|$. Будем обозначать через $S[i]$ i-й символ в строке S, считая, что начальный символ имеет номер 1, а не 0.

В поиске в базах данных большую роль играют шаблоны, представляющие собой строки со специальными символами «`?`» и «`*`»: они позволяют с помощью одного запроса находить записи, не имеющие ровно одно значение, а находящиеся в некотором множестве значений. Специальный символ «`?`» обозначает ровно один произвольный символ, а специальный символ «`*`» — произвольное количество произвольных символов (не исключается случай пустого символа). При этом если требуется найти строку с символами «`?`» и «`*`», то в запросе перед ними следует указать символ «\». Например, по запросу `"abc?d*e\?"` могут найтись записи `"abcxde?"`, `"abcydpqre?"`, но не `"abczdef"`.

В [1] указана следующая формула для вычисления расстояния между строками S_1 и S_2:

$$\rho_0(S_1,\ S_2) = \delta(|S_1|,\ |S_2|),$$

где

$$\delta(i,j) = \begin{cases} i, & j = 0, i \geqslant 0, \\ j, & i = 0, j > 0, \\ \min(\delta(i,j-1)+1,\ \delta(i-1,j)+1, & \\ \delta(i-1,j-1)+m(i,j)), & i > 0, j > 0, \end{cases} \quad (1)$$

где функция $m(i,j)$ имеет значение 0, если символы $S_1[i]$ и $S_2[j]$ равны, и 1 в противном случае. Доказательство приведено в [1]. Там же описан алгоритм Вагнера-Фишера, вычисляющий расстояние в соответствии с этой формулой.

Идея алгоритма Вагнера-Фишера состоит в следующем. Пусть $|S_1| = M$, $|S_2| = N$. Рассмотрим матрицу D со строками от 0 до N и столбцами от 0 до M. Заполним нулевую строку числами 0, 1,..., M. Заполним нулевой столбец числами 0, 1,..., N. Затем обойдем строки от первой до N-й, последовательно вычисляя значения в каждой из них в позициях 1,..., M по формуле (1). Тогда искомым расстоянием будет последний элемент матрицы, т. е. $D[N, M]$.

В качестве примера рассмотрим вычисление расстояния между строками S_1 = "*abcde*" и S_2 = "*axce*". Здесь $M = 5$, $N = 4$, и матрица будет выглядеть следующим образом:

Таблица 1. Матрица, демонстрирующая алгоритм Вагнера-Фишера

	0	**1**	**2**	**3**	**4**	**5**
0	0	1	2	3	4	5
1	1	0	1	2	3	4
2	2	1	1	2	3	4
3	3	2	2	1	2	3
4	4	3	3	2	2	2

Из этой матрицы видно, что $\rho_0(S_1,\ S_2) = D[4,5] = 2$. Действительно, для того чтобы перевести S_1 в S_2, нужно два элементарных преобразования: заменить второй символ на `"x"` и удалить четвертый.

Замечание. Результатом выполнения алгоритма Вагнера-Фишера является матрица $(N+1)\times(M+1)$: она может использоваться для нахождения редакционного предписания, т. е. последовательности преобразований, переводящей одну строку в другую [1]. Здесь мы исследуем задачу вычисления расстояний, сводящуюся к поиску последнего по порядку построения элемента матрицы. Для этого достаточно на каждом шаге построения матрицы держать в памяти предыдущую строку и уже найденные элементы текущей, т. е. можно вместо двумерного массива $(N+1)\times(M+1)$ использовать два одномерных массива размерности $(N+1)$. Однако в дальнейшем для пояснения работы алгоритма мы будем приводить полные матрицы.

Перейдем к обобщению введенного понятия расстояния.

Пусть, например, $S_1 = "abcxydz"$, $T_1 = "abc*d?"$. Строка S_1 соответствует шаблону T_1, если вместо звездочки взять последовательность `"xy"`, а вместо вопросительного знака — символ `"z"`. Рассмотрим теперь строку $S_2 = "abcxydz"$ и шаблон $T_2 = "qbc*d?"$. Заметим, что S_2 не соответствует T_2, однако она будет соответствовать этому шаблону, если сделать одно элементарное преобразование, а именно, заменить символ `"a"` на `"q"`. Поэтому естественно считать, что расстояния от S_1 до T_1 и от S_2 до T_2 равны соответственно 0 и 1. Таким образом, возникает вопрос об определении расстояния от строки до шаблона и о способе его вычисления.

2 Определение расстояния от строки до шаблона и его вычисление для одного частного случая задачи

Шаблон определяет некоторое множество строк. Естественно ввести расстояние следующим образом.

Определение 3. *Расстоянием от строки S до шаблона T будем называть минимум из расстояний от строки S до*

строк, соответствующих шаблону T:

$$\rho(S,\ T) = \min_{t \in T} \rho_0(S,\ t)$$

Сначала ограничимся рассмотрением шаблонов, содержащих в качестве специальных символов вопросительные знаки, но не звездочки. Очевидно, все строки, соответствующие любому такому шаблону, имеют одинаковую длину. Поэтому для вычисления расстояния от строки до такого шаблона пригодна формула (1) со следующим изменением: $m(i, j) = 0$ не только для равных символов $S[i]$ и $T[j]$, но и в случае, когда $T[j] =$ "?".

Рассмотрим вычисление расстояния от строки $S =$ "$abcde$" до шаблона $T =$ "$a?ce$". С учетом названного изменения формулы (1) по аналогии с алгоритмом Вагнера-Фишера можно построить следующую матрицу:

Таблица 2. Вычисление расстояния от строки до простого шаблона

	0	**1**	**2**	**3**	**4**	**5**
0	0	1	2	3	4	5
1	1	0	1	2	3	4
2	2	1	0	1	2	3
3	3	2	1	0	1	2
4	4	3	2	1	1	1

Значит, $\rho(S,\ T) = 1$. Действительно, S будет соответствовать T после одного элементарного преобразования, а именно, удаления четвертого символа.

3 Алгоритм для общего случая

В шаблоне, введенном пользователем, символы `"*"` и `"?"` могут быть как специальными символами, используемыми для задания множества строк, так и символами в обычном

смысле. Поэтому организован предварительный анализ введенного шаблона: по нему строятся новая строка, массив целых чисел и целое число `ast_count`, равное количеству специальных символов `"*"` в шаблоне. Новая строка содержит те же символы, что и исходная, но без знаков "экранирования" `"\"`, а массив содержит на i-й позиции 0, 1 или 2, если символ на этой позиции в новой строке является соответственно обычным символом, специальным символом `"?"` или специальным символом `"*"`. Например, если пользователь ввел строку `"abc?d\?e*f\*g\\h"`, то будут построены строка `"abc?d?e*f*g\h"` и массив $[0, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0]$.

Алгоритм для произвольного количества специальных символов `"*"` является весьма ресурсоемким. Однако практика показывает, что, как правило, пользовательский ввод содержит не более одного такого символа, а в этом случае алгоритм вычисления расстояния удается значительно оптимизировать. Кроме того, можно предложить некоторую оптимизацию для случая, когда шаблон содержит ровно две звездочки, и они находятся на его краях.

В контексте исходной задачи о поиске в базе данных можно утверждать, что не требуется вычислять в точности расстояния от каждой строки до шаблона — надо выявить только те строки, для которых это расстояние не превосходит выбранный пользователем целый параметр ε. Поэтому целесообразно ввести функцию расстояния, имеющую в качестве множества значений не все множество $\mathbb{N} \cup \{0\}$, а множество $\{0,\ 1, \ldots,\ \varepsilon,\ \text{«}\textit{много}\text{»}\}$, т. е.

$$\rho_\varepsilon(S,T) = \begin{cases} \rho(S,T), & \rho(S,T) \leqslant \varepsilon, \\ \text{«}\textit{много}\text{»}, & \rho(S,T) > \varepsilon \end{cases}$$

Такое определение расстояния позволяет значительно сократить количество арифметических действий. Смысл «много» будет иметь значение -1.

Рассмотрим подробнее части алгоритма, соответствующие

разным подклассам входных данных.

3.1 Случай 1: `ast_count=0`

Для вычисления расстояния в случае, когда шаблон не содержит специальных символов «*», пригоден алгоритм, описанный в разделе «Определение расстояния от строки до шаблона и его вычисление для одного частного случая задачи».

Сократить количество арифметических действий можно, если вычислять $\rho_\varepsilon(\cdot)$, а не $\rho(\cdot)$. Действительно, из формулы (1) видно, что если все элементы очередной строки матрицы превосходят ε, то и в следующей строке все элементы будут превосходить ε. Поэтому соответствующая проверка позволяет досрочно выйти из цикла и возвратить значение -1 (т. е. «много»), если расстояние заведомо превосходит ε.

3.2 Случай 2: `ast_count=1`

Прежде чем рассматривать шаблоны со звездочками, дадим определение подшаблона.

Определение 4. *Рассмотрим шаблон T, содержащий специальный символ «*» в позиции i. Подшаблонами этого шаблона назовем шаблоны, которые получатся, если в T вместо $T[i]$ подставлять произвольное количество специальных символов «?» или пустой символ.*

Составим алгоритм вычисления расстояния от строки S до шаблона T, содержащего ровно один специальный символ «*». Этот специальный символ обозначает произвольное количество произвольных обычных символов. Значит, множество строк, соответствующих T, является объединением множеств строк, соответствующих всем подшаблонам T. Расстоянием от строки S до шаблона T можно считать минимум из расстояний от этой строки до каждого из подшаблонов T. При этом для его вычисления нужно перебрать только конечное число вариантов. Действительно, расстояние заведомо не превосходит

величину $(|T| - 1)$ (в случае, когда никакие символы из S не совпадают с обычными символами из T). В этом случае под звездочкой подразумевается последовательность из $|S|$ вопросительных знаков. Если увеличивать количество специальных символов «?» в этой последовательности, то расстояние будет только увеличиваться. Если же перебирать более короткие последовательности, то оно может уменьшиться.

Рассмотрим следующий пример. Пусть надо вычислить расстояние от строки $S = "abcdxyz"$ до шаблона $T = "a * xz"$. Заметим, что $\rho(S,T) = \min\limits_{0 \leqslant k \leqslant 7} \rho(S, T_k)$, где $T_0 = "axz"$, $T_1 = "a?xz"$, $T_2 = "a??xz"$, $T_3 = "a???xz"$, $T_4 = "a????xz"$, $T_5 = "a?????xz"$, $T_6 = "a??????xz"$, $T_7 = "a???????xz"$. Последовательно рассматривая эти подшаблоны, применим к ним алгоритм, аналогичный алгоритму Вагнера-Фишера.

Для каждого из шаблонов T_i построим матрицу M_i, $i = 0$, 2, ..., 7 (таблицы 3–10).

Таким образом, $\rho(S,T) = \min(4,\ 3,\ 2,\ 1,\ 1,\ 1,\ 2,\ 3) = 1$.

Внесем в перебор вариантов некоторую оптимизацию.

1. Предварительная оптимизация состоит в том, чтобы использовать один массив $(|S| + |T|) \times (|S| + 1)$, вычисляя в нем последовательно все нужные матрицы. (Это существенно, если переопределения размеров массива являются сложными действиями, и для сокращения их количества допустим перерасход памяти.)

2. Пусть специальный символ «*» в шаблоне находится на месте i. Как видно из приведенных матриц (таблицы 3–10), соответствующие строки с малыми номерами у них совпадают. Действительно, во всех подшаблонах T_k первые (i-1) символов равны, а значит, равны соответствующие строки во всех матрицах.

3. Нумерация подшаблонов T_k именно в порядке возрастания количества вопросительных знаков на месте звездоч-

Таблица 3. Матрица M_0

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	1	0	1	2	3	4	5	6
2	2	1	1	2	3	3	4	5
3	3	2	2	2	3	4	4	4

Таблица 4. Матрица M_1

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	1	0	1	2	3	4	5	6
2	2	1	0	1	2	3	4	5
3	3	2	1	1	2	2	3	4
4	4	3	2	2	2	3	3	3

Таблица 5. Матрица M_2

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	1	0	1	2	3	4	5	6
2	2	1	0	1	2	3	4	5
3	3	2	1	0	1	2	3	4
4	4	3	2	1	1	1	2	3
5	5	4	3	2	2	2	2	2

Таблица 6. Матрица M_3

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	1	0	1	2	3	4	5	6
2	2	1	0	1	2	3	4	5
3	3	2	1	0	1	2	3	4
4	4	3	2	1	0	1	2	3
5	5	4	3	2	1	0	1	2
6	6	5	4	3	2	1	1	1

Таблица 7. Матрица M_4

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	1	0	1	2	3	4	5	6
2	2	1	0	1	2	3	4	5
3	3	2	1	0	1	2	3	4
4	4	3	2	1	0	1	2	3
5	5	4	3	2	1	0	1	2
6	6	5	4	3	2	1	1	2
7	7	6	5	4	3	2	2	1

Таблица 8. Матрица M_5

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	1	0	1	2	3	4	5	6
2	2	1	0	1	2	3	4	5
3	3	2	1	0	1	2	3	4
4	4	3	2	1	0	1	2	3
5	5	4	3	2	1	0	1	2
6	6	5	4	3	2	1	0	1
7	7	6	5	4	3	2	1	1
8	8	7	6	5	4	3	2	1

Таблица 9. Матрица M_6

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	1	0	1	2	3	4	5	6
2	2	1	0	1	2	3	4	5
3	3	2	1	0	1	2	3	4
4	4	3	2	1	0	1	2	3
5	5	4	3	2	1	0	1	2
6	6	5	4	3	2	1	0	1
7	7	6	5	4	3	2	1	0
8	8	7	6	5	4	3	2	1
9	9	8	7	6	5	4	3	2

Таблица 10. Матрица M_7

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	1	0	1	2	3	4	5	6
2	2	1	0	1	2	3	4	5
3	3	2	1	0	1	2	3	4
4	4	3	2	1	0	1	2	3
5	5	4	3	2	1	0	1	2
6	6	5	4	3	2	1	0	1
7	7	6	5	4	3	2	1	0
8	8	7	6	5	4	3	2	1
9	9	8	7	6	5	4	3	2
10	10	9	8	7	6	5	4	3

ки в исходном шаблоне позволяет улучшить оптимизацию предыдущего пункта. Действительно, у матриц, соответствующих шаблонам T_{k-1} и T_k, совпадают строки до $(i + k - 2)$ включительно.

4. Если $i < (|T| + 1)/2$, то символы в S и в T рассматриваются в обратном порядке. Это позволяет увеличить часть матрицы, общую для всех подшаблонов $\{T_k\}$, и, следовательно, сократить количество вычислений.

5. Как и при рассмотрении шаблонов, не содержащих звездочек, проверка условия, что в очередной строке все элементы превосходят ε, позволяет не строить полностью матрицу для подшаблона, расстояние до которого заведомо больше ε.

6. Если расстояние до очередного подшаблона оказалось равно нулю, то оставшиеся проверять не нужно.

3.3 Случай 3: `ast_count=2`

Составим алгоритм вычисления расстояния от строки S до шаблона T, в котором первый и последний символ являются специальными символами «*», тогда как на других позициях специальных символов «*» нет. Аналогично предыдущему разделу, здесь нужно подставить вместо этих звездочек последовательности из таких количеств вопросительных знаков, которые будут минимизировать расстояние от S до T.

Если никакие символы из S не совпадают ни с какими неспециальными символами из T, то $\rho(S,T) = |T| - 2$, причем наиболее близкими к строке S будут те шаблоны, в которых количества вопросительных знаков, подставленных вместо первой звездочки и вместо второй звездочки в сумме равны $|S|$. Значит, в общем случае $\rho(S,T)$ заведомо не превосходит $(|T| - 2)$. При этом суммарное количество подставленных в шаблон вопросительных знаков не превосходит $|S|$.

Рассмотрим следующий пример. Пусть нужно вычислить расстояние от строки $S = "xaby"$ до шаблона $T = "*ac*"$. Рассуждая по аналогии с предыдущим разделом, представим эту величину в виде $\rho(S,T) = \min\limits_{0 \leqslant k \leqslant 14} \rho(S, T_k)$, где $T_0 = "ac", T_1 = "ac?", T_2 = "ac??", T_3 = "ac???", T_4 = "ac????"$, $T_5 = "?ac"$, $T_6 = "?ac?"$, $T_7 = "?ac??"$, $T_8 = "?ac???"$, $T_9 = "??ac"$, $T_{10} = "??ac?"$, $T_{11} = "??ac??"$, $T_{12} = "???ac"$, $T_{13} = "???ac?"$, $T_{14} = "????ac"$. Построив соответствующие матрицы, получим:

$$\rho(S,T) = \min(3,\ 2,\ 2,\ 2,\ 2,\ 2,\ 1,\ 1,\ 2,\ 2,\ 2,\ 2,\ 2,\ 2,\ 2) = 1.$$

Заметим, что для перебора вариантов организованы вложенные циклы: внешний цикл — по количествам вопросительных знаков на месте первой звездочки, внутренний — по количествам вопросительных знаков на месте второй звездочки. На каждой итерации внешнего цикла вычисление аналогично вычислению в случае `ast_count=1`, поэтому применимы те же

приемы оптимизации (кроме обхода символов в обратном порядке).

3.4 Случай 4: остальные виды шаблонов

Остались нерассмотренными многочисленные, но не имеющие большого практического значения, виды шаблонов. А именно, шаблоны, содержащие более двух звездочек, а также те, которые содержат ровно две звездочки, но не на краях. Как и в других случаях, здесь надо найти минимум из расстояний от строки S до шаблонов, получающихся из исходного шаблона T заменой звездочек на последовательности вопросительных знаков или пустые символы (суммарное количество этих вопросительных знаков не превосходит $|S|$).

Для реализации такого подхода удобно использовать рекурсию.

Требуемое расстояние будет минимумом из расстояний от S до шаблонов $T_0,\ T_1, \ldots,\ T_{|S|}$, полученных из T подстановкой вместо первой звездочки соответственно пустого символа, одного знака вопроса и т. д. до последовательности из $|S|$ знаков вопроса. Каждый из этих шаблонов содержит на одну звездочку меньше, чем T, и к нему применима аналогичная процедура, но множества вопросительных знаков, подставляемых вместо первой звездочки, будут зависеть от номера шаблона. Для T_0 максимальное количество вопросительных знаков в такой последовательности равно $|S|$, для T_1 оно равно $(|S| - 1)$ и т. д. Для $T_{|S|}$ это множество состоит из одного элемента — пустого символа.

На некотором шаге рекурсии будут составлены шаблоны, не содержащие звездочек, и для них будет применим алгоритм, описанный в разделе, соответствующем случаю `ast_count=0`.

Рассмотрим следующий пример. Пусть $S = "abcd"$, $T = "*b*d"$. Приведем соответствующую схему рекурсивных вызовов (рисунок 1). Запись вида (`T`, N) будет означать, что на очередном шаге обрабатывается шаблон `T`, а N —

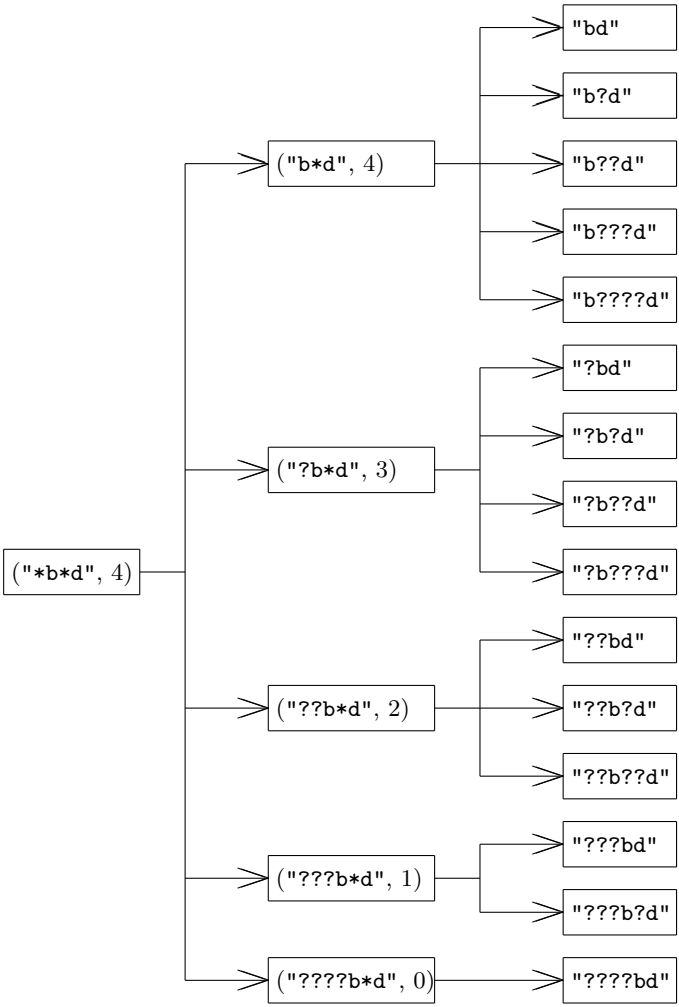


Рис. 1. Схема рекурсивных вызовов

максимальное количество вопросительных знаков, которые будут подставлены в `T` вместо первой звездочки).

В целях оптимизации можно организовать досрочный выход из цикла на очередном шаге рекурсии, если на какой-то итерации оказалось, что расстояние до очередного шаблона равно нулю. Кроме того, очевидно, остается в силе оптимизация для алгоритма, который обрабатывает шаблоны, не содержащие звездочек.

4 Реализация в традиционной архитектуре

Усовершенствованный алгоритм нечеткого поиска был реализован не только в облачной архитектуре, но и в традиционной — на языке Visual Basic for Applications (VBA). Во-первых, разработка сначала на VBA сделала процесс отладки более удобным. Во-вторых, интересна и сама реализация на этом языке, поскольку при работе в реальной системе CRM значительную роль играет предварительная подготовка баз данных в таблицах Excel с помощью макросов. Например, востребованы макросы, разделяющие адреса на составляющие (улица, дом и прочие) с учетом разнообразных форматов, встречающихся в исходных базах данных, макросы, проводящие преобразования строк специального вида, и макросы, основанные на поиске по нетривиальным критериям.

Здесь разработка приложения на VBA [3] была не основной, а вспомогательной задачей, поэтому оно имеет демонстрационный характер. Оно устроено следующим образом.

Будем считать, что в первом столбце записаны строки, а во втором — шаблоны, расстояния до которых требуется найти. При запуске макроса `main_handler` появляется диалоговое окно, содержащее следующие элементы управления:

1. Окошки для ввода трех чисел: номер строки, где начина-

ется обработка, номер строки, где завершается обработка, и погрешность. Обозначим эти числа соответственно через n_1, n_2 и ε.

2. Рамку-раздел «Вычисление расстояний», содержащий:
   (a) список для выбора одного из двух значений: «Индивидуальные шаблоны» (выбрано по умолчанию) и «Общий шаблон»,
   (b) окошко для ввода общего шаблона,
   (c) кнопку «Вычислить расстояния».
3. Рамку-раздел «Нечеткий поиск», содержащий:
   (a) окошко «Шаблон»,
   (b) рамку-подраздел «Искать» со значениями для выбора «Первое соответствие» (выбрано по умолчанию) и «Все соответствия»,
   (c) кнопку «Найти».
4. Кнопки «Справка» и «Закрыть».

Обозначим через $L[i, j]$ ячейку рабочего листа на пересечении строки i и столбца j.

Кнопка «Вычислить расстояния» запускает следующее преобразование. Если выбран пункт «Индивидуальные шаблоны», то для всех строк (i) от n_1 до n_2 будет вычислено расстояние от строки из $L[i, 1]$ до шаблона из $L[i, 2]$ и записано в ячейку $L[i, 3]$. Здесь и далее в этом разделе имеется в виду расстояние $\rho_\varepsilon(\cdot)$, где «погрешность» ε пользователь вводит в соответствующее окошко. Поэтому если $\rho(L[i, 1], L[i, 2])$ окажется достаточно большим, то $L[i, 3]$ будет присвоено значение -1. Если выбран пункт «Общий шаблон», то будут проведены аналогичные вычисления, но вместо значений из $L[i, 2]$ везде будет взят шаблон из соответствующего окошка в диалоговом окне.

Кнопка «Найти» запускает поиск в подмножестве первого столбца рабочего листа, которое определяется границами n_1 и n_2. При этом используется следующий критерий: $\rho_\varepsilon(L[i,1],T) \geqslant 0$, где T — шаблон, введенный пользователем в окошко «Шаблон». Если пользователь выбрал пункт «Первое соответствие», то поиск завершается после нахождения первой строки, удовлетворяющей названному критерию. Если пользователь выбрал пункт «Все соответствия», то макрос обходит все строки от n_1 до n_2 и выявляет те, которые удовлетворяют названному критерию. Результаты поиска выводятся в новом диалоговом окне.

Скриншот запущенного приложения показан на рисунке 2.

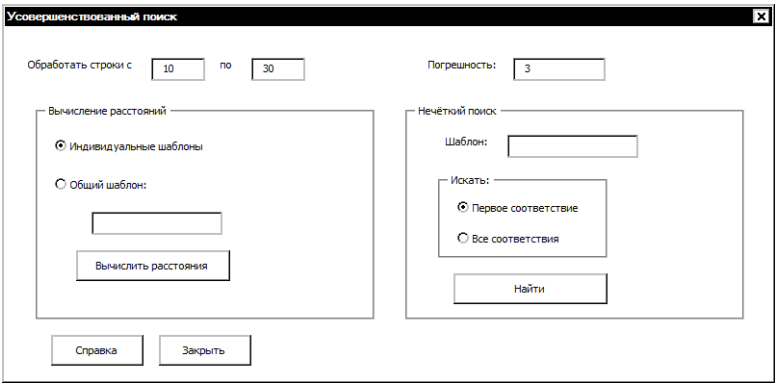


Рис. 2. Диалоговое окно приложения на VBA

5 Реализация в облачной архитектуре

Реализация алгоритма в облачной архитектуре не ограничилась переводом его с языка VBA на Apex: понадобилось несколько изменений, учитывающих специфику архитектуры Salesforce.

Основная задача работы состоит в том, чтобы создать страницу Visualforce и класс, которые будут обеспечивать следую-

щую функциональность. Пользователь вводит на этой странице шаблон T и погрешность ε, а затем нажатием кнопки запускает поиск в базе данных тех записей `{x}` об организациях, у которых поле названия `Name` близко к T: $\rho_\varepsilon(x.Name, T) \geqslant 0$. При этом найденные записи `{x}` должны быть отсортированы по возрастанию расстояний до T.

Прежде чем обсудить организацию вычислений, поясним общие принципы работы страниц Visualforce [4]. Они обеспечивают диалог с пользователем. К странице можно подключить класс-контроллер («controller») на языке Apex, содержащий методы, к которым будет обращаться страница. Эти методы могут использоваться как для организации реакций на события на странице, так и для получения информации из базы данных или записи в нее. Apex во многом напоминает язык Java, при этом имеются средства для обращения к базе данных.

С точки зрения реализации поиска важнейшее отличие облачной архитектуры от традиционной в том, что она значительно более чувствительна к сложности алгоритма. Если макрос может, проработав несколько минут, выдать ответ, то во время выполнения функции на Apex'е может возникнуть ошибка превышения допустимого количества арифметических действий. Поэтому облачная реализация ограничивается функциональностью для обработки шаблонов, содержащих не более одной звездочки (такие шаблоны составляют большинство реальных запросов). Кроме того, организованы относительно простые предварительные проверки строк, чтобы не вычислять расстояния для тех, которые заведомо не удовлетворяют критерию.

Предварительные проверки делаются следующим образом.

1. Проверка на этапе запроса. Для записей организаций создается поле-формула `NameLength` (т. е. поле, значение которого вычисляется автоматически), равное длине поля `Name`. В функции на Apex'е вводится целая переменная `t_eps`, которой присваивает-

ся значение, получаемое вычитанием из длины шаблона количества специальных символов «*» в нем и параметра ε (если получается отрицательное число, то `t_eps` полагается равным нулю). Затем делается запрос организаций из базы данных следующим образом: `List<Account> AllAccounts = [select Id, Name from Account where NameLength >=: t_eps];` (`List` — структура данных, аналогичная `ArrayList` в языке C#; `Id` — поле, которое представляет собой последовательность символов специального вида и служит уникальным ключом записи). Действительно, если длина строки меньше `t_eps`, то расстояние от нее до шаблона заведомо превосходит ε.

2. Первая грубая оценка. Здесь создается множество из символов, составляющих шаблон, за исключением специальных. Затем для очередной записи просматривается поле `Name`, и если это поле содержит более ε символов, не входящих в названное множество, то запись выводится из дальнейшего рассмотрения.

3. Вторая грубая оценка. На этом шаге создается множество из тех подстрок шаблона, которые имеют длину 2 и не содержат специальных символов. Например, по шаблону `T="abdc?ezt*d"` будет составлено множество `{«ab», «bd», «dc», «ez», «zt»}`. Затем у поля `Name` очередной записи двухсимвольные подстроки сопоставляются с построенным множеством по аналогии с предыдущим пунктом.

После этих проверок остаются записи, для которых расстояния $\rho_\varepsilon(\cdot)$ вычисляются непосредственно способами, описанными в предыдущих разделах.

На страницу полученные записи выводятся таблицами по 15 записей; если найдено более 15 записей, предусмотрены ссылки для перелистывания списка. В таблицах сделаны ссылки

на страницы, представляющие соответствующие записи в базе данных — для этого в запросе было указано поле `Id`.

Отметим, что другой важной особенностью разработки для Salesforce является необходимость тестирования специальными средствами этой системы. Для этого создаются классы и методы с заголовками специального вида. В них следует запускать методы из тестируемого класса с такими наборами входных данных, чтобы было выполнено максимально возможное количество строк из проверяемых методов (требуется запустить хотя бы 75 процентов проверяемого кода). В Salesforce имеются кнопки для запуска тестовых методов.

При этом рекомендуется проверять значения, полученные в результате выполнения тестируемых методов, с помощью специального метода `System.assertEquals`. Пусть, например, создан экземляр класса `c`, в котором есть метод `f`, возвращающий значение 1 при некоторых входных данных. Тогда проверку можно организовать следующим образом: `System.assertEquals(c.f(/*аргументы*/), 1);`. Если в методе есть ошибка, и он возвращает не 1, то после выполнения теста будет выдано соответствующее сообщение.

В целях тестирования в Salesforce был загружен ряд записей об организациях с похожими названиями. На рисунке 3 показан скриншот страницы Visualforce с результатами поиска по шаблону.

6 Заключение

В работе введено понятие расстояния от строки до шаблона. Предложен алгоритм вычисления этого расстояния, рассмотрены способы его реализации как в облачной, так и в традиционной архитектуре. Результаты исследования могут быть использованы для организации поиска в базах данных по нетривиальным критериям, в том числе, в системах, реализованных в облачной архитектуре.

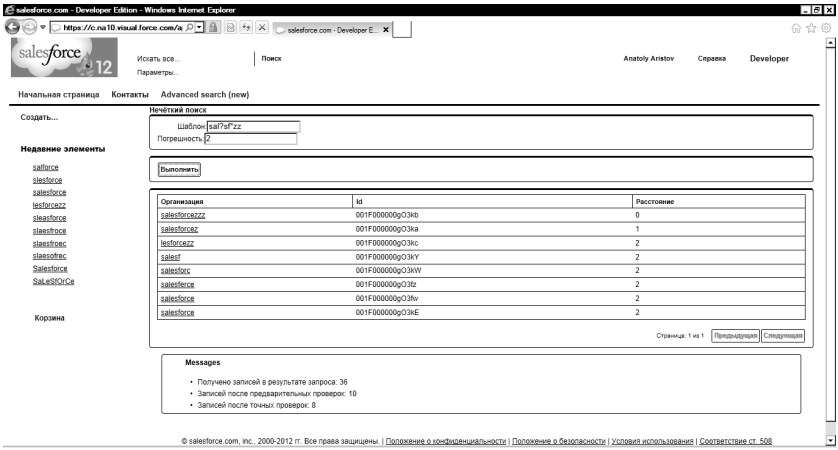


Рис. 3. Страница в Salesforce

Список литературы

[1] Энциклопедическая статья «Расстояние Левенштейна». — http://ru.wikipedia.org.

[2] *В. И. Левенштейн.* Двоичные коды с исправлением выпадений и вставок символа 1 // Проблемы передачи информации. — 1965. — том 1, выпуск 1. — С. 12-25.

[3] *А. Ю. Гарнаев.* Самоучитель VBA. — С.-Пб.: БХВ-Петербург, 2004 г. — 542 с. — ISBN 5-94157-410-X.

[4] *J. Ouellette.* Development with the Force.com Platform. — Boston: Addisson-Wesley, 2011 г. — 472 с.

[5] *Р. Седжвик.* Алгоритмы на C++. — М.: Вильямс, 2011 г. — 1056 с. — ISBN 978-5-8459-1650-1.

УДК 517.977.1

Задача оптимального управления для математической модели финансового кризиса

© 2012 г. М. А. Губанова

gubanovamargarita@gmail.com

Кафедра оптимального управления

1 Введение

На протяжении почти двухвекового периода становления и развития мирового индустриального общества в экономике многих стран происходили кризисы, во время которых наблюдался нарастающий спад производства, скопление нереализованных товаров на рынке, падение цен, крушение системы взаимных расчетов, крах банковских систем, разорение промышленных и торговых фирм, резкий скачок безработицы.

В девятнадцатом веке и первой половине двадцатого мир пережил несколько международных финансовых кризисов. Экономические кризисы до XX века ограничивались пределами одной, двух или трех стран, затем стали приобретать международный характер. Несмотря на то, что в последние десятилетия мировым сообществом созданы механизмы по предотвращению мировых кризисов (укрепление государственного регулирования хозяйственных процессов, создание международных финансовых организаций, проведение мониторинга и др.), как свидетельствует история мировых экономических катаклизмов, ни точно предсказать, ни тем более избежать их невозможно.

Все эти события показывают, что наше экономическое будущее небезопасно, мировая экономика является очень неустойчивой системой и может быть сломлена. Это также приводит

к вопросам: можно ли этих событий избежать, что может быть сделано для недопущения или предотвращения подобных событий в будущем, или смягчения их последствий. Поэтому сейчас как никогда важно владеть актуальной информацией о современном экономическом кризисе. Необходимо понять, что полезного можно извлечь из сложившейся ситуации, когда и какое решение стоит принять, чтобы правильно и вовремя отреагировать на те или иные изменения в работе, политике, экономике, обществе.

В связи с этим большой интерес представляют методы, основанные на построении и одновременном изучении математических моделей экономических процессов. Использование математического моделирования позволяет оценить аспекты и последствия, связанные с воздействием всевозможных кризисных и экстремальных ситуаций. В свою очередь, среди большого количества методов экономико-математического моделирования следует выделить математическую теорию оптимального управления, которая позволяет находить решения сложных экономико-прикладных задач.

Аппарат математической теории оптимального управления очень часто используется и в медицине для исследования процессов заболевания. Примером является задача о распространении эпидемии в неоднородном сообществе, контролируемая проведением вакцинации восприимчивых к заболеванию людей и карантина [1]. Наиболее распространенные противоэпидемические средства - это вакцинация населения, подверженного заболеванию, и медикаментозное лечение инфицированных людей. При некоторых тяжелых заболеваниях контроль за ходом эпидемии осуществляется с помощью изоляции инфицированных людей. Другим методом контроля является проведение программы «Здоровье», которая состоит в организации теле- и радиопередач, лекций, посвященных этой проблеме, и т. д. Эти меры можно рассматривать в качестве управляющих воздействий на ход эпидемии. Модели учитывают наличие имму-

нитета (постоянного или временного), факторы естественной рождаемости и смертности, латентный период заболевания и другие параметры.

Статья [8] предлагает рассмотреть модель финансового кризиса на основе модели распространения заболевания. В данной работе математическая модель взаимодействия фирм в условиях финансового кризиса изучается на основе управляемой модели процесса распространения эпидемии. Предположим, что экономика - это совокупность взаимодействующих различных фирм. В соответствии с математической моделью управления эпидемий разделим их на две группы: стабильные («здоровые») - фирмы, подверженные влиянию кризиса, и нестабильные («больные») - фирмы, пострадавшие от влияния кризиса. Считаем, что нестабильные фирмы имеют финансовые трудности (т.е. не в состоянии исполнить свои финансовые обязательства). Также отсутствует приток новых участников в систему и фирмы принадлежат только к одной из указанных групп. Кроме того, они взаимодействуют и оказывают влияние друг на друга.

2 Описание математической модели и свойства её переменных

Рассмотрим математическую модель взаимодействия фирм в условиях финансового кризиса:

$$\begin{cases} \dot{x}(t) = -\beta x(t)y(t) - \delta x(t) + v(t), \\ \dot{y}(t) = \beta x(t)y(t) + \delta x(t) - \gamma y(t), \quad t \in [0, T], \\ x(0) = x_0,\ y(0) = y_0; \quad x_0, y_0 > 0. \end{cases} \tag{1}$$

Здесь:
T - фиксированное время, на котором рассматривается процесс взаимодействия фирм;
$x(t)$ - количество фирм, подверженных влиянию кризиса в момент времени t (стабильная группа);

$y(t)$ - количество фирм, пострадавших от влияния кризиса в момент времени t (нестабильная группа);
δ - параметр, характеризующий скорость увеличения нестабильных фирм;
β - параметр, характеризующий частоту взаимодействия двух групп фирм;
γ - параметр, характеризующий число фирм, которые в результате влияния кризиса становятся неплатежеспособными.

Управление взаимодействием фирм осуществляется с помощью вложения денежных средств : $v(t)$ - управляющая функция (скорость вложения денежных средств в стабильную группу фирм). Под классом допустимых управлений $D(T)$ понимаем всевозможные измеримые функции $v(t)$, удовлетворяющие при почти всех $t \in [0, T]$ неравенствам $0 \leqslant v(t) \leqslant v_{max}$.

Исследуем свойства модели (1). Имеет место следующее утверждение.

Теорема 1. *Пусть задана произвольная управляющая функция $v(\cdot) \in D(T)$. Тогда отвечающее этому управлению $v(t)$ решение $(x(t), y(t))$ системы (1) определено на всем отрезке времени $[0, T]$ и удовлетворяет неравенствам:*

$$0 < x(t) < x_{max}, \quad 0 < y(t) < y_{max},$$

где $\quad x_{max} = x_0 + v_{max}T, \quad y_{max} = e^{\beta x_{max} T}(y_0 + x_{max}T).$

Доказательство. Пусть $(x(t), y(t))$ — решение системы (1), отвечающее управлению $v(t)$, где $t \in \Delta \subseteq [0, T]$. Здесь Δ - наибольший полуинтервал существования решения [4].

Рассмотрим первое уравнение системы (1) с соответствующим начальным условием:

$$\begin{cases} \dot{x}(t) = -(\beta y(t) + \delta)x(t) + v(t), \\ x(0) = x_0. \end{cases}$$

Проинтегрируем его методом вариации произвольной постоянной. Находим решение:

$$x(t) = e^{-\int\limits_0^t (\beta y(t)+\delta)d\xi} \left(x_0 + \int\limits_0^t e^{\int\limits_0^s (\beta y(t)+\delta)d\xi} v(t)ds \right). \qquad (2)$$

Отсюда видно, что $x(t) > 0$ для всех $t \in \Delta$.
Проводя аналогию, выпишем второе уравнение системы (1) с соответствующим начальным условием:

$$\begin{cases} \dot{y}(t) = (\beta x(t) + \gamma)y(t) + \delta x(t), \\ y(0) = y_0. \end{cases}$$

Тогда решение $y(t)$ имеет вид:

$$y(t) = e^{\int\limits_0^t (\beta x(t)-\gamma)d\xi} \left(y_0 + \int\limits_0^t e^{-\int\limits_0^s (\beta x(t)-\gamma)d\xi} \delta x(t)ds \right). \qquad (3)$$

Видно, что $y(t) > 0$ при всех $t \in \Delta$.

Оценим сверху выражение, стоящее в правой части уравнения (2). Имеет место неравенство:

$$x(t) \leqslant x_0 + v_{max}T.$$

Следовательно, неравенства $0 < x(t) < x_{max}$ выполняются для всех $t \in \Delta$. Проведем подобные рассуждения для выражения, стоящего в правой части уравнения (3):

$$y(t) < e^{\beta x_{max}T} y_0 + x_{max}Te^{\beta x_{max}T}.$$

Таким образом, неравенства $0 < y(t) < y_{max}$ выполняются для всех $t \in \Delta$.

Из приведённых оценок и леммы (§14, глава 4) [2] следует, что решение существует на всем отрезке. Обоснование теоремы завершено. ■

Для исследуемой системы (1) рассмотрим задачу минимизации функционала вида:

$$J(v) = \int\limits_0^T [y(t) + \sigma v(t)]dt \to \min_{v(\cdot)\in D(T)}. \tag{4}$$

Этот функционал показывает, что в условиях финансового кризиса необходимо минимизировать количество нестабильных фирм, а также количество вкладываемых денежных средств на отрезке времени $[0, T]$.

Из результатов работ [3, 7] вытекает существование в задаче (1),(4) оптимального решения — оптимального управления $v_*(t)$, $t \in [0, T]$ и соответствующей ему оптимальной траектории $(x_*(t), y_*(t)), t \in [0, T]$.

3 Принцип максимума Понтрягина

Для анализа задачи оптимального управления (1), (4) применим принцип максимума Понтрягина [5]. В соответствии с ним для оптимального управления $v_*(t)$ и оптимальной траектории $(x_*(t), y_*(t))$ существует нетривиальное решение $(\psi_*(t), \varphi_*(t))$ сопряженной системы:

$$\begin{cases} \dot{\psi}(t) = \beta y_*(t)\psi(t) + \delta\psi(t) - \beta y_*(t)\varphi(t) - \delta\varphi(t), \\ \dot{\varphi}(t) = \beta x_*(t)\psi(t) - \beta x_*(t)\varphi(t) + \gamma\varphi(t) + 1, \\ \psi(T) = 0, \quad \varphi(T) = 0, \end{cases} \tag{5}$$

для которого оптимальное управление $v_*(t)$ имеет вид:

$$v_*(t) = \begin{cases} v_{max} & , L(t) > 0, \\ \forall v(t) \in [0, v_{max}], & L(t) = 0, \\ 0, & L(t) < 0. \end{cases} \tag{6}$$

Здесь $L(t) = \psi(t) - \delta$, $t \in [0, T]$ — функция переключений, поскольку от её поведения зависит вид управления $v_*(t), t \in [0, T]$.

Системы (1), (5) и соотношение (6) образуют краевую задачу принципа максимума. Изучим ее более подробно.

Имеет место следующее утверждение.

Лемма 1. *Существует момент времени $\theta_* \in [0, T)$, что на интервале $(\theta_*, T]$ оптимальное управление $v_*(t)$ принимает значение нуль.*

Доказательство. Для функции переключения справедливо соотношение $L(T) = -\delta < 0$. Поскольку $L(t)$ является непрерывной функцией, то существует значение $\theta_* \in [0, T)$, что на интервале $(\theta_*, T]$ она отрицательна. Тогда из формулы (6) следует, что $v_*(t)$ равна нулю. Доказательство завершено. ■

Перепишем сопряженную систему (5) в виде:

$$\begin{cases} \dot{\psi}(t) = (\beta y_*(t) + \delta)(\psi(t) - \varphi(t)), \\ \dot{\varphi}(t) = \beta x_*(t)(\psi(t) - \varphi(t)) + (\gamma\varphi(t) + 1). \end{cases} \tag{7}$$

Введем в рассмотрение вспомогательные функции:

$$\begin{cases} G(t) = \psi(t) - \varphi(t), \\ P(t) = \gamma\varphi(t) + 1. \end{cases} \tag{8}$$

Тогда перепишем систему (7) в виде системы для функции $L(t)$ и вспомогательных функций $G(t)$, $P(t)$:

$$\begin{cases} \dot{L}(t) = (\beta y_*(t) + \delta)G(t), \\ \dot{G}(t) = (\beta y_*(t) - \beta x_*(t) + \delta)G(t) - P(t), \\ \dot{P}(t) = \gamma\,\beta\,x_*(t)\,G(t) + \gamma P(t), \\ L(T) = -\delta, G(T) = 0, P(T) = 1. \end{cases} \tag{9}$$

Исследуем функцию переключения $L(t)$. Для нее установим справедливость следующего утверждения.

Лемма 2. *Функция переключения $L(t)$ не может обратиться в нуль на интервале $\Delta \subset [0, T]$.*

Доказательство. Предположим противное. Пусть существует интервал $\Delta \subset [0, T]$ на котором $L(t) = 0$. Следовательно, $\dot{L}(t) = 0$ для $t \in \Delta$. Тогда из первого уравнения системы (9) следует, что $G(t) = 0$, и значит $\dot{G}(t) = 0$ для $t \in \Delta$. Из второго уравнения системы (9) находим, что функция $P(t) = 0$. Откуда следуют равенства $L(t) = G(t) = P(t) = 0$ для $t \in \Delta$. Из линейности и однородности системы (9) вытекает, что $L(t) = G(t) = P(t) = 0$ на всем отрезке $[0, T]$, что противоречит начальным условиям. Наше предположение неверно. Функция $L(t)$ не может обратиться в нуль на интервале $\Delta \subset [0, T]$. Доказательство леммы завершено. ■

Далее, имеет место одно из основных утверждений.

Лемма 3. *Функция переключения $L(t)$ обращается в нуль не более, чем в двух точках отрезка $[0, T]$.*

Доказательство. В системе (9) выполним замену переменных: $\eta(t) = G(t), \chi(t) = P(t) + h(t)G(t)$, где $h(t)$ — функция, которую нужно найти. Имеем:

$$\begin{cases} \dot{L}(t) = (\beta y_*(t) + \delta)\eta(t), \\ \dot{\eta}(t) = (\beta(y_*(t) - x_*(t)) + \delta + h(t))\eta(t) - \chi(t), \\ \dot{\chi}(t) = (\beta\gamma x_*(t) - \gamma h(t) + \dot{h}(t) + (\beta y_*(t) - x_*(t)) + \\ +\delta)h(t) + h^2(t))\eta(t)(h(t) - \gamma)\chi(t). \end{cases} \tag{10}$$

Выберем функцию $h(t)$ так, чтобы обращалось в нуль выражение при $\eta(t)$ в последнем уравнении. Тогда система (10) перепишется в виде:

$$\begin{cases} \dot{L}(t) = (\beta\bar{y}(t) + \delta)\eta(t), \\ \dot{\eta}(t) = (\beta(\bar{y}(t) - \bar{x}(t)) + h(t) + \delta)\eta(t) - \chi(t), \\ \dot{\chi}(t) = -(h(t) - \gamma)\chi(t). \end{cases} \tag{11}$$

Уравнение для $h(t)$ выглядит следующим образом:

$$\dot{h}(t) + (\beta(y_*(t) - x_*(t)) + \delta - \gamma)h(t) + h^2(t) + \beta\gamma x_*(t) = 0. \quad (12)$$

Пусть $h(t)$ — решение (12), которое определено на наибольшем полуинтервале $\Delta \subseteq [0, T]$. Покажем, что уравнение (12) имеет решение, определенное на всем отрезке $[0, T]$. Предположим противное, то есть пусть произвольное решение $h(t)$ системы (12) определено на интервале $[0, t_1)$, $t_1 \in (0, T]$, который является наибольшим интервалом существования этого решения. Тогда для решения $h(t)$ из леммы (§14, глава 4) [2]. вытекает соотношение

$$\lim_{t \to t_1 - 0} |h(t)| = +\infty.$$

Из него следует существование числа $\rho > 0$ и значения $t_0 \in [0, t_1)$, для которых справедливо включение $h(t) \in \Lambda$ при всех $t \in [t_0, t_1)$, где $\Lambda = \{q \in \mathbb{R}^1 \colon |q| \geqslant \rho\}$.

Вычислим на множестве Λ производную функции $|h(t)|$ в силу уравнения (12):

$$\frac{d}{dt}|h(t)| = \frac{h(t)\dot{h}(t)}{|h(t)|} = -[\beta(\bar{y}(t) - \bar{x}(t)) + (\delta - \gamma)]\frac{h^2(t)}{|h(t)|} - \\ - \frac{h^3(t)}{|h(t)|} - \beta\gamma\bar{x}(t)\frac{h(t)}{|h(t)|}. \quad (13)$$

Оценим сверху полученное соотношение (13):

$$\frac{d}{dt}|h(t)| \leqslant [(\beta\bar{y}(t) + \delta) + \beta\bar{x}(t) + \gamma]|h(t)| + h^2(t) + \beta\gamma\bar{x}(t) \leqslant \\ \leqslant |h(t)|^2 + [(\beta y_{max} + \delta) + (\beta x_{max} + \gamma)]|h(t)| + \beta\gamma x_{max}. \quad (14)$$

Введем следующие обозначения:

$$A = (\beta y_{max} + \delta) + (\beta x_{max} + \gamma), \quad B = \beta\gamma x_{max}.$$

Учитывая их, перепишем дифференциальное неравенство (14):

$$\frac{d}{dt}|h(t)| \leqslant |h(t)|^2 + A|h(t)| + B, \quad t \in [t_0, t_1).$$

Рассмотрим уравнение:

$$K^2 - AK + B = 0. \tag{15}$$

Определим знак его дискриминанта:

$$D = ((\beta y_{max} + \delta) + (\beta x_{max} + \gamma))^2 - \\ - 4\beta\gamma x_{max} > (\beta x_{max} + \gamma)^2 - 4\beta\gamma x_{max} = (\beta x_{max} - \gamma)^2 \geqslant 0.$$

Значит, величина дискриминанта положительна.

Введем функцию Ляпунова $V(h) = |h| + K_0$, где K_0 — наибольший корень уравнения (14). Тогда имеем

$$K_0 = \frac{A + \sqrt{A^2 - 4B}}{2},$$

Легко заметить, что $2K_0 > A$.

Перепишем теперь дифференциальное неравенство (14) для функции $V(h)$:

$$\frac{d}{dt}V(h(t)) \leqslant (V(h(t)) - K_0)^2 + A(V(h(t)) - K_0) + B,$$

или после преобразований

$$\frac{d}{dt}V(h(t)) \leqslant V^2(h(t)) - (2K_0 - A)V(h(t)) + (K_0^2 - AK_0 + B).$$

Учитывая, что $K_0^2 - AK_0 + B = 0$, окончательно находим:

$$\frac{d}{dt}V(h(t)) \leqslant V^2(h(t)) - (2K_0 - A)V(h(t)), \quad t \in [t_0, t_1). \tag{16}$$

Рассмотрим вспомогательную задачу Коши:

$$\begin{cases} \dot{\xi}(t) = \xi^2(t) - (2K_0 - A)\xi(t), & t \in [0, T], \\ \xi(0) = \xi_0, \quad \xi_0 > K_0 + \rho. \end{cases} \tag{17}$$

Видно, что для значения ξ_0 выполняется неравенство $\xi_0 > 2K_0 - A$.

Найдем решение задачи Коши (17). Для этого решим уравнение Бернулли. Получаем:

$$\xi(t) = \left(\left[\frac{1}{\xi_0} - \frac{1}{2K_0 - A} \right] e^{(2K_0 - A)t} + \frac{1}{2K_0 - A} \right)^{-1}, t \in [0, T].$$

Из неравенства $\xi_0 > 2K_0 - A$ имеем отрицательность выражения в квадратных скобках. Поэтому функция $\xi(t)$ является положительной и возрастающей на интервале $[0, T]$. Тогда находим неравенство:

$$\xi(t) < \xi(T), \quad t \in [0, T).$$

Поэтому из дифференциального неравенства (16) вспомогательной задачи Коши (17) и теоремы Чаплыгина (§1, глава 1) [6], при условии $\xi_0 = V(h_0) = |h_0| + K_0$, имеем неравенства:

$$|h(t)| < \xi(t) - K_0 \leqslant \xi(T) - K_0, \quad t \in \Delta.$$

Тогда, в силу следствия из леммы (§14, глава 4) [2], решение $h(t)$ уравнения (12) существует на всем отрезке $[0, T]$. Поэтому система уравнений (11) также определена на $[0, T]$.

Покажем, что $L(t)$ имеет не более двух нулей. Пусть функция переключения $L(t)$ обращается в нуль по крайней мере в трех точках. Тогда из [9] следует, что $G(t)$ имеет по крайней мере два нуля. Следовательно, и $\eta(t)$ обращается в нуль по крайней мере в двух точках τ_1 и τ_2: $0 \leqslant \tau_1 \leqslant \tau_2 < T$. Проинтегрируем выражение для $\eta(t)$ из системы (11), откуда

получаем:

$$0 = -\int\limits_{\tau_1}^{\tau_2} \chi(s) e^{-\int\limits_{\tau_1}^{s}((\beta(\bar{x}-\bar{y})+h+\delta))d\xi} ds,$$

следовательно, существует $r \in (\tau_1, \tau_2)$ такое, что $\chi(r) = 0$. Тогда из третьего уравнения системы (11) следует, что $\chi(t) = 0$. Следовательно, верны равенства $\chi(t) = G(t) = P(t) = 0$, откуда следует, что $\eta(t) = 0$ и $L(t) = 0$, $t \in [0, T]$, а тогда по лемме 2 такого быть не может. Значит наше предположение неверно. Следовательно, функция переключения $L(t)$ имеет не более двух нулей. Утверждение доказано. ■

Справедливо утверждение, вытекающее из формулы (6) и лемм $1, 3$.

Теорема 2. *Оптимальное управление $v_*(t)$ на отрезке $[0, T]$ имеет следующий вид*:

$$v_*(t) = \begin{cases} 0, & \text{если} \quad 0 \leqslant t < \theta_1^*, \\ v_{max}, & \text{если} \quad \theta_1^* \leqslant t \leqslant \theta_2^*, \\ 0, & \text{если} \quad \theta_2^* < t \leqslant T, \end{cases}$$

где $\theta_1^*, \theta_2^* \in (0, T)$ — моменты переключений.

На его основе можно сделать вывод о том, что для достижения к заданному моменту времени T минимального значения функционала, требуется в зависимости от соотношений между параметрами модели на интервале времени $[\theta_1^*, \theta_2^*] \subset [0, T]$ вкладывать максимально возможное количество денежных средств в стабильную группу фирм, и, соответственно, ничего не вкладывать на остальных интервалах времени.

4 Результаты численного исследования

В настоящей работе средствами программы Maple13 реализована численная схема решения задачи оптимального управ-

ления (1),(4), которая обеспечивает получение результатов сопоставимых с моделируемой системой.

Численный алгоритм решения системы дифференциальных уравнений (1) получен непосредственным применением метода Рунге-Кутты четвертого порядка точности.

Введем множество $\Lambda(T) = \{\theta = (\theta_1, \theta_2) \in R^2 \colon 0 \leqslant \theta_1 \leqslant \theta_2 \leqslant T\}$. Для любой точки $\theta \in \Lambda(T)$ определим управление:

$$v_\theta(t) = \begin{cases} 0, & \text{если} \quad 0 \leqslant t < \theta_1, \\ v_{max}, & \text{если} \quad \theta_1 \leqslant t \leqslant \theta_2, \\ 0, & \text{если} \quad \theta_2 < t \leqslant T. \end{cases}$$

Подставляя его в исходную систему (1), получаем оптимальную траекторию $(x_\theta(t), y_\theta(t))$. Далее, с помощью подстановки функции $y_\theta(t)$ в функционал (4) вычисляем значение функционала $J(v_\theta)$. Таким образом, возникает функция двух переменных $F(\theta_1, \theta_2) = J(v_\theta)$, $\theta \in \Lambda(T)$. В результате, задача оптимального управления (1),(4) свелась к задаче конечномерной условной оптимизации:

$$F(\theta_1, \theta_2) \to \min_{(\theta_1, \theta_2) \in \Lambda(T)}.$$

Для реализации численных расчетов задаем значения параметров модели:
$T = 2$;
$t_1 = 0{,}3$;
$t_2 = 0{,}7$;
$\beta = 1$;
$\gamma = 0{,}2$;
$\delta = 0{,}2$;
$v_{max} = 10$.
Тогда график оптимального управления $v_*(t)$ имеет вид, представленный на рисунке 1. Графики соответствующих оптимальных траекторий $x_*(t), y_*(t)$ изображены на рисунке 2.

Рассмотрим траекторию $(x(t), y(t))$ системы (1), расположенную на одной координатной плоскости, в зависимости от

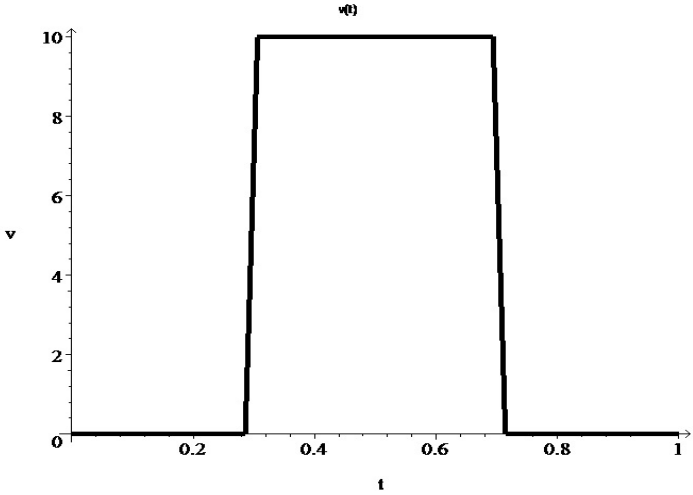


Рис. 1. График зависимости оптимального управления от времени.

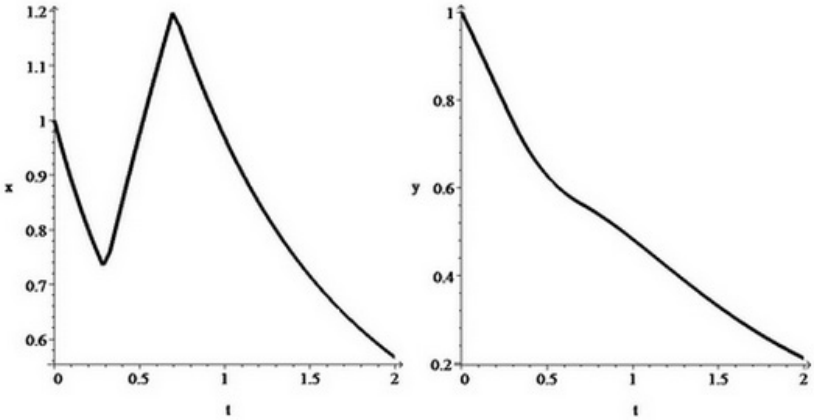


Рис. 2. Графики зависимости оптимальных траекторий от времени.

постоянного размера вложения денежных средств $v(t)$, где $x(t)$ — сплошная линия, $y(t)$ — штриховая линия. На рисун-

ке 3 продемонстрирован график зависимости изменения коли-

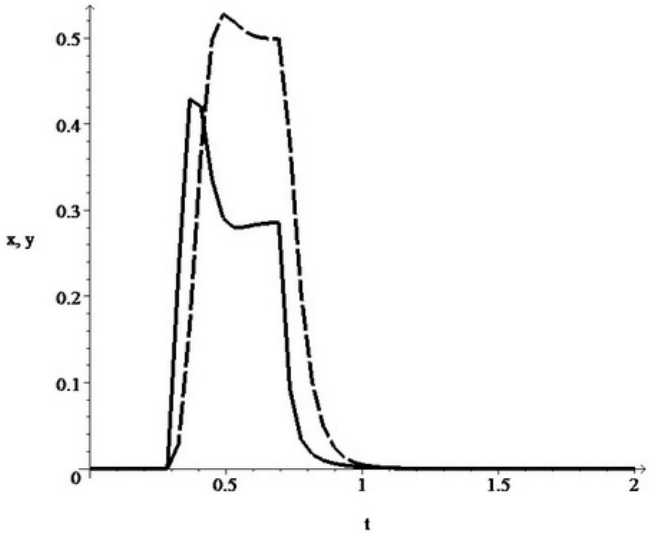


Рис. 3. Графики зависимости изменения количества фирм $x(t), y(t)$ от времени при $v(t) = 0$.

чества фирм от времени при $v(t) = 0$. Далее, на рисунке 4 представлен график зависимости изменения количества фирм от времени при управлении $v(t) = 4$. Наконец, график зависимости изменения количества фирм от времени при $v(t) = 10$ изображен на рисунке 5.

Анализ зависимости изменения количества фирм от размера вложения денежных средств привел к следующему выводу, отражающему адекватность модели. Количество стабильных фирм $x(t)$ возрастает и количество нестабильных фирм $y(t)$ уменьшается при увеличении размера вложения денежных средств $v(t)$ в данную систему.

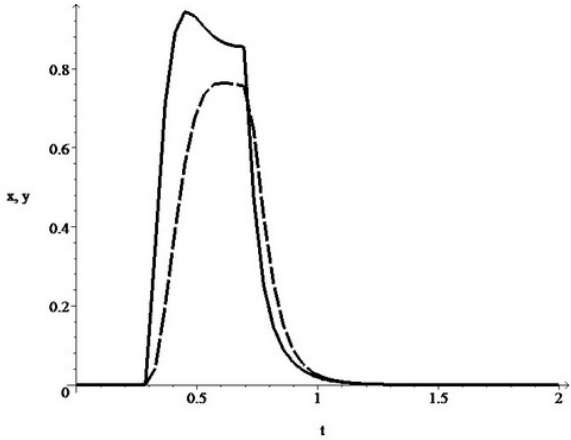


Рис. 4. Графики зависимости изменения количества фирм $x(t), y(t)$ от времени при $v(t) = 4$.

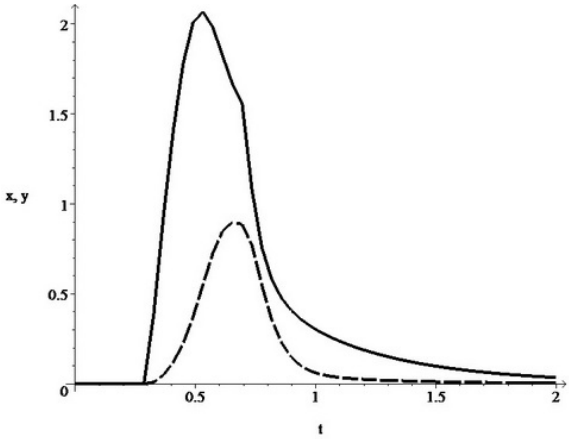


Рис. 5. График зависимости изменения количества фирм $x(t), y(t)$ от времени при $v(t) = 10$.

5 Заключение

В данной работе была разработана и исследована математическая модель, описывающая динамику взаимодействия фирм

в условиях финансового кризиса (на примере взаимодействия двух групп фирм); получена численная схема решения задачи оптимального управления с целью минимизации количества нестабильных фирм, а также затрат в виде вложения денежных средств.

Эта математическая модель финансового кризиса представляет собой большое упрощение и имеет недостатки по сравнению с чертами реальной экономики. Однако следует учесть сложность этой системы, которая состоит в том, что построение математической модели разумной точности, которую можно использовать для количественного прогнозирования, представляет собой задачу огромной сложности. Основной задачей данной работы было продемонстрировать, что такое моделирование возможно, показать взаимосвязь факторов, имеющие принципиальное значение и влияние в экономике, а также метод решения задачи оптимального управления данного типа.

Список литературы

[1] Андреева Е. А., Цирулева В. М. Вариационное исчисление и методы оптимизации. - М.: Высшая школа, 2006.

[2] Демидович Б. П. Лекции по математической теории устойчивости. - М.: Наука, 1967

[3] Ли Э. Б., Маркус Л. Основы теории оптимального управления. - М.: Наука, 1972

[4] Понтрягин Л. С. Обыкновенные дифференциальные уравнения. - М.: Наука, 1974

[5] Понтрягин Л. С., Болтянский В. Г., Гамкрелидзе Р. В. Мищенко Е. Ф. Математическая теория оптимальных процессов. - М.: Наука, 1983.

[6] Тихонов А. Н., Васильева А. Б., Свешников А. Г. Дифференциальные уравнения. - М.: Наука 1998.

[7] Флеминг У. Ришел Р. Оптимальное управление детерминированными и стохастическими системами. - М. Мир, 1978

[8] Andrei Korobeinikov. Financial crisis: An attempt of mathematical modelling. Appl. Math. Lett. 22(12): 1882-1886 (2009)

[9] Dmitruk A. V. A generalized estimate on the number of zeros for solution of a class of linear differential equations // SIAM Journal of Control and optimization. 1992 Vol30, №5. P 1087-1091.

УДК 517.956.227

Об одной смешанной задаче для уравнения теплопроводности, приводящей к спектральной задаче с граничными условиями третьего рода, одно из которых содержит спектральный параметр

© 2012 г. Д. А. Гуляев

gulden@cs.msu.ru

Кафедра функционального анализа и его применений

Рассмотрим смешанную задачу для уравнения теплопроводности

$$U_t(x,t) = a^2 U_{xx}(x,t) \tag{1}$$

в области $D = \{(x,t); \quad 0 < x < 1, \quad 0 < t < T\}$ с граничными условиями

$$U'_x(o,t) = bU(0,t), \quad U'_x(1,t) = -dU'_t(1,t), \quad b \neq 0, \quad d > 0 \tag{2}$$

и начальным условием

$$U(x,0) = f(x). \tag{3}$$

возникающую, например при моделировании процесса теплопереноса в однородном стержне, на одном из концов которого задается линейная взаимосвязь значений теплового потока и температуры, а на другом конце помещена сосредоточенная теплоемкость [1]. Решая задачу (1)-(3) методом разделения переменных, получим спектральную задачу

$$X''(x) + \lambda X(x) = 0, \quad 0 < x < 1, \tag{4}$$

$$X'(0) = bX(0), \quad X'(1) = d\lambda X(1) \tag{5}$$

Спектральная задача (4)-(5), рассмотренная в работе [2], не имеет нулевого собственного значения, поэтому общее решение уравнения (4) в случае $\lambda \neq 0$, удовлетворяющее первому граничному условию, можно записать в виде:

$$X(x) = \frac{b \sin \sqrt{\lambda} x}{\sqrt{\lambda}} + \cos \sqrt{\lambda} x$$

Присоединив второе граничное условие для этой функции, получим характеристическое уравнение задачи (4)-(5)

$$(1 + bd)\sqrt{\lambda} \sin \sqrt{\lambda} = (b - d\lambda) \cos \sqrt{\lambda}. \qquad (6)$$

Если $bd = -1$, то уравнение (6) имеет один отрицательный корень $\lambda = -1/d^2$ и бесконечное множество положительных корней $\lambda = [\pi/2 + \pi(n-1)]^2, n = 1, 2, 3, ...$ В случае $bd > -1$, $b > 0$, все корни уравнения (6) – положительные. При значениях параметров удовлетворяющих условиям $bd < -1$ или $bd > -1$, $b < 0$ также имеется одно отрицательное собственное значение, а все остальные собственные числа расположены в положительной части действительной оси.

Присвоим нулевой индекс любому собственному значению, а все остальные занумеруем в порядке возрастания. Собственные функции задачи (4)-(5) определяются формулой:

$$X_n(x) = \sqrt{2} \left[\frac{b \sin \sqrt{\lambda_n} x}{\sqrt{\lambda_n}} + \cos \sqrt{\lambda_n} x \right], \quad n = 0, 1, 2, ...$$

$\longleftrightarrow$ (в случае $\lambda_n < 0$ синус и косинус гиперболические). Функции биортонормированной системы $\{\psi_m(x)\}, m = 1, 2, 3, ...$, к системе $\{X_n(x)\}, n = 1, 2, 3, ...$, имеют вид

$$\psi_m(x) = \left[X_m(x) - \frac{X_m(1)}{X_0(1)} X_0(x) \right] / \left[\int_0^1 X_m^2(x) dx + d X_m^2(1) \right].$$

Используя результат, полученный в теореме 2 статьи [2], можно доказать следующе утверждение

Теорема 1. *Пусть $f(x)$ - функция из класса Гельдера* $^{\alpha}[0,1], \alpha > 0$. *Тогда решение задачи (1)-(3) можно представить в виде билинейного ряда*

$$U(x,t) = \sum_{n=0}^{\infty} \left[df(1)X_n(1) + \int_0^1 f(t)X_n(t)dt \right] \times$$
$$\times \left[dX_n^2(1) + \int_0^1 X_n^2(t)dt \right]^{-1} X_n(x)e^{-a^2\lambda_n t}. \qquad (7)$$

Доказательство. Согласно результату теоремы 2 из работы [2] любую функцию $f(x) \in C^{\alpha}[0,1], \alpha > 0$ можно разложить в равномерно сходящийся на отрезке [0,1] ряд

$$f(x) = \sum_{n=0}^{\infty} \left[df(1)X_n(1) + \int_0^1 f(t)X_n(t)dt \right] \times$$
$$\times \left[dX_n^2(1) + \int_0^1 X_n^2(t)dt \right]^{-1} X_n(x). \qquad (8)$$

Применяя метод разделения переменных, решение задачи (1)-(3) формально записывается в виде ряда (7). При $t > 0$ этот ряд можно сколь угодно раз почленно дифференцировать, а в силу известного признака Дирихле-Абеля для равномерной сходимости рядов равенство (7) при $t \to 0 + 0$ переходит в соотношение (8).

В связи с рассматриваемым в статье вопросом отметим работы [3,4].

Автор благодарит академика Моисеева Е. И. и Капустина Н. Ю., за постановку задачи и полезное обсуждение результатов.

Литература

1. Тихонов А.Н., Самарский А.А. Уравнения математической физики. М.: Наука 1972.

2. Гуляев Д.А. О равномерной сходимости спектарльных разложений для спектральной задачи с граничными условиями третьего рода, одно из которых содержит спектральный параметр. //Дифференциальные уравнения, 2011, Т.47, №10, С.1503-1507

3. Капустин Н.Ю., Моисеев Е.И. О сходимости спектральных разложений функций из класса Гёльдера для двух задач со спектральным параметров в граничном условии. //Дифференциальные уравнения, 2000, Т.36, №8, С.1069-1074.

4. Капустин Н.Ю. О равномерной сходимости в классе C1 ряда Фурье для спектральной задачи с квадратом спектрального параметра в граничном условии. //Дифференциальные уравнения, 2011, Т.47. №10. С.1394-1399.

УДК 004.023

Об одном подходе к решению задачи поиска нот музыкальных произведений

© 2012 г. Ю. С. Корухова, М. В. Мытрова

yulia@cs.msu.ru, Serpuhovichok@yandex.ru

Кафедра алгоритмических языков

Работа посвящена проблеме поиска нот музыкальных произведений, хранящихся в электронных библиотеках в формате MusicXML. Данный формат является одним из самых распространенных и поддерживается многими программами - нотными редакторами. Большинство современных поисковых систем работают с библиотеками аудио- и видеофайлов, содержащих записи музыкальных произведений, а поиск, как правило, ведется только по названию и автору композиции. В данной работе исходными данными для поиска является фрагмент нотной записи произведения, а библиотека произведений представляет собой набор MusicXML файлов, содержащих ноты. Данный формат позволяет хранить ноты как структурированную информацию, однако традиционные методы поиска оказываются к нему неприменимы из-за специфики предметной области: при поиске нот результатом может быть файл, даже не имеющий общих фрагментов с запросом, например, из-за того, что мелодия записана в другой тональности. В работе предложен новый подход к решению задачи поиска, учитывающий специфику предметной области.

1 Введение

В современный век информационных технологий практически в любой области данные начинают храниться и распространяться в электронном виде. Музыкальная область – не исключение. Множество аудио, видео и текстовой музыкальной

информации хранится на просторах сети Интернет. С появлением библиотек файлов возникает задача поиска информации в них. В рамках данной работы мы рассматриваем файлы, содержащие нотные записи музыкальных произведений. Такие файлы, как правило, создаются с помощью программ – нотных редакторов. Двумя наиболее универсальными форматами, в которых хранятся нотные записи, являются pdf и MusicXML. Несмотря на свою распространенность, формат pdf имеет серьезный недостаток: ноты в нем не модифицируются, тогда как формат MusicXML [13] позволяет редактировать ноты, так как он поддерживается более, чем 150 программами-редакторами (MuseScore, Finale, Guitar Pro и др.). Внесение изменений в таком редакторе, например, транспонирование мелодии в другую тональность, может быть выполнено автоматически и не представляет проблем для пользователя.

В настоящее время музыкальные поисковые системы ориентированы только на поиск нот по названию и фамилии автора [15, 16]. Задача поиска музыкальных файлов на основе анализа их содержимого решалась только для аудиофайлов. При этом поиск поводился либо по фрагменту мелодии (акустическому отпечатку) [12], либо, как в поисковой системе Musipedia [14], по фрагменту нотной записи, мелодическому контуру, ритму. Последний подход представляет особый интерес, так как для корректной работы такого метода поиска существенной является информация о предметной области, учитывается тональность произведений, несоответствие ритма и других характеристик. Однако поиск проводится для аудиофайлов в формате midi.

Исходными данными для предлагаемого в данной работе метода поиска является запись фрагмента произведения в виде MusicXML файла. В качестве результата представляется последовательность файлов, содержащих нотные записи, и их степени соответствия запросу. Запись нот в виде MusicXML файла представляет собой структурированный документ, однако

традиционные методы поиска, разработанные для структурированных файлов [8], оказываются неприменимыми. В методе должна учитываться специфика предметной области. Например, одна и та же мелодия, записанная в разных тональностях, представляется файлами с (формально) разным содержимым. При записи мелодии в более быстром темпе можно поменять информацию о темпе произведения, а можно записать его нотами другой длительности, что также приведет к появлению файлов с разным содержимым для одной мелодии. Поэтому необходимо специальное представление нотных записей, позволяющее учесть эти особенности.

В информационном поиске мы сталкиваемся с задачей сравнения двух фрагментов (запроса и библиотечного файла), которые в чем-то являются похожими. Требуется ответить на вопрос о том, являются ли они представлением одного произведения. Подобная ситуация возникает в математике при проведении доказательства методом математической индукции. При выполнении шага индукции имеется предположение, которое считается истинным, и требуется доказать истинность заключения индукции, которое похоже на гипотезу. Одним из методов, с помощью которого можно автоматизировать такое доказательство, является метод волновых правил [1,2]. Метод предлагает выделить общие части (основу) гипотезы и заключения и отметить различия — выполнить аннотирование выражений. Затем применять в доказательстве только те правила, которые уменьшают различия в соответствии с некоторой мерой. При успешном исходе доказательства все различия будут устранены, и истинность заключения индукции будет доказана. Для использования данного метода при поиске необходимо определить правила аннотирования запроса и библиотечного файла, определить правила, с помощью которых производится преобразования и задать меру.

В следующих разделах работы будут рассмотрены основные составляющие предлагаемого метода и его реализация. В

разделе 2 приводится информация о внутреннем представлении нотных записей, раздел 3 рассказывает о сравнении музыкальных фрагметов с использованием волновых правил. Реализация метода описана в разделе 4. В заключении сформулированы результаты работы.

2 Внутреннее представление нот

Предлагаемый метод поиска работает с файлами в формате MusicXML. Данный вид файлов является подмножеством XML файлов, в котором зафиксирован набор тегов разметки содержимого. Для описания информации о нотах и произведении введены, например, теги `<note>`, `<octave>`, `<pitch>`, `<duration>`, позволяющие выделить ноту, октаву, в которой она находится, высоту и длительность ноты соответственно. В данном формате переписывание имеющейся мелодии, например, на два тона выше приведет к изменению содержимого файла. Однако, с музыкальной точки зрения, мелодия остается той же. Поскольку задачей является именно нахождение похожих мелодий, необходимо внутреннее представление, которое было бы одинаковым для разных файлов, содержащих одну мелодию. Подходы к решению данной проблемы предложены в [3, 4, 6].

Музыкальный фрагмент представляется в виде списка нот и пауз. Ноты хранятся в формате: `(относительная высота, относительная длительность)`, паузы характеризуются только относительной длительностью.

Относительная высота ноты — это число, равное смещению (в тонах) относительно самой низкой ноты фрагментов (точки отсчета). При этом точка отсчета не обязательно окажется тоникой. Выбор такого способа расстановки относительных высот обусловлен тем, что в запросе пользователь может не указать тональность набранного им фрагмента. В этом случае могут возникнуть проблемы с автоматическим определением тоники, а определение самой низкой ноты фрагментов

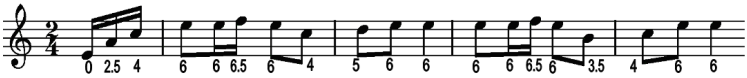


Рис. 1. Пример вычисления относительной высоты нот.

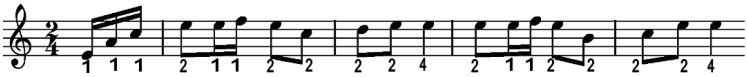


Рис. 2. Пример вычисления относительной длительности нот.

затруднений не вызывает.

Рассмотрим расстановку относительных высот на примере фрагмента «Польки» С. Рахманинова (рисунок 1). Здесь нижнее ми получает относительную высоту 0, а все остальные относительные высоты отсчитываются от нее.

Относительная длительность ноты — это отношение длительности ноты к длительности самой короткой ноты фрагментов. Длительность самой короткой ноты фрагментов считаем равной 1. Если ноты одной высоты связаны лигой, то во внутреннем представлении они считаются одной нотой, и их длительности складываются. Относительные длительности пауз вычисляются одновременно с относительными длительностями нот по такому же правилу. Расстановка относительных длительностей для фрагмента «Польки» С. Рахманинова приведена на рисунке 2.

Данное представление позволяет получить одинаковую формальную запись для одинаковых мелодий, таким образом, упрощая их сравнение. Однако одного лишь изменения представления оказывается недостаточно для решения задачи поиска. Наличие несовпадающих внутренних представлений не обязательно означает, что рассматриваемые музыкальные записи не относятся к одному произведению. Во–первых, несовпадение может быть вызвано разными длинами запроса и файла из нотной библиотеки. Во-вторых, при формировании запроса

по слуху, в нем могут отразиться особенности, характерные для конкретного инструмента, а при наличии в библиотеке нот для другого инструмента, запись будет выглядеть по–другому. И, наконец, в некоторых нотах запроса могут появиться случайно сделанные пользователем ошибки. Поэтому поиска мелодии по точному соответствию на практике оказывается недостаточно, требуется искать также и похожие, но не совпадающие с запросом композиции.

3 Сравнение музыкальных фрагментов

Сравнение музыкальных фрагментов, уже переведенных во внутреннее представление, происходит в три этапа. Сначала выполняется поиск наибольшей общей подпоследовательности нот и аннотация фрагментов. Затем выполняются преобразования с помощью волновых правил. На последнем этапе вычисляется оценка похожести фрагментов.

3.1 Аннотирование выражений

Чтобы выяснить, имеет ли смысл доказывать похожесть двух фрагментов, они проходят предварительную проверку. Пусть у нас есть фрагменты F_1 и F_2. Найдем их самую длинную общую подпоследовательность, используя алгоритм, описанный в [7]. Длиной нотной последовательности будем называть количество нот и пауз в ее внутреннем представлении. Если длина самой длинной общей подпоследовательности нот (Longest Common Subsequences of Notes или LCSN) не превышает половины длины меньшего из фрагментов, то похожесть фрагментов приравнивается к нулю и дальнейшие действия не производятся. В противном случае составляется аннотированная формула

$$F_1[LCSN] = F_2[LSCN] \tag{1}$$

где $F[L]$ обозначает последовательность, содержащую L. Выражение

$$LCSN = LCSN \tag{2}$$

является основой, а все остальные последовательности символов отмечаются как волновой фронт. В процессе преобразования формулы (1) изменение основы не допускается, а волновые фронты можно изменять, перемещать или удалять в соответствии с правилами преобразования.

3.2 Использование волновых правил

Целью метода волновых правил является преобразование формулы (1) в формулу (2). Преобразования проводятся с помощью известных заранее правил переписывания. Правила строятся на основе закономерностей, известных из теории музыки и правил гармонии [10, 11], а также на основе практического опыта использования нотной поисковой системы. В данной работе применяются три категории правил: правила исправления ошибок, правила альтернативной записи и правила сокращения.

Правила исправления ошибок предназначены для того, чтобы устранить возможные ошибки, допущенные пользователем во время записи нот мелодии по слуху. Так, например, при подборе достаточно большого интервала (квинта, секста, септима) пользователь может ошибиться на полтона как в сторону уменьшения интервала, так и в сторону его увеличения. Поэтому такие расхождения в крайних нотах волновых фронтов, прилегающих к одинаковым частям основы, считаются опечаткой и приводятся к одной ноте.

Один и тот же музыкальный фрагмент можно записать и исполнить по–разному, при этом мелодия будет узнаваема. Такое явление можно встретить в вариациях и аранжировках. Для того, чтобы сравнивать различные вариации одной мелодии, вводятся *правила альтернативной записи*, преобразующие альтернативные способы записи фрагмента друг к другу.

Они предусматривают различные варианты расстановки пауз и использования неаккордовых звуков [10].

После применения правил, исправляющих ошибки, и правил альтернативной записи может увеличиться наибольшая общая подпоследовательность нот в заданных фрагментах. Так как при применении правил изменять основу нельзя, мы не можем добавить в нее появившиеся общие ноты, поэтому такие ноты удаляются. Но, как и в случае применения правил исправления ошибок, происходит удаление не всех общих нот волновых фронтов, а только тех, которые граничат с одинаковыми частями основы. Подобные преобразования отнесены к категории *правил сокращения*.

При работе поисковой системы в ее базе содержится большое количество правил, многие из которых можно применять как в прямом, так и в обратном направлении. Случайный выбор правил может зациклить процесс преобразования или сильно увеличить время его выполнения. Поэтому нужно определить меру, с помощью которой можно «измерять» количество оставшихся различий. На каждом шаге, после применения правила, мера должна уменьшаться. Процесс преобразования останавливается в одном из двух случаев: либо когда мера становится равной нулю, то есть все различия в выражениях устранены, либо когда ни одно из правил преобразования не применимо. При работе с музыкальными фрагментами в качестве меры предлагается использовать количество нот в волновом фронте. В процессе преобразования же выбираются и применяются только те правила, которые уменьшают длину волнового фронта.

3.3 Оценка похожести фрагментов

Похожесть фрагментов обратно пропорциональна длине волнового фронта, который остался после применения правил, а также зависит от количества примененных правил, поскольку правила применялись для устранения различий, то есть исход-

ные фрагменты не были одинаковыми. Для оценки похожести фрагментов F_1 и F_2 мы предлагаем формулу

$$similarity(F_1, F_2) = \frac{L_1 - TWFL_1 + L_2 - TWFL_2}{L_1 + L_2 + \alpha * Nrules}, \qquad (3)$$

где L_1 — длина фрагмента F_1,

L_2 — длина фрагмента F_2,

$TWFL_1$ — длина волнового фронта фрагмента F_1,

$TWFL_2$ — длина волнового фронта фрагмента F_2,

$Nrules$ — число примененных правил преобразования,

α — эвристический коэффициент.

Поскольку в музыкальной области нет формальной оценки сходства мелодий, в основном оно определяется на слух, хотя часто и основано на наличии общих фрагментов, эвристический коэффициент подбирался экспериментально. Чем больше значение этого коэффициента, тем большую роль имеет количество примененных правил преобразования. Практическое использование метода показало, что при значении коэффициента $\alpha = 0.5$ оценка совпадает с оценкой, сделанной специалистом.

Результатом вычисления похожести фрагментов может быть значение из отрезка $[0, 1]$, где 0 соответствует полному несовпадению фрагментов, а 1 – их полному совпадению. В действительности диапазон возможных значений меньше, так как сильно отличающиеся музыкальные фрагменты были устранены ранее, после определения самой длинной общей подпоследовательности нот.

После прохождения указанных этапов поиска получается список композиций, которые сортируются в порядке убывания значений, вычисленных по формуле (3), и объявляются результатами поиска.

4 Реализация метода

Предложенный метод поиска был реализован в прототипной системе, выполняющей поиск нотных записей. В качестве

входных данных система принимает файл MusicXML, содержащий запрос, в качестве результата на экране отображается список соответствующих запросу файлов с указанием степени их соответствия запросу. Система имеет web-интерфейс. В настоящее время поиск проводится для библиотеки файлов, хранящейся на локальной машине, но процесс добавления файлов в библиотеку автоматизирован, то есть библиотека является расширяемой. Одна из задач, которую, как правило, требуется решить поисковым системам, это индексирование содержимого библиотеки.

4.1 Индексирование

Проблема индексирования музыкальных файлов заключается в том, что нельзя однозначно разбить мелодию на части и построить полноценный словарь. Если при работе с текстовым файлом разбить его на отдельные слова не составляет труда, то разделить музыкальное произведение на осмысленные фрагменты автоматически затруднительно. Паузы и тактовые черты не всегда являются разделителями законченных музыкальных фрагментов, поэтому полагаться только на них нельзя. Для разбиения музыкального произведения используется так называемый метод «скользящего окна» [5]. Задается размер (суммарная длительность) окна и размер шага, определяющий степень перекрытия двух последовательно выделенных фрагментов. Затем окно проходит с заданным шагом от начала произведения до его конца, тем самым разбивая его на фрагменты, совпадающие по размеру с окном. Фрагменты музыкального произведения, содержащие репризы и вольты, проходятся окном столько раз, сколько это необходимо в соответствии с правилами игры по нотам.

Часть мелодии, выражающая законченную музыкальную мысль, в элементарной теории музыки называется периодом. Простейший тип периода состоит из восьми тактов. Период делится на две части по четыре такта, называющиеся предложе-

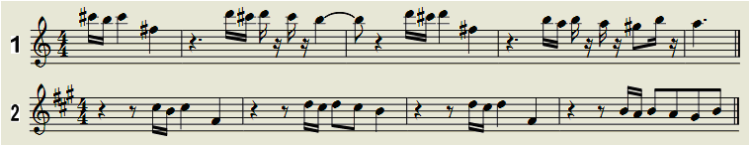


Рис. 3. Фрагменты «Europe - The Final Countdown», взятые из разных источников.

ниями [11]. Предложения, в свою очередь, можно разбить на двухтактные фразы [9]. Следуя такому принципу разделения мелодии, произведение проходится трижды – по одному разу окнами размером в 8, 4 и 2 такта. Полученные фрагменты переводятся во внутреннее представление и заносятся в словарь. В словаре они хранятся в виде инвертированного файла [8]. То есть для каждого фрагмента указывается список произведений, в которых он встречается, и число вхождений в каждое из этих произведений. В силу особенностей сравнения музыкальных фрагментов, при занесении в базу и подсчете вхождений фрагмента в произведение рассматривается не точное совпадение нового фрагмента и имеющегося в базе, а такое совпадение, при котором величина, полученная при вычислении формулы (3), не меньше порогового значения. В результате экспериментов для текущей программной реализации системы выбрано пороговое значение 0.9.

4.2 Пример работы системы

Рассмотрим применение волновых правил в системе при сравнении двух фрагментов мелодии «Europe - The Final Countdown», один из которых задан как запрос, второй хранится в словаре системы (рисунок 3). После приведения к относительной высоте и относительной длительности они будут выглядеть как показано на рисунке 4. Далее система находит наибольшую общую подпоследовательность нот и аннотирует фрагменты. Результат представлен на рисунке 5. После

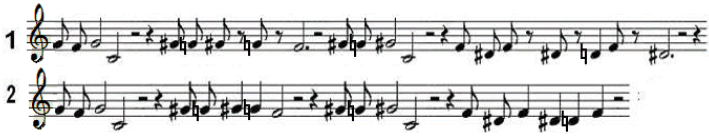


Рис. 4. Фрагменты «Europe - The Final Countdown», приведенные к относительной высоте и относительной длительности.

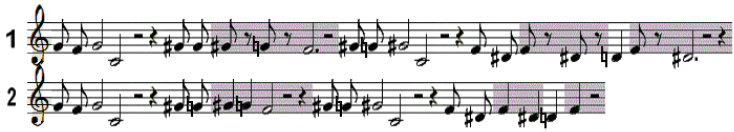


Рис. 5. Аннотированные фрагменты. На белом фоне – ноты основы, на сером фоне – волновой фронт.

аннотирования можно применять правила преобразования. К первому фрагменту можно несколько раз применить правило-альтернативу (рисунок 6). После пятикратного применения правила – альтернативы первый фрагмент будет выглядеть, как показано на рисунке 7. К полученным фрагментам можно применить правила сокращения (4) и (5).

$$[\ldots]aXYZc[\ldots] = [\ldots]aXFZc[\ldots] => \\ => [\ldots]aYc[\ldots] = [\ldots]aFc[\ldots] \quad (4)$$

$$[\ldots]X[pause] = [\ldots]Y[pause] => [\ldots]X = [\ldots]Y, \quad (5)$$

где $[\ldots]$ – любые последовательности нот (они не меняются данными правилами), a, c — последовательности нот основы, X, Y, Z, F – последовательности нот волнового фронта, $[pause]$ – паузы любой длительности, не обязательно одинаковой.

В итоге фрагменты примут вид, изображенный на рисунке 8. Так как в базе не осталось правил вывода, которые можно применить к этим фрагментам, процесс преобразования останавливается, не достигнув нулевой меры. По сравнению с состоянием фрагментов до применения правил (суммарная длина

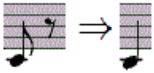


Рис. 6. Правило – альтернатива.

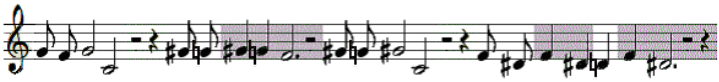


Рис. 7. Первый фрагмент после пятикратного применения правила – альтернативы.

волнового фронта равнялась 22) мера уменьшилась более, чем в четыре раза (теперь она равна 5).

4.3 Оценка работы метода

Для оценки поиска, выполняемого системой рассмотрим пример, предложенный в [4]. В базе системы, состоящей из 45610 фрагментов, находились следующие фрагменты произведения «Roslin Castle» (рисунок 9).

Последовательно был выполнен поиск первых четырех фрагментов, результаты работы представлены в таблице 1. Словом «ошибка» отмечены результаты, не являющиеся фрагментами «Roslin Castle». Из таблицы 1 можно увидеть, что минусом системы является неполная выдача: находятся не все возможные варианты, что хорошо видно на примере первого

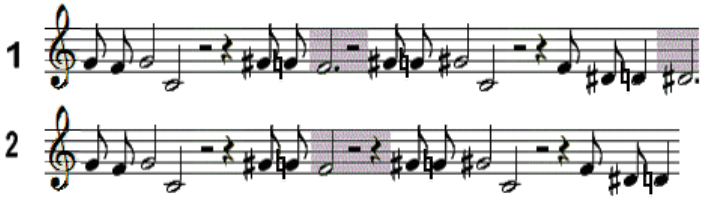


Рис. 8. Фрагменты после применения правил сокращения.

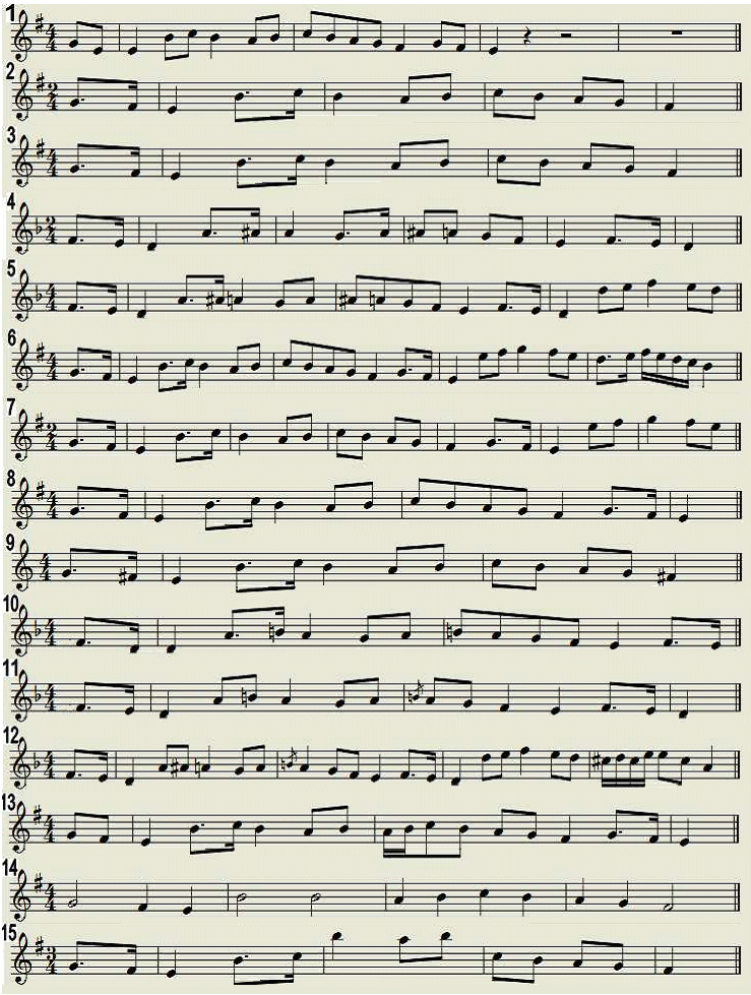


Рис. 9. Фрагменты произведения «Roslin Castle» (некоторые записаны с ошибками), находящиеся в базе поисковой системы.

Номер запроса	Выдача	Значение функции похожести
1	1	1
1	ошибка	0.5405405405405
2	2	1
2	3	0.9285714285714
2	9	0.9285714285714
2	8	0.8387096774193
2	5	0.7222222222222
2	7	0.7222222222222
2	4	0.7096774193548
2	10	0.6666666666666
2	6	0.6046511627907
2	11	0.6
3	3	1
3	9	1
3	8	0.8387096774193
3	4	0.8148148148148
3	5	0.7222222222222
3	10	0.6666666666666
3	6	0.6190476190476
3	11	0.6
3	ошибка	0.51612903225806
4	4	1
4	8	0.8235294117647
4	5	0.7179487179487
4	7	0.7179487179487
4	3	0.7096774193548
4	9	0.7096774193548
4	6	0.6086956521739
4	11	0.6060606060606
4	10	0.6060606060606

Таблица 1. Результаты работы системы на запросах 1-4.

запроса. Для него причиной выдачи небольшого количества результатов послужило отсутствие правила – альтернативы, переводящего ритмический рисунок «восьмая нота + восьмая нота» в пару «восьмая нота с точкой + шестнадцатая нота». Подобные недостатки можно устранить, дополнив базу правил вывода, если недостающее правило вывода представляется целесообразным использовать. Плюсом системы является невысокая доля ошибок в поисковой выдаче.

5 Заключение

В работе предложен метод поиска информации в структурированных файлах специального вида – формата MusicXML, используемого для хранения нот музыкальных произведений. В рамках предложенного метода решены следующие задачи: выбрано внутреннее представление для хранения нотных записей, позволяющее проводить их сравнение в соответствии со спецификой предметной области, и составлены правила преобразования, применяющиеся для устранения различий в похожих фрагментах. Метод поиска реализован в прототипной системе, позволяющей также пополнять библиотеку файлов, в которых ведется поиск, и автоматически индексировать содержимое библиотеки. Эксперименты с системой позволили подобрать эвристические коэффициенты для оценки сходства запроса и библиотечного фрагмента. Метод позволяет находить ноты музыкальных произведений по заданному фрагменту и оценивать соответствие результатов поиска введенному запросу.

Список литературы

[1] *Bundy A., Basin D., Hutter D., Ireland A.* Rippling: Meta–level Guidance for Mathematical Reasoning. — Cambridge University Press, 2005.

[2] Rippling: Meta–level Guidance for Mathematical Reasoning. www.rippling.org

[3] *Typke R.* Music Retrieval Based on Melodic Similarity — Ph.D. Thesis, University of Utrecht, 2007 www.cs.uu.nl/groups/MG/multimedia/publications/art/typke-thesis.pdf

[4] *Typke R., Giannopoulos P., Veltkamp R.C., Wiering F., van Oostrum R.* Using Transportation Distances for Measuring Melodic Similarity // Proceedings of the Fourth International Conference on Music Information Retrieval — 2003 —pp. 107–114.

[5] *Xing Wei, Bruce Croft W.* Modeling Term Associations for Ad-hoc Retrieval Performance within Language Modeling Framework // Proceedings of the 29-th European conference on IR research — Springer-Verlag Berlin, Heidelberg, 2007, — pp. 52–63 — ISBN 978-3-540-71494-1.

[6] *Yang X., Chen Q., Wang X.* A Novel Approach Based on Fault Tolerance and Recursive Segmentation to Query by Humming // Advances in Computer Science and Information Technology: AST/UCMA/ISA/ACN 2010 Conferences Joint Proceedings. — Springer-Verlag Berlin, Heidelberg, 2010, — pp. 544–557 — 978-3-642-13576-7.

[7] *Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.* Алгоритмы: построение и анализ, 2-е издание.: Пер. с англ. —М.: Издательский дом «Вильямс», 2005. — 1296 с. — ISBN 5-8459-0857-4 (рус.)

[8] *Маннинг К.Д., Рагхаван П., Шютце Х.* Введение в информационный поиск.: Пер. с англ. — М.: ООО «И.Д. Вильямс», 2011 — 528 с. — ISBN 978-5-8459-1623-5 (рус.)

[9] *Зарипов Р. Х.* Кибернетика и музыка. —М: Изд-во «Знание», 1963.

[10] *Дубовский И., Евсеев С., Способин И., Соколов В.* Учебник гармонии. —М: Изд-во «Музыка», 1965.

[11] *Вахромеев В. А.* Элементарная теория музыки. —М.: МУЗГИЗ, 1961.

[12] Echoprint: Open source music identification. — `echoprint.me`

[13] MusicXML format. — `www.makemusic.com/musicxml`

[14] Musipedia: Melody Search Engine. — `www.musipedia.org`

[15] Нотомания - ноты для различных инструментов. — `www.notomania.ru`

[16] MusicXML sheets library — `www.wikifonia.org`

УДК 004.932.72'1

Оценка фона изображения для автоматического обнаружения разметки и дефектов дорожного покрытия

© 2012 г. А. С. Лебедев

alebedev@graphics.cs.msu.ru

Кафедра автоматизации систем вычислительных комплексов

1 Аннотация

В работе рассматриваются различные подходы для решения задачи автоматического обнаружения разметки и дефектов дорожного покрытия, которые используются при видеопаспортизации дорог и в системах помощи водителю. Производится анализ алгоритмов с точки зрения качества используемых признаков для классификации областей изображения. Особое внимание уделяется признакам, построенным на основе оценки общего фона изображения. Предлагается новый алгоритм оценки фона изображения, который позволяет получать более дискриминативные признаки для классификации, нежели существующие методы. Также в работе представлена новая схема сегментации, которая позволяет обнаруживать дефекты дорожного покрытия в низкоконтрастных частях изображения.

2 Введение

В настоящее время задача контроля качества дорожного покрытия является очень актуальной. Существуют различные подходы для решения данной задачи. Например, корпорация Google представила сервис [1], который позволяет интегрально оценивать качество дорожного покрытия на основе сбора данных о тряске с мобильных телефонов автовладельцев. Другим

подходом является видеопаспортизация дорог, целью которой является фотографирование дорожного покрытия и разметка областей дорожного покрытия: автоматическая [2] или полуавтоматическая [3]. В дальнейшем мы будем рассматривать только подход, связанный с распознаванием дефектов полотна по фотоизображениям. Среди дефектов полотна, как правило, выделяют два типа: отсутствие дорожной разметки и наличие заплаток или ям (т.н. площадные дефекты). Алгоритмы обнаружения дефектов чаще всего работают с ректифицированными изображениями дорог, которые получаются из обычных путем применения специального преобразования, обратного тому, что выполняется в фотокамере [4]. В данной работе предлагаемый алгоритм использует только ректифицированные изображения дорожного полотна. Для тестирования алгоритма и сравнения с существующими использовались базы фотографий дорожного полотна, описанные в работах [2] и [4]. На рисунке 1 представлены примеры ректифицированных изображений из этих баз. Часто ректифицированные изображения содержат помимо дорожного полотна посторонние объекты (обочина, автомобили). Для фильтрации посторонних объектов используются специальные классификаторы. В настоящее время существует

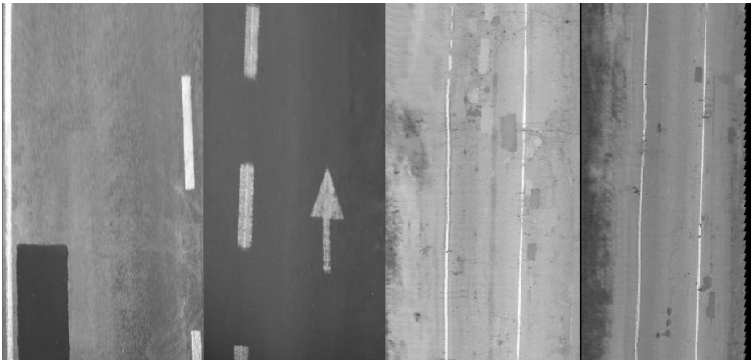


Рис. 1. Примеры используемых для тестирования ректифицированных изображений

большое число алгоритмов для обнаружения дефектов дорожного полотна. Описание и анализ существующих методов приводится в п.3. Одной из основных целей данной работы является предоставление новых признаков для обнаружения дефектов, которые могут использоваться в любой уже существующей системе, повышая её точность. Предлагаемые признаки основаны на идее построения общего фона дорожного покрытия. Алгоритмы построения признаков описаны в п.4. В п.5 приводится описание существующих признаков, которые неявно используют идею построения фона дорожного полотна, а также проводится сравнение существующих признаков с предложенными. Нами была разработана схема сегментации дорожного полотна, которая позволяет более точно обнаруживать дефекты, по сравнению с существующими подходами (см. п.6). Наконец, в п.7 обсуждаются достоинства и недостатки предложенных алгоритмов, а также их возможные применения в других областях.

3 Обзор существующих методов

В настоящее время существует множество подходов к решению задачи обнаружения дефектов дорожного покрытия. Сначала рассмотрим алгоритмы по обнаружению дорожной разметки, которые используются в автомобильных системах помощи водителю.

Детекторы дорожных разметок используют различную информацию: яркость, цвет, границы, форма. В работе [5] описан алгоритм на основе вычисления градиента изображения. В зависимости от ориентации и величины градиента находятся пиксели, принадлежащие дорожной разметке. К сожалению, данный метод часто обнаруживает ложные объекты, не являющиеся разметкой.

В работе [6] в качестве априорной информации предлагается использовать форму разметки. Для нахождения линий раз-

метки применяется преобразование Хафа. Несмотря на устойчивость этого метода к условиям освещения, он не устойчив к шумам. Существуют и более робастные методы для поиска линий разметки, например [7]. Еще один метод [8], использующий информацию об определенной направленности линии разметки – применяет специальные детекторы, фильтрующие пиксели, локальная окрестность которых имеет заранее известную ориентацию. Все указанные методы в явном виде ищут прямые линии на изображениях, однако в общем случае дорожные разметки являются более сложными объектами по форме. Что касается площадных дефектов дорожного покрытия, то их априорная форма также неизвестна, поэтому данные методы неприменимы для этой задачи.

Другая часто используемая в алгоритмах априорная информация – информация о разнице цвета между общим фоном дорожного покрытия и искомыми дефектами или разметкой. В [2] приведен обзор и сравнение существующих признаков для обнаружения разметок. Результаты исследования показали, что признаки на основе адаптивной бинаризации показывают лучший результат, чем остальные. Фактически данные признаки неявно используют оценку общего фона дорожного покрытия. Эта идея применяется и для построения наших признаков, алгоритм расчета которых представлен в п.4. В п.5. приводится описание существующих признаков, использующих оценку фона, а также сравнение с предложенными признаками.

В работе [4] представлено два алгоритма для обнаружения дорожных дефектов. Первый алгоритм сегментирует исходное изображение и рассчитывает различные признаки для областей: на основе цвета, текстуры, относительной яркости. На основе признаков строится каскад классификаторов, который определяет области, являющиеся разметкой или площадными дефектами. Авторы статьи сравнили предложенный алгоритм с подходом попиксельной классификации с последующим пре-

образованием Платта и показали, что их подход в среднем немного лучше, т.к. сегментация более устойчива к шуму. Однако попиксельный подход потенциально лучше сохраняет детали найденных объектов. Для получения точных границ областей используется метод активных контуров, который требует ручной инициализации контура. В п.6 приводится описание схемы работы алгоритма, которая является компромиссом между попиксельным подходом и классификацией сегментов с точки зрения точности разметки и устойчивости к шуму.

4 Предлагаемый алгоритм для расчета признаков

Основная идея алгоритма заключается в наиболее точной оценке общего фона изображения дорожного полотна. Априорно алгоритм основывается на следующих суждениях относительно входных данных:

- Дорожная разметка локально всегда ярче, чем само дорожное покрытие
- Искомые дефекты дорожного покрытия темнее, чем дорожное полотно
- Яркость дорожного полотна является достаточно гладкой функцией

Последнее требование означает, что в изображении отсутствуют резкие перепады яркости. На практике это требование выполняется не всегда. Например, в случае наличия резкой тени на дороге. Для таких случаев можно успешно применять алгоритмы [9], которые строят маску тени, и оценивать фон дорожного полотна независимо в области тени и вне неё. Предлагаемый алгоритм оценки фона состоит из следующих шагов:

- Определение узловых точек
- Построение аппроксимирующего сплайна типа тонкой пластины [10]
- Расчет фона по уравнению сплайна

Основная цель этапа построения узловых точек состоит в определении точек, которые наиболее вероятно являются точками фона. В качестве кандидатов рассматриваются точки, удовлетворяющие соотношению: $|I_{blurred}(x,y) - I(x,y)| < T$, где $I_{blurred}$ — размытое фильтром Гаусса исходное изображение, I — исходное изображение, T — некоторый порог. Радиус размытия и порог фиксируются в зависимости от разрешения изображения и максимально возможного размера дефекта. Далее узловые точки выбираются из кандидатов произвольным образом так, чтобы равномерно покрыть всю площадь изображения. Число точек выбирается из соображений скорости работы алгоритма, т.к. сложность построения сплайна пропорциональна квадрату числа узловых точек.

На втором этапе строится аппроксимирующий сплайн типа тонкой пластины по узловым точкам. Особенностью данного сплайна является возможность удобного регулирования параметра гладкости, который фиксируется в зависимости от разрешения изображения. Алгоритм построения сплайна является итерационным и немного различается для случаев поиска светлых областей (разметка) и темных (площадные дефекты). Каждая итерация состоит из построения сплайна по текущим значениям яркости в узловых точках и последующей коррекции с учетом реальных значений яркости изображения. Изначально значения яркости в узловых точках равны яркости изображения в этих точках. На этапе коррекции значений узловых точек их яркость изменяется по формуле:

$$I_n(p_i) = max(TPS(I_{n-1}(p_1),\ I_{n-1}(p_2), \ldots,\ I_{n-1}(p_l)),\ I(p_i))$$

в случае поиска темных областей,

$$I_n(p_i) = min(TPS(I_{n-1}(p_1),\ I_{n-1}(p_2), \ldots,\ I_{n-1}(p_l)),\ I(p_i))$$

в случае поиска светлых областей,

$$I_0(p_i) = I(p_i),$$

где $I(p_i)$ — значение яркости исходного изображения в i-ой узловой точке ($i = 1, 2, \ldots, l$), $I_n(p)$ — значение яркости в узловой точке после коррекции на n-ой итерации, $TPS(I_{n-1}(p_1), I_{n-1}(p_2), \ldots, I_{n-1}(p_l))$ — значение сплайна в узловой точке, построенного по значениям яркости узловых точек, взятых с предыдущей итерации. Необходимое количество итераций может быть заранее фиксировано (оно зависит от предполагаемых размеров дефекта), также можно указать дополнительное условие останова:

$$\sum_{i=1}^{l} |I_n(p_i) - I_{n-1}(p_i)| < \delta$$

На третьем этапе происходит расчет значений сплайна в каждой точке изображения. Данная операция может оказаться наиболее вычислительно затратной, если разрешение изображения и количество узловых точек велико. Однако поскольку построенная оценка фона является довольно гладкой, то можно производить расчет значений сплайна для уменьшенного изображения, а затем интерполировать яркость для исходного размера изображения с использованием более быстрых методов (например, используя билинейную интерполяцию).

Восстановив фон дорожного полотна можно построить простой детектор дефектов. Для этого нужно вычесть из исходного изображения фон в случае дорожной разметки или из фона вычесть исходное изображение в случае площадных дефектов. Данную разность можно рассматривать в качестве основного признака интересующей области. Если же исходное изображение было сегментировано, то в качестве признаков для сегмента можно использовать среднее значение разности по сегменту, а также различные квантили, например, 25%, 50% и 75%.

В п.5 будут описаны существующие признаки, неявно использующие идею построения фона, а также будет проведено сравнение с предлагаемыми признаками.

5 Существующие признаки на основе оценки фона

В данном разделе рассмотрим два признака: разница с размытым фильтром Гаусса изображением и на основе симметричного локального порога [2]. Основная идея первого признака — оценить фон дорожного полотна как среднее (со сверткой Гаусса) по окрестности. Радиус фильтра зависит от размера искомых областей. Данный фильтр часто применяется для компенсации освещения при распознавании текста, когда ищутся объекты априорно более темные, чем фон. В случае поиска светлых областей на более темном фоне признак определяется как разность исходной яркости изображения и размытой яркости.

Признак на основе симметричного локального порога рассчитывается для каждой линии изображения независимо от других. Пусть имеется точка с координатой x в какой-то линии. Затем рассматривается разница яркости в точке x и среднего значения яркости в правой окрестности x в той же линии, а также разница яркости в точке x и среднего значения яркости левой окрестности x. Максимум из двух чисел и есть значение признака. Стоит отметить, что описанные признаки ищут объекты, которые светлее фона. Чтобы нацелиться на поиск темных объектов нужно, наоборот, из оценки фона вычесть исходную яркость.

Для сравнения предлагаемых признаков использовались ректифицированные изображения из баз, описанных в работах [2] и [4]. На некоторых изображениях помимо дорожного полотна присутствуют и другие объекты: обочина, автомобили и т.п. Для фильтрации таких объектов использовалась схема алгоритма, описанная в п.6, в которой после получения отдельных сегментов применялись простые классификаторы на основе цвета. Поскольку исходные изображения в базе цветные, то отличить обочину (или автомобиль) от дорожного по-

лотна достаточно просто. Фильтрация посторонних объектов применялась для всех алгоритмов. На рисунке 2 представлены результаты работы детекторов дорожной разметки на основе симметричного локального порога, разницы с размытым изображением и предлагаемого алгоритма. На рисунке 3 представлены аналогичные результаты по обнаружению площадных дорожных дефектов. Для объективного сравнения полученных

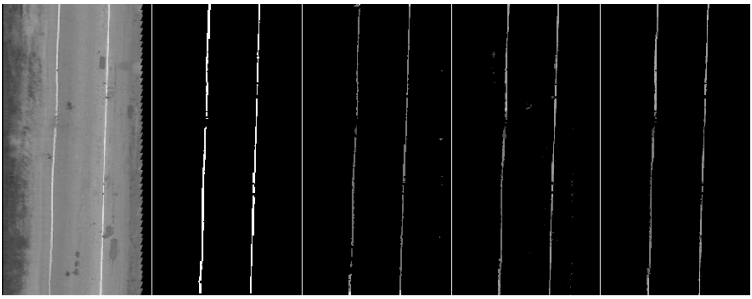


Рис. 2. Слева направо — исходное изображение, вручную размеченное изображение разметки, результат разницы с размытым изображением, результат симметричного локального порога, результат предложенного алгоритма

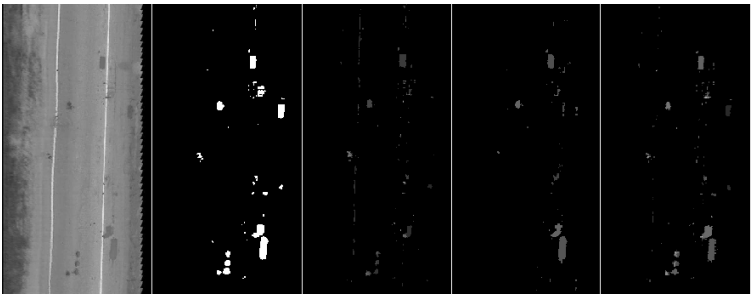


Рис. 3. Слева направо — исходное изображение, вручную размеченное изображение площадных дефектов, результат разницы с размытым изображением, результат симметричного локального порога, результат предложенного алгоритма

результатов с ручной разметкой производился расчет количества ложно найденных и ненайденных пикселей изображения. Предварительно полученные значения признаков (рисунки 2 и 3) бинаризовались путем применения набора порогов из диапазона [0, 255]. На рисунке 4 представлены ROC-кривые для всех рассмотренных алгоритмов поиска разметки, построенных для различных значений порогов. На рисунке 5 приведена аналогичная ROC-кривая для поиска дефектов. Как видно из гра-

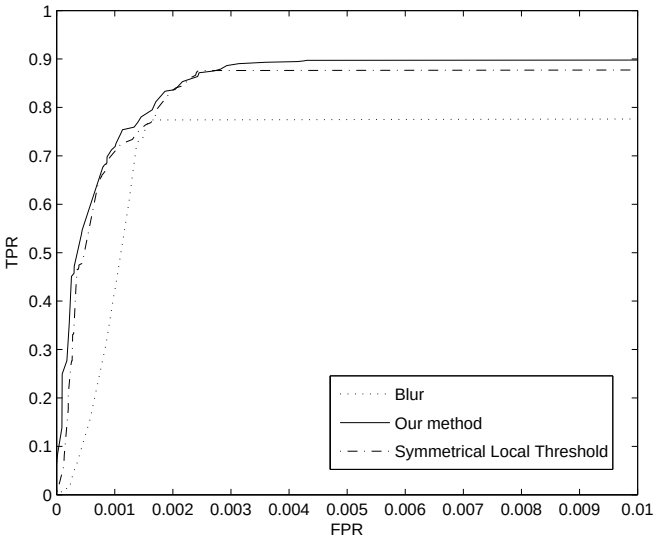


Рис. 4. ROC-кривые алгоритмов поиска дорожной разметки: разница с размытым изображением, симметричный локальный порог, предложенный алгоритм

фиков, предложенный алгоритм показывает лучшие результаты. В случае поиска разметки качество алгоритмов отличается не очень сильно, т.к. дорожная разметка однообразна по форме и цвету, в то время как дорожные дефекты могут сильно варьироваться и по форме, и по цвету.

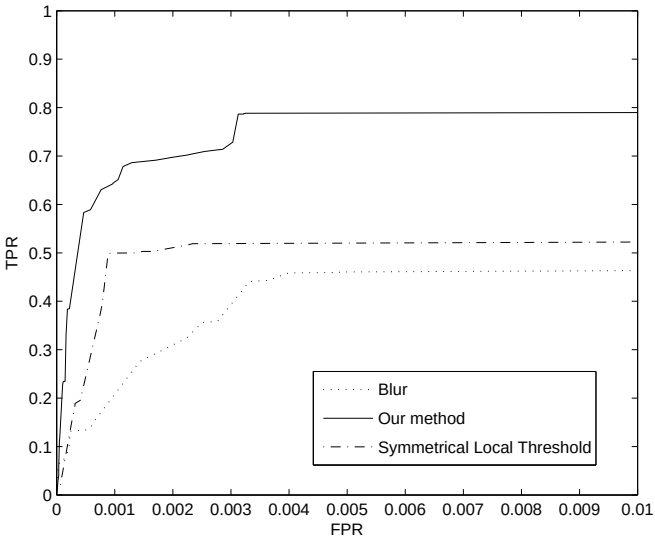


Рис. 5. ROC – кривые алгоритмов поиска дорожных дефектов: разница с размытым изображением, симметричный локальный порог, предложенный алгоритм

6 Предлагаемая схема сегментации дорожного полотна

В настоящее время распространены два подхода к поиску объектов на фотоизображениях. В первом подходе исходное изображение сегментируется, для каждого сегмента вычисляются признаки, затем производится классификация сегментов. Второй подход состоит в попиксельной классификации изображения. Достоинством первого подхода является возможный учет формы искомого объекта (можно ввести соответствующий признак для сегмента), в сглаживании шумов в изображении; недостатком — возможные пропуски мелких объектов из-за сглаживающего эффекта сегментации. На примере задачи

поиска дефектов дорожного полотна сегментация может пропустить низкоконтрастные дефекты. Для решения этой проблемы могут применяться различные предобработки: повышение локального контраста, ретинекс-фильтры [11] и т.п., однако проблемы часто остаются.

Попиксельная классификация изображения позволяет более точно находить интересующие объекты, но она является неустойчивой к шуму, а также не способна учитывать форму искомых объектов.

Нами предлагается использовать следующую схему сегментации, которая рассчитана на поиск объектов, отличающихся по цвету от фона. На первом этапе происходит оценка фона изображения (см. п. 4, 5). Далее рассматривается разница между исходным изображением и фоном, которая затем сегментируется. Предлагается использовать алгоритм Colour Structure Code [12] для сегментации, который имеет интуитивно понятный яркостный порог для фильтрации шумных областей. Для полученных сегментов рассчитываются интересующие признаки, по ним производится классификация областей. В частности, нами использовались дополнительные цветовые признаки для фильтрации обочины и других посторонних объектов в фотоизображении. Предложенная схема сегментации позволяет с одной стороны более точно детектировать даже мелкие объекты, с другой — позволяет отфильтровать шумы.

Ранее в [4] было показано, что подход с сегментацией в среднем работает лучше, чем попиксельная классификация. Для сравнения подхода посегментной классификации с предложенным был проведен следующий эксперимент. Мы вручную разметили дефекты на фотографиях, выделяя соответствующие сегменты на изображениях. Для сегментации изображения использовались различные алгоритмы: Mean-Shift [13], Color Structure Code [12], Edge Flow [14], пирамидальная сегментация [15]. Настройки алгоритмов подбирались таким образом, чтобы каждый из дефектов распадался не более чем на 4-5

сегментов (в случае большего числа сегментов преимущества использования посегментной классификации пропадают). Для полученных посегментных ручных разметок была построены ROC-кривые на основе сравнения с попиксельными ручными разметками. Далее для каждой фотографии был выбран лучший показатель TPR по всем сегментациям и усреднен по всем снимкам. На рисунке 6 представлено сравнение полученной ROC-кривой с ROC-кривой нашего алгоритма. Как видно,

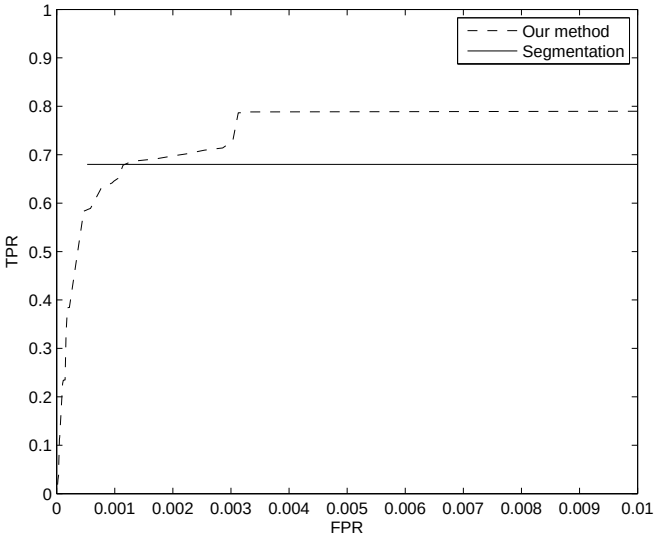


Рис. 6. ROC – кривые (дорожные дефекты): ручная посегментная разметка, предложенный алгоритм

предложенная схема сегментации потенциально лучше обычного подхода с сегментацией. Это происходит из-за того, что не теряются мелкие или низконтрастные объекты, границы объектов становятся более точными. Стоит отметить, что предложенную схему сегментации можно обобщить: вместо сегментации карты разности можно сегментировать карту вероятности,

полученную с использованием метода попиксельной классификации. Это позволяет применить данную схему для более широкого класса задач.

7 Результаты

В данной статье нами были предложены новый алгоритм построения признаков для поиска дорожной разметки и площадных дорожных эффектов. Нами было показано, что данные признаки являются более дискриминативными, нежели существующие. Также была предложена новая схема алгоритма сегментации дефектов, которая позволяет добиться большей точности в обнаружении объектов, чем существующие подходы на основе посегментной и попиксельной классификации.

Стоит отметить некоторые недостатки предложенного алгоритма. Априори мы предполагали, что дорожные дефекты темнее, чем общий фон дорожного полотна. На практике встречаются случаи, когда данное предположение не является верным — дорожные дефекты могут быть темнее только на границах области. В этом случае предложенный алгоритм обнаружит только границы дефекта. В этом случае имеет смысл применять алгоритмы на основе поиска границ дефектов. Наиболее совершенный из таких алгоритмов был рассмотрен в работе [4]. Тем не менее, предложенные в статье подходы по поиску дефектов могут с успехом применяться в других областях, где есть априорная информация об относительной яркости искомых объектов: распознавание текста, номерных знаков, поиск объектов в медицинских изображениях и т.п.

Список литературы

[1] *Проект Google «Русские дороги»* http://www.rusdorogi.ru/

[2] *Veit T., Tarel J.P., Nicolle P., Charbonnier P.* Evaluation of

road marking feature extraction // 11th International IEEE Conference on Intelligent Transportation Systems. 2008. pp. 174-181.

[3] *Barinova O., Shapovalov R., Sudakov S., Velizhev A., Konushin A.* Efficient road mapping via interactive image segmentation // 3D City Models, Road Databases and Traffic Monitoring (CMRT 2009). 2009.

[4] *Судаков С., Семашко А., Баринова О., Конушин А., Киншаков В., Крылов А.* Алгоритмы детектирования разметки и дефектов дорожного покрытия // Proceedings of GraphiCon' 08. 2008.

[5] *Lu W., Wang H., Wang Q.* A synchronous detection of the road boundary and lane marking for intelligent vehicles // In Proc. of SPND. 2007. pp. 741-745.

[6] *McDonald J.* Detecting and tracking road markings using the Hough transform // Proc. of the Irish Machine Vision and Image Processing Conference. 2001.

[7] *Borkar A., Hayes M., Smith M.* A novel lane detection system with efficient ground truth generation // Intelligent Transportation Systems. 2012. 13. N 1. pp. 365-374.

[8] *McCall J., Trivedi M.* An integrated, robust approach to lane marking detection and lane tracking // In Proc. IEEE Intelligent Vehicles Symposium. 2004. pp. 533-537.

[9] *Hara T., Aoki Y.* A robust road marking detection method with pseudo-top-view images for creating database // 17th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV). 2011. pp. 1-5.

[10] *Bartoli A., Perriollat M., Chambon, S.* Generalized thin-plate spline warps // International journal of computer vision. 2010. 8. N 1. pp. 85-110.

[11] *Yi S., Jung C., Kim J.K., Jung K., Hur N.* Novel retinex method using selective smoothing filter based on local histogram analysis // ITC-CSCC. 2010. p. 397.

[12] *Rehrmann V., Priese L.* Fast and robust segmentation of natural color scenes // Computer Vision—ACCV. 1997. pp. 598-606.

[13] *Comaniciu D., Meer P.* Mean shift: A robust approach toward feature space analysis // Pattern Analysis and Machine Intelligence. 2002. 24. N 5, pp. 603-619.

[14] *Ma W., Manjunath B.* Edge flow: a framework of boundary detection and image segmentation // Computer Vision and Pattern Recognition. 1997. pp. 744-749.

[15] *Rezaee M., Van Der Zwet P., Lelieveldt B., Van Der Geest R., Reiber J.* A multiresolution image segmentation technique based on pyramidal segmentation and fuzzy clustering // IEEE Transactions on Image Processing. 2000. 9. N 7. pp. 1238-1248.

УДК 517.977

Управление эллипсоидальным контейнером

© 2012 г. А. И. Месяц

month_october@mail.ru

Кафедра системного анализа

1 Введение

Работа посвящена рассмотрению задачи, появляющейся в теории группового (коллективного, командного) управления, занимающейся построением и исследованием математических моделей, описывающих ситуации, когда группе объектов необходимо решить общую задачу, при этом взаимодействуя, централизованно или нет, друг с другом. Такие ситуации появляются в прикладных инженерных и биологических задачах [5–7].

Особый интерес представляет собой ситуация, когда группе агентов необходимо, находясь недалеко друг от друга, передвигаться из начального положения в конечное, минуя известные препятствия. Примером может быть, например, группа беспилотных пожарных вертолетов, которым необходимо долететь до места пожаротушения. В [1] предложен следующий метод решения подобной задачи: описать вокруг агентов некоторый (виртуальный) контейнер и разбить дальнейшее движение на две компоненты:

1. Движение контейнера, который должен переместиться из начального положения в конечное, меняя свою форму и избегая столкновения с внешними препятствиями;
2. Движение агентов внутри контейнера, которые должны оставаться внутри него на протяжении всего движения, избегая столкновений друг с другом.

При этом ясно, что форма контейнера должна в принципе допускать возможность расположения объектов внутри себя, т.е. он не должен “сплющиваться”, вырождаться.

В настоящей работе предлагается постановка задачи позиционного управления подобным контейнером, имеющим вид эллипсоида. Его динамика описывается системой обыкновенных дифференциальных уравнений, для центра — векторным, для матрицы конфигураций — матричным [4]. Движение контейнера при этом должно удовлетворять фазовым ограничениям на минимальным размер.

Работа ораганизована следующим образом. В разделе 2 ставятся основные рассматриваемые задачи — задача совмещения конфигураций и задача движения контейнера. Разделы 3 и 5 посвящены задаче совмещения конфигураций. В разделе 4 описано, как можно соблюсти ограничения на размер контейнера, подобрав соответствующим образом ограничения на управления. В разделе 6 приводится алгоритм для решения задачи движения эллипсоидального контейнера на плоскости.

2 Постановка задачи

Рассмотрим эллипсоид $\mathcal{E}\left(q(t), Q(t)\right) \equiv \{x \in \mathbb{R}^n : \ \langle x, q(t)\rangle + \sqrt{\langle x, Q(t)x\rangle} \leqslant 1\}$, динамика которого описывается системой дифференциальных уравнений:

$$\begin{cases} \dot{Q}(t) = A(t)Q(t) + Q(t)A^T(t) + V(t,q,Q), \\ Q(t_0) = Q_0 = Q_0^T > 0, \\ \ddot{q} = u(t,q,Q), \\ q(t_0) = q_1, \ \dot{q}(t_0) = 0, \ q(t_1) = q_1, \ \dot{q}(t_1) = 0. \end{cases} \tag{1}$$

Здесь управляющие параметры — матрицы A, V и вектор u. Далее предполагается, что эти матрицы имеют следующую

структуру:

$$A = \begin{bmatrix} 0 & a_{12} & a_{13} & \dots & a_{1n} \\ a_{12} & 0 & a_{23} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{1n} & a_{2n} & a_{3n} & \dots & 0 \end{bmatrix}, V = \begin{bmatrix} v_{11} & 0 & \dots & 0 \\ 0 & v_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & v_{nn} \end{bmatrix}.$$

Иными словами, матрица A — симметричная, с нулями на диагонали, а V — диагональная. Управление подчинено геометрическому ограничению, $\|A(t)\| \leqslant \mu$, $\|V(t)\| \leqslant \nu$, $\forall t \in [t_0, t_1]$. Здесь и всюду далее под матричной нормой понимается матричная норма Фробениуса,

$$\|A\| = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^2}, \ A \in \mathbb{R}^{n \times n},$$

порождаемая матричным скалярным произведением $\langle A, B \rangle = \operatorname{tr} B^T A$. Для такой системы рассматриваются две задачи.

Задача совмещения конфигураций. Требуется найти позиционные управления $A(Q), V(Q)$, переводящие на некотором конечном интервале времени $[t_0, t_1]$ матрицу конфигураций $Q(t)$ в целевое состояние Q_1, а также оценить сверху время t_1, для этого необходимое. При этом должны выполняться следующие условия:

1. В любой момент времени матрица Q является матрицей некоторого невырожденного эллипсоида:

$$Q(t) = Q^T(t) > 0 \ \forall \ t \in [t_0, t_1]. \tag{2}$$

2. Выполнено условие на минимальный размер эллипсоида:

$$\lambda_{\min}(Q(t)) \geqslant \lambda_0 > 0 \ \forall \ t \in [t_0, t_1], \tag{3}$$

где через $\lambda_{\min}(Q)$ обозначено наименьшее собственное число матрицы Q.

Задача движения эллипсоидального контейнера. Найти позиционные управления $u(t,q,Q)$ и $A(t,q,Q)$, $V(t,q,Q)$, переводящее на интервале времени $[t_0,t_1]$ эллипсоид $\mathcal{E}\left(q(t),Q(t)\right)$ в целевое состояние $\mathcal{E}\left(q_1,Q_1\right)$. При этом должны выполняться условия (2), (3), а так же — фазовые ограничения: заданы эллипсоиды $\mathcal{E}\left(p_i,P_i\right)$ $(i=1,2,\ldots,m)$, и требуется, что бы

$$\mathcal{E}\left(q(t),Q(t)\right)\cap \operatorname{int}\mathcal{E}\left(p_i,P_i\right)=\varnothing\ \forall\ t\in[t_0,t_1],\ \forall i=1,2,\ldots,m. \tag{4}$$

Считаем, что начальная и конечная конфигурации $Q_1,\ Q_0$ удовлетворяют этим ограничениям.

3 Решение в $\mathbb{R}^n$

Предварительно для удобства введем линейные операторы $[\cdot]^d$ и $[\cdot]^o$, отображающие $\mathbb{R}^{n\times n}$ в себя. A^d определяется как диагональная матрица, диагональ которой совпадает с диагональю A, а A^o — как матрица, совпадающая с A во всех компонентах, кроме диагональной, где у нее стоят нули. В этих обозначениях для управлений A,V справедливы тождества $V^d=V,\ A^o=A$.

Рассмотрим функцию $\mathcal{V}(Q)=\langle Q-Q_1,Q-Q_1\rangle$ и будем выбирать управления так, что бы оно минимизировало ее значения на траекториях системы. Имеем

$$\frac{1}{2}\frac{d\mathcal{V}}{dt}\bigg|_{(1)}=\langle Q-Q_1,AQ+QA+V\rangle\equiv\langle U,AQ+QA+V\rangle=$$
$$=\langle A,QU+UQ\rangle+\langle V,U\rangle\equiv\langle A,R\rangle+\langle V,U\rangle\,.$$

Отсюда приходим к двум задачам минимизации

$$\begin{cases}\langle U,V\rangle\to\min\limits_{V},\\ \|V\|\leqslant\nu.\end{cases}\qquad\begin{cases}\langle R,A\rangle\to\min\limits_{A},\\ \|A\|\leqslant\mu.\end{cases} \tag{5}$$

Эти задачи имеют своим решением

$$V = -\frac{\nu U^d}{\|U^d\|},\ \left\|U^d\right\| \neq 0,\ A = -\frac{\mu R^o}{\|R^o\|},\ \|R^o\| \neq 0. \qquad (6)$$

Лемма 1. *Если $Q = Q^T > 0$, то $\left\|U^d\right\| = 0$ и $\|R^o\| = 0$ тогда и только тогда, когда $\|U\| = 0$.*

Доказательство. Поскольку $Q = Q^T > 0$, то справедливо представление $Q = S\Lambda S^T$, где $SS^T = I$, а Λ — диагональная матрица с положительными диагональными элементами. (Здесь и далее через I обозначена единичная матрица.)

Пусть $\left\|U^d\right\| = 0$ и $\|R^o\| = 0$. Тогда $U = U^o$ и

$$\begin{aligned} 0 = \langle U, R^o\rangle = \langle U^o, R\rangle = \langle U^o, QU + UQ\rangle = \\ = \langle QU^o, U\rangle + \langle U^oQ, U\rangle = 2\langle U^o, QU^o\rangle = \\ = 2\left\langle U^o, S\Lambda S^T U^o\right\rangle \ \Rightarrow\ S^T U^o = 0 \ \Rightarrow\ U^o = 0. \end{aligned}$$

Но если $U^d = 0, U^o = 0$, то $U = 0$. В другую сторону соотношение очевидно. □

Таким образом, в любой точке траектории хотя бы одна из систем (5) имеет решение, задаваемой формулами (6). Значит, нам надо доопределить управления в тех случаях, когда $\left\|U^d\right\| = 0$ или $\|R^o\| = 0$. Доопределеим в таких точках V нулем.

Для доопределения A будем действовать следующим способом. Рассмотрим множество

$$\mathcal{R} = \{Q \in \mathbb{R}^{n\times n} : R^o \equiv (R(Q))^o = 0\}.$$

Будем строить управление так, что бы система попала на $\mathcal{R}$ и дальше двигалась по нему до конечной точки Q_1 (ясно, что $Q_1 \in \mathcal{R}$).

Для начала рассмотрим подробнее структуру множества $\mathcal{R}$. Оно задается через систему скалярных уравнений, $r_{ij}(Q) = 0,\ 1 \leqslant i < j \leqslant n$. Посмотрим, когда эта система задает

гладкое многообразие. Для этого рассмотрим нормали, $N_{ij} = \left(\frac{\partial r_{ij}(Q)}{\partial q_{ab}}\right)_{a,b=1}^{n}$, (при $i < j$) причём

$$\frac{\partial r_{ij}(Q)}{\partial q_{ab}} = \begin{cases} a \neq i \text{ и } b \neq j: & 0, \\ a = i \text{ и } b \neq j \text{ или } a = j \text{ и } b \neq i: & 2q_{bj} - q^1_{bj}, \\ a \neq i \text{ и } b = j \text{ или } a \neq j \text{ и } b = i: & 2q_{ia} - q^1_{ia}, \\ a = i \text{ и } b = j: & 2(q_{aa} + q_{bb}) - q^1_{aa} - q^1_{bb}. \end{cases} \tag{7}$$

Если в окрестности точки Q множество $\mathcal{R}$ образует гладкое многообразие, то условие движения по $\mathcal{R}$ будет иметь вид $\dot{Q} \in \in \mathcal{T}_{\mathcal{R}}(Q)$, или $\left\langle \dot{Q}, N_{ij} \right\rangle = 0,\ 1 \leqslant i < j \leqslant n$, откуда

$$\langle A, QN_{ij} + N_{ij}Q \rangle = -\langle N_{ij}, V \rangle,\ 1 \leqslant i < j \leqslant n. \tag{8}$$

Полученное соотношение есть система линейных алгебраических уравнений размерности $\frac{n(n-1)}{2}$.

Нам потребуется вспомогательная [10]

Лемма 2. *Матричное уравнение $AX - BX = C$ имеет единственное решение для любой правой части C тогда и только тогда, когда у матриц A и B нет общих собственных чисел.*

Лемма 3. *В окрестности точки Q_1 множество $\mathcal{R}$ образует гладкое многообразие.*

Доказательство. Предположим, что система (7) из $\frac{n(n-1)}{2}$ элементов линейно зависима в точке Q, т.е. существуют такие не равные все нулям числа λ_{ij}, что $\sum_{i<j} \lambda_{ij} N_{ij}(Q) = 0$. Рассмотрим элемент этой линейной комбинации в позиции $(a, b), a < b$:

$$\sum_{j \neq b} (2q_{bj} - q^1_{bj})\lambda_{aj} + \sum_{i \neq a} (2q_{ia} - q^1_{ia})\lambda_{ib} + \lambda_{ab}(2(q_{aa} + q_{bb}) - q^1_{aa} - q^1_{bb}) =$$

$$= \sum_{j=1}^{n} (2q_{bj} - q^1_{bj})\lambda_{aj} + \sum_{i=1}^{n} (2q_{ia} - q^1_{ia})\lambda_{ib} \equiv$$

$$\equiv (\Lambda(2Q - Q_1))_{ab} + ((2Q - Q_1)\Lambda)_{ab}.$$

Если положить $\lambda_{ii} = 0,\ i = 1, 2, \ldots, n$, то полученное выражение остается верным для диагональных компонент, что позволяет записать систему уравнений

$$\begin{cases} \Lambda(2Q - Q_1) + (2Q - Q_1)\Lambda = 0,\ \Lambda = \Lambda^T = \Lambda^o, \\ ((Q - Q_1)Q + Q(Q_1 - Q))^o = 0. \end{cases} \tag{9}$$

Из первого соотношения получим, что набор нормалей линейно независим в точке Q_1. Действительно, пусть это не так, тогда для любой матрицы X получаем: $0 = \langle X, \Lambda Q_1 + Q_1\Lambda\rangle = \langle \Lambda, Q_1 X + X Q_1\rangle$. По лемме 2 найдется такое X, что $Q_1 X + X Q_1 = \Lambda$, откуда $\Lambda = 0$.

Таким образом, в некоторой окрестности Q_1 нормали линейно независимы, и, следовательно, множество $\mathcal{R}$ представляет собой гладкое многообразие размерности $\frac{n(n-1)}{2}$. □

В общем случае, система (8) может иметь более одного решения из-за линейной зависимости нормалей. Будем определять значение $A(Q)$ как решение задачи

$$\begin{cases} \langle A, A\rangle \to \min, \\ \langle A, QN_{ij} + N_{ij}Q\rangle = -\langle N_{ij}, V\rangle,\ 1 \leqslant i < j \leqslant n. \end{cases} \tag{10}$$

Заметим, что обнуление всех N_{ij} возможно лишь при $2Q = Q_1$, но из второго уравнения (9) получаем, что $Q_1^o = Q^o = 0$. Значит, Q, Q_1 — диагональные, и можно положить $A = 0$. В противном случае, множество допустимых решений в задаче (10) ограничено.

Лемма 4. *Пусть $Q = Q^T > 0$. Тогда в любой точке $\mathcal{R}$, где матрица $2Q - Q_1 \neq 0$, (следовательно, и в Q_1) задача* (10) *имеет, и при том единственное, решение.*

Доказательство. Задачу (10) можно рассмотреть как проблему моментов [3, 9]. Домножим каждое равенство в (9) на λ_{ij} и

сложим их, формально полагая $\lambda_{ii} = 0$. Соблюдение системы равенств (9) равносильно выполнению соотношения

$$\left\langle A, Q\left(\sum_{i<j}\lambda_{ij}N_{ij}\right) + \left(\sum_{i<j}\lambda_{ij}N_{ij}\right)Q\right\rangle = -\left\langle V, \sum_{i<j}\lambda_{ij}N_{ij}\right\rangle$$

Из предыдущей леммы мы знаем, что $\sum_{i<j}\lambda_{ij}N_{ij} = \Lambda(2Q - -Q_1)+(2Q-Q_1)\Lambda \equiv \Psi(\Lambda)$. Обозначим $\|A\| = \alpha$. Тогда решение будет иметь вид

$$A = \alpha\frac{Q\Psi + \Psi Q}{\|Q\Psi + \Psi Q\|},\ \alpha = \sup_{\Lambda}\frac{\langle V, \Psi\rangle}{\|Q\Psi + \Psi Q\|}. \tag{11}$$

Для существования решения у этой задачи [9, гл.8, §11] необходимо и достаточно, что бы

$$\inf_{\langle\Psi,V\rangle=1}\|Q\Psi + \Psi Q\| > 0. \tag{12}$$

Рассмотрим выражение, стоящее под нормой, Пусть условие (12) не выполняется. Тогда при Λ, на котором достигается точная нижняя грань, для любой матрицы X справедливо $0 = \langle X, \Psi Q + Q\Psi\rangle = \langle \Psi, XQ + QX\rangle$, но по лемме 2 найдется такой X, что $XQ + QX = \Psi$, откуда $\Psi = 0$, что противоречит с тем, что $\langle\Psi, V\rangle = 1$. □

Заметим, что при осуществленном выборе управлений система является асимптотически устойчивой, и, следовательно, в любую окрестность Q_1 траектория попадет за конечное время. Покажем, что управление (11) приводит нас в Q_1 за конечное время.

Пусть на Ψ^0 достигается точная верхняя грань в (11). Тогда управление будет иметь вид

$$A = \frac{\langle V, \Psi_0\rangle}{\|Q\Psi^0 + \Psi^0 Q\|^2}\left(Q\Psi^0 + \Psi^0 Q\right).$$

Подставляя это выражение в систему, получим:

$$\dot{Q} = \frac{\langle V, \Psi_0 \rangle}{\|Q\Psi^0 + \Psi^0 Q\|^2} \left(2Q\Psi^0 Q + \Psi^0 Q^2 + Q^2 \Psi^0\right) + V,$$

и, кроме того, $\dot{U}^d = \dot{Q}^d \equiv \langle V, \Psi_0 \rangle \Gamma + V$. По лемме 1, нам надо показать, что за конечное время обнулится U^d. Обозначим $y(t) = \left\langle U^d, U^d \right\rangle$. Имеем:

$$\dot{y} = 2\left\langle U^d, \dot{U}^d \right\rangle = 2\left\langle U^d, \langle V, \Psi_0 \rangle \Gamma + V \right\rangle =$$
$$= -2\mu \left(\frac{\left\langle U^d, \Gamma \right\rangle \left\langle U^d, \Psi^0 \right\rangle}{\|U^d\|} + \frac{\left\langle U^d, U^d \right\rangle}{\sqrt{\left\langle U^d, U^d \right\rangle}} \right) \leqslant$$
$$\leqslant -2\mu \left(-\sqrt{y} \left\|\Psi^0\right\| \cdot \|\Gamma\| + \sqrt{y}\right) \leqslant -2\mu\sqrt{y}(1 - \left\|\Psi^0\right\| \cdot \|\Gamma\|).$$

Таким образом, если мы покажем, что в некоторой окрестности Q_1 величина $\left\|\Psi^0\right\| \cdot \|\Gamma\|$, отделена от единицы, то мы придем к мажорирующему уравнению $\dot{y} = -\sqrt{y} \cdot \mathrm{const}$. Траектории этого уравнения приходят в ноль за конечное время. Такую оценку проведем в пункте 5 для $n = 2$.

4 Соблюдение условия невырожденности

Введем множество

$$M_{\lambda_0} = \left\{ Q \in \mathbb{R}^{n \times n} : \ Q = Q^T > 0, \ \lambda_{\min}(Q) \geqslant \lambda_0 \right\}.$$

Используя теорему Куранта-Фишера [8], несложно показать, что M_{λ_0} — выпуклое множество. Пусть $Q_0, \ Q_1 \in \mathrm{int} M_{\lambda_0}$. Введем следующие множества: $S_{\lambda_0} = B_{d(Q_1, \partial M_{\lambda_0})}(Q_1)$, $F_{\lambda_0} = B_{d(Q_1, \partial M_{\lambda_0})}(Q_0)$. При этом $S_{\lambda_0} \subset M_{\lambda_0}$, $F_{\lambda_0} \subset M_{\lambda_0}$, поскольку мы предположили, что начальное и конечное положение лежат внутри M_{λ_0}. Ясно, что если траектория полностью лежит в одном из этих множеств, то она удовлетворяет условиям (2), (3). Кроме того, если траектория попадет в S_{λ_0}, то она его уже не покинет, и нарушить условия (2), (3) не сможет.

Пусть t_F — момент покидания траекторией системы множества F_{λ_0}, а t_S — время попадания траектории во множество S_{λ_0}. Если мы сможем гарантировать, что $t_S < t_F$, то решение задачи будет удовлетворять фазовым ограничениям. Введем обозначения: $d_F = d(Q_0, \partial M_{\lambda_0})$, $d_S = d(Q_1, \partial M_{\lambda_0})$, $U_0 = \|Q_0 - Q_1\|$, $M^1 = \max\limits_{B_{U_0}(Q_0)} \|Q\|$, $M_2 = \min\limits_{Q \in B_{U_0}(Q_0) \setminus S_{\lambda_0}} (\nu \left\|U^d(Q)\right\|) + \mu \|R^o(Q)\|)$. Заметим, что по лемме$\tilde{1}$ $M_2 > 0$.

Теорема 1. *Если ограничения на управления μ, ν удовлетворяют соотношению*

$$(U_0^2 - d_S^2)(\mu + 2\nu M^1) \leqslant d_F M_2 \tag{13}$$

то $t_F \geqslant t_S$.

Доказательство. Оценим время, которое потребуется системе на выход из множества F_{λ_0}. Пусть $Q \in F_{\lambda_0}$. Тогда

$$d_F = \left\| \int\limits_{t_0}^{t_F} \dot{Q}(\tau) d\tau \right\| \leqslant \int\limits_{t_0}^{t_F} \|AQ + QA + V\| \, d\tau = (\nu + 2\mu M^1)(t_F - t_0),$$

откуда

$$t_F \geqslant t_0 + \frac{d_F}{\nu + 2\mu M^1}.$$

Оценим теперь время перехода на S_{λ_0}. Воспользуемся формулой Ньютона-Лейбница:

$$d_S^2 = \|Q(t_S) - Q_1\|^2 = \mathcal{V}(t_S) = \mathcal{V}(t_0) + \int\limits_{t_0}^{t_S} \dot{\mathcal{V}}(\tau) d\tau,$$

$$t_S \leqslant t_0 + \frac{\mathcal{V} - d_S^2}{\min\limits_{Q \in B_{U_0}(Q_0) \setminus S_{\lambda_0}} (\nu \|U^d(Q)\|) + \mu \|R^o(Q)\|)} \equiv t_0 + \frac{U_0^2 - d_S^2}{M_2}$$

Записывая условие $t_S < t_F$, получаем условие (13). □

В условии (13) фигурирует расстояние до границы множества M_{λ_0}. Искать его придется численно; однако поиск наименьшего собственного значения у матрицы размерности от трех и выше — задача численно трудоемкая. Поэтому вместо исследования поведения минимального собственного значения будем исследовать поведение некоторой величины, оценивающей это собственное значение снизу.

Лемма 5. *Пусть* $Q \in \mathbb{R}^{n\times n}$, $Q = Q^T > 0$. *Тогда*

$$\lambda_{\min}(Q) > \frac{\det Q}{\Delta_1 + \Delta_2 + \ldots + \Delta_n},$$

где $\Delta_1, \Delta_2, \ldots, \Delta_n$ *— алгебраические дополнения диагональных элементов матрицы* Q.

Доказательство. Введём матрицу $Q(\lambda) = Q - \lambda I$, где I — единичная матрица, и пусть $f(\lambda) = \det Q(\lambda)$, а $\hat{Q}$ — присоединенная к Q матрица. Воспользуемся формулой Лиувилля и тем фактом, что $Q^{-1} = \frac{\hat{Q}}{\det Q}$:

$$(\det Q(\lambda))' = \det Q(\lambda)\,\mathrm{tr}\left[\frac{dQ(\lambda)}{d\lambda}Q^{-1}(\lambda)\right] \Rightarrow$$

$$f'(0) = (\det Q(0))' = \det Q(0)\,\mathrm{tr}\left[\frac{dQ(0)}{d\lambda}Q^{-1}(\lambda)\right] =$$

$$= \det Q\,\mathrm{tr}\left[\begin{bmatrix} -1 & 0 & \ldots & 0 \\ 0 & -1 & \ldots & 0 \\ 0 & 0 & \ldots & -1 \end{bmatrix} Q^{-1}\right] =$$

$$= -\det Q\,\mathrm{tr}\left[\frac{\hat{Q}}{\det Q}\right] = -(\Delta_1 + \Delta_2 + \ldots + \Delta_n).$$

В силу положительной определенности матрицы Q, у $f(\lambda)$ есть набор вещественных корней суммарной кратности n, и, следовательно, на $(-\infty, \lambda_{\min}(Q))$ направление выпуклости она не меняет. Как λ^{2n}, так и $-\lambda^{2n+1}$ на этом участке выпуклы, и,следовательно, мы можем взять ноль касательной за искомую оценку. □

Заметим, что предложенную оценку можно рассматривать как первый шаг в решении уравнения $f(\lambda) = 0$ методом касательных (Ньютона). Продолжая этот процесс дальше, можно получить более тонкие оценки.

5 Решение в $\mathbb{R}^2$

Рассмотрим теперь более подробнее задачу при $n = 2$. Тогда $\mathcal{R}$ будет задаваться единственным уравнением

$$r_{12}(Q) \equiv 2q_{12} \operatorname{tr} U + u_{12} \operatorname{tr} Q = 0,$$

откуда ясно, что система всегда за конечное время будет попадать на $\mathcal{R}$. В системе (5) при $u_{11}^2 + u_{22}^2 = 0$ будем брать

$$v_{11} = v_{22} = 0. \tag{14}$$

Единственная нормаль будет иметь вид

$$N_{12} = \frac{\partial r_{12}}{\partial Q} = \begin{bmatrix} 2q_{12} - q_{12}^1 & 2\operatorname{tr} Q - \operatorname{tr} Q_1 \\ 2\operatorname{tr} Q - \operatorname{tr} Q_1 & 2q_{12} - q_{12}^1 \end{bmatrix}$$

Заметим, что если $N_{12} = 0$, то $Q = 2Q_1$ что в сочетании с $Q \in \mathcal{R}$ означает, что $q_{12}^1 = 0$ и, следовательно, $u_{12} = 0$, и можно положить $a_{12} = 0$.

Пусть теперь $N_{12} \neq 0$. Из системы (8) пполучаем

$$a_{12} = \frac{\nu \operatorname{tr} U}{\sqrt{u_{11}^2 + u_{22}^2}} \cdot \frac{2q_{12} - q_{12}^1}{4q_{12}(2q_{12}^1 - q_{12}^1) + 2\operatorname{tr} Q(2\operatorname{tr} Q - \operatorname{tr} Q_1)} \equiv$$
$$\equiv \frac{\nu \operatorname{tr} U}{\sqrt{u_{11}^2 + u_{22}^2}} \cdot K(Q). \tag{15}$$

Исследуем, когда эта формула не верна, т.е. когда знаменатель в последнем соотношении обнуляется. Это возможно в двух случаях:

1. $u_{11}^2 + u_{22}^2 = 0$, что влечет $U = 0$, и, следовательно, мы достигли желаемой цели;
2. Обнуляется знаменатель $K(Q)$. Запишем это условие вместе с условием $Q \in \mathcal{R}$:
$$\begin{cases} 4q_{12}(2q_{12} - q_{12}^1) + 2\operatorname{tr} Q(2\operatorname{tr} Q - \operatorname{tr} Q_1) = 0, \\ q_{12}(2\operatorname{tr} Q - \operatorname{tr} Q_1) - q_2^1 \operatorname{tr} Q = 0. \end{cases}$$

Если $2\operatorname{tr} Q - \operatorname{tr} Q_1 = 0$, то $q_{12}^1 = 0$ и $q_{12} = 0$, следовательно, можно положить $a_{12} = 0$. Аналогично при $q_{12}^1 = 0$.

Пусть теперь $2\operatorname{tr} Q - \operatorname{tr} Q_1 \neq 0$, $q_{12}^1 \neq 0$. Домножим первое уравнение на q_2, второе на $2\operatorname{tr} Q$ и вычтем одно из другого:

$$2q_{12}^2(2q_{12} - q_{12}^1) + q_{12}^1(\operatorname{tr} Q)^2 = 0. \tag{16}$$

В пределе имеем

$$2(q_{12}^1)^2 + (\operatorname{tr} Q_1)^2 \neq 0,$$

что означает, что с определенного момента знаменатель $K(Q)$ в ноль не обращается. До этого момента в таком случае можно положить $a_{12} = 0$.

Чтобы управление было допустимым, надо потребовать, что бы $|a_{12}| \leqslant \frac{\mu}{\sqrt{2}}$. При этом

$$K(Q_1) = \frac{(q_{12}^1)^2}{2(q_{12}^1)^2 + \operatorname{tr}^2 Q_1} = \frac{1}{2 + \frac{\operatorname{tr}^2 Q_1}{(q_{12}^1)^2}} \leqslant \frac{1}{2 + 2\frac{q_{11}^1 q_{22}^1}{(q_{12}^1)^2}} \leqslant \frac{1}{4}.$$

С определенного момента времени справедлива оценка

$$\left| \frac{\nu \operatorname{tr} U}{\sqrt{(u_{11}^0)^2 + (u_{22}^0)^2}} \cdot K(Q) \right| \leqslant \mu \frac{\sqrt{2}}{4}, \tag{17}$$

и нам достаточно потребовать, что бы

$$\frac{\nu}{\sqrt{2}} \geqslant \mu \frac{\sqrt{2}}{4}. \tag{18}$$

Заметим, что условие (18) не включает в себя абсолютные величины матриц Q_0, Q_1.

Покажем теперь, что при таком выборе a_{12} за конечное время обновлятся диагональные отклонения. Рассмотрим

$$\frac{d}{dt}\left(u_{11}^2+u_{22}^2\right)=2(u_{11}\dot{u}_{11}+u_{22}\dot{u}_{22})=$$
$$=2\left(\frac{-\nu(u_{11}^2+u_{22}^2)}{\sqrt{u_{11}^2+u_{22}^2}}+\frac{\nu K(Q)(u_{11}+u_{22})^2}{\sqrt{u_{11}^2+u_{22}^2}}\right)\leqslant$$
$$\leqslant\{(a+b)^2\leqslant 2(a^2+b^2)\}\leqslant$$
$$\leqslant 2\nu\left(-\sqrt{u_{11}^2+u_{22}^2}+2K\sqrt{u_{11}^2+u_{22}^2}\right)=$$
$$=-2\nu\sqrt{u_{11}^2+u_{22}^2}(1-2K),$$

но, с определенного момента, $1-2K\geqslant\frac{1}{2}$. Тогда, обозначая $u_{11}^2+u_{22}^2=y$, мы приходим к мажорирующему уравнению $\dot{y}=-A\sqrt{y}$. Его траектории приходят в ноль за время $2\sqrt{y(t_0)}/\nu$.

При этом за оценку времени достижения цели можно взять величину

$$T_*=t_0+\frac{2\sqrt{(u_{11}^0)^2+(u_{22}^0)^2}}{\nu}+\frac{|q_{12}^0|\sqrt{2}}{\mu\min\limits_{B_{U_0}(Q_0)\cap M_{\lambda_0}}\operatorname{tr}Q}. \qquad (19)$$

Эта оценка справедлива в том случае, когда знаменатель в формуле (15) не обнуляется. Если же это так, то к T^* надо добавить интервал времени, начиная с которого он не обнуляется. Оценим это время. Для обнуления требуется выполнение (16). Введём функцию $h(Q)=2q_{12}^2(2q_{12}-q_{12}^1)+q_{12}^1(\operatorname{tr}Q)^2$; $h(Q_1)\neq 0$, и мы можем оценить радиус шарика с центром в Q_1, внутри которого $|h(Q)|>0$. Пусть $\|Q-Q_1\|<\varepsilon$. Тогда

$$|h(Q)|\geqslant|q_2^1||(\operatorname{tr}Q)^2|-2|q_{12}|^2\cdot|2q_{12}-q_{12}^1|\geqslant$$
$$\geqslant|q_{12}^1|(\operatorname{tr}Q_1-2\varepsilon)^2-2(q_{12}^1+\varepsilon)^2\cdot(2|q_{12}-q_{12}^1|+|q_{12}^1|)\geqslant$$
$$\geqslant|q_{12}^1|(\operatorname{tr}Q_1-2\varepsilon)^2-2(|q_{12}^1|+\varepsilon)^2(2\varepsilon+|q_{12}^1|) \qquad (20)$$

Таким образом, нам подойдет $\hat{\varepsilon} = \max\{\varepsilon > 0 : \ |q_{12}^1|(\operatorname{tr} Q_1 - 2\varepsilon)^2 - 2(|q_{12}^1| + \varepsilon)^2(2\varepsilon + |q_{12}^1|) > 0\}$. При этом, если $\hat{\varepsilon} > \|Q_0 - Q_1\| \equiv U_0$, то поправку на обнуление знаменателя можно не делать.

Оценим время t', необходимое системе для достижения $B_{\hat{\varepsilon}}(Q_1)$. Воспользуемся формулой Ньютона-Лейбница для функции $V(Q(t))$:

$$V(t') - V(t_0) = \int\limits_{t_0}^{t'} \dot{V}(Q(\tau))d\tau \leqslant (t' - t_0) \max_{\tau\in[t_0,t']} \dot{V}(Q(\tau)),$$

$$t' \leqslant t_0 + \frac{U_0^2 - \hat{\varepsilon}^2}{\max\limits_{\tau\in[t_0,t']} \dot{V}(Q(\tau))} =$$

$$= t_0 + \frac{U_0^2 - \hat{\varepsilon}^2}{2\min\{2|r_{12}| + \nu\sqrt{u_{11}^2 + u_{22}^2}\}} \leqslant t_0 + \frac{U_0^2 - \hat{\varepsilon}^2}{2\nu\hat{\varepsilon}}.$$

С учетом этой поправки, можно усовершенствовать оценку (19) до

$$T^* = t_0 + \frac{2\sqrt{(u_{11}^0)^2 + (u_{22}^0)^2}}{\nu} + \frac{|q_{12}^0|\sqrt{2}}{\mu \min \operatorname{tr} Q} + \max\left\{\frac{U_0^2 - \hat{\varepsilon}^2}{2\nu\hat{\varepsilon}}, 0\right\} \tag{21}$$

Подводя итог, сформулируем теорему.

Теорема 2. *Для любых симметричных и положительно определенных матриц Q_0, Q_1 закон управления, определяемый формулами* (5), (14), (15) *и удовлетворяющий условию* (18), *переводит Q_0 в Q_1 за конечное время t_1, для которого справедлива оценка* (21)*: $t_1 \leqslant T^*$.*

Заметим, что, вообще говоря, выбор управления V в особом режиме при условии $u_{11}^2 + u_{22}^2 = 0$ можно осуществлять не обязательно по формуле (14), а любым образом, не препятствующим выходу на многообразие $\mathcal{R}$.

6 Задача движения контейнера

В этом пункте мы опишем простой алгоритм, решающий задачу движения эллипсоидального контейнера при ограничениях (4).

Относительно препятствий сделаем предположение:

$$\min_{i \neq j} d(\mathcal{E}\left(p_i, P_i\right), \mathcal{E}\left(p_j, P_j\right)) = \Lambda > 0. \tag{22}$$

Предполагается, что препятсвия "существенны" в том смысле, что нам выгодно двигаться между ними, а не объезжать их все сразу.

Для решения задачи введем вспомогатльные эллипсоиды $\mathcal{E}\left(r_i, R_i\right),\ i = 1, 2, \ldots m$, такие, что

$$d(\mathcal{E}\left(r_i, R_i\right), \mathcal{E}\left(p_i, P_i\right)) \geqslant \varepsilon,\ \Lambda > 3\varepsilon,\ \forall i,$$
$$\mathcal{E}\left(r_i, R_i\right) \cup \mathcal{E}\left(r_j, R_j\right) = \varnothing,\ i \neq j.$$

Будем называть эти эллипсоидлы безопасными зонами, их границы — орбитами препятствий.

Кроме того, введем шар $F = B_d(q_1)$, где $d = \min_i d(q_1, \mathcal{E}\left(r_i, R_i\right))$. Этот шар будем предполагать невырожденным и будем интерпретировать как множество, при попадании всего контейнера в которое можно не обращать внимания на препятствия.

Будем обозначать через $\mathcal{E}(d, l_1, l_2)$ эллипсоид с собственными значениями l_1 l_2 и собственным вектором d, отвечающим собственному значению l_1 ("направление").

Опишем общий алгоритм решения. Обозначим текущее состояние контейнера через $\mathcal{E}\left(q, Q\right)$.

1. Проложить луч r от q до q_1. Если он не пересекает ни одной орбиты препятствий, то принимаем форму $\mathcal{E}(q_1 - q, l, \varepsilon)$, где l — некоторое удобное нам число. После этого по прямой едем до F, и, полностью оказавшись в нем, беспрепятственно принимаем форму Q_1.

2. Если луч пересек какую-либо орбиту $\partial\mathcal{E}\left(r_k, R_k\right)$ в точке M, то находим на орбите точку M' такую, что

$$M' = \operatorname{Argmin}\Big\{ d(L, M) : \ L \in \partial\mathcal{E}\left(r_k, R_k\right), \\ \text{луч } r(M, M - q_1) \cap \mathcal{E}\left(r_k, R_k\right) = \varnothing \Big\}$$

Пусть касательный вектор N в точке M смотрит в сторону наименьшей дуги эллипса MM'. Тогда принимаем форму $\mathcal{E}(N, 3\varepsilon, \varepsilon)$ и передвигаемся в точку M, где скорость центра должна быть сонаправлена с N'.

3. Перемещаемся по орбите из M в M' следующим образом. Параметризуем $\partial\mathcal{E}\left(r_k, R_k\right)$:

$$\partial\mathcal{E}\left(r_k, R_k\right) = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^2 : \ \begin{bmatrix} x \\ y \end{bmatrix} = V \begin{bmatrix} \sqrt{\lambda_1}\sin(\alpha) \\ \sqrt{\lambda_2}\cos(\alpha) \end{bmatrix} V^T \right\},$$

где столбцы матрицы V — собственные вектора $\mathcal{E}\left(r_k, R_k\right)$, отвечающие собственным значениям $\lambda_1,\ \lambda_2$, а $\alpha \in [0, 2\pi)$. Пусть в этой параметризации M отвечает угол α, M' — угол β, R — матрица поворота на угол $\alpha - \beta$. Тогда, принимая форму $R\mathcal{E}\left(N, 3\varepsilon, \varepsilon\right)R^T$, двигаемся из M в M', положив при этом

$$u = -V \begin{bmatrix} \sqrt{\lambda_1}A^2\cos(\alpha + A(t - t_0)) \\ \sqrt{\lambda_2}A^2\sin(\alpha + A(t - t_0)) \end{bmatrix} V^T,$$

где t_0 — время первоначального попадания на орбиту, A — частота поворота, определяемая из соотношения

$$\beta = \alpha + A(t_1 - t_0),$$

где t_1 — время, к которому закончится смена матрицы конфигураций.

4. Оказавшись в M', переходим к пункту 1.

Управление центром можно строить обычными методами оптимального управления или динамического программирования [2, 3].

7 Численные примеры

Проиллюстрируем работу управления, полученного в разделах 3, 5. Пусть $t_0 = 0$, $\mu = \nu = 1$. Для первого примера возьмем при $n = 2$

$$Q_0 = \begin{bmatrix} 2 & 0,1 \\ 0,1 & 1 \end{bmatrix}, Q_1 = \begin{bmatrix} 8 & 2 \\ 2 & 8 \end{bmatrix},$$

Траектория достигает $\varepsilon = 0,4 \cdot 10^{-4}$ –окрестности нуля за $6,9968$. Второй пример, $n = 3$:

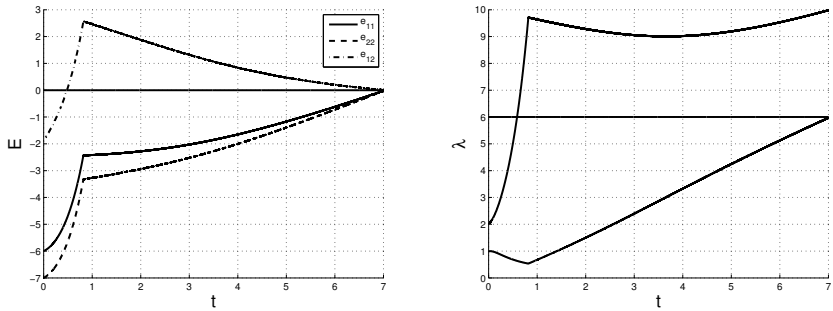


Рис. 1. Результаты работы на первом примере: отклонения компонент матрицы и ее собственные значения.

$$Q_0 = \begin{bmatrix} 8 & 1 & -1 \\ 1 & 1 & 0 \\ -1 & 0 & 2 \end{bmatrix}, Q_1 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 6 \end{bmatrix}.$$

Траектория достигает конечной точки к моменту $6,8764$. Результаты работы изображены на рисунке 2.

В обоих примерах точке излома траекторий отвечает момент выхода на множество $\mathcal{R}$.

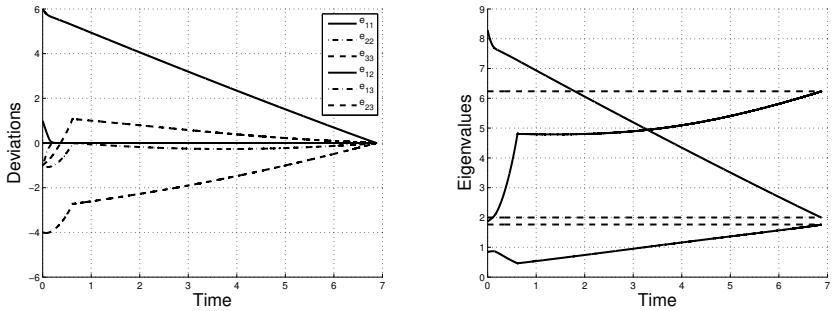


Рис. 2. Результаты работы на втором примере: отклонения компонент матрицы и ее собственные значения.

8 Заключение

В настоящей работе предложена математическая модель управления эллипсоидальным контейнером в $\mathbb{R}^n$, приводится метод решения задачи совмещения конфигураций в $\mathbb{R}^n$, даются более подробные результаты для $\mathbb{R}^2$. Сформулировано достаточное условие для соблюдения ограничений на размер контейнера.

Список литературы

[1] *Kurzhanski A.B., Varaiya P.* On synthesizing target controls under obstacle and collision avoidance // Journal of Franklin Institute. — 2010. — Vol. 347, no 1. – P.130–145. — doi:10.1016/j.franklin.2009.11.002

[2] *Куржанский А.Б.* Избранные труды. М., Издательство МГУ, 2009. — 756 с. — ISBN 978-5-211-05758-6.

[3] *Красовский Н.Н.* Теория управления движением. М., Наука, 1968. — 476 с.

[4] *Чебунин И.В.* Условия управляемости для уравнения Рикатти. // Дифференциальные уравнения. — 2003. — Т.39, №12. — С. 1654-1661.

[5] *Pettersen K.Y., Gravdahl J.T., Nijmeijer H.* (Eds.) Group Coordination and Cooperative Control. Berlin: Springer, 2006. — 323 p. — ISBN 978-3540334682.

[6] *Козлов Р.И., Максимкин Н.Н., Кисилев Л.В., Ульянов С.А.* Устойчивость конфигураций группового движения автономных поводных роботов в условиях неопределенности // Подводные роботы и робототехника. — 2010. — №19. — С.40–46

[7] *Reza Olfati-Saber.* Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory. // IEEE Transactions on automatic control. — 2006. — Vol. 51, no. 3 — P.401–420.

[8] *Тыртышников Е.Е.* Матричный анализ и линейная алгебра. М., ФИЗМАТЛИТ, 2007. — 476 с. — ISBN 978-5-9221-0778-5.

[9] *Васильев Ф.П.* Методы оптимизации, том 2. М., МЦНМО, 2011. — 433 с. — ISBN 978-5-94057-708-9

[10] *Гантмахер Ф.Р.* Теория матриц. М., Физматлит, 2010. — 366 с. — ISBN 5-9221-0524-8.

УДК 004.891

Схемы оценки уверенности в экспертных системах

© 2012 г. Н. А. Моросанова

nmorosanova@gmail.com

Кафедра алгоритмических языков

1 Введение

Важным компонентом экспертной системы является ее интерфейсная часть, которая обеспечивает получение информации от пользователя и представление результата в виде, понятном для пользователя. Обычно интерфейс экспертной системы использует некоторое подмножество естественного языка [1]. Полученная от пользователя информация может обладать следующими свойствами [2]:

1. быть верной только с некоторой вероятностью;
2. быть верной по мнению эксперта только в некоторой степени;
3. быть неточной по своей природе;
4. быть неполной или противоречивой.

Для экспертных систем, основанных на правилах, дополнительно рассматривается неопределенность в правилах. Ее источниками могут служить в частности [3]:

1. ошибки при составлении отдельных правил;
2. проблемы совместимости правил: избыточность, противоречивость правил и др.

Перечисленные обстоятельства приводят к необходимости иметь некоторый способ описания уверенности в полученной информации, как при формулировке правил, так и в процессе консультации пользователя. Для этого при создании экспертной системы заранее определяется некоторая схема оценки уве-

ренности. Эта схема используется в процессе извлечения знаний для составления базы знаний, а также может влиять на организацию логического вывода. Как правило, схема оценки уверенности состоит из нескольких частей:

1. способа описания степени уверенности;
2. правил преобразования введенной пользователем информации для описания уверенности указанным способом;
3. правил преобразования полученной в результате вывода информации для получения вида, удобного для пользователя.

2 Схемы оценки уверенности

Подходы к описанию уверенности развивались и до появления экспертных систем, достаточно привести, например, Байесовскую теорию, теорию Демпстера-Шефера и теорию размытых множеств. Эти подходы применялись в некоторых экспертных системах [4–6].

Однако в связи развитием экспертных систем появились другие подходы, например, коэффициенты уверенности (CF), логические программы с неопределенностью, теория подтверждения (endorsement theory). Рассмотрим далее эти подходы к описанию уверенности более подробно.

Схема оценки уверенности, основанная на коэффициентах уверенности, была предложена в экспертной системе MYCIN [7], и также она называется схемой Шортлиффа.

Для описания степени уверенности используется число из отрезка $[-1; 1]$, именуемое коэффициентом уверенности, причем -1 означает полную уверенность в опровержении информации, а 1 — полную уверенность в ее подтверждении.

Над коэффициентами уверенности определены следующие операции и отношения:

— $\min(a, b)$, $\max(a, b)$, $\mathrm{not}(a)$ — операции для логического вывода, обобщающие конъюнкцию, дизъюнкцию и отри-

цание.

— Отношения сравнения с пороговыми значениями $\pm 0{,}2$. Эти пороговые значения определяют значимость соответствующей информации (т.е. если $|CF| < 0{,}2$, то информация не значима и не используется в дальнейшем выводе).
— $\mathrm{tmx}(a, b)$ — операция ослабления для учета уверенности в правилах, с помощью которых происходит вывод.
— $\mathrm{cmb}(a, b)$ — операция комбинирования при наличии нескольких источников информации.

Преобразование коэффициентов уверенности в понятный для пользователя вид требуется в случае ввода информации пользователем-экспертом и вывода результата.

При вводе информация от пользователя поступает в виде ответов на вопросы системы. Ответ состоит из двух частей: смысловая часть («да», «нет», «неизвестно») и уверенность в ответе, представленная в виде целого числа от -10 до 10. Коэффициент уверенности определяется следующим образом: полученное число делится на 10, а в том случае, если дан ответ «неизвестно», то он равен нулю.

Для вывода информации определяются пороговые значения для коэффициентов уверенности, полученных в результате логического вывода:

1. $|CF| \geqslant 0{,}8$: «с уверенностью»,
2. $0{,}4 \leqslant |CF| < 0{,}8$: «предположительно»,
3. $|CF| < 0{,}4$: «слабое предположение».

Использование пороговых значений позволяет переводить количественные значения, используемые в процессе логического вывода, в текстовый вид.

Следующий подход [8] основан на добавлении неопределенности в аппарат логических программ для того, чтобы использовать этот аппарат для организации экспертных систем. Для описания степени уверенности используются коэффициенты уверенности — числа из полуинтервала (0; 1]. Над этими числами определяется функция уверенности f, аргументом

которой служит конечное множество коэффициентов уверенности, а результатом является коэффициент уверенности. Эта функция обязана удовлетворять следующим требованиям:

1. $f(S \cup \{1\}) = f(\{1\})$;
2. если $S < S'$, то $f(S) < f(S')$, где $S < S'$ означает, что существует такое множество Y, для которого $|S| = |Y| = n$, $S' \subset Y$ и возможно разбить элементы множеств Y и S на непересекающиеся пары так, что для всех пар $s_i < y_i$.

Логическая программа с неопределенностью P записывается как конечное множество пар вида $< A \leftarrow B, f >$. Функция f определяет уверенность в самом правиле $< A \leftarrow B >$ как $f(\emptyset)$ и также способ получения уверенности заключения по значениям коэффициентов уверенности посылки.

Для управления выводом вводится пороговое значение r коэффициента уверенности, которое используется для отсечения тех ветвей вычислений, которые не дают значение коэффициента уверенности для цели вывода, большее, чем указанное значение. Результат вывода представляется в виде: «В рамках P цель A может быть доказана с уверенностью C, причем $C > r$».

Подход, определяемый в рамках теории подтверждения [9], основан на том, что степень уверенности не представляется в количественном виде. Способ описания степени уверенности заключается в следующем:

— выделяются наименования для групп убеждений (например, «сильные», «слабые», «опровержимые»);
— определяются наименования, получающиеся в результате комбинирования исходных для некоторой информации-гипотезы наименований (например, «вероятный», «противоречивый»);
— определяется порядок введенных наименований по степени уверенности;
— определяется достаточный уровень уверенности для того, чтобы считать гипотезу подтвержденной. Этот уровень

может отличаться для разных гипотез.

Поскольку при таком подходе вывод оперирует значениями-названиями, то преобразований для интерфейсной части не требуется.

3 Выбор схемы оценки уверенности

Выбор конкретной схемы оценки уверенности является подзадачей, возникающей при разработке экспертной системы.

С одной стороны, можно изначально зафиксировать некоторую схему для того, чтобы она использовалась экспертами и в процессе извлечения знаний, и при последующем использовании системы. При таком подходе можно использовать для вывода эту же выбранную схему. Преимуществом такого подхода является возможность использования единой схемы на всех этапах создания и использования экспертной системы. Недостатком может стать то обстоятельство, что разные эксперты используют разные схемы оценки своей уверенности в фактах, однако оказываются вынужденными использовать выбранную заранее схему.

С другой стороны, можно попробовать подобрать нужную схему для каждого эксперта, а для вывода и хранения в базе знаний преобразовывать данные к какому-либо единому виду. При данном подходе возникает задача выбора нужной для эксперта схемы. Для того чтобы упростить решение этой задачи, можно

— либо искать схемы в рамках некоторого класса схем,
— либо составить некоторый список модельных задач для выявления используемой схемы.

Преимуществом этого подхода может стать большая гибкость в описании степени уверенности. Недостатком этого подхода является возникновение дополнительных задач выбора схем и составления списка вопросов для выявления схемы, используемой экспертом, а также способа преобразования схем оценки

уверенности к общему виду.

Задача преобразования данных, представленных с помощью различных схем оценки уверенности, возникает также при необходимости совместного использования различных схем, например, в распределенной экспертной системе [10].

4 Изменяемая схема Шортлиффа

С помощью преобразований схем оценки уверенности возможно совместное использование различных схем. Рассмотрим общие требования к определению механизма таких преобразований, а также более подробно преобразования конкретной схемы — схемы Шортлиффа.

4.1 Преобразования схем оценки уверенности

Рассмотрим задачу преобразования схем оценки уверенности друг в друга. В рамках каждой схемы определяется способ описания уверенности:

— множество значений, которые описывают степень уверенности в факте,
— допустимые операции, определенные над этим множеством.

Если рассматривать схемы с количественным описанием степени уверенности, то в них, как правило, определяется операция комбинирования оценок уверенности, которая используется для совмещения информации об одном факте из различных источников. Это одна из основных операций, определяющих результат логического вывода в экспертной системе.

Пусть операция комбинирования может быть представлена в виде функции $\mathrm{cmb}(x, y) = z$, где x, y и z принадлежат одному множеству значений. Все такие функции могут быть классифицированы по значению z [11] на:

1. конъюнктивные: $z \leqslant \min(x, y)$;

2. дизъюнктивные: $z \geqslant \max(x, y)$;
3. компромиссные: $\min(x, y) < z < \max(x, y)$.

Можно отметить, что некоторые из операций комбинирования нельзя отнести только к одному из перечисленных классов в том случае, если отношение значений $\min(x, y)$, $\max(x, y)$ и $\mathrm{cmb}(x, y)$ зависит от значений аргументов x и y. Например, комбинирование в схеме Шортлиффа:

1. конъюнктивно, если $x, y \in [-1; 0)$;
2. дизъюнктивно, если $x, y \in [0; 1]$;
3. является компромиссным, если x и y имеют разный знак.

Также операции комбинирования могут отличаться наличием некоторых из следующих свойств:

— ассоциативность;
— коммутативность;
— идемпотетность;
— монотонность.

Задача преобразования схем оценки уверенности друг в друга может быть решена с помощью отображения множества значений одной схемы на множество значений другой схемы. В таком множестве, как правило, присутствуют некоторые особо выделенные значения [10]: «истина», «ложь» и нейтральный элемент ϵ, такой, что для любых x $\mathrm{cmb}(x, \epsilon) = x$, а также на этом множестве задано отношение порядка. Пусть отображение задается с помощью функции h. Тогда к h можно предъявить следующие требования [10]:

1. сохранение результата комбинирования: $h(\mathrm{cmb}_1(x, y)) = \mathrm{cmb}_2(h(x), h(y))$;
2. сохранение «особых» значений: $h(\text{«истина»}) = \text{«истина»}$, $h(\text{«ложь»}) = \text{«ложь»}$, $h(\epsilon_1) = \epsilon_2$;
3. сохранение отношения порядка: если $x > y$, то $h(x) > h(y)$.

Перечисленные требования вводились [10] для решения задачи совместного использования различных схем оценки уверенности, поэтому необходимо было только соответствие в накопле-

нии информации об уверенности в фактах для возможности обмена ей. Это соображение привело к тому, что в первом из требований указана только операция комбинирования. В случае, когда требуется полное соответствие схем оценки уверенности для проведения логического вывода, подобные требования могут быть предъявлены и к другим определенным операциям.

Таким образом, h — это гомоморфизм. Если рассматривать только взаимно-однозначные отображения, то h станет изоморфизмом.

Совокупность первого и третьего требований означает, что преобразование схем уверенности друг в друга возможно, только если операции комбинирования в них принадлежат одному классу по значению $\mathrm{cmb}(x, y) = z$. Другие свойства операций комбинирования, такие, как ассоциативность или идемпотентность, так же сохраняются при отображении h. Это означает, что совпадение основных свойств операций комбинирования, определенных в рамках двух схем оценки уверенности, является необходимым условием существования изоморфной трансформации между этими схемами.

4.2 Изоморфные трансформации схемы Шортлиффа

Рассмотрим изоморфные трансформации одной из распространенных схем оценки уверенности, называемой также схемой Шортлиффа. В литературе [10] рассматривается изоморное преобразование этой схемы для доказательства (с некоторыми ограничениями) ее соответствия схеме оценки уверенности, основанной на байесовском подходе. Рассмотрим произвольные изоморфные трансформации этой схемы.

С помощью взаимно-однозначных отображений $h : [-1; 1] \to \Re$, где $\Re$ — некоторое множество значений, задаются трансформации схемы Шортлиффа [12]. Такие трансформации порождают целый класс изоморфных схем оценки уверенности. Задача выбора конкретной трансфор-

мации возникает в том случае, если требуется использовать несколько различных схем оценки уверенности совместно. Эта задача распадается на несколько подзадач.

Во-первых, требуется выбрать множество $\Re$ — множество значений, которые может принимать коэффициент уверенности факта. Эти значения могут зависеть, в том числе, от природы фактов, которые рассматриваются, от значений фактов, имеющих какое-то количественное описание. Во многих случаях это множество может быть определено в явном виде.

Во-вторых, имея уже определенное множество $\Re$, требуется выбрать функцию h, от которой зависит, какими будут образы операций и отношений над коэффициентами уверенности. Таким образом, h определяет, как коэффициент уверенности результата правила зависит от коэффициентов уверенности в его посылке.

На основе имеющихся данных можно попробовать подобрать функцию h. Для проверки пригодности функций-кандидатов можно составить некоторое количество модельных продукционных правил и сравнивать требуемые результаты с результатами вывода в трансформации схемы Шортлиффа, задаваемой с помощью функции-кандидата.

Для того, чтобы определить этот класс функций-кандидатов, можно сформулировать некоторые требования к функции h, например:

— монотонно возрастающая;
— достаточно гладкая;
— сохраняющая существенную разницу между пороговыми значениями.

Исходя из условий конкретной задачи, могут быть сформулированы какие-либо дополнительные требования.

Будем обозначать операции в трансформации схемы Шортлиффа, получаемой при отображении $h(x)$ с помощью префикса «$h|$». Так как $h(x)$ является изоморфизмом, то

$$h(\mathrm{cmb}(a,b)) = h|\,\mathrm{cmb}(h(a),h(b)). \qquad (1)$$

В схеме Шортлиффа определяются особые значения-пороги, которые имеют четко заданный смысл. Поэтому эти значения, возможно, наиболее просты для идентификации в случае поиска трансформации схемы.

Рассмотрим некоторые из ситуаций, которые возникают при решении задачи выбора конкретной трансформации:

1. Известны $h(x)$, $h(\mathrm{cmb}(x,x))$, $h(\mathrm{cmb}(x,\mathrm{cmb}(x,x)))$ и т.д., где $x = \pm 0,2$, то есть значения коэффициентов уверенности в том случае, если несколько независимых источников информации свидетельствуют о значении коэффициента уверенности, соответствующем пороговому значению.
2. Известен вид функции комбинирования $h|\,\mathrm{cmb}$, возможно в параметрическом виде.

Тогда на основе равенства (1) можно получить некоторую функцию-кандидат на h. Рассмотрим примеры обеих ситуаций.

Пример 1

Пусть коэффициенты уверенности являются натуральными числами от 1 до 100 (100-балльная система) и известны некоторые значения: $h(-1) = 1$, $h(0) = 50$ и $h(1) = 100$, а также значения, являющиеся порогами подтверждения и опровержения из нескольких (от 1 до 4) источников:

Таблица 1. Известные пороговые значения для $h(x)$

Подтверждение	Опровержение	Количество источников
75	38	1
80	35	2
85	30	3
90	20	4

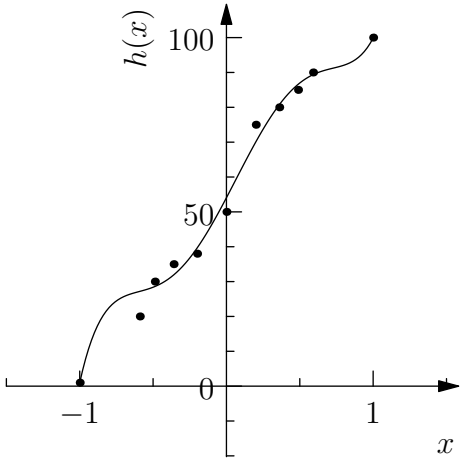
Приближение функции h можно получить, например, в ви-

де многочлена с помощью полиномиальной регрессии:

$$h(x) = 73{,}93(x+1)^5 - 393{,}8(x+1)^4 + \\ + 731{,}25(x+1)^3 - 549{,}73(x+1)^2 + \\ + 191{,}41(x+1) + 1.$$

Имеющиеся значения и приближение $h(x)$ на координатной плоскости выглядят следующим образом:

Рис. 1. Приближение h(x)



Пример 2

Для $h|\,\mathrm{cmb} = h(a) + h(b)$ и $\Re = [-\infty; +\infty]$ получаем [12] параметрическое семейство отображений h:

$$h(x,\beta) = \begin{cases} -\infty & x = -1; \\ +\log_\beta(1+x) & -1 < x < 0; \\ -\log_\beta(1-x) & 0 \leqslant x < 1; \\ +\infty & x = 1. \end{cases}$$

5 Заключение

Существует множество схем оценки уверенности для использования в экспертной системе, среди которых схема Шортлиффа является одним из известных примеров. Аппарат изоморфных трансформаций этой схемы оценки уверенности позволяет использовать индивидуальные схемы оценки уверенности для каждого пользователя, а также использовать имеющиеся системы, использующие эту схему, совместно с системами, использующими какую-либо другую схему. При этом возникает дополнительная задача определения отображения, осуществляющего требуемую трансформацию классической схемы Шортлиффа, которая может быть решена на основе некоторой дополнительной информации.

Поскольку аппарат трансформаций применим к любой схеме оценки уверенности, использующей количественное представление степени уверенности, то возможно его применение для других схем оценки уверенности. Интересной задачей является обнаружение изоморфных соответствий между различными схемами оценки уверенности.

Список литературы

[1] *Hendler J. A.* Expert Systems: The User Interface. — Norwood, NJ: Ablex Pub., 1988. — 324 p. — ISBN 0893914290.

[2] *Grzymala-Busse J. W.* Managing Uncertainty in Expert Systems. — Boston: Kluwer Academic, 1991. — 224 p. — ISBN 079239169.

[3] *Джарратано Д., Райли Г.* Экспертные системы: принципы разработки и программирование. — М.: Вильямс, 2006. — 1152 с. — ISBN 978-5-8459-1156-8, 0-534-38447-1.

[4] *Petrovic S., Mishra N., Sundar S.* A novel case based reasoning approach to radiotherapy planning // Expert Systems with Applications. — 2011. — Vol. 38, no. 9. — P. 10759–10769. — doi: 10.1016/j.eswa.2011.01.109.

[5] *Spiegelhalter David J., Dawid A. Philip, Lauritzen Steffen L., Cowell Robert G.* Bayesian Analysis in Expert Systems // Statistical Science. — 1993. — Vol. 8, no. 3. — P. 219–247. — doi: 10.1214/ss/1177010888.

[6] *Cheung W., Pitcher T., Pauly D.* A Fuzzy Logic Expert System to Estimate Intrinsic Extinction Vulnerabilities of Marine Fishes to Fishing // Biological Conservation. — 2005. — Vol. 124, no. 1. — P. 97–111. — doi:10.1016/j.biocon.2005.01.017.

[7] *Buchanan B. G., Shortliffe E. H.* Rule-based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. — Reading, MA: Addison-Wesley, 1984. — 748 p. — ISBN 0201101726.

[8] *Shapiro Ehud Y.* Logic programs with uncertainties: a tool for implementing rule-based systems // International joint conference on Artificial intelligence: Proceedings of the eighth international conference. — 1983. — Vol. 1. — P. 529–532.

[9] *Ng K.-C., Abramson B.* Uncertainty Management in Expert Systems // IEEE Expert. — 1990. — Vol. 5, no. 2. — P. 29–48. — doi:10.1109/64.53180.

[10] *Luo X., Zhang C., Leung H.-F.* Information sharing between heterogeneous uncertain reasoning models in a multi-agent environment: A case study // International Journal of Approximate Reasoning. — 2001. — Vol. 27, no. 1. — P. 27–59. — doi:10.1016/S0888-613X(01)00032-9.

[11] *Bloch I.* Information combination operators for data fusion: a comparative review with classification // IEEE Transactions

on Systems, Man, and Cybernetics — Part A: Systems and Humans In Systems. — 1996. — Vol. 26, no. 1. — P. 52–67. — doi:10.1109/3468.477860.

[12] *Моросанова Н. А., Соловьев С. Ю.* Формальные свойства схемы Шортлиффа // Управление большими системами. — 2012. — Вып. 36. — С. 5–38.

УДК 519.688

Построение множеств достижимости некоторых управляемых систем пиксельным методом

© 2012 г. А. О. Новикова

novikova.a.o@gmail.com

Кафедра оптимального управления

1 Введение

Оптимальное управление охватывает широкий круг задач, в которых при определенных ограничениях на ресурсы требуется минимизировать (максимизировать) заданный критерий качества. Задачи оптимального управления встречаются в различных областях науки, техники, медицины, экономики, экологии. Например, актуальными являются задачи ядерной энергетики (управление охлаждением реактора), робототехники (движение роботов, управление всевозможными станками и автоматами), механики полета (самонаводящиеся ракеты, автопилоты, автоматическая стыковка на орбите, управление самолетом), экономики (задачи долговременного планирования), экологии (расчет допустимого воздействия на экосистему), биофизики и медицины (модель распространения эпидемии) и т. д.

В задачах оптимального управления важнейшую роль играет множество достижимости. Оно характеризует все возможные положения управляемой системы в каждый момент времени.

В данной работе изучен пиксельный метод построения множеств достижимости для плоских управляемых систем. Так как эти алгоритмы позволяют эффективно разделить процедуру вычислений на множество независимых параллельных процессов, то для программирования используется тех-

нология CUDA (Compute Unified Device Architecture) [5, 6] с Си-подобным языком программирования. Использование такой технологии [4] помогает уменьшить время вычислений в несколько десятков раз.

2 Постановка задачи оптимального управления

В рассматриваемой задаче оптимального управления [1] требуется перевести объект из множества начальных состояний M_0 на множество конечных состояний M_1 за счет выбора допустимого управления $u = u(t)$ из класса кусочно-непрерывных функций допустимых управлений Y_U с областью управления U так, чтобы функционал J принимал минимальное значение. Управление $u(t)$, решающее поставленную задачу, оптимально в смысле интегрального функционала

$$J = \int_{t_0}^{t_1} f^0(t, x(t), u(t))dt.$$

Траектория $x(t)$, отвечающая оптимальному управлению $u(t)$, называется оптимальной траекторией. Таким образом, постановка задачи оптимального управления принимает следующий вид:

$$\begin{cases} \dot{x} = f(t, x, u), & u(t) \in U, \\ x(t_0) \in M_0, & x(t_1) \in M_1, \\ J \to \min\limits_{u(\cdot) \in Y_U} . \end{cases}$$

Рассматриваемая задача требует задания следующего набора исходных данных: $\{ f_0, f, Y = Y_U; M_0, M_1, t_0 \}$, где f_0, f — непрерывные по совокупности переменных x, u и непрерывно дифференцируемые по x функции [1], множества M_0, M_1 выпуклы и компактны, множество U компактно.

3 Множество достижимости

3.1 Определение

При изучении задачи оптимального управления большое значение имеет множество достижимости $X(t_0, t, M_0)$. Это множество зависит от следующего набора параметров:

- M_0 — множество начальных значений
- t_0 — начальный момент времени
- $t \geqslant t_0$ — рассматриваемый момент времени
- f — нелинейная функция, (в линейном случае зависящая от матрицы A)
- $Y = Y_U$ — класс допустимых управлений

Множество достижимости отвечает на вопрос: куда можно перейти к моменту времени t по траекториям дифференциального уравнения $\dot{x} = f(t, x, u)$, исходящим в начальный момент времени t_0 из различных точек множества начальных значений M_0 при использовании всевозможных управлений $u(\cdot) \in Y$. Множество концов всех таких траекторий в момент времени t образует множество достижимости — $X(t_0, t, M_0)$.

Получим формулу для аналитического представления множества достижимости в случае линейной управляемой системы:

$$\begin{cases} \dot{x} = Ax + u, \quad u(t) \in U, \\ x(t_0) \in M_0. \end{cases}$$

Рассмотрим задачу Коши

$$\begin{cases} \dot{x} = Ax + u(t), \quad t_0 \leqslant t, \\ x(t_0) = x_0, \end{cases}$$

где функция $u(t)$ задана.

Формула Коши для её решения выглядит следующим образом:

$$x(t) = e^{(t-t_0)A}x_0 + \int_{t_0}^{t} e^{(t-s)A}u(s)ds.$$

Тогда формула для множества достижимости имеет вид:

$$X(t_0, t, M_0) = \bigcup_{\substack{x_0 \in M_0 \\ u(\cdot) \in Y}} \left\{ e^{(t-t_0)A} x_0 + \int_{t_0}^{t} e^{(t-s)A} u(s) ds \right\},$$

или

$$X(t) = X(t_0, t, M_0) = e^{(t-t_0)A} M_0 + \int_{t_0}^{t} e^{(t-s)A} Y_U(s) ds.$$

3.2 Основные свойства множеств достижимости линейных управляемых систем

- Компактность и выпуклость
  - если множество начальных значений $M_0 \in \Omega(E^n)$ компактно, то множество достижимости тоже является компактным $X(t) \in \Omega(E^n)$ [2]
  - если множество начальных значений есть выпуклый компакт $M_0 \in \operatorname{conv} \Omega(E^n)$, то множество достижимости тоже является выпуклым компактом $X(t) \in \operatorname{conv} \Omega(E^n)$, здесь $\operatorname{conv} \Omega(E^n)$ — множество всех непустых выпуклых компактов E^n [2]
- Непрерывность
  - множество достижимости $X(t)$ непрерывно зависит от времени t: $h(X(t'), X(t)) \to 0$ при $t' \to t$, здесь h — метрика Хаусдорфа

Замечание 1. *Множество достижимости $X(t_0, t, M_0)$ никак не зависит от множества конечных состояний M_1.*

4 Пиксельный метод построения множеств достижимости

Пиксельный метод связан с покрытием фазового пространства сеткой [7]. Сетка строится в виде разбиения рассматри-

ваемой области на квадраты (примитивы). Множество достижимости закрашивается пикселями в форме квадратов. Работа с пикселями множеств достижимости осуществляется при помощи характеристических матриц множества. Обращение к элементам матрицы осуществляется через элементы характеристической матрицы. Каждому элемент множества достижимости сопоставляется только один элемент матрицы и наоборот. Для правильного построения характеристической матрицы множества важно учитывать расположение квадратов на множестве, чтобы получить корректное соответствие элемента множества его позиции.

4.1 Алгоритм

- Для начального множества M_0 характеристическая матрица строится следующим образом: множество разбивается на примитивы, в матрице на соответствующую i, j позицию примитива ставится 1, если примитив принадлежит множеству M_0, в противном случае 0.
- По матрице для множества M_0 восстанавливаются значения элементов и подставляются в формулы для решения задачи Коши. Таким образом вычисляются точки множества достижимости на каждом шаге по времени.

4.2 Методы решения задачи Коши

Задача Коши решается одним из трех методов:
- метод Эйлера
- метод Рунге-Кутта второго порядка точности
- метод Рунге-Кутта четвертого порядка точности

4.3 Характеристическая матрица

В каждый момент времени t_i множеству $X(t_i)$ соответствует матрица A, элементы которой a_{mn} определяются следующим образом с помощью узлов сетки λ_{mn}:

$$a_{mn} = \begin{cases} 1, & \text{если } \lambda_{mn} \in X(t_i) \\ 0, & \text{иначе.} \end{cases}$$

По этой матрице A в каждый момент времени можно однозначно построить приближение к множеству достижимости $X(t_i)$.

4.4 Метод Эйлера для решения задачи Коши

Пусть дана задача Коши для уравнения первого порядка:

$$\begin{cases} \dot{x} = f(t, x), \\ x(t_0) = x_0, \end{cases}$$

где функция f определена в некоторой области $D \subset R^2$. Решение разыскивается на интервале $(x_0, b]$. На этом интервале введем узлы

$$x_0 < x_1 < \cdots < x_n \leqslant b.$$

Приближенное решение в узлах t_i, которое обозначим через x_i определяется по формуле:

$$x_i = x_{i-1} + (t_i - t_{i-1}) f(t_{i-1}, x_{i-1}), \quad i = 1, 2, 3, \ldots, n.$$

Интерпретация для работы с множествами достижимости.

Рассмотрим задачу Коши:

$$\begin{cases} \dot{x} = f(t, x, u), \quad u(t) \in U, \\ x(t_0) \in M_0, \end{cases}$$

f — непрерывная функция, множество M_0 выпуклое и компактное, множество U компактно.

Для момента времени $t_1 = t_0 + \Delta_t$ множество достижимости для данной системы может быть получено следующим

образом:

$$X(t_1) = X(t_1, X(t_0)) = \bigcup_{x(t_0)\in M_0} \left\{ x(t_0) + \Delta_t \bigcup_{u\in U} f(t_0, x(t_0), u) \right\}.$$

Таким образом, используя свойство эволюционности множества достижимости, в момент времени t_{i+1} множество достижимости определяется формулой:

$$X(t_{i+1}) = X(t_{i+1}, X(t_i)) = \\ = \bigcup_{x(t_i)\in X(t_i)} \left\{ x(t_i) + \Delta_t \bigcup_{u\in U} f(t_i, x(t_i), u) \right\}.$$

4.5 Метод Рунге-Кутта для решения задачи Коши

Для более точного построения множества достижимости можно воспользоваться формулой Рунге-Кутта **второго** порядка точности:

$$X(t_{i+1}) = X(t_{i+1}, X(t_i)) = \bigcup_{x(t_i)\in X(t_i)} \left\{ x(t_i) + \right. \\ \left. + \Delta_t \bigcup_{u\in U} f\left(t_i + \frac{\Delta_t}{2}, x(t_i) + \Delta_t \frac{f(t_i, x(t_i), u)}{2}, u\right) \right\}.$$

Большую точность дает численный метод с использованием формулы Рунге-Кутта **четвертого** порядка точности:

Рассмотрим задачу Коши

$$\begin{cases} \dot{x} = f(t, x), \\ x(t_0) = x_0, \end{cases}$$

Тогда приближенное значение в последующих точках вычисляется по итерационной формуле:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$$

где h — величина шага сетки по t. Вычисление нового значения проходит в четыре стадии:

$$\mathbf{k}_1 = h\ \mathbf{f}(t_n, \mathbf{x}_n)\,, \quad \mathbf{k}_2 = h\ \mathbf{f}\left(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{1}{2}\mathbf{k}_1\right),$$

$$\mathbf{k}_3 = h\ \mathbf{f}\left(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{1}{2}\mathbf{k}_2\right), \quad \mathbf{k}_4 = h\ \mathbf{f}(t_n + h, \mathbf{x}_n + \mathbf{k}_3)\,.$$

Интерпретация предыдущей формулы для работы с множествами достижимости:

$$X(t_{i+1}) = X(t_{i+1}, X(t_i), u) = \\ = \bigcup_{\substack{x(t_i)\in X(t_i)\\ u\in U}} \left\{ x(t_i) + \Delta_t \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \right\},$$

$$k_1 = \Delta_t f\left(t_i, x(t_i), u)\right), \quad k_2 = \Delta_t f\left(t_i + \frac{\Delta_t}{2}, x(t_i) + \frac{1}{2}k_1, u\right),$$

$$k_3 = \Delta_t f\left(t_i + \frac{\Delta_t}{2}, x(t_i) + \frac{1}{2}k_2, u\right),$$

$$k_4 = \Delta_t f\left(t_i + \Delta_t, x(t_i) + k_3, u\right).$$

На основании данного метода разработана программа для построения множеств достижимости двумерных управляемых систем.

5 Результаты

Численные расчеты приведены для задач, решения которых известны и хорошо изучены.

5.1 Задача о тележке

Сформулируем условия рассматриваемой задачи:

- $T \geqslant t_0$ — рассматриваемый момент времени
- $A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ — матрица системы
- $\{u \in E^2 \colon u_1 = 0, |u_2| \leqslant 1\}$ — область управления, вертикальный отрезок
- $t_0 = 0$ — начальный момент времени
- $M_0 = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\}$ — множество начальных значений.

Параметры, задаваемые в разработанной программе:

$$x_1 \in [-5, 5], \quad x_2 \in [-5, 5], \quad \Delta_x = 0{,}0125,$$
$$\Delta_t = 0{,}3, \qquad \Delta_u = 0{,}025, \quad t_0 = 0, T = 2{,}7$$

На рисунке 1 изображено полученное пиксельным методом для рассматриваемой задачи множество достижимости.

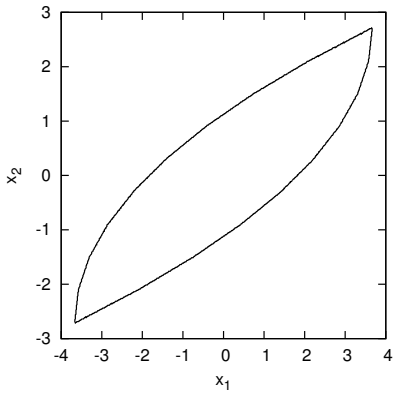


Рис. 1. Множество достижимости для задачи о тележке.

Представлены параметры, при которых не достигается достаточная точность, но есть возможность наглядно оценить расстояния между множествами, полученными при различных

методах решения задачи Коши:

$$x_1 \in [-5,5], \quad x_2 \in [-5,5], \quad \Delta_x = 0{,}03,$$
$$\Delta_t = 0{,}3, \qquad \Delta_u = 0{,}05, \qquad t_0 = 0, T = 2{,}7$$

На рисунке 2 изображены:

1. Множество достижимости, полученное с использованием метода Рунге-Кутта второго порядка (серое множество), и точная граница множества достижимости, которая почти полностью совпадает с данной аппроксимацией.
2. Граница множества достижимости, полученного с использованием метода Эйлера (изображена пунктиром), не совпадающая с точной границей множества достижимости.

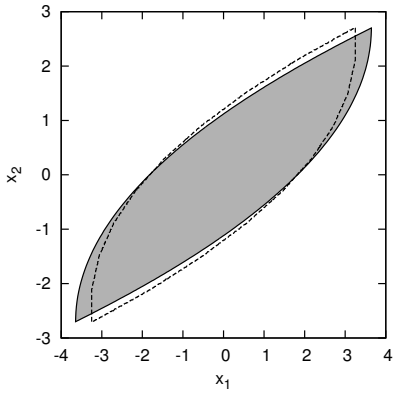


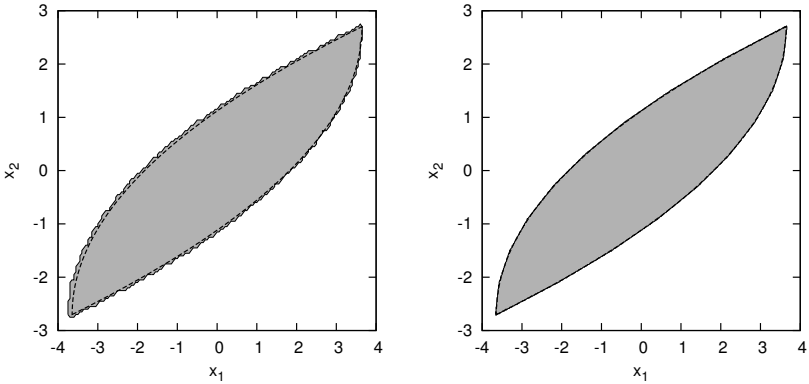
Рис. 2. Множества достижимости для задачи о тележке, полученные с применением разных методов для решения задачи Коши: метода Рунге-Кутта второго порядка точности (множество серого цвета), совпадающее с границей точного множества, и метода Эйлера (граница множества изображена пунктиром).

Расстояние Хаусдорфа между множествами достижимости, полученными при решении задачи методами Рунге-Кутта вто-

рого и четвертого порядков точности, составляет 0,0125. Визуально эти два множества отличить нельзя.

Для рассматриваемой задачи существуют наборы параметров, которые задают дискретную аппроксимацию достаточно точно, но есть и те параметры, при которых результат работы программы отличается от точного множества. На рисунке 3 изображены две пары множеств достижимости:

1. Множество достижимости, достаточно точно аппроксимирующее искомое множество, и точное множество достижимости. Расстоянием Хаусдорфа между данными множествами составляет 0,049.
2. Множество достижимости, полученное при других параметрах, и точное множество достижимости с расстоянием Хаусдорфа между ними, превышающим указанное выше и составляющим 0,093.



(*а*) (*б*)

Рис. 3. Дискретная аппроксимация множества достижимости для задачи о тележке различной точности: хорошей (*а*) и плохой (*б*) в сравнении с границей точного множества.

5.2 Задача о математическом маятнике

Сформулируем условия рассматриваемой задачи:

- $T \geqslant t_0$ — рассматриваемый момент времени
- $A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ — матрица системы
- $\{u \in E^2 \colon u_1 = 0, |u_2| \leqslant 1\}$ — область управления, вертикальный отрезок
- $t_0 = 0$ — начальный момент времени
- $M_0 = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\}$ — множество начальных значений.

Параметры, задаваемые в разработанной программе:

$$x_1 \in [-3, 3], \quad x_2 \in [-3, 3], \quad \Delta_x = 0{,}003,$$
$$\Delta_t = 0{,}15, \qquad \Delta_u = 0{,}05, \qquad t_0 = 0, T = 3{,}3$$

На рисунке 4 изображено полученное для рассматриваемой задачи множество достижимости.

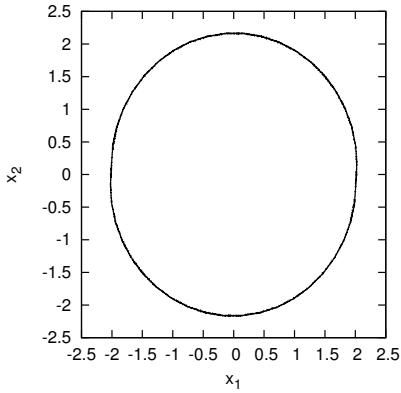


Рис. 4. Множество достижимости для задачи о математическом маятнике.

Далее представлены результаты исследований, аналогичные проведенным для контрольной задачи о тележке. Для на-

бора параметров проиллюстрированы различия в работе методов Рунге-Кутта второго порядка и Эйлера на рисунке 5. На рисунке 6 изображены дискретные аппроксимации точного множества достижимости для математического маятника, составляющие с ним расстояния Хаусдорфа, равные 0.0229 и 0.0346.

$$x_1 \in [-3,3], \quad x_2 \in [-3,3], \quad \Delta_x = 0{,}025,$$
$$\Delta_t = 0{,}15, \quad \Delta_u = 0{,}025, \quad t_0 = 0, T = 3{,}3$$

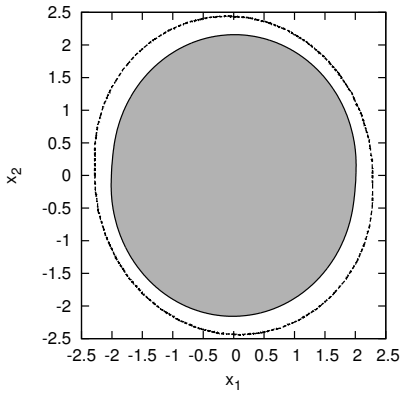
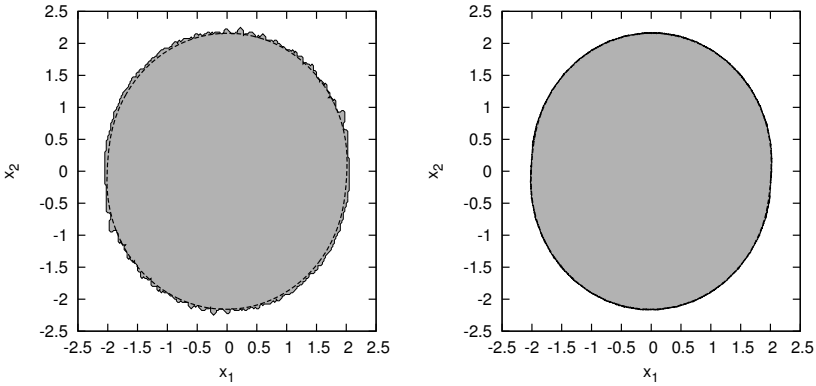


Рис. 5. Множества достижимости для задачи о математическом маятнике, полученные с применением разных методов для решения задачи Коши: метода Рунге-Кутта второго порядка точности (множество серого цвета), совпадающее с границей точного множества, и метода Эйлера (граница множества изображена пунктиром).

5.3 Модифицированный пример Ли-Маркуса

Сформулируем условие рассматриваемой задачи:

- $T \geqslant t_0$ — рассматриваемый момент времени
- $B(x) = \begin{pmatrix} x_2 & -x_1 \\ -x_1 & -x_2 \end{pmatrix}$ — матрица системы [3]



(*а*) (*б*)

Рис. 6. Дискретная аппроксимация множества достижимости для задачи о математическом маятнике различной точности: хорошей (*а*) и плохой (*б*) в сравнении с границей точного множества.

- $\{u \in E^2 \colon u_1^2 + 25u_2^2 \leqslant 1\}$ — область управления, эллипс
- $t_0 = 0$ — начальный момент времени
- $M_0 = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$ — множество начальных значений.

Параметры, задаваемые в разработанной программе:

$$x_1 \in [-2, 2], \quad x_2 \in [-2, 2], \quad \Delta_x = 0{,}0083,$$
$$\Delta_t = 0{,}3, \qquad \Delta_u = 0{,}16, \qquad t_0 = 0, T = 2{,}7$$

На рисунке 7 изображено полученное для рассматриваемой задачи множество достижимости.

При следующих параметрах на рисунке 8

$$x \in [-2, 2], \quad y \in [-2, 2], \quad \Delta_x = 0{,}025,$$
$$\Delta_t = 0{,}3, \qquad \Delta_u = 0{,}16, \quad t_0 = 0, T = 2{,}7$$

построены множества достижимости, полученные методом Эйлера и методом Рунге-Кутта второго порядка точности. Иссле-

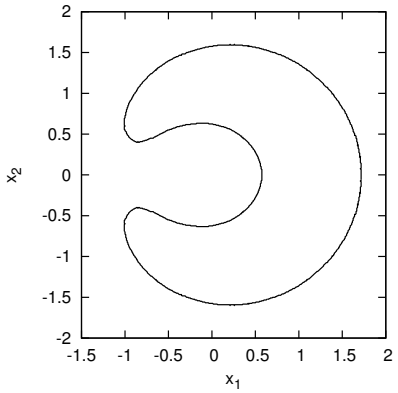


Рис. 7. Множество достижимости для задачи Ли-Маркуса.

дования на рисунке 9 аналогичны проведенным в первых двух рассмотренных контрольных задачах.

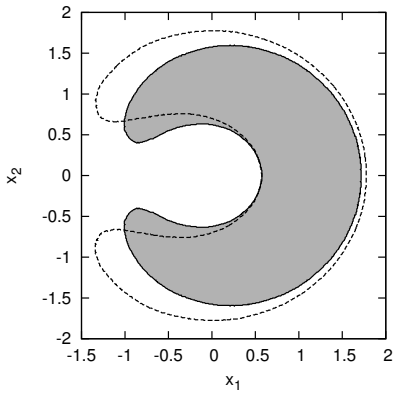
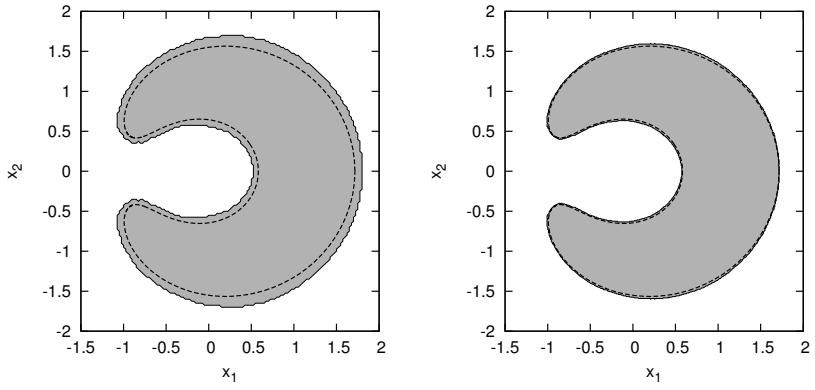


Рис. 8. Множества достижимости для задачи Ли-Маркуса, полученные с применением разных методов для решения задачи Коши: метода Рунге-Кутта второго порядка точности (множество серого цвета), совпадающее с границей точного множества, и метода Эйлера (граница множества изображена пунктиром).



(*а*) (*б*)

Рис. 9. Дискретная аппроксимация множества достижимости для задачи Ли-Маркуса различной точности: хорошей (*а*) и плохой (*б*) в сравнении с границей точного множества

6 Статистика

В таблице 1 приведены значения метрики для оценки расстояния между каждым из построенных пиксельным методом множеств и точным множеством.

Таблица 1. Значения метрики Хаусдорфа для точного множества достижимости и дискретной аппроксимации с применением одного из методов решения задачи Коши

	Метод Эйлера	Метод РК-2	Метод РК-4
Ли-Маркус	0,41	0,07	0,07
Маятник	0,34	0,03	0,02
Тележка	0,39	0,04	0,04

Для построения множества достижимости пиксельным методом рационально использовать метод Рунге-Кутта второго

порядка точности, т.к. полученная данным образом аппроксимация не существенно отличается от построенной с применением метода Рунге-Кутта четвертого порядка точности, но метод четвертого порядка требует больших вычислительных затрат.

7 Заключение

Реализован пиксельный метод построения множества достижимости для управляемых систем. Программа разработана с использованием параллельных вычислений на графических процессорах, что позволяет существенно сократить время вычислений [4]. Проведены исследования полученных результатов, показывающие эффективность решения поставленной задачи.

Список литературы

[1] *Понтрягин Л. С., Болтянский В. Г., Гамкрелидзе Р. В., Мищенко Е. Ф.* Математическая теория оптимальных процессов: Учебное пособие. — М.: Наука, 1983.

[2] *Киселев Ю. Н., Аввакумов С. Н., Орлов М. В.* Оптимальное управление. Линейная теория и приложения: Учебное пособие. — М.: МАКС Пресс, 2007.

[3] *Киселев Ю. Н.* Построение точных решений для нелинейной задачи быстродействия специального вида: Фундаментальная и прикладная математика. — 1997. Т. 3. Выпуск 3. с. 847–868

[4] *Боресков А. В., Харламов А. А.* Основы работы с технологией CUDA: Учебное пособие. — М.: ДМК-Пресс, 2010.

[5] NVIDIA CUDA — неграфические вычисления на графических процессорах `http://www.ixbt.com/video3/cuda-1.shtml`

[6] CUDA Zone — официальный сайт CUDA
`http://www.nvidia.ru/object/cuda_home_new_ru.html`

[7] *Винников Е. В.* Численное построение множеств достижимости нелинейных управляемых систем в среде Matlab. Современные методы теории краевых задач: материалы весенней математической школы «Понтрягинские чтения — XX». — Воронеж, ВГУ, 2009.

УДК 519.1

О числе множеств, k-свободных от нуля, в группах простого порядка

© 2012 г. В. Г. Саргсян

vahe_sargsyan@ymail.com

Кафедра математической кибернетики

Подмножество A элементов группы $\mathbf{G}$ называется *k-свободным от нуля*, если уравнение $x_1 + \cdots + x_k = 0$ не имеет решения в множестве A. Доказано, что логарифм числа множеств, k-свободных от нуля, в группе простого порядка p, асимптотически равен p/k.

1 Введение

Пусть $\mathbf{G}$ – множество с определенной на нем операцией сложения. Подмножество $A \subseteq \mathbf{G}$ называется *(k,l)-свободным от сумм* , если уравнение $x_1 + \cdots + x_k - y_1 - \cdots - y_l = 0$ не имеет решения в множестве A. Множество, $(2,1)$-свободное от сумм, называется просто *свободным от сумм*, а множество, $(k,0)$-свободное от сумм, назовем *k-свободным от нуля.* Семейство всех подмножеств $A \subseteq \mathbf{G}$, (k,l)-свободных от сумм, обозначим через $SF_{k,l}(\mathbf{G})$. Для натуральных чисел m и n обозначим через $[m,n]$ множество натуральных чисел x, таких, что $m \leqslant x \leqslant n$.

В 1988 г. П. Камерон (P. Cameron) и П. Эрдеш (P. Erdos) [1] предположили, что $SF_{2,1}([1,n]) = O(2^{n/2})$. В частности, они доказали, что существуют константы c_0 и c_1, такие, что $|SF_{2,1}([[n/3],n])| \sim c_0 2^{n/2}$ для четных n и $|SF_{2,1}([[n/3],n])| \sim c_1 2^{n/2}$ для нечетных n . В [2] и [3] доказано, что $\log|SF_{2,1}([1,n])| \lesssim n/2$ (здесь и далее логарифмы берутся по основанию два). Б. Грин (B. Green) [4], А. А. Сапоженко [5] нашли асимптотику для $|SF_{2,1}([1,n])|$.

Группу вычетов по простому модулю p обозначим через $\mathbf{Z}_p$. В 2002 г. В. Лев (V. Lev) и Т. Шон (T. Schoen) [6] доказали, что если p – достаточно большое простое число, то справедлива оценка

$$2^{\lfloor (p-2)/3 \rfloor}(p-1)(1+O(2^{-\varepsilon_1 p})) \leqslant |SF_{2,1}(\mathbf{Z}_p)| \leqslant 2^{p/2-\varepsilon_2 p},$$

где ε_1 и ε_2 – положительные константы.

В 2005 г. Б. Грин (B. Green) и И. Ружа (I. Ruzsa) [7] получили асимптотику логарифма числа множеств, свободных от сумм, в абелевых группах. В частности, они доказали, что для любой абелевой группы $\mathbf{G}$ порядка n

$$\log |SF_{2,1}(\mathbf{G})| \backsim \mu(\mathbf{G}),$$

где $\mu(\mathbf{G})$ – максимальный размер множества, свободного от сумм, в группе $\mathbf{G}$. $\mu(\mathbf{G})$ определяется следующим образом: $\mu(\mathbf{G}) = 1/3 + 1/(3p)$, если p является наименьшим простым делителем n вида $p \equiv 2 \pmod 3$, если такого делителя нет, но $3|n$, то $\mu(\mathbf{G}) = 1/3$, в противном случае, $\mu(\mathbf{G}) = 1/3 - 1/(3m)$, где m – экспонента группы $\mathbf{G}$.

В 2009 г. А. А. Сапоженко [8] доказал следующую теорему:

Теорема 1. *Для любого $\alpha \in \{-1, 1\}$ существует константа c_α, такая, что для любого $\varepsilon > 0$ существует натуральное число N, такое, что для любого простого p вида $p \equiv \alpha \pmod 3$, такого, что $p > N$, выполняются неравенства*

$$1 \leqslant \frac{|SF_{2,1}(\boldsymbol{Z}_p)|}{c_\alpha (p-1) 2^{\lfloor (p-2)/3 \rfloor}} < 1 + \varepsilon.$$

Вместе с тем интенсивно рассматривались обобщения проблемы Камерона–Эрдеша. В частности, речь шла о числе множеств, (k, l)-свободных от сумм. В 1996 г. Н. Калкин (N. Calkin) и А. Тейлор (A. Taylor) [9] доказали, что существует константа C_k, $k \geqslant 3$, такая, что $|SF_{k,1}([1, n])|$ не превосходит $C_k 2^{(k-1)n/k}$. В 1998 г. Ю. Билу (Yu. Bilu) [10] доказал, что

$|SF_{l+1,l}([1,n])| = (1+o(1))2^{\lfloor (n+1)/2 \rfloor)}$, а Н. Калкин (N. Calkin) и Дж. Томсон (J. Thomson) [11] доказали, что существует константа $C_{k,l}$, $k \geqslant 4l-1$, такая, что $SF_{k,l}([1,n])$ не превосходит $C_{k,l}2^{(k-l)n/k}$. В 2000 г. Т. Шон (T. Schoen) [12] получил асимптотику для $|SF_{k,l}([1,n])|$ при некоторых ограничениях на k, l. В 2003 г. В. Лев (V. Lev) [13] получил верхную оценку для $|SF_{k,l}([1,n])|$. Семейство всех подмножеств, k-свободных от нуля, в группе $\mathbf{Z}_p$, обозначим через $S_k(p)$ и $s_k(p) = |S_k(p)|$. Целью работы является получение оценки числа $s_k(p)$. В настоящей работе доказывается следующая теорема.

Теорема 2. *Пусть p — простое число и $k \geqslant 3$ — натуральное число. Тогда существует положительная константа C, такая, что выполняются неравенства*

$$2^{\lfloor (p-2)/k \rfloor}(p-1)(1+O(2^{-Cp})) \leqslant s_k(p) \leqslant 2^{(p/k)+(k-3)+o(p)}. \quad (1)$$

При достаточных больших p из (1) вытекает асимптотика числа $\log s_k(p)$.

2 Вспомогательные утверждения

Пусть $\mathbb{R}$ — множество действительных чисел, $f_i \colon \mathbf{Z}_p \to \mathbb{R}$, $i = 1, \ldots, k \geqslant 2$, и $x \in \mathbf{Z}_p$, тогда положим

$$(f_1 * \cdots * f_k)(x) = \sum_{x_1 \in \mathbf{Z}_p} \cdots \sum_{x_{k-1} \in \mathbf{Z}_p} f_1(x_1) \cdots f_{k-1}(x_{k-1}) \times$$

$$\times f_k(x - x_1 - \cdots - x_{k-1}).$$

Пусть

$$\widehat{f}(x) = \sum_{y \in \mathbf{Z}_p} f(y) e^{2\pi i \frac{xy}{p}},$$

и для всякого $A \subseteq \mathbf{Z}_p$ и любого целого числа d положим $d \star \star A = \{da \mid a \in A\}$. Функция $\widehat{f}(x)$ называется *преобразованием*

Фурье функции f, а $d \star A$ назовем *растяжением* множества A. Заметим, что для любого $x \in \mathbf{Z}_p$

$$(f_1 \widehat{* \cdots *} f_k)(x) = \widehat{f}_1(x) \cdots \widehat{f}_k(x). \tag{2}$$

Действительно,

$$(f_1 \widehat{* \cdots *} f_k)(x) = \sum_{y \in \mathbf{Z}_p} (f_1 * \cdots * f_k)(y) e^{2\pi i \frac{yx}{p}} =$$

$$= \sum_{y \in \mathbf{Z}_p} \sum_{y_1 \in \mathbf{Z}_p} \cdots \sum_{y_{k-1} \in \mathbf{Z}_p} f_1(y_1) \cdots f_{k-1}(y_{k-1}) \times$$

$$\times f_k(y - y_1 - \cdots - y_{k-1}) \cdot e^{2\pi i \frac{y_1 x}{p}} \cdots e^{2\pi i \frac{y_{k-1} x}{p}} \cdot e^{2\pi i \frac{(y - y_1 - \cdots - y_{k-1})x}{p}} =$$

$$= \sum_{y_1 \in \mathbf{Z}_p} f_1(y_1) \cdot e^{2\pi i \frac{y_1 x}{p}} \cdots \sum_{y_{k-1} \in \mathbf{Z}_p} f_{k-1}(y_{k-1}) \cdot e^{2\pi i \frac{y_{k-1} x}{p}} \times$$

$$\times \sum_{y \in \mathbf{Z}_p} f_k(y - y_1 - \cdots - y_{k-1}) \cdot e^{2\pi i \frac{(y - y_1 - \cdots - y_{k-1})x}{p}} = \widehat{f}_1(x) \cdots \widehat{f}_k(x).$$

Пусть $A_1, \ldots, A_k$ — непустые подмножества группы $\mathbf{Z}_p$, а характеристические функции множеств $A_1, \ldots, A_k$ обозначим соответственно через $\chi_{A_1}(x), \ldots, \chi_{A_k}(x)$. Тогда $(\chi_{A_1} * \ldots * * \chi_{A_k})(x)$ — это количество наборов $(x_1, \ldots, x_k) \in A_1 \times \cdots \times A_k$, таких, что $x = x_1 + \cdots + x_k$. Положим $A_1 + \cdots + A_k = \{x_1 + \cdots + x_k \mid x_1 \in A_1, \ldots, x_k \in A_k\}$ и $S_{h,k}(A_1, \ldots, A_k) = \{x \in \mathbf{Z}_p \mid (\chi_{A_1} * \cdots * \chi_{A_k})(x) \geq h\}$, где $h > 0$. Пусть далее для любого целого i и всякого $A \subseteq \mathbf{Z}_p$ через iA обозначим множество $\underbrace{A + \cdots + A}_{i}$, а через $-A$ обозначим множество $\{p - x \mid x \in A\}$.

Теорема 3. [6] *Пусть $A \subseteq \mathbf{Z}_p$ — подмножество последовательных элементов и $|A| < p/3 + 1$. Тогда*

$$|\{d \star B \mid B \subseteq A, d \in \mathbf{Z}_p\}| = 2^{|A|}(1 - 2^{-|A \setminus (-A)| - 1})(p - 1) +$$

$$+ O(2^{5|A|/6} p^2).$$

Теорема 4. (Cauchy-Davenport, [14]) *Пусть* $A_1, \ldots, A_k$ — *непустые подмножества группы* $\boldsymbol{Z}_p$. *Тогда*

$$|A_1 + \cdots + A_k| \geqslant \min(p, |A_1| + \cdots + |A_k| - (k-1)).$$

Теорема 5. (Pollard, [15]) *Пусть* A_1, A_2 — *непустые подмножества группы* $\boldsymbol{Z}_p$. *Тогда*

$$|S_{1,2}(A_1, A_2)| + \cdots + |S_{t,2}(A_1, A_2)| \geqslant t \min(p, |A_1| + |A_2| - t),$$

где $t \leqslant \min(|A_1|, |A_2|)$.

Из теорем 4, 5 вытекают следующие два утверждения.

Лемма 1. *Пусть* $A_1, \ldots, A_k$ — *непустые подмножества группы* $\boldsymbol{Z}_p$. *Тогда*

$$|S_{1,k}(A_1, \ldots, A_k)| + \cdots + |S_{t,k}(A_1, \ldots, A_k)| \geqslant$$
$$\geqslant t \min(p, |A_1| + \cdots + |A_k| - t - k + 2),$$

где $t \leqslant \min(|A_1|, \ldots, |A_k|)$.

Доказательство. Предположим, что $|A_1| = \min(|A_1|, \ldots, |A_k|)$. В силу теоремы 5 имеем

$$|S_{1,2}(A_1, (A_2 + \cdots + A_k))| + \cdots + |S_{t,2}(A_1, (A_2 + \cdots + A_k))| \geqslant$$
$$\geqslant t \min(p, |A_1| + |A_2 + \cdots + A_k| - t), \tag{3}$$

где $t \leqslant |A_1|$.

С другой стороны, в силу теоремы 4 имеем

$$|A_2 + \cdots + A_k| \geqslant \min(p, |A_2| + \cdots + |A_k| - (k-2)). \tag{4}$$

Подставив (4) в (3) получим, что

$$|S_{1,k}(A_1, \ldots, A_k)| + \cdots + |S_{t,k}(A_1, \ldots, A_k)| \geqslant$$
$$\geqslant |S_{1,2}(A_1, (A_2 + \cdots + A_k))| + \cdots + |S_{t,2}(A_1, (A_2 + ... + A_k))| \geqslant$$
$$\geqslant t \min(p, |A_1| + \cdots + |A_k| - t - k + 2). \qquad \square$$

Лемма 2. *Пусть $A_1, \ldots, A_k$ — непустые подмножества группы $\boldsymbol{Z}_p$ и $h \leqslant \min(|A_1|, \ldots, |A_k|)$. Тогда*

$$|S_{h,k}(A_1, \ldots, A_k)| \geqslant \min(p, |A_1| + \cdots + |A_k| - k + 2) - 2(hp)^{1/2}.$$

Доказательство. Заметим, что при $i \leqslant j$ выполняется $|S_{i,k}(A_1, \ldots, A_k)| \geqslant |S_{j,k}(A_1, \ldots, A_k)|$. Тогда если выберем $h \leqslant \leqslant t \leqslant \min(|A_1|, \ldots, |A_k|)$, то в силу леммы 1 имеем

$$t \min(p, |A_1| + \cdots + |A_k| - t - k + 2) \leqslant$$
$$\leqslant |S_{1,k}(A_1, \ldots, A_k)| + \cdots + |S_{t,k}(A_1, \ldots, A_k)| \leqslant$$
$$\leqslant hp + t|S_{h,k}(A_1, \ldots, A_k)|.$$

Полагая $t = (hp)^{1/2}$, получим

$$\min(p, |A_1| + \cdots + |A_k| - k + 2) - 2(hp)^{1/2} \leqslant$$
$$\leqslant \min(p, |A_1| + \cdots + |A_k| - k - (hp)^{1/2} + 2) - (hp)^{1/2} \leqslant$$
$$\leqslant |S_{h,k}(A_1, \ldots, A_k)|. \qquad \square$$

Лемма 3. *Пусть $\boldsymbol{T}_{k,s}(\boldsymbol{Z}_p) = \{A \subset \boldsymbol{Z}_p \mid |A| \leqslant p/ks\}$. Тогда существует s, такое, что*

$$|\boldsymbol{T}_{k,s}(\boldsymbol{Z}_p)| \leqslant 2^{p/k}.$$

Доказательство. Пусть n и m — целые числа, $1 \leqslant m \leqslant n$. Тогда [16]

$$\sum_{0 \leqslant i \leqslant m} \binom{n}{i} \leqslant \left(\frac{en}{m}\right)^m. \tag{5}$$

Выберем s, удовлетворяющее условию

$$esk \leqslant 2^s. \tag{6}$$

Тогда в силу (5) имеем ($n = p$ и $m = p/ks$)

$$|\mathbf{T}_{k,s}(\mathbf{Z}_p)| = \sum_{0 \leqslant i \leqslant p/ks} \binom{p}{i} \leqslant (esk)^{p/ks} \leqslant (2^s)^{p/ks} = 2^{p/k}. \qquad \square$$

Пусть L — натуральное число. Для каждого $y \in \{0, \ldots, p - 1\}$ определим разбиение $\mathbf{R}_{y,L}$ группы $\mathbf{Z}_p$ на интервалы вида $J_i^y = \{(iL + 1 + y), \ldots, ((i+1)L + y)\}, 0 \leqslant i \leqslant \lfloor p/L \rfloor - 1$. Все интервалы J_i^y разбиения $\mathbf{R}_{y,L}$ имеют длину L, а множество $J_y = \mathbf{Z}_p \setminus \bigcup_i J_i^y$ имеет мощность $p - L\lfloor p/L \rfloor < L$. Множество $A \subseteq \mathbf{Z}_p$ называется *L-гранулированным* [17], если для некоторого целого числа $d < p$ и некоторого разбиения $\mathbf{R}_{y,L}$ растяжение $d \star A$ является объединением нескольких интервалов J_i^y разбиения $\mathbf{R}_{y,L}$ (отличных от J_y). Обозначим множество L-гранулированных подмножеств группы $\mathbf{Z}_p$ через $\mathbf{G}_L(\mathbf{Z}_p)$.

Лемма 4. *Справедливо неравенство*

$$|\boldsymbol{G}_L(\boldsymbol{Z}_p)| \leqslant p2^{p/L}. \tag{7}$$

Доказательство. Число подмножеств интервалов разбиения $R_{y,L}$ группы $\mathbf{Z}_p$ обозначим через $g(R_{y,L})$, а число различных разбиений $R_{y,L}$ через $r(L)$. Очевидно, что

$$|\mathbf{G}_L(\mathbf{Z}_p)| \leqslant g(R_{y,L})r(L). \tag{8}$$

Заметим, что число интервалов разбиения $\mathbf{R}_{y,L}$ группы $\mathbf{Z}_p$ равно $\lfloor p/L \rfloor$, а количество различных разбиений $\mathbf{R}_{y,L}$ группы $\mathbf{Z}_p$ равно p. Отсюда и из неравенства (8) вытекает справедливость неравенства (7). □

Лемма 5. *Пусть $A \subseteq \boldsymbol{Z}_p$ — подмножество мощности αp, $\varepsilon_1, \varepsilon_2, \varepsilon_3$ — положительные действительные числа, $k \geqslant 2$ и L — натуральные числа, а p — простое число, такое, что выполняется неравенство*

$$p > (\sqrt{8k}L)^{4^{2k}\alpha^{2(k-1)}\varepsilon_1^{-2k}\varepsilon_2^{-2}\varepsilon_3^{-1}}. \tag{9}$$

Тогда существует $A' \subseteq \boldsymbol{Z}_p$ со следующими свойствами:

(i) *A' является L-гранулированным;*

(ii) $|A \setminus A'| \leqslant \varepsilon_1 p$;

(iii) *множество kA содержит все элементы $x \in \boldsymbol{Z}_p$, для которых $(\underbrace{\chi_{A'} * \cdots * \chi_{A'}}_{k})(x) \geqslant \varepsilon_2 p^{k-1}$, за исключением не более $\varepsilon_3 p$ элементов.*

Доказательство. Пусть $h \in \{0, \ldots, p-1\}$ и $\mathbf{R}_{h,L}$ — разбиение группы $\mathbf{Z}_p$.

(i) Множество A' определим как объединение интервалов J_i^h разбиения $\mathbf{R}_{h,L}$, таких, что $|A \cap J_i^h| \geqslant \varepsilon_1 L/2$. По определению очевидно, что A' — L-гранулированное.

(ii) Пусть $x \in A \setminus A'$, тогда либо $x \in J_h$, либо $x \in A \cap J_i^h$, $(i = 0, \ldots, \lfloor p/L \rfloor - 1)$, и $|A \cap J_i^h| \leqslant \varepsilon_1 L/2$. В первом случае имеем $|J_h| < L$, а из неравенства (9) следует, что $L \leqslant \varepsilon_1 p/2$. Итак

$$|A \setminus A'| \leqslant \frac{\varepsilon_1 L}{2} \cdot \frac{p}{L} + L \leqslant \varepsilon_1 p.$$

(iii) Пусть $x \in \mathbf{Z}_p$ и $\widehat{\chi_A}(x)$ — преобразование Фурье характеристической функции χ_A множества A. Тогда

$$\widehat{\chi_A}(x) = \sum_{y \in \mathbf{Z}_p} \chi_A(y) e^{2\pi i \frac{yx}{p}} = \sum_{y \in A} e^{2\pi i \frac{yx}{p}}.$$

Определим δ равенством $\delta = 4^{-k} \varepsilon_1^k \varepsilon_2 \varepsilon_3^{1/2} \alpha^{-k+3/2}$, где $\varepsilon_1, \varepsilon_2$ и ε_3 — из неравенства (9). Положим $\mathbf{D} = \{x \neq 0 \mid |\widehat{\chi_A}(x)| \geqslant \delta p\}$. Определим функцию $f(x)$ следуюшим образом:

$$f(x) = \frac{1}{2L-1} \sum_{j=-(L-1)}^{L-1} e^{2\pi i \frac{jqx}{p}}.$$

В дальнейшем, для доказательства пункта (iii) нам понадбится показать, что существует $q \in \mathbf{Z}_p \setminus \{0\}$, такое, что для всех $x \in \mathbf{Z}_p$ выполняется неравенство

$$|\widehat{\chi_A}(x)||1 - f^k(x)| \leqslant \delta p. \tag{10}$$

Неравенство (10), очевидно, имеет место для $x = 0$, так как $f(0) = 1$, а также в случае, когда $|\widehat{\chi_A}(x)| \leqslant \delta p$, поскольку $f(x) \in \in [-1, 1]$. Итак, осталось показать существование q, такого, что неравенство (10) выполнено для всех $x \in \mathbf{D}$. Оценим величину $1 - f(x)$. Обозначим через $\langle x \rangle$ расстояние x от ближайшего целого числа. Воспользуемся тем, что $1 - \cos(2\pi x) \leqslant 2\pi^2 \langle x \rangle^2$. Имеем

$$1 - f(x) = \frac{2}{2L-1} \sum_{j=1}^{L-1} \left(1 - \cos \frac{2\pi jqx}{p}\right) \leqslant \frac{4\pi^2}{2L-1} \sum_{j=1}^{L-1} \left\langle \frac{jqx}{p} \right\rangle^2 \leqslant$$

$$\leqslant \frac{4\pi^2}{2L-1} \left\langle \frac{qx}{p} \right\rangle^2 \sum_{j=1}^{L-1} j^2 \leqslant \frac{2\pi^2 L^2}{3} \left\langle \frac{qx}{p} \right\rangle^2. \qquad (11)$$

Воспользуемся тем, что при условии $|x| \leqslant 1$

$$1 - x^k = (1-x)(1 + x + x^2 + \cdots + x^{k-1}) \leqslant k(1-x). \qquad (12)$$

Из (11) и (12) следует, что

$$|\widehat{\chi_A}(x)||1 - f^k(x)| \leqslant k|\widehat{\chi_A}(x)||1 - f(x)| \leqslant 8kL^2 \langle qx/p \rangle^2 |\widehat{\chi_A}(x)|.$$

Заметим, что при выполнении неравенства

$$\left\langle \frac{qx}{p} \right\rangle \leqslant \frac{1}{\sqrt{8k}L} \left(\frac{\delta p}{|\widehat{\chi_A}(x)|} \right)^{1/2} \qquad (13)$$

для некоторого $q \in \mathbf{Z}_p \setminus \{0\}$ и для всех $x \in \mathbf{D}$, то неравенство (10) также выполняется.

Докажем, что q такое, существует. По определению имеем

$$\langle qx/p \rangle = \min\left((qx \pmod p)/p; (p - qx \pmod p)/p\right).$$

Положим $d = |\mathbf{D}|$, $\mathbf{D} = \{r_1, \ldots, r_d\}$. Обозначим величину $(1/\sqrt{8k}L)(\delta p/|\widehat{\chi_A}(r_i)|)^{1/2}$ через a_i. Тогда неравенство (13) можно переписать в виде

$$\min\left(qr_i \pmod p; p - qr_i \pmod p\right) \leqslant pa_i; i = 1, \ldots, d. \qquad (14)$$

Обозначим множество $\{(x_1, \ldots, x_d) \mid x_1, \ldots, x_d \in \mathbf{Z}_p\}$ через $\mathbf{Z}_p^d$. Разобьем $\mathbf{Z}_p^d$ на непересекающиеся подмножества

$$\mathbf{Z}_p^d = \bigcup_{(i_1,\ldots,i_d)} \mathbf{Q}_{i_1,\ldots,i_d},$$

где

$$\mathbf{Q}_{i_1,\ldots,i_d} = \{(x_1, \ldots, x_d) \mid i_j p a_j < x_j \leqslant (i_j + 1) p a_j; j = 1, \ldots, d\}.$$

Пусть μ_d — количество различных множеств $\mathbf{Q}_{i_1,\ldots,i_d}$. С учетом того, что $0 \leqslant i_j \leqslant 1/a_j - 1, j = 1, \ldots, d$, имеем

$$\mu_d \leqslant \prod_{i=1}^{d} \frac{1}{a_i}.$$

Рассмотрим следующие $p - 1$ элементы из $\mathbf{Z}_p^d$:

$$(qr_1 \pmod p, \ldots, qr_d \pmod p),$$

где $r_1, \ldots, r_d \in \mathbf{D}$, $q = 1, \ldots, p - 1$.

Покажем, что если

$$p > \prod_{i=1}^{d} \frac{1}{a_i}, \tag{15}$$

то существует q, такое, что для всех $r_i \in \mathbf{D}$, $i = 1, \ldots, d$, выполняется неравенство (14). Рассмотрим два случая:

(A) Пусть $\mu_d = p - 1$, тогда в качестве q выберем $q_0 \in \mathbf{Z}_p \setminus \{0\}$, такое, что $(q_0 r_1 \pmod p, \ldots, q_0 r_d \pmod p) \in \mathbf{Q}_{0,\ldots,0}$. Нетрудно заметить, что при $q = q_0$ неравенство (14) выполняется для всех $r_i \in \mathbf{D}, i = 1, \ldots, d$;

(B) Пусть $\mu_d < p-1$, тогда по принципу Дирихле существуют q_1, $q_2 \in \mathbf{Z}_p \setminus \{0\}$, такие, что векторы $(q_1 r_1 \pmod p, \ldots, q_1 r_d \pmod p)$ и $(q_2 r_1 \pmod p, \ldots, q_2 r_d \pmod p)$ принадлежат одному и тому же множеству из $\mathbf{Q}_{i_1,\ldots,i_d}$. Очевидно, что при

$q = q_1 - q_2$ неравенство (14) выполняется для всех $r_i \in \mathbf{D}, i = 1, \ldots, d$.

Теперь покажем, что неравенство (15) является следствием условия (9). Действительно, в силу равенства Парсеваля имеем

$$p^{-1} \sum_{x \in \mathbf{Z}_p} |\widehat{\chi_A}(x)|^2 = \sum_{x \in \mathbf{Z}_p} |\chi_A(x)|^2 = \alpha p. \tag{16}$$

Из (16) следует, что

$$\sum_{x \in \mathbf{D}} |\widehat{\chi_A}(x)|^2 \leqslant \alpha p^2. \tag{17}$$

Из неравенства (17) и неравенства о среднем арифметическом и среднем геометрическом имеем

$$\left(\prod_{x \in \mathbf{D}} |\widehat{\chi_A}(x)|^2 \right)^{1/d} \leqslant \frac{1}{d} \sum_{x \in \mathbf{D}} |\widehat{\chi_A}(x)|^2 \leq \frac{\alpha p^2}{d}.$$

Отсюда следует, что

$$\prod_{x \in \mathbf{D}} |\widehat{\chi_A}(x)| \leqslant \left(\frac{\alpha p^2}{d} \right)^{d/2}. \tag{18}$$

Из (18) имеем

$$(\sqrt{8k}L)^d \left(\prod_{x \in \mathbf{D}} \frac{|\widehat{\chi_A}(x)|}{\delta p} \right)^{1/2} \leqslant (\sqrt{8k} L \alpha^{1/4} \delta^{-1/2} d^{-1/4})^d. \tag{19}$$

Нетрудно заметить, что правая часть неравенства (19) является возрастающей функцией от d при $d < 64k^2 L^4 \alpha / \delta^2 e$. С другой стороны, из (17) имеем $d\delta^2 p^2 \leqslant \alpha p^2$, а значит, $d \leqslant \alpha/\delta^2$. Следовательно,

$$(\sqrt{8k} L \alpha^{1/4} \delta^{-1/2} d^{-1/4})^d \leqslant (\sqrt{8k} L)^{\alpha/\delta^2}.$$

Полагая $\delta = 4^{-k}\varepsilon_1^k\varepsilon_2\varepsilon_3^{1/2}\alpha^{-k+3/2}$, получаем, что существует q, такое, что неравенство (10) выполняется. При этом без ограничения общности можно считать $q = 1$(этого можно добиться выбором соответствующего растяжения множества A).

Определим функцию $\chi_1(x)$ следующим образом:

$$\chi_1(x) = \frac{1}{|P|}(\chi_A * \chi_P)(x),$$

где $P = \{-(L-1), \ldots, L-1\}$. Из определеня свертки следует, что

$$\chi_1(x) = \frac{1}{|P|}|A \cap (P + x)|,$$

а из равенства (2) имеем $\widehat{\chi_1}(x) = \widehat{\chi_A}(x)f(x)$. Отсюда и также в силу равенства Парсеваля и равенства (2) следует

$$\sum_{x\in\mathbf{Z}_p}\left|(\underbrace{\chi_A * \cdots * \chi_A}_{k})(x) - (\underbrace{\chi_1 * \cdots * \chi_1}_{k})(x)\right|^2 =$$

$$= p^{-1}\sum_{x\in\mathbf{Z}_p}\left|(\underbrace{\chi_A \widehat{*\cdots*} \chi_A}_{k})(x) - (\underbrace{\chi_1 \widehat{*\cdots*} \chi_1}_{k})(x)\right|^2 =$$

$$= p^{-1}\sum_{x\in\mathbf{Z}_p}\left|\widehat{\chi_A}^k(x) - \widehat{\chi_1}^k(x)\right|^2 = p^{-1}\sum_{x\in\mathbf{Z}_p}|\widehat{\chi_A}(x)|^{2k}\left|1 - f^k(x)\right|^2 \leqslant$$

$$\leqslant p^{-1}\left(\sup_{x\in\mathbf{Z}_p}|\widehat{\chi_A}(x)|^{k-1}\left|1 - f^k(x)\right|\right)^2\sum_{x\in\mathbf{Z}_p}|\widehat{\chi_A}(x)|^2. \qquad (20)$$

По определению следует

$$|\widehat{\chi_A}(x)| = \left|\sum_{y\in A} e^{2\pi i\frac{yx}{p}}\right| \leqslant \sum_{y\in A}\left|e^{2\pi i\frac{yx}{p}}\right| = \alpha p. \qquad (21)$$

Из (10), (16), (20) и (21) имеем

$$\sum_{x\in\mathbf{Z}_p}\left|(\underbrace{\chi_A*\cdots*\chi_A}_{k})(x)-(\underbrace{\chi_1*\cdots*\chi_1}_{k})(x)\right|^2\leqslant$$

$$\leqslant\left(\sup_{x\in\mathbf{Z}_p}|\widehat{\chi_A}(x)|\left|1-f^k(x)\right|\right)^2\alpha^{2k-3}p^{2k-3}\leqslant\alpha^{2k-3}\delta^2p^{2k-1}. \quad (22)$$

Если $x\in A'$, то существует интервал длины L, который содержит x, и следовательно, содержится в интервале $[x-(L-1),\ldots,x+(L-1)]$. Из определения множества A' следует, что этот интервал содержит не менее $\varepsilon_1L/2$ точек из A, а из определения функции $\chi_1(x)$ следует, что $\chi_1(x)\geqslant\varepsilon_1/4$. Заметим, что для всех $x\in\mathbf{Z}_p$ справедливо неравенство $\chi_1(x)\geqslant\varepsilon_1\chi_{A'}(x)/4$. Из этого неравенства и определения свертки вытекает, что для всех $x\in\mathbf{Z}_p$

$$(\underbrace{\chi_1*\cdots*\chi_1}_{k})(x)\geqslant\varepsilon_1^k(\underbrace{\chi_{A'}*\cdots*\chi_{A'}}_{k})(x)/4^k. \quad (23)$$

При условии, что

$$(\underbrace{\chi_{A'}*\cdots*\chi_{A'}}_{k})(x)\geqslant\varepsilon_2p^{k-1} \quad (24)$$

из (23) имеем

$$(\underbrace{\chi_1*\cdots*\chi_1}_{k})(x)\geqslant{\varepsilon_1}^k\varepsilon_2p^{k-1}/4^k.$$

Покажем, что количество элементов $x\in\mathbf{Z}_p$, таких, что $(\underbrace{\chi_A*\cdots*\chi_A}_{k})(x)=0$ и удовлетворяющих неравенству (24), не превосходит ε_3p. Семейство таких элементов обозначим через $\mathbf{F}$. Заметим, что для всякого $x\in\mathbf{F}$

$$|(\underbrace{\chi_A*\cdots*\chi_A}_{k})(x)-(\underbrace{\chi_1*\cdots*\chi_1}_{k})(x)|^2\geqslant\frac{{\varepsilon_1}^{2k}{\varepsilon_2}^2p^{2(k-1)}}{4^{2k}}. \quad (25)$$

Из (22) и (25) имеем

$$\alpha^{2k-3}\delta^2 p^{2k-1} \geqslant \sum_{x\in\mathbf{Z}_p}\left|(\underbrace{\chi_A * \cdots * \chi_A}_{k})(x) - (\underbrace{\chi_1 * \cdots * \chi_1}_{k})(x)\right|^2 =$$

$$= \sum_{x\in\mathbf{F}}\left|(\underbrace{\chi_A * \cdots * \chi_A}_{k})(x) - (\underbrace{\chi_1 * \cdots * \chi_1}_{k})(x)\right|^2 +$$

$$+ \sum_{x\in(\mathbf{Z}_p\setminus\mathbf{F})}\left|(\underbrace{\chi_A * \cdots * \chi_A}_{k})(x) - (\underbrace{\chi_1 * \cdots * \chi_1}_{k})(x)\right|^2 \geqslant$$

$$\geqslant |\mathbf{F}|\frac{{\varepsilon_1}^{2k}{\varepsilon_2}^2 p^{2(k-1)}}{4^{2k}}.$$

Остюда следует, что

$$|\mathbf{F}| \leqslant \frac{4^{2k}\alpha^{2k-3}\delta^2}{{\varepsilon_1}^{2k}{\varepsilon_2}^2}p \leqslant \varepsilon_3 p. \qquad \square$$

3 Максимальная мощность множества, k-свободного от нуля, в группе $\mathbf{Z}_p$

Пусть $k \geqslant 3$ — натуральное число. Через $\mu_k(p)$ обозначим максимальную мощность множества, k-свободного от нуля, в $\mathbf{Z}_p$.

Теорема 6. *Пусть p — простое число, а $k \geqslant 3$ — натуральное число. Тогда выполняется равенство*

$$\mu_k(p) = \left\lfloor \frac{p-2}{k} \right\rfloor + 1.$$

Доказательство. Пусть $A \subseteq \mathbf{Z}_p$ — подмножество, k-свободное от нуля. Из того, что $0 \notin kA$ следует $|kA| \leqslant p-1$. С другой

стороны, в силу теоремы 4 имеем $k|A| - (k-1) \leqslant |kA|$. Из предыдущих неравенств получим, что $\mu_k(p) \leqslant \lfloor (p-2)/k \rfloor + 1$.

Теперь докажем, что $\mu_k(p) \geqslant \lfloor (p-2)/k \rfloor + 1$. Более того, покажем, что существует множество последовательных элементов, k-свободное от нуля, мощности $\lfloor (p-2)/k \rfloor + 1$, в группе Z_p. Положим, $c = \lfloor (p-2)/k \rfloor$. Вполне очевидно, что $kc \leqslant p-2$. Известно, что существуют такие целые числа u и v, что $1 = ku + pv$. Рассмотрим следующее множество $A = \{u, u+1, \ldots, u+c\}$ последовательных элементов группы Z_p. Заметим, что для любого $i \in [0, \ldots, kc]$ выполняется $1 \leqslant 1+i \leqslant 1+kc \leqslant p-1$. Теперь рассмотрим множество $B = \{1+i \mid 0 \leqslant i \leqslant kc\}$ как подмножество группы Z_p. Заметим, что $0 \notin B$, и для ku как элемента группы Z_p выполняется $ku = 1 - pv = 1$. С другой стороны, $kA = \{ku + i \mid 0 \leqslant i \leqslant kc\} = B$. Итак, множество A мощности $\lfloor (p-2)/k \rfloor + 1$, является k-свободным от нуля, так как $0 \notin kA$. Отсюда следует, что $\mu_k(p) \geqslant \lfloor (p-2)/k \rfloor + 1$. $\square$

4 Доказательство теоремы 2

4.1 Нижняя оценка

Пусть p — достаточно большое простое число. Как мы видели в параграфе 3, существует множество A последовательных элементов, k-свободное от нуля, мощности $\lfloor (p-2)/k \rfloor + 1$, в группе Z_p. Заметим, что любое подмножество множества A тоже является k-свободным от нуля, и число растяжений всех его подмножеств дает нижную оценку. Из теоремы 3 получим, что существует положительная константа C, такая, что

$$2^{\lfloor (p-2)/k \rfloor}(p-1)(1 + O(2^{-Cp})) \leqslant s_k(p).$$

4.2 Верхняя оценка

Пусть $k \geqslant 3$ — натуральное число, а s — удовлетворяет условию (6). Рассмотрим разбиение множества $\mathbf{S}_k(p)$ на две

части:

$$\mathbf{S}_k(p) = \mathbf{S}'_{k,s}(p) \cup \mathbf{S}''_{k,s}(p), \tag{26}$$

где

$$\mathbf{S}'_{k,s}(p) = \{A \in \mathbf{S}_k(p) \mid |A| \leqslant p/ks\},$$
$$\mathbf{S}''_{k,s}(p) = \{A \in \mathbf{S}_k(p) \mid |A| > p/ks\}.$$

Из (26) следует, что

$$s_k(p) = |\mathbf{S}'_{k,s}(p)| + |\mathbf{S}''_{k,s}(p)|. \tag{27}$$

В силу леммы 3 имеем

$$\left|\mathbf{S}'_{k,s}(p)\right| \leqslant |\mathbf{T}_{k,s}(\mathbf{Z}_p)| \leqslant 2^{p/k}. \tag{28}$$

Теперь оценим сверху $|\mathbf{S}''_{k,s}(p)|$. Пусть $A \in \mathbf{S}''_k(p)$ и p — простое число, такое, что для натуральных чисел $k \geqslant 3$, L и положительных действительных чисел $\varepsilon_1, \varepsilon_2$ и ε_3 выполняется условие (9). Тогда по лемме 5 существует подмножество A', обладающее свойствами $(i) - (iii)$. Оценим $|\mathbf{S}''_{k,s}(p)|$ путем подсчета количество пар (A', A).

Пусть $A' \in \mathbf{G}_L(\mathbf{Z}_p)$ — выбрано. Рассмотрим два случая: $|A'| \geqslant p/k$ и $|A'| < p/k$.

В первом случае в силу утверждения (iii) леммы 5 имеем

$$|S_{\varepsilon_2 p^{k-2},k-1}(\chi_{A'}, \ldots, \chi_{A'}) \setminus (k-1)A| \leqslant \varepsilon_3 p.$$

Так как A — множество, k-свободное от нуля, что то же самое, что $(k-1)A \cap (-A) = \emptyset$, получим, что

$$|A| \leqslant p - |S_{\varepsilon_2 p^{k-2},k-1}(\chi_{A'}, \ldots, \chi_{A'})| + \varepsilon_3 p.$$

В силу леммы 2 имеем

$$|S_{\varepsilon_2 p^{k-2},k-1}(\chi_{A'}, \ldots, \chi_{A'})| \geqslant \min(p, (k-1)|A'| - k + 3) - 2(\varepsilon_2 p^{k-1})^{1/2}.$$

При условии $|A'| \geqslant p/k$, получаем

$$|S_{\varepsilon_2 p^{k-2},k-1}(\chi_{A'}, \ldots, \chi_{A'})| \geqslant (k-1)p/k - 2{\varepsilon_2}^{1/2} p^{(k-1)/2} - (k-3).$$

Отсюда следует, что число способов выбора A для заданного A' мощности, превышающей p/k, не превосходит

$$2^{(p/k)+(k-3)+(2\varepsilon_2{}^{1/2}p^{(k-3)/2}+\varepsilon_3)p}. \tag{29}$$

Если $|A'| < p/k$, то в силу утверждения (ii) леммы 5 имеем $|A \setminus A'| \leqslant \varepsilon_1 p$. Отсюда следует, что $|A| \leqslant |A'| + \varepsilon_1 p$. Итак, число способов выбора A для заданного A' мощности, не превышающей p/k, не превосходит

$$2^{(p/k)+\varepsilon_1 p}. \tag{30}$$

В силу (29), (30), леммы 4 и применяя лемму 5 с параметрами $\varepsilon_1 = \varepsilon_3 = \varepsilon$, $\varepsilon_2 = \varepsilon^2 p^{3-k}$ и $L = 1 + \lfloor 1/\varepsilon \rfloor$, получим, что

$$|\mathbf{S}''_{k,s}(p)| \leqslant 2^{(p/k)+(k-3)+o(p)}. \tag{31}$$

Из неравенств (27), (28) и (31) следует, что

$$s_k(p) \leqslant 2^{p/k} + 2^{(p/k)+(k-3)+o(p)} = 2^{(p/k)+(k-3)+o(p)}.$$

5 Заключение

Целью статьи является доказательство неравенств для числа $s_k(p)$ множеств k-свободных от нуля (см. Теорему 2). Из полученных неравенств вытекает асимптотика величины $\log_2 s_k(p)$.

Автор выражает признательность своему научному руководителю профессору А. А. Сапоженко за постановку задачи и внимание к этой работе.

Работа выполнена при поддержке РФФИ, проект №10–01–00768-а.

Список литературы

[1] *Cameron P. J., Erdos P.* On the number of sets of integers with various properties // Number Theory. Proc. First Conf.

Can. Number Th. Ass. / Ed. by R. A. Mollin. Berlin: de Gruyter. — 1990. — P. 61–79.

[2] *Alon N.* Independent sets in regular graphs and sum-free subsets of abelian groups // Israel J. Math. — 1991. — Vol. 73. — P. 247–256.

[3] *Calkin N. J.* On the number of sum-free set // Bull. London Math. Soc. — 1990. — Vol. 22. — P. 141–144.

[4] *Green B.* The Cameron–Erdos conjecture // Bull. London Math. Soc. — 2004. — Vol. 36, no 6. — P. 769–778.

[5] *Сапоженко А. А.* Гипотеза Камерона–Эрдеша // Докл. РАН. — 2003. — Т. 393, № 6. — С. 749–752.

[6] *Lev V. F., Schoen T.* Cameron–Erdos modulo a prime // Finite Fields Appl. — 2002. — Vol. 8, no 1. — P. 108–119.

[7] *Green B., Ruzsa I.* Sum-free sets in abelian groups // Israel J. Math. — 2005. — Vol. 147. — P. 157–188.

[8] *Сапоженко А. А.* Решение проблемы Камерона–Эрдеша для групп простого порядка // ЖВМиМФ. — 2009. — Т. 49, № 8. — С. 1–7.

[9] *Calkin N. J., Taylor A. C.* Counting sets of integers, no k of which sum to another // J. Number Theory. — 1996. — Vol. 57. — P. 323–327.

[10] *Bilu Yu.* Sum-free sets and related sets //Combinatorica. — 1998. — Vol. 18. — P. 449–459.

[11] *Calkin N. J., Thomson J. M.* Counting generalized sum-free sets // J. Number Theory. — 1998. — Vol. 68. — P. 151–160.

[12] *Schoen T.* A note on the number of (k, l)-sum–free sets // Elect. J. Combin. — 2000. — Vol. 7. — P. 1–8.

[13] *Lev V. F.* Sharp estimates for the number of sum-free sets // J. Reine Angew. Math. — 2003. — Vol. 555. — P. 1–25.

[14] *Nathanson M. B.* Additive number theory: Inverse problems and the geometry of sumsets, Graduate Texts in Mathematics. **165**. — New York: Springer-Verlag, 1996.

[15] *Pollard J. M.* A generalization of the theorem of Cauchy and Davenport // J. London Math. Soc. — 1974. —Vol. 8, no. 2. — P. 460–462.

[16] *Сапоженко А. А.* Проблема Дедекинда и метод граничных функционалов. — М.:ФИЗМАТЛИТ, 2009.

[17] *Green B., Ruzsa I.* Counting sum-sets and sum-free sets modulo a prime // Studia Sci. Math. Hungarica. — 2004. — Vol. 41. — P. 285–293.

УДК 517.518.85

Алгоритм оценки кривой доходности по нескольким группам облигаций с оптимальным выбором весов

© 2012 г. И. С. Сорокин, В. А. Лапшин

igor_sorokin@mail.ru, victor.lapshin@gmail.com

Кафедра системного анализа

1 Введение

Определение срочной структуры процентных ставок (бескупонная кривая доходности или кривая доходности «спот») для стран Еврозоны является актуальной проблемой финансовой инженерии и риск-менеджмента, не имеющей на текущий момент единого общепринятого решения [1]. В России широко используется кривая доходности к погашению, где набор облигаций представляется как набор точек на координатной плоскости с осями «срок погашения» – «доходность к погашению» или «дюрация» – «доходность к погашению». Однако также не имеется общепризнанной модели.

В настоящей работе предлагается новый метод для построения кривой бескупонной доходности «спот», который может быть применим не только для одной страны (например, России), но и для облигаций разного кредитного качества (например, государственные облигации стран Еврозоны). Цель данной работы — развитие подхода, предложенного в [2]. Главным недостатком метода [2] является поведение кривой доходности в окрестности нуля. В данной статье удалось устранить этот недостаток и получить экономически непротиворечивую форму кривой доходности для группы стран Еврозоны и для каждой страны в отдельности. Для определения кривой доходности отдельной страны выбран алгоритм стандарта EFFAS-EBC [3],

который предполагает, что специфическая для каждой группы облигаций кривая получается параллельными сдвигом фиксированной базовой кривой из семейства относительных кривых безрисковой доходности «спот».

Под облигацией будем понимать долговое обязательство, расписание платежей по которому заранее известно (т. е. мы рассматриваем облигации только с фиксированным купоном). В качестве упрощающих, но общепринятых предположений, рассмотрим «идеальный» рынок, на котором торгуются облигации со всеми возможными номиналами и временами до погашения той же степени рискованности и ликвидности, что и данные. Рассмотрим бескупонную облигацию с единовременной выплатой основного долга со временем до погашения T и номиналом 1. Обозначим ее цену в текущий момент через $d(T)$. Эта зависимость называется функцией дисконтирования. Задача определения срочной структуры процентных ставок непосредственно связана с задачей нахождения функции дисконтирования. Предполагается, что стоимость любой купонной облигации равна стоимости портфеля бескупонных облигаций с соответствующими сроками до погашения. Тогда зависимость цены P облигации от сроков t_i и объемов выплат F_i, $i = 0, \ldots, n$ имеет вид $P = \sum_{i=0}^{n} d(t_i)F_i$, где $d(t)$ — функция дисконтирования. Пусть на рынке торгуются N облигаций m стран с ценами P_k и объемами выплат F_{ki}, в моменты времени t_i, $i = 0, \ldots, n$, $k = 1, \ldots N$. Таким образом, приходим к задаче: найти функцию дисконтирования $d(t)$, $t \geqslant 0$, удовлетворяющую следующим условиям:

1. $d(t)$ не возрастает на всей области определения;
2. $d(t) > 0$, $d(0) = 1$;
3. $\sum_{i=0}^{n} F_{ki} d(t_i) = P_k$ для всех $k = 1, \ldots, N$.

Связь между кривой доходности и функцией дисконтирования определяется способом начисления процентов. В российских банках чаще всего используется начисление процентов раз в год. Для нас удобнее будет непрерывное начисление процентов.

В этом случае кривая доходности $r(t)$ и функция дисконтирования $d(t)$ связаны соотношением $d(t) = \exp(-r(t)t)$.

В такой постановке эта задача является некорректно поставленной. Из приведенных условий можно определить только значения функции дисконтирования в точках разбиения, т. е. определить $d(t_i)$, причем единственным образом это можно сделать лишь в случае, когда система в условии 3. является определенной, что априори можно гарантировать только в тривиальном случае, когда ни одна облигация не имеет купонных выплат (времена погашения облигаций обычно не совпадают). Иначе требуется специальным образом сформированная выборка облигаций, что сильно ограничивает общность метода. В общем случае однозначному определению не поддаются даже $d(t_i)$, не говоря уже о значениях $d(t)$ в промежуточных точках. Кроме того, нетрудно убедиться в неустойчивости задачи даже в фиксированных точках. Принципы решения некорректно поставленных задач описаны в [4]. В соответствии с принципами, описанными там, а также с учетом того, что гладкость срочной структуры процентных ставок — экономически разумное требование [2], для регуляризации в дальнейшем мы наложим на структуру процентных ставок условие гладкости.

2 Обзор литературы

Подходы, традиционно используемые для определения срочной структуры процентных ставок (нахождения функции дисконтирования) можно разделить на три группы: инженерные методы, параметрическое оценивание и интерполяционные методы. К инженерным методам относятся, например, использование доходности похожей облигации, использование кривой доходности к погашению, метод пошагового определения процентных ставок (bootstrapping). Самыми известными и широко используемые среди параметрических методов являются методы Нельсона-Зигеля [5] , Свенсона [6]. На российском рынке

также используется G-кривая, в плане параметризации являющаяся вариацией метода Нельсона-Зигеля [7].

Среди интерполяционных методов можно выделить сплайновый подход, который был впервые предложен в работе [8]. В [8] использовались квадратичные, в [9] – кубические сплайны. В [10] впервые были использованы B-сплайны, в [11] – экспоненциальные, в [12] – L-сплайны. В работе [1] были использованы сглаживающие сплайны. Путем комбинации экспоненциальных сплайнов и требования гладкости для регуляризации задачи, в [2] были получены экспоненциально-синусоидальные сплайны.

Параметрические методы имеют ряд недостатков: экономически противоречивые результаты (использование параметрических методов, в частности, широко известных методов Нельсона-Зигеля и Свенсона, иногда приводит к результатам, абсурдным с экономической точки зрения); некорректные результаты в случае отсутствия сделок по бумагам с малыми или большими сроками до погашения; численная неустойчивость результата (часто отбрасывание одной бумаги существенно меняет форму кривой, особенно на коротком ее конце, около $t = 0$).

К недостаткам сплайновых (и непараметрических в целом) методов можно отнести высокую сложность получаемых моделей и связанную с ней трудоемкость оценки параметров модели. Также зачастую результат трудно интерпретируется с экономической точки зрения.

3 Новый подход к задаче

Следуя идеям [2], [3], сделаем замену переменной: $d(t) = \exp(-z(t))$. Тогда получим систему:

$$\begin{cases} \dot{z} = f(x), \\ \dot{x} = u. \end{cases}$$

$f(x)$ имеет экономический смысл — это мгновенная форвардная ставка. Поэтому необходимо потребовать, чтобы $f(x) \geqslant 0$. Так как $d(t)$ — фукнция дисконтирования, то $z(0) = 0$. Для существование аналитического решения данной системы в [2] в качестве $f(x)$ рассматривается x^2.

Занумеруем облигации разных стран от 1 до N, где N – общее количество облигаций, то есть $1 \dots K_1$ – облигации первой страны, $K_1 + 1 \dots K_2$ – облигации второй страны, $\dots$, $K_{m-1} + 1, \dots, K_m = N$ – облигации m-й страны. Обозначим спрэд l-й страны через r_l, $l = 1, \dots, m$. Так как спрэд не зависит от времени, то $\dot{r}_l = 0$, $l = 1, \dots, m$. Так как облигаций N, а стран m, $m < N$, то для дальнейших рассуждений вместо r_l иногда будем писать r_k, имея в виду спрэд эмитента k-й облигации, $k = 1, \dots, N$.

Рассмотрим функционал:

$$J(u) = \frac{\alpha^2}{2} \int\limits_0^T \tau u^2(\tau)\, d\tau + \sum_{k=1}^{N} \frac{w_k}{2} \left(\sum_{i=0}^{n} F_{ki} e^{-z(t_i) - r_k t_i} - P_k \right)^2, \tag{1}$$

здесь α – параметр, отвечающий за гладкость полученного решения, w_k – весовые коэффициенты. В качестве w_k будем брать квадрат обратного значения разности между ценой спроса и предложения [3].

Второе слагаемое в данном функционале отвечает за точность получаемого решения и его вид получается из простых рассуждений. Первое слагаемое отвечает за гладкость и форму кривой доходности. Кроме того за счет выбора первого слагаемого можно добиться аналитического решения данной задачи. В [2] в качестве первого слагаемого рассматривается:

$$\frac{\alpha^2}{2} \int\limits_0^T u^2(\tau)\, d\tau.$$

При таком выборе функционала производная кривой доходности в нуле $r'(0)$ всегда равна нулю, а сама кривая имела перегиб. Такой вид кривой доходности является экономически необоснованным. Чтобы избежать этого, в данной статье предлагается умножить $u^2(\tau)$ под интегралом на τ. Тогда при малом t значение функционала $J(u(t))$ не будет зависеть от гладкости решения (от первого слагаемого), а только от второго слагаемого, поэтому форма кривой доходности в окрестности нуля определяется только исходными данными, а не требованием гладкости, как это было в [2]. Таким образом приходим к задаче:

$$\begin{cases} \dot{z} = x^2, \quad z(0) = 0, \\ \dot{x} = u, \\ \dot{r}_l = 0, \quad l = 1, \ldots, m, \\ J(u) = \dfrac{\alpha^2}{2} \displaystyle\int_0^T \tau u^2(\tau) d\tau + \sum_{k=1}^{N} \frac{w_k}{2} \left(\sum_{i=0}^{n} F_{ki} e^{-z(t_i) - r_k t_i} - P_k \right)^2, \\ J(u) \to \min\limits_u \end{cases}$$

Для решения данной задачи введем вспомогательные переменные $y_k(t)$, а второе слагаемое в функционале занесем под интеграл.

$$y_k(t) = \sum_{i\colon t_i < t} F_{ki} e^{-z(t_i) - r_k t_i}.$$

Тогда получим задачу:

$$\begin{cases} \dot{z} = x^2, \quad z(0) = 0, \\ \dot{x} = u, \\ \dot{y}_k = \sum\limits_{i=0}^{n} F_{ki} e^{-z-r_k t} \delta(t - t_i), \quad y_k(0) = 0, \\ \dot{r}_l = 0, \quad l = 1, \ldots, m, \\ J(u) = \dfrac{\alpha^2}{2} \int\limits_0^T \tau u^2(\tau) + \\ \qquad + \sum\limits_{k=1}^{N} w_k \left(y_k(\tau) - P_k\right) \sum\limits_{i=0}^{n} F_{ki} e^{-z-r_k t} \delta_w(t - t_i)\, d\tau, \\ J(u) \to \min\limits_u \end{cases}$$

Здесь $\delta(t)$ – дельта-функция Дирака. Для применения принципа максимума Понтрягина [13] воспользуемся «разрывной» заменой времени. Введем функцию $\nu(t)$, останавливающую время.

$$\begin{cases} \dot{z} = x^2 \cdot \nu(t), \quad z(0) = 0, \\ \dot{x} = u \cdot \nu(t), \\ \dot{y}_k = \sum\limits_{i=0}^{n} F_{ki} e^{-z-r_k t} (1 - \nu_i(t)), \quad y_k(0) = 0, \\ \dot{r}_l = 0, \quad l = 1, \ldots, m, \\ J(u) = \dfrac{\alpha^2}{2} \int\limits_0^T \tau u^2(\tau) \cdot \nu(\tau) + \\ \qquad + \sum\limits_{k=1}^{N} w_k \left(y_k(\tau) - P_k\right) \sum\limits_{i=0}^{n} F_{ki} e^{-z-r_k t} (1 - \nu_i(t))\, d\tau, \\ J(u) \to \min\limits_u \end{cases} \tag{2}$$

Здесь

$$\nu(t) = \min_{i=0,\dots,n} \nu_i(t),$$

$$\nu_i(t) = \begin{cases} 0, & t \in [t_i + i, t_i + i + 1], \quad i = 0, \dots, n, \\ 1, & \text{иначе.} \end{cases}$$

«Остановленное» время будет иметь динамику $\dot{\tau} = \nu(t)$. Выпишем Гамильтониан системы (2):

$$H(z, x, u, \overline{\psi}, y_k, r_k, t) = \psi_0 \frac{\alpha^2}{2} t u^2 \cdot \nu(t) +$$
$$+ \psi_0 \sum_{k=1}^{N} \frac{w_k}{2} (y_k - P_k) \sum_{i=0}^{n} F_{ki} e^{-z - r_k t} (1 - \nu_i(t)) +$$
$$+ \psi_x x^2 \cdot \nu(t) + \psi_y u \cdot \nu(t) + \sum_{k=1}^{N} \psi_k \sum_{i=0}^{n} F_{ki} e^{-z - r_k t} (1 - \nu_i(t)).$$

Так как получилась задача со свободным правым концом, то $\psi_0 = -1$, [13]. Сопряженная система:

$$\begin{cases} \dot{\psi}_x = -\sum_{k=1}^{N} w_k (y_k - P_k) \sum_{i=0}^{n} F_{ki} e^{-z - r_k t} (1 - \nu_i(t)) + \\ \quad + \sum_{k=1}^{N} \psi_k \sum_{i=0}^{n} F_{ki} e^{-z - r_k t} (1 - \nu_i(t)), \\ \dot{\psi}_y = -2 \psi_x x \cdot \nu(t), \\ \dot{\psi}_k = w_k \dot{y}_k, \\ \dot{\varphi}_l = \sum_{k} [w_k (y_k - P_k) - \psi_k] \sum_{i=0}^{n} F_{ik} e^{-z} e^{-r_k t} (1 - \nu_i(t)) t. \end{cases} \tag{3}$$

Сумма берется по всем k таким, что k-я облигация принадлежит l-й стране. Условия трансверсальности:

$$\psi_x(T) = \psi_y(0) = \psi_y(T) = \psi_k(T) = \varphi_k(0) = \varphi_k(T) = 0. \tag{4}$$

Определим опмимальную функцию $u(t)$ из условия максимума [13]:

$$H(z, x, u^*, \overline{\psi}, y_k, r_k, t) = \max_u H(z, x, u, \overline{\psi}, y_k, r_k, t), \tag{5}$$

$$u^*(t) = \frac{\psi_y(t)}{\alpha^2}, \text{ при } \nu(t) = 1. \tag{6}$$

Заметим, что управление $u(t)$ не определено в нуле. Доопределим по непрерывности:

$$u(0) = \lim_{t \to 0+} \frac{\psi_y(t)}{\alpha^2 t} = \left\{ \frac{0}{0} \right\} = \lim_{t \to 0} \frac{\dot{\psi}_x}{\alpha^2} = -\frac{2\psi_x(0)}{\alpha^2} x(0).$$

В силу условий трансверсальности (4) имеем, что производная мгновенной форвардной процентной ставки $f(t)$ на правом конце равна нулю:

$$f'(T) = \frac{d}{dt} x^2 \mid_{t=T} = 2x(T)\dot{x}(T) = 0. \tag{7}$$

Введем новую переменную $q_k = y_k(T)$ – цена k-й облигации, полученная дисконтированием потока платежей по кривой доходности.

$$\psi_k = w_k(y_k - q_k),$$

$$\dot{\psi}_x = -\sum_{k=1}^{N} w_k(q_k - P_k) \sum_{i=0}^{n} F_{ki} e^{-z - r_k t}(1 - \nu_i(t)).$$

$$\psi_x = \sum_{k=1}^{N} w_k(q_k - P_k) \sum_{i}^{t_i > t} F_{ki} e^{-z(t_i) - r_k t_i}, \tag{8}$$

то есть ψ_x - кусочно постоянная функция. Тогда $\psi_x = \psi_i$ на $[t_{i-1}, t_i)$, где

$$\psi_i = \sum_{k=1}^{N} w_k(q_k - P_k) \sum_{s=i}^{n} F_{ks} e^{-z(t_s) - r_k t_s}.$$

Найдем уравнение для r_l. В силу условий трансверсальности (4) мы знаем, что $\varphi_l(t)$ на концах обращается в ноль, и мы знаем выражение для производной $\dot{\varphi}_l(t)$ (3), поэтому:

$$\varphi_l(T) = \int\limits_0^T \dot{\varphi}_l(\tau)\, d\tau =$$

$$= \sum_k \sum_{i=0}^n \left[w_k(y_k(t_i) - P_k) - \psi_k(t_i)\right] F_{ik} e^{-z(t_i) - r_l t_i} t_i =$$

$$= \sum_k \sum_{i=0}^n \left[w_k(y_k(t_i) - P_k) - w_k y_k(t_i) + w_k q_k\right] F_{ik} e^{-z(t_i) - r_l t_i} t_i =$$

$$= \sum_k w_k \sum_{i=0}^n \left[q_k - P_k\right] F_{ik} e^{-z(t_i) - r_l t_i} t_i = 0.$$

Отсюда получаем уравнение на r_l, $l = 1, \ldots, m$:

$$\sum_k \sum_{i=0}^n \left[q_k - P_k\right] F_{ik} e^{-z(t_i) - r_l t_i} t_i = 0. \tag{9}$$

В силу (2), (6) имеем:

$$\dot{x} = u \cdot \nu(t) = \frac{\psi_y(t) \cdot \nu(t)}{\alpha^2 t} \Rightarrow \dot{x} t = \frac{\psi_y \cdot \nu(t)}{\alpha^2} \Rightarrow$$

$$\Rightarrow \ddot{x} t + \dot{x} = -\frac{2\psi_x \cdot \nu(t)}{\alpha^2} x, \quad \lambda = -\frac{2\psi_x}{\alpha^2}. \tag{10}$$

Таким образом получили задачу для $x(t)$:

$$\ddot{x} t + \dot{x} = \lambda x \cdot \nu(t), \tag{11}$$

$$\dot{x}(0) = \lambda_1 x(0), \quad \dot{x}(T) = 0. \tag{12}$$

Можно показать, что общим решением такого уравнения

будет

$$x(t) = \begin{cases} C_{2i-2}Y_0(2\sqrt{-\lambda_i t}) + C_{2i-1}J_0(2\sqrt{-\lambda_i t}), & \lambda_i < 0, \\ C_{2i-2}K_0(2\sqrt{\lambda_i t}) + C_{2i-1}I_0(2\sqrt{\lambda_i t}), & \lambda_i > 0, \\ C_{2i-2}\ln t + C_{2i-1}, & \lambda_i = 0, \\ t \in [t_{i-1}, t_i), & i = 1, 2, \ldots, n. \end{cases}$$

Здесь $Y_0(t)$, $J_0(t)$, $K_0(t)$, $I_0(t)$ — функции Бесселя и модифицированные функции Бесселя первого и второго рода [14]. Так как $Y_0(0)$ и $K_0(0)$ равны плюс и минус бесконечности, следовательно $C_0 = 0$. Для определение коэффициентов C_i, $i = 1, 2, \ldots, 2n-1$ выпишем условия непрерывности и непрерывной дифференцируемости $x(t)$, которые следуют из того, что $x''(t)$ кусочно непрерывна. Для этого необходимо знать производные функций Бесселя. Как показано в [14] для функций Бесселя справедливы следующие соотношения:

$$\begin{cases} \dfrac{d}{dt}J_0(2\sqrt{-\lambda t}) = -\sqrt{\dfrac{-\lambda}{t}}J_1(2\sqrt{-\lambda t}), \\ \dfrac{d}{dt}Y_0(2\sqrt{-\lambda t}) = -\sqrt{\dfrac{\lambda}{t}}Y_1(2\sqrt{-\lambda t}), \\ \dfrac{d}{dt}I_0(2\sqrt{\lambda t}) = \sqrt{\dfrac{\lambda}{t}}I_1(2\sqrt{\lambda t}), \\ \dfrac{d}{dt}K_0(2\sqrt{\lambda t}) = -\sqrt{\dfrac{\lambda}{t}}K_1(2\sqrt{\lambda t}). \end{cases}$$

Таким образом, при $t \in [t_0, t_1)$ получается, что

$$x'(t) = \begin{cases} -C_1\sqrt{\frac{-\lambda}{t}}J_1(2\sqrt{-\lambda t}), & \lambda_1 < 0, \\ C_1\sqrt{\frac{\lambda}{t}}I_1(2\sqrt{\lambda t}), & \lambda_1 > 0, \\ C_1, & \lambda_1 = 0. \end{cases}$$

Если $\lambda_1 > 0$, то

$$x'(0) = C_1\sqrt{-\lambda_1}\lim_{t\to 0+}\frac{J_1(2\sqrt{-\lambda_1 t})}{\sqrt{t}} = \\ = C_1\sqrt{-\lambda_1}\lim_{t\to 0+}\frac{\sqrt{-\lambda_1 t}+\overline{o}(\sqrt{t})}{\sqrt{t}} = \\ = \lambda_1 C_1 = \lambda_1 x(0). \quad (13)$$

Аналогично, если $\lambda_1 < 0$, то

$$x'(0) = C_1\sqrt{\lambda_1}\lim_{t\to 0+}\frac{I_1(2\sqrt{\lambda_1 t})}{\sqrt{t}} = \\ = C_1\sqrt{\lambda_1}\lim_{t\to 0+}\frac{\sqrt{\lambda_1 t}+\overline{o}(\sqrt{t})}{\sqrt{t}} = \\ = \lambda_1 C_1 = \lambda_1 x(0). \quad (14)$$

Дважды продифференцируем соотношение $z(t) = r(t)t$, возьмем его в точке $t = 0$ и воспользуемся соотношениями (13)-(14):

$$\dot{r}(0) = \lambda_1 x^2(0). \quad (15)$$

Анализируя выражение для $\dot{r}(0)$ можно заметить, что оно зависит от разности между реальными ценами P_k и ценами, полученными дисконтированием потока платежей по кривой доходности (это следует из формул (8), (10)), и не зависит от требований гладкости (как это было в [2]).

Выражение для $x(t)$ зависит от $z(t_i)$, которые в свою очередь зависят от $x(t)$, поэтому будем решать данную задачу численно. Опишем простой численный метод для поиска решения. Временно́й промежуток $[0, T]$ разбивается на отрезки $[t_{i-1}, t_i]$, $i = 1, \ldots, n$. Затем получаются выражения для $u(t_i)$, $z(t_i)$ через $f(t_i)$, интеграл заменяется на сумму. И запускается многомерная оптимизация по $f(t_i)$. Более продвинутый численный метод для поиска $z(t_i)$ описан в [12]. В качестве начального

приближения выбирается плоская процентная ставка $r(t) \equiv r$, которая является решением одномерной задачи минимизации:

$$\sum_{k=1}^{N} \frac{w_k}{2} \left(\sum_{i=0}^{n} F_{ki} e^{-rt_i} - P_k \right)^2 \to \min_r .$$

Для решения уравнения (9) воспользуемся методом Ньютона. Обозначим левую часть через $f(r_l)$. В качестве начального приближения выбирается $r_l^0 = 0$. Тогда

$$r_l^{n+1} = r_l^n - \frac{f(r_l^n)}{f'(r_l^n)}.$$

4 Иллюстрации

Проиллюстрируем рассмотренную выше теорию на реальных данных. Будем использовать данные о котировках 490 еврооблигаций за август 2011 года одиннадцати стран (Австрия, Бельгия, Германия, Испания, Финляндия, Франция, Ирландия, Италия, Нидерланды, Словакия и Люксенбург), предоставляемых информационным агенством Bloomberg.

5 Заключение

В статье рассматривается новых подход к определению временно́й структуры процентных для облигаций с разным кредитным качеством эмитентов. Данный подход использует идеи, предложенные в [2], [3]. Авторы ставили своей целью изменить функционал так, чтобы получались экономически обоснованные результаты.

Из условия равенство нулю мгновенной форвардной ставки $f'(T) = 0$ (7) можно вывести, что кривая доходности при увеличении горизонта T будет стремиться к константе. Так как производная в нуле кривой доходности $\dot{r}(0)$ не зависит от требований гладкости (это следует из формул (8), (10), (15)), то

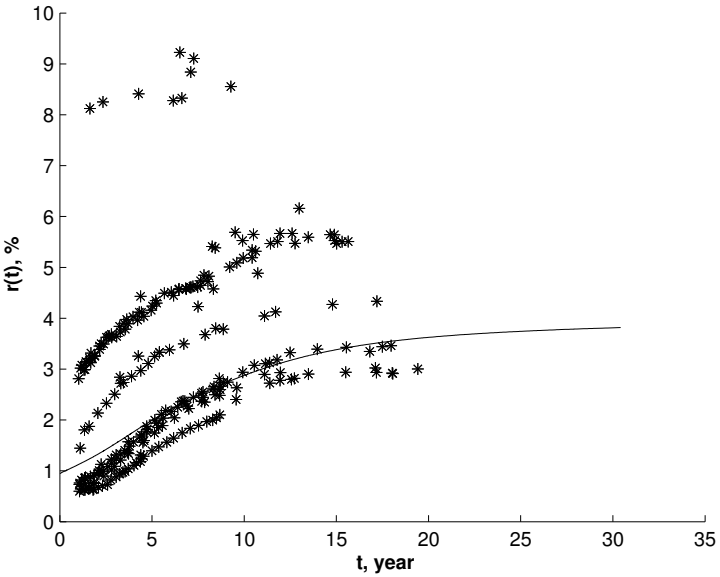


Рис. 1. Базовая кривая доходности для одиннадцати стран Еврозоны за 22 августа 2011 года. Звездочками обозначены точки «дюрация» – «доходность к погашению».

поведение кривой доходности на коротком конце $t = 0$ определяется только исходными данными.

Рассмотренная модель спрэда как параллельный сдвиг базовой кривой имеет свои плюсы и минусы. Главный плюс — это простота. При рассмотрении нескольких групп эмитентов у нас добавилось минимальное количество новых параметров (ровно m). Но благодаря такой простой модели возможно построить кривую доходности для стран с небольшим количеством облигаций (например, рисунок 4) и для стран с различным по ликвидности выпусками облигаций (например, рисунок 5). К минусам можно отнести, что в силу простоты модели, в некоторых случаях форма кривой доходности получается не совсем

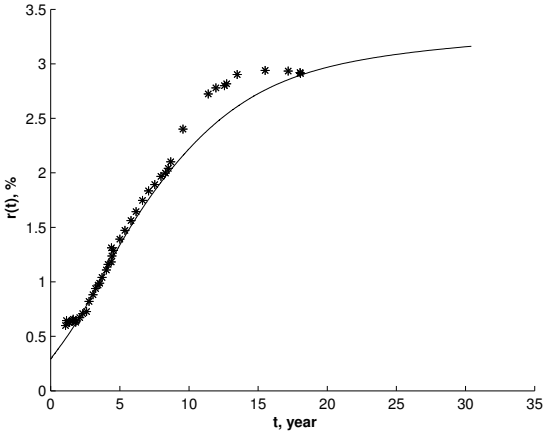


Рис. 2. Кривая доходности Германии за 22 августа 2011 года. Звездочками обозначены точки «дюрация» – «доходность к погашению».

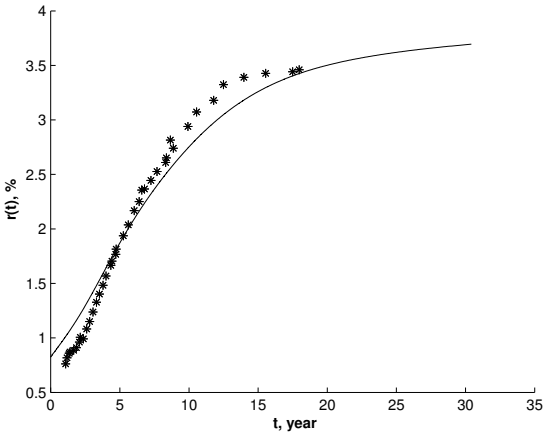


Рис. 3. Кривая доходности Франции за 22 августа 2011 года. Звездочками обозначены точки «дюрация» – «доходность к погашению».

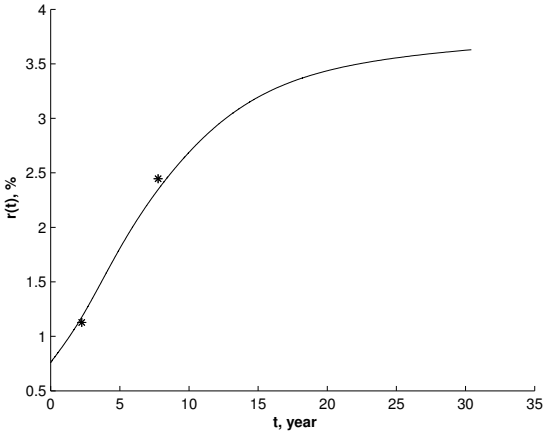


Рис. 4. Кривая доходности Люксембурга за 22 августа 2011 года. Звездочками обозначены точки «дюрация» – «доходность к погашению».

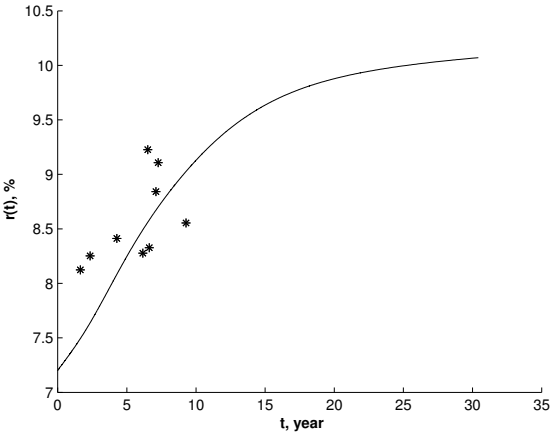


Рис. 5. Кривая доходности Ирландии за 22 августа 2011 года. Звездочками обозначены точки «дюрация» – «доходность к погашению».

точной (например, рисунок 3).

Форма кривой доходности, полученная в данной статье, является экономически обоснованной. А использование простейшей модели спрэда позволяет использовать данный метод определения срочной структуры процентных ставок для группы облигаций разного кредитного качества.

Авторы выражают благодарность Тарасовой Полине за предоставленные данные.

Список литературы

[1] *Adams K., Van Deventer D.* Fitting yield curves and forward rate curves with maximum smoothness // J. Fixed Income. 1994. 1. N 4. P. 52-62.

[2] *Лапшин В. А.* Определение срочной структуры процентных ставок // Вестник Московского университета. Серия 15, Вычислительная математика и кибернетика. No.4. – 2009. – С. 37-43.

[3] *Косьяненко А. В., Лапшин В. А., Смирнов С. Н.* Методика построения кривой безрисковой доходности «спот» и кредитных спрэдов для группы облигаций с неоднородным кредитным качеством эмитентов: стандарт EFFAS-EBC для стран Еврозоны. // WP16/2007/03. — М.: ГУ ВШЭ, 2007. — 40 с.

[4] *Тихонов А. Н., Арсенин В. Я.* Методы решения некорректных задач. М.: Наука, 1979.

[5] *Nelson C., Siegel A.* Parsimonious modeling of yield curves // J. Bus. 1987. N 60. P. 473-489.

[6] *Svensson L.* Estimating forward interest rates with extended Nelson-Siegel method // Sveriges Riskbank Quartely Review. 1995. N 3. P. 13-26.

[7] *Балабушкин А., Гамбаров Г., Никитин А., Шевчук И.* Кривая бескупонной доходности на рынке ГКО-ОФЗ // Рынок ценных бумаг. 2006. №3. С. 68-77.

[8] *McCulloch J.* Measuring the term structure of interest rates // J. Bus. 1971. 44. N 1. P. 19-31.

[9] *McCulloch J.* The tax adjusted yield curve // J. Finance. 1975. N 30. P. 811-829.

[10] *Steeley J.* Estimating the gilt-edged term structure: Basis splines and confidence intervals // J. Bus. Fin. Acc. 1991. 18. N 4. P. 513-529.

[11] *Vasicek O., Fong G.* Term structure modeling using exponential splines // J. Finance. 1982. 37. N 2. P. 339-348.

[12] *Wahba, G.* (1990). Spline Models for Observational Data. SIAM, CBMS-NSF Regional Conference Series in Applied Mathematics, V59, Philadelphia.

[13] *Понтрягин Л. С., Болтянский В. Г., Гамкрелидзе Р. В., Мищенко Е. Ф.* Математическая теория оптимальных процессов, 4-е изд. М.: Наука, 1983. - 393 с.

[14] *Ватсон Г. Н.* Теория бесселевых функций, часть 1. М.: Издательство иностранной литературы, 1945. Перевод с 2-го английского издания Бермана В. С.

УДК 681.518

Алгоритм восстановления глобального контекста

© 2012 г. Д. Е. Стельмашенко

dashanikolaeva@gmail.com

Кафедра алгоритмических языков

1 Введение

Теория формальных понятий [1] позволяет построить понятийную структуру проблемной области при помощи так называемых формальных контекстов. По сути формальный контекст является отчетом о полевых исследованиях, в котором перечисляются встречающиеся в проблемной области объекты, признаки и отношения между ними. Как правило, подобные отчеты не бывают полными, что влечет за собой необходимость объединения нескольких отчетов-контекстов. Корректное объединение контекстов предполагает учет ограниченных возможностей приборов наблюдения, а также то, что информацию приносят несвязанные между собой участники экспедиций. Настоящая статья, также как и работы [2, 3], демонстрирует конструктивный подход к решению описанной задачи при помощи аппарата формальных понятий [1].

В настоящей работе представлен алгоритм объединения нескольких формальных контекстов в один глобальный контекст. Этот алгоритм, как и сама возможность такого объединения, доказанная в [2], является актуальным продолжением исследований свойств локальных и глобальных контекстов, проведенных в [4].

2 Формальные контексты и формальные понятия

Будем использовать букву N для обозначения множества элементов, именуемых признаками. Множество N состоит из подмножеств N_1, N_2, ..., N_k, причем условие $N_i \cap N_j = \emptyset$ не обязательно.

Объектами будем называть подмножества признаков. Для их обозначения будем использовать строчные буквы латинского алфавита: a, b, c и т. д. $a \subseteq N$, $b \subseteq N$ и т. д. Любой объект a может быть представлен в виде объединения подмножеств: $a = a_1 \cup a_2 \cup \ldots \cup a_k$, где
$a_1 = a \cap N_1$, $a_2 = a \cap N_2$, ..., $a_k = a \cap N_k$.

Классом назовем непустое подмножество объектов. Обозначать классы будем заглавными буквами латинского алфавита: A, B, C и т. д. Пусть A — класс, c — множество признаков. Введем операции $Con(A)$ и $A[c]$:

$$Con(A) = \bigcap_{a \in A} a, \qquad A[c] = \{\ a \in A \mid c \subseteq a\ \}.$$

Определение 1. *Формальным контекстом или просто контекстом называется пара $< D, N >$, где D — класс объектов, N — множество признаков и $D \subseteq 2^N$.*

Не ограничивая общности, будем предполагать, что

$$Con(D) = \emptyset. \tag{1}$$

Это требование не повлияет на ход дальнейших рассуждений, поскольку при наличиии у всех объектов контекста общей части ее можно предварительно вывести за рамки рассмотрения, а затем вернуть в нужный момент.

Контекст $< D, N >$ можно задавать простым перечислением элементов D (объектов), поскольку множество признаков N однозначно им определяется:

$$N = \{\ n\ \in\ d \mid d\ \in\ D\ \}.$$

Пример 1. *Приведем пример задания формального контекста:*

$D_S = \{\ \{\ 0,\ 2,\ 4,\ 6,\ 8\ \},\ \{\ 0,\ 2,\ 5,\ 7,\ 8\ \},\ \{\ 0,\ 3,\ 4,\ 7,\ 9\ \},$
$\{\ 1,\ 2,\ 5,\ 6,\ 9\ \},\ \{\ 0,\ 3,\ 5,\ 6,\ 9\ \}\ \},$
$N_S = \{\ 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9\ \}.$

Как уже было замечено выше, перечисление объектов N_S при задании контекста не обязательно.

Определение 2. *Формальным понятием в контексте $< D, N >$ называется такой класс объектов A, $A \subseteq D$, для которого справедливо равенство $A = \{\ d \in D \mid Con(A)\ \subseteq d\ \}$. При этом множество $Con(A)$ называется содержанием формального понятия A.*

Классическое представление формального понятия [1] подразумевает расссмотрение его как пары: объем (экстент, экстенсионал, ext) и содержание (интент, интенсионал, int). Объем понятия соответствует набору входящих в него объектов, а содержание отображает общие для этих объектов признаки. В настоящей работе, как и в работах [2, 3], объем понятия и его содержание рассматриваются как равносильные множества, каждое из которых однозначно определяет понятие. Пара множеств A и $Con(A)$, где A — понятие, именуется понятием-в-полной-форме или пф-понятием.

Использование операции $D[\cdot]$ позволяет ввести два эквивалентных определения формального понятия:

1. В контексте $< D, N >$ класс A есть формальное понятие тогда и только тогда, когда $A\ =\ D[Con(A)]$.

2. В контексте $< D, N >$ класс $D[c]$ есть формальное понятие тогда и только тогда, когда $c\ =\ Con(D[c])$.

Согласно требованию (1), в любом формальном контексте $< D, N >$ класс D является формальным понятием, а его содержание есть пустое множество $\emptyset$.

Пусть $< D, N >$ — формальный контекст.

Будем обозначать

D^F — множество всех понятий в контексте D;

D^C — множество содержаний понятий из контекста D;

$D^C = \{ \ Con(A) \mid A \in D^F \ \}$.

Пример 2. *Формальный контекст из примера* 1 *порождает двадцать одно формальное понятие, причем:*

$D_S^F = \{ \ \{(\emptyset)\}; \{(1,2,5,6,9)\}; \{(0,3,5,6,9)\}; \{(0,3,4,7,9)\}; \{(0,2,5,7,8)\}; \{(0,2,4,6,8)\}; \{(1,2,5,6,9), (0,3,5,6,9)\}; \{(0,3,4,7,9), (0,3,5,6,9)\}; \{(0,2,4,6,8), (0,2,5,7,8)\}; \{(0,2,4,6,8), (1,2,5,6,9)\}; \{(0,2,5,7,8), (1,2,5,6,9)\}; \{(0,2,5,7,8), (0,3,4,7,9)\}; \{(0,2,4,6,8), (0,3,5,6,9)\}; \{(0,2,5,7,8), (0,3,5,6,9)\}; \{(0,2,4,6,8), (0,3,4,7,9)\}; \{(0,3,4,7,9), (1,2,5,6,9), (0,3,5,6,9)\}; \{(0,2,4,6,8), (1,2,5,6,9), (0,3,5,6,9)\}; \{(0,2,5,7,8), (1,2,5,6,9), (0,3,5,6,9)\}; \{(0,2,4,6,8), (0,2,5,7,8), (1,2,5,6,9)\}; \{(0,2,4,6,8), (0,2,5,7,8), (0,3,4,7,9), (0,3,5,6,9)\}; \{(0,2,4,6,8), (0,2,5,7,8), (0,3,4,7,9), (1,2,5,6,9), (0,3,5,6,9)\}\}$.

$D_S^C = \{ \ (\emptyset), (0), (2), (5), (6), (9), (0,4), (0,5), (0,6), (0,7), (2,5), (2,6), (0,2,8), (0,3,9), (5,6,9), (0,2,4,6,8), (0,2,5,7,8), (0,3,4,7,9), (0,3,5,6,9), (1,2,5,6,9), (0,1,2,3,4,5,6,7,8,9) \ \}$.

3 Глобальные и локальные формальные контексты

Рассмотрим контекст $< D, N >$. Если множесво его признаков N задано двумя подмножествами: N_1 и N_2 ($N = N_1 \cup N_2$), то эти подмножества однозначно порождают пару контекстов $< D_1, N_1 >$ и $< D_2, N_2 >$, где

$D_1 = \{d_1 \mid d_1 + d_2 \in D\}$ и $D_2 = \{d_2 \mid d_1 + d_2 \in D\}$. Контексты $< D_1, N_1 >$ и $< D_2, N_2 >$ будем называть локальными, а контекст $< D, N >$ — глобальным. Количество локальных контекстов прямо пропорционально количеству подмножеств N_1, N_2, ..., N_k множества N. Как было показано выше, подмножества N_1, N_2, ..., N_k однозначно конструируют локальные контексты $< D_1, N_1 >$, $< D_2, N_2 >$, ..., $< D_k, N_k >$. Однако обратная задача конструирования глобального контекста по известным локальным контекстам, вообще говоря, не имеет однозначного решения. Для восстановления глобального контекста по известным локальным контекстам требуется дополнительная информация, проявляющаяся в закономерностях связей между локальными и глобальными контекстами.

В работах [2–4] подробно изучаются свойства локальных и глобальных контекстов, а также связи между наборами их формальных понятий. В [2] приводятся доказательства двух принципов, лежащих в основе решения задачи однозначного восстановления глобального контекста по известным локальным контекстам. Это принципы тестирования и альтернативного конструирования.

4 Принцип альтернативного конструирования

Пусть заданы наборы локальных контекстов, каждый из которых получался бы из неизвестного глобального контекста при решении прямой задачи. Количество локальных контекстов в наборе может быть любым — от 2 до k. Идея принципа альтернативного конструирования состоит в конструировании всех возможных вариантов объединений формальных понятий контекстов из каждого набора и дальнейшем пересечении полученных множеств.

Будем считать, что списки формальных понятий каждого локального контекста представлены в виде D^C. В та-

ком случае необходимо построить множество R, состоящее из всех возможных объединений элементов множеств D^C локальных контекстов. В случае двух локальных контекстов такое объединение проще всего представить в виде таблицы R, первая строка и первый столбец которой содержат списки формальных понятий первого и второго локальных контекстов соответственно. Каждая клетка таблицы в таком случае заполняется элементом R_i, j, равным объединению соответствующих элементов множеств D^C локальных контекстов: $R_i, j = Con(A_i) \cup Con(B_j)$, где $Con(A_i)$ и $Con(B_j)$ — элементы множеств D^C локальных контекстов (то есть формальные содержания соответствующих формальных понятий локальных контекстов).

В [2] показано, что множество R обязательно содержит в себе все формальные понятия искомого глобального контекста. Однако, это множество содержит и много псевдопонятий — элементов, не входящих в список формальных понятий глобального контекста. Решение задачи восстановления глобального контекста заключается в исключении всех псевдопонятий из множества R.

Идея принципа альтернативного конструирования состоит в том, чтобы строить множества R для различных наборов локальных контекстов и пересекать их между собой. Пусть множества R_i и R_j соответствуют i-му и j-му наборам локальных контекстов. Очевидно, что те элементы множеств R_i и R_j, которые не войдут в их общую часть $R_i \cap R_j$, не могут быть формальными понятиями глобального контекста в силу того, что любое множество R содержит в себе полный список формальных понятий глобального контекста. Итеративным построением и пересечением между собой множеств R для различных наборов локальных контекстов можно восстановить набор формальных понятий глобального контекста (а, следовательно, и сам глобальный контекст). Подробно принцип альтернативного конструирования описан в [2].

5 Принцип тестирования

Идея принципа тестирования состоит в проверке элементов множества R на объектах некоторого локального контекста с последующей экстраполяцией результатов этой проверки на глобальный контекст $< D, N >$.

Пусть C — подлежащий тестированию элемент R, c' — фиксированное множество признаков, а $< D_t, N_t > = < D_t, c \cup c' >$ - локальный контекст. Тогда, согласно [2], возможны три случая:

1. $D_t[c] = \emptyset$. Тогда $D[c] = \emptyset$ и $D[c + c_x] = \emptyset$ для любого c_x из N.

2. $D_t[c] \neq \emptyset$ и $c \subset Con(D_t[c])$. Тогда $D_t[c] = (D[c])_t$ и $c \subset Con(D_1[c]) \subseteq Con(D[c])$, а также множество $c \cup c''$, где $c'' \subseteq N \setminus N_t$, не является содержанием какого-либо понятия в глобальном контексте $< D, N >$. Однако в контексте $< D, N >$ существует понятие $D[c]$, содержание которого есть $c \cup c_x$, где c_x — непустое множество признаков.

3. $D_t[c] \neq \emptyset$ и $c = Con(D_t[c])$. Тогда в глобальном контексте $< D, N >$ существует $D[c]$, содержание которого есть $c \cup c_x$, где c_x — некоторое множество признаков.

Подробно принцип тестирования описан в [2].

6 Алгоритм восстановления глобального контекста

Опишем алгоритмически процесс восстановления неизвестного глобального контекста по известным наборам локальных контекстов. Начнем с постановки задачи: требуется однозначно восстановить глобальный контекст $< D, N >$ по имеющимся наборам локальных контекстов

$\{< D_1^1, N_1^1 >, < D_1^2, N_1^2 >, \ldots, < D_1^n, N_1^n >\}, \ldots,$
$\{< D_k^1, N_k^1 >, < D_k^2, N_k^2 >, \ldots, < D_k^m, N_k^m >\}.$
Количество локальных контекстов в каждом наборе произвольно и определяется количеством разбиений $N^1, \ldots, N^n$ множества признаков глобального контекста N.

Алгоритм решения поставленной задачи подразумевает описание алгоритмов принципа тестирования и принципа альтернативного конструирования, а также описание алгоритма итеративного вызова этих принципов в ходе работы программы. Каждый из принципов может решить поставленную задачу самостоятельно, однако, допускается и попеременная работа принципов, причем смена используемого принципа может происходить на любой итерации. Задача оптимального выбора следующего набора локальных контекстов и используемого метода (принципа) для каждой итерации остается пока нерешенной и является целью дальнейших исследований в данной области. В алгоритмическом описании решения задачи восстановления глобального контекста ограничимся введением некоторой функции `optimum`, которая по неизвестному пока алгоритму должна определять оптимальный для следующего шага выбор локальных контекстов и метода (принципа тестрования или принципа альтернативного конструирования).

6.1 Алгоритм альтернативного конструирования

Процедура `Construct` строит $R[\,]$ — список кандидатов в содержания формальных понятий глобального контекста. $R[\,]$ состоит из всех возможных объединений элементов множеств D^c для k локальных контекстов, входящих в один набор. На вход процедуре `Construct` подается массив $S[1..k]$ из списков D^c для k локальных контекстов ($S[i]$ — это список D^c для i-ого локального контекста рассматриваемого набора).

```
Procedure Construct(In: S[1..k]; Out: R[ ])
i ← 1
R[ ] ← Constr(S[i], S[i+1])
```

```
If k ≠ i+1 then  // см. замечание 1
    { R[ ] ← constr(R[ ], S[i+2])
     i ← i + 1 }
Return R[ ]
```

Замечание 1. k — это число элементов в списке $S[1..k]$.

Процедура `Construct` использует процедуру `Constr`, которая строит множество всех возможных объединений элементов множеств D^c для двух локальных контекстов. Опишем алгоритм работы этой процедуры. На вход процедуре `Constr` подаются два списка $A[1..m]$ и $B[1..n]$. Результатом работы этой процедуры является список $O[1..(m \times n)]$, содержащий все возможные варианты объединения элементов списков $A[1..m]$ и $B[1..n]$.

```
Procedure Constr(In: A[1..m], B[1..n]; Out: O[ ])
v ← 0
for i from 1 to m // см. замечание 1
    { for j from 1 to n // см. замечание 2
        { v ← v + 1
        O[v] ← A[i] ∪ B[j] }
    }
Return O[ ]
```

Замечание 1. m — это число элементов в списке $A[1..m]$.

Замечание 2. n — это число элементов в списке $B[1..n]$.

Алгоритм построения для контекста множества D^c хорошо изучен [1, 5]. В связи с этим будем считать этот алгоритм известным.

Пусть имеется всего z наборов локальных контекстов. Каждый набор представляет собой список $S[1..k]$, в свою очередь также состоящий из списков, элементы которых соответствуют формальным содержаниям понятий соответствующего локального контекста ($S[i] = D_i^c$). Пусть массив $G[1..z]$ содержит все z известных наборов локальных контекстов:

$G[1..z] = [(S_1^1 \dots S_k^1), \dots, (S_1^z \dots S_l^z)]$.

В таком случае алгоритм решения задачи восстановления

глобального контекста по известным локальным контекстам может быть записан следующим образом:

```
Result ← Construct(G[1])
for i from 1 to z // см. замечание 1
{ Result ← Intersection(Result, Construct(G[i+1]))
If Correct(Result) then
Return Result }
```

Замечание 1. z — это число элементов в списке $G[\,]$.

Алгоритм решения задачи восстановления глобального контекста использует две процедуры: процедуру `Intersection` и процедуру `Correct`.

Процедура `intersection(A[ ], B[ ])` возвращает общие для списков $A[\,]$ и $B[\,]$ элементы.

Процедура `Correct` делает следующее:

1. Строит формальный контекст по списку `Result` = D^c.

2. Разделяет признаки построенного контекста согласно тому, как они делились в любом из наборов локальных контекстов.

3. Сравнивает полученные локальные контексты с исходными.

Далее, если контексты совпадут, значит список `Result` совпадает с множеством D^c искомого глобального контекста, то есть контекст восстановлен правильно и задача решена. В таком случае функция `Correct` выдаст результат `True`, после чего описанный выше алгоритм вернет в качестве результата список `Result`. Если же полученные локальные контексты не совпадут с исходными, значит в списке `Result` есть ложные понятия, функция `Correct` вернет значение `False` и алгоритм продолжит свою работу.

Приведем пример работы алгоритма альтернативного конструирования.

Пример 3. *Пусть в роли неизвестного глобального контекста выступает контекст* $< D_S, N_S >$ *из примера* 1 *и пусть в роли первого набора локальных контекстов выступает*
$G[1] = (< D_S^1, N_S^1 >, < D_S^2, N_S^2 >)$, *а в роли второго набора*
$G[2] = (< D_S^3, N_S^3 >, < D_S^4, N_S^4 >)$, *где*
$D_S^1 = \{\{0,2\},\{0,3\},\{1,2\}\}$,
$D_S^2 = \{\{4,6,8\},\{5,7,8\},\{4,7,9\},\{5,6,9\}\}$,
$D_S^3 = \{\{0,5\},\{0,4\},\{1,5\},\{0,5\}\}$,
$D_S^4 = \{\{2,6,8\},\{2,7,8\},\{3,7,9\},\{2,6,9\},\{3,6,9\}\}$.

Таблица 1. Множество кандидатов в формальные понятия глобального контекста, полученное функцией `Construct(G[1])`

	$\emptyset$	0	02	03	12	2
$\emptyset$	$\emptyset$	0	02	03	12	2
4	4	04	024	034	124	24
468	468	0468	02468	03468	12468	2468
479	479	0479	02479	03479	12479	2479
5	5	05	025	035	125	25
569	569	0569	02569	03569	12569	2569
578	578	0578	02578	03578	12578	2578
6	6	06	026	036	126	26
7	7	07	027	037	127	27
8	8	08	028	038	128	28
9	9	09	029	039	129	29

Изначально `Result` получает значение `Construct(G[1])`. Это первое множество кандидатов в формальные содержания понятий глобального контекста, оно содержит 66 элементов. В описании принципа альтернативного конструирования оно обозначалось как R. Там же было замечено, что проще всего представить себе построение этого множества для набора из двух локальных контекстов в виде таблицы, первая строка и первый столбец которой соответствуют множествам D^c локальных контекстов, а клетки заполняются объединением со-

ответствующих элементов. Таблица 1 представляет построение множества `Construct(G[1])`.

На следующем шаге алгоритма производится операция `Result` $\leftarrow$ `Intersection(Result, Construct(G[i+1]))`, равносильная `Intersection(Construct(G[1]), Construct(G[2]))`.

Аналогично тому, как было построено множество `Construct(G[1])`, строится множество `Construct(G[2])` (см. Таблицу 2)

Таблица 2. Множество кандидатов в формальные понятия глобального контекста, полученное функцией `Construct(G[2])`

	$\emptyset$	0	04	05	15	5
$\emptyset$	$\emptyset$	0	04	05	15	5
2	2	02	024	025	125	25
26	26	026	0246	0256	1256	256
268	268	0268	02468	02568	12568	2568
269	269	0269	02469	02569	12569	2569
278	278	0278	02478	02578	12578	2578
28	28	028	0248	0258	1258	258
369	369	0369	03469	03569	13569	3569
379	379	0379	03479	03579	13579	3579
39	39	039	0349	0359	1359	359
6	6	06	046	056	156	56
69	69	069	0469	0569	1569	569
7	7	07	047	057	157	57
9	9	09	049	059	159	59

Далее находится пересечение полученных списков кандидатов в формальные понятия, то есть производится операция `Intersection(Construct(G[1]), Construct(G[2]))`. Работу этой операции демонстрирует Таблица 3. Она получена путем удаления из Таблицы 1 (первого списка кандидатов) элементов, не вошедших в множество `Construct(G[1])` $\cap$ `Construct(G[2])`. Таким образом,

Таблица 3 отображает результат сокращения количества кандидатов в формальные понятия глобального контекста после первой итерации принципа альтернативного конструирования.

Таблица 3. Оставшиеся кандидаты в понятия глобального контекста после первой итерации метода.

	$\emptyset$	0	02	03	12	2
$\emptyset$	$\emptyset$	0	02			2
4		04	024			
468			02468			
479				03479		
5	5	05	025		125	25
569	569	0569	02569	03569	12569	2569
578			02578		12578	2578
6	6	06	026			26
7	7	07				
8			028			28
9	9	09		039		

Если в первом множестве `Construct(G[1])` содержалось 66 элементов, а во втором множестве `Construct(G[2])` — 84 элемента, то после пересечения этих множеств осталось 33 элемента, каждый из которых является кандидатом в элементы множества D^c глобального контекста.

Расчеты показывают, что для однозначного восстановления глобального контекста в условиях настоящего примера необходимо произвести три итерации предложенного алгоритма принципа альтернативного конструирования, одна из которых уже проведена. Пусть для второй итерации

$G[3] = < D_S^5, N_S^5 >, < D_S^6, N_S^6 >$,

а для третьей итерации

$G[4] = < D_S^7, N_S^7 >, < D_S^8, N_S^8 >$

где $N_S^5 = 1589$, $N_S^6 = 023467$ и

$N_S^7 = 4567$, $N_S^8 = 012389$.

В таком случае на второй итерации после пересечения промежуточного `Result` с $G[3]$ останется 26 элементов, а на третьей итерации после пересечения промежуточного `Result` с $G[4]$ останется 20 элементов. Именно столько формальных понятий содержит искомый глобальный контекст, следовательно, на третьей итерации функция `Correct(Result)` выдаст значение `True` и алгоритм завершит свою работу.

Отметим, что производительность предложенного алгоритма сильно зависит от выбора набора локальных контекстов. Оптимальная стратегия выбора следующего набора локальных контекстов на каждой итерации пока остается неизвестной. В связи с этим будем предполагать, что некая функция `optimum` будет находить оптимальный выбор этого набора для каждого шага алгоритма.

6.2 Алгоритм тестирования

Пусть c — элемент R, выбранный для тестирования. Среди известных локальных контекстов выбираем тот, множество признаков N_t которого удовлетворяет соотношению $N_t = c \cup c'$, где c' — любое множество признаков: $< D_t, N_t > = < D_t, c \cup c' >$. Пусть элементами списка $F[\,]$ являются объекты множества D^c контекста $< D_t, N_t >$ и R — список кандидатов в элементы множества D^c глобального контекста.

Процедура `test` порождает $m[\,]$ — список элементов, подлежащих тестированию. На вход процедуре `test` подается $R[\,]$ — список кандидатов в формальные понятия глобального контекста и c — элемент таблицы R, выбранный для тестирования.

```
procedure test(In: R[1..t], c; Out: m[ ])
i ← 0
for a from 1 to t do // см замечание 1
    { If subseteq(c, R[a]) then // см замечание 2
        { i ← i+1;  m[i] ← R[a] } }
Return m[ ]
```

Замечание 1. t — это число элементов в списке $R[\,]$.

Замечание 2. Процедура `subseteq(c, R[a])` выдает значение `True` если элемент $c \subseteq R[a]$.

Процедура `testing` выполняет одну итерацию (один шаг) принципа тестирования. Она генерирует список $H[\,]$, состоящий из двух элементов: новой версии множества R и множества R^+ — списка достоверно известных формальных понятий глобального контекста. На вход процедуре `testing` подается $R[\,]$ — список кандидатов в элементы множества D^c глобального контекста, тестируемый элемент m, $F[\,]$ — множество D^c выбранного локального контекста и N'_t — список признаков выбранного локального контекста без тех признаков, которые входят в тестируемый элемент m. Отметим, что алгоритм генерации множества R описан в алгоритме принципа альтернативного конструирования.

```
procedure testing(In: R[1..t],m,F[1..z],N'_t; Out: H[ ])
   R+[ ] ← ∅ // случай №3
   for i from 1 to z // см. замечание 1
   {If m = F[i] then
   {If ‖Find(m, R[ ])‖ = 1 then // см. замечания 2,3
     R+[ ] ← R+[ ] + m } // см. замечание 4
   H[1] ← R[ ]
   H[2] ← R+[ ]
   return H[ ] }
   // случай №2
   for i from 1 to z // см. замечание 1
       If subset(m, F[i]) then // см. замечание 5
       { w ← Find(m, R[ ]) // см. замечания 2,6
         s ← 1
         for g from 1 to ‖w[ ]‖ do // см. замечание 3
           { for k from 1 to ‖N'_t[ ]‖ do
   { if subseteq(N'_t[k], w[g]) then // см. замечание 7
      { min[s] ← w[g] // см. замечание 6
       s ← s+1 }}}
```

```
R[ ] ← R[ ] - m - (w[ ] - min[ ]) // см. замечание 4
if ‖(Find(m, R[ ]))‖ = 1 then // см. замечания 2,3
R+[ ] ← R+[ ] + Find(m, R[ ]) // см. замечание 2,4
H[1] ← R[ ]
H[2] ← R+[ ]
return H[ ] }
// случай №1 (Dt[m] = ∅)
{R[ ] ← R[ ] - Find(m, R[ ]) // см. замечания 2,4
H[1] ← R[ ]
H[2] ← R+[ ]
return H[ ] }
```

Замечание 1. z — это число элементов в списке $F[\,]$.

Замечание 2. Процедура `Find(m, R[ ])` находит все элементы списка $R[\,]$, содержащие m.

Замечание 3. Запись $\|A[\,]\|$ обозначает число элементов в списке $A[\,]$.

Замечание 4. Записи вида $R[\,] \pm a$ обозначают добавление или удаление элемента a из списка $R[\,]$. Аналогичные обозначения будем использовать для добавления или удаления из одного списка всех элементов другого списка: $A[\,] \pm B[\,]$.

Замечание 5. Процедура `subset(a, b)` возвращает значение `True` если $a \subset b$.

Замечание 6. В процедуре `testing` вводятся две локальные переменные: `w[ ]` и `min[ ]`. Список `w[ ]` содержит те элементы списка $R[\,]$, которые содержат m. Список `min[ ]` содержит те элементы списка `w[ ]`, в которых присутствует хотя бы один признак из множества $N'_t[\,]$.

Замечание 7. Процедура `subseteq(a, b)` возвращает значение `True` если $a \subseteq b$.

Приведем пример работы алгоритма тестирования.

Пример 4. *Будем предполагать, что в роли неизвестного глобального контекста выступает контекст* $< D_S, N_S >$ *из примера* 1, *что в качестве множества* R *используется мно-*

жество `construct(G[1])` *из примера* 3 *(Таблица* 1*) и что для тестирования выбрано множество* $N_t = \{0, 3, 8\} = 038$.

Тогда $D_t = \{\{0, 8\}, \{0, 3\}, \emptyset\}$ и
$D^c = F_t[] = \{\{0, 8\}, \{0, 3\}, \{0\}, \emptyset\}$.

Процедура `test` найдет пять элементов множества R, подлежащих тестированию: 0, 03, 038, 08 и 8.

Результаты тестирования:

1. $c = 038$ — случай 1, $R[038] = \{038,\ 03468,\ 03578\}$, элементы $R[038]$ исключаются из R.

2. $c = 8$ — случай 2, множество $R[8]$ содержит 18 элементов, из R исключаются: 8, 12468, 12578, 128, 2468, 2578, 28, 468, 578.

3. $c = 0$ — случай 3, множество $R[0]$ содержит 33 элемента.

4. $c = 03$ — случай 3, множество $R[03]$ содержит 11 элементов.

5. $c = 08$ — случай 3, множество $R[08]$ содержит 9 элементов.

Расчеты показывают, что для построения искомого множества $(D_s)^c$ достаточно построить R и выполнить четыре итерации тестирования:
Построение множества R. $N_1 = 0123$, $N_2 = 456789$.
Первоначально множество R содержит 66 элементов.
1 *итерация.* $N_t = 038$. Результат: R содержит 54 элемента.
2 *итерация.* $N'_t = 028$. Результат: R содержит 43 элемента.
3 *итерация.* $N''_t = 03479$. Результат: R содержит 26 элементов.
4 *итерация.* $N'''_t = 12569$. Результат: R содержит 20 элементов.

Как уже было замечено, алгоритм наилучшего выбора элемента для тестирования и локального контекста пока остается неизвестным. Будем предполагать наличие некой функции

`Optimum`, результатом работы которой является выявление оптимальных значений этих параметров для каждой новой итерации метода.

7 Заключение

В статье предложено решение задачи однозначного восстановления глобального контекста по известным наборам локальных контекстов. Сформулированы два алгоритма для решения этой задачи - это алгоритмы, реализующие принцип альтернативного конструирования и принцип тестирования. Показано, что оба предолженных алгоритма приводят к нахождению решения задачи, однако, и алгоритм принципа альтернативного конструирования, и алгоритм принципа тестирования сильно зависят от выбора набора локальных контекстов и элементов для тестирования на каждой итерации. Задача нахождения оптимальной стратегии выбора этих параметров на каждом шаге алгоритмов остается открытой и является объектом дальнейших исследований в этой области.

Список литературы

[1] *Ganter B., Wille R.* Formal Concept Analysis: Mathematical Foundations. // Springer — 1999.

[2] *Соловьев С. Ю., Стельмашенко Д. Е.* Подходы к исследованию формальных контекстов. // Информационные процессы — 2011 — том 11, №2 — С. 277-290.

[3] *Стельмашенко Д. Е.* Свойства формальных контекстов. // Информационные процессы — 2011 — том 1, №1 — С. 86-89.

[4] *Valtchev P., Missaoui R., Lebrun P.* A partition-based approach towards constructing Galois (concept) lattices. //

Discrete Mathematics — 2002 — vol. 256, no.3. — P. 801-829.

[5] *Kuznetsov S. O., Obiedkov S. A.* Comparing Performance of Algorithms for Generating Concept Lattices. // Journal of Experimental and Theoretical Artificial Intelligence — 2002 — vol. 14, — P. 189-216.

УДК 517.977

Гарантированное оценивание в задаче поиска на плоскости

© 2012 г. В. А. Степанович

kolhizin@gmail.com

Кафедра системного анализа

1 Введение

В работе рассматривается задача поиска объекта на плоскости. Движение объекта удовлетворяет известным законам движения, возможно с неопределенностью. В частности, были рассмотрены следующие случаи: неподвижный объект; движение вдоль известной кривой с неизвестной, но ограниченной известными величинами, скоростью; движение на плоскости в соответствии с линейным дифференциальным уравнением, в которое входит неизвестная функция — помеха. Источником информации является «измерительное устройство», которое в дискретные моменты времени производит наблюдения. Наблюдение проводится следующим образом:

1. информационное множество покрывается прямоугольниками, возможно пересекающимися;
2. «измерительное устройство» с некоторой ошибкой определяет, которым из прямоугольников принадлежит объект;
3. «измерительное устройство» сообщает номера прямоугольников.

В работе исследованы условия, при которых моменты времени проведения наблюдений являются оптимальными. Получены оценки на размер информационного множества после наблюдений. Рассмотрена задача идентификации параметров системы по результатам наблюдений.

2 Постановка задачи

На плоскости рассматривается точка M, движение которой удовлетворяет некоторому закону. В начальный момент времени t_0 ее положение известно с точностью до известного множества $\mathcal{D}(t_0)$, которое является прямоугольником с **центром** $y(t_0)$ и **размером** $d(t_0)$, т.е.

$$\mathcal{D}(t_0) = [y_1(t_0) - d_1(t_0), y_1(t_0) + d_1(t_0)] \times \\ \times [y_2(t_0) - d_2(t_0), y_2(t_0) + d_2(t_0)] \,.$$

Далее под $\mathcal{D}(t)$ будем понимать минимальный прямоугольник, который гарантированно содержит точку M. $\mathcal{D}(t)$ будем также называть **рамкой**.

Доступен измерительный прибор, который действует следующим образом. Пусть дан некоторый момент времени θ. В этот момент в соответствии с размером $\mathcal{D}(\theta-)$ (будем обозначать $\mathcal{D}(\theta-) = \lim_{h\to-0}\mathcal{D}(\theta+h)$, $\mathcal{D}(\theta+) = \lim_{h\to+0}\mathcal{D}(\theta+h)$) выбираются четыре **измерительных прямоугольника** $Y_i^{est}(\theta)$ с центром $y_i^{est}(\theta)$, $i = 1, \ldots, 4$ и размером $s^{est}(\theta)$, которые покрывают $\mathcal{D}(\theta-)$. После этого **измерительный прибор** производит измерение

$$\mathcal{I}(\theta) = \left\{ i \,\middle|\, M \in Y_i^{est}(\theta) + \xi_i(\theta) \right\},$$

где $|\xi_{i,j}(\theta)| \leqslant \mu_j(\theta)$, $i = 1, \ldots, 4$, $j = 1, 2$. Кроме того считаем, что

$$\mu_j(\theta) = \beta d_j(\theta-),\ j = 1, 2, \tag{1}$$
$$s_j^{est}(\theta) = \alpha(\theta)\frac{d_j(\theta-)}{2},\ j = 1, 2 \quad 1 < \alpha(\theta) < 2.$$

Таким образом, мы выбираем не размер измерительных прямоугольников, а $\alpha(\theta)$. После этого на основании имеющейся информации происходит выбор нового прямоугольника $\mathcal{D}(\theta+)$. С помощью описанных процедур требуется установить положение точки M.

2.1 Выбор размера измерительных прямоугольников

Поскольку для функционирования измерительного устройства измерительные прямоугольники должны в объединении покрывать $\mathcal{D}(\theta-)$, на $\alpha(\theta)$ должны быть наложены ограничения. Необходимое условие этого можно записать в следующем виде:

$$\mathcal{D}(\theta-) \subset \bigcup_{i=1,\dots,4} \left\{Y_i^{est}(\theta) \dot{-} \Xi(\theta)\right\},$$

где «$\dot{-}$» — разность Минковского, $\Xi(\theta) = \{\xi \,||\xi_j(\theta)| \leqslant \mu_j(\theta),\, j = 1, 2\}$. Отсюда легко видеть, что на размер измерительных прямоугольников должно быть наложено условие

$$s_j^{est}(\theta) \geqslant \frac{d_j(\theta-)}{2} + \mu_j(\theta), \quad j = 1, 2. \tag{2}$$

Тогда можно выбрать центры измерительных прямоугольников так, чтобы они покрывали $\mathcal{D}(\theta-)$ при любых ошибках. Из (1) и (2) имеем, что $\alpha(\theta) \geqslant 1 + 2\beta$. А из ограничений на $\alpha(\theta)$ получаем ограничение на β: $\beta < \frac{1}{2}$.

2.2 Оценка размера рамки после измерения

После проведения измерения мы получаем множество индексов $\mathcal{I}(\theta)$, содержащее номера измерительных прямоугольников, в которые попала точка M. При этом ясно, что если $M \in Y_i^{est}(\theta) + \xi_i(\theta) = Z_i^{est}(\theta)$, то $M \in \hat{Z}_i^{est}(\theta)$, где

$$\hat{Z}_i^{est}(\theta) = Y_i^{est}(\theta) + \Xi(\theta).$$

Тогда выбираем $D(\theta+) = \bigcap_{i\in\mathcal{I}(\theta)} \hat{Z}_i^{est}(\theta)$. После этого берем $\mathcal{D}(\theta+) = \mathcal{D}(\theta-) \bigcap D(\theta+)$.

Выпишем размер прямоугольника:

$$2d_i(\theta+) \leqslant \min_{k\in\mathcal{I}(\theta)} \{y_{k,i}^{est}(\theta)\} - \max_{k\in\mathcal{I}}\{y_{k,i}^{est}(\theta)\} + 2\mu_i(\theta) + 2s_i^{est}(\theta).$$

Заметим, что ни одна величина от $\{\xi\}$ не зависит. Поэтому ошибки влияют только на состав множества $\mathcal{I}(\theta)$.

Для оценки множества $\mathcal{D}(\theta+)$ используем худший случай, а именно пересечение $D(\theta+)$ и $\mathcal{D}(\theta-)$ совпадает с $D(\theta+)$, а также $|\mathcal{I}(\theta)| = 1$. Таким образом, получаем:

$$d_i(\theta+) \leqslant s_i^{est}(\theta) + \mu_i(\theta) \leqslant d_i(\theta-)\left(\frac{\alpha(\theta)}{2} + \beta\right). \qquad (3)$$

Помимо этого проведем оценку снизу на размер $\mathcal{D}(\theta+)$. Для этого предположим, что реализовался наилучший случай, а именно $|\mathcal{I}(\theta)| = 4$, то есть точка M попала во все прямоугольники одновременно. Кроме того, предполагаем, что $s_i^{est}(\theta) \leqslant d_i(\theta-)$.

Тогда $\mathcal{D}(\theta+) = D(\theta+)$. В противном случае существовала бы точка вне $\mathcal{D}(\theta-)$, но внутри пересечения рамок. Сопоставляя два факта: объединение рамок содержит $\mathcal{D}(\tau_j-)$, а пересечение рамок не вложено в $\mathcal{D}(\theta-)$, приходим к противоречию с условием на размеры. Таким образом, достаточно оценить снизу пересечение $\bigcap Z_i^{est}(\theta)$. Так как все прямоугольники пересекаются, то выполняется соотношение

$$\max_k\{y_{k,i}^{est}(\theta)\} - \min_k\{y_{k,i}^{est}(\theta)\} \leqslant 2s_i^{est}(\theta), \quad i = 1, 2,$$

а тогда

$$2d_i(\theta+) = \min_k\{y_{k,i}^{est}(\theta)\} - \max_k\{y_{k,i}^{est}(\theta)\} + 2\mu_i(\theta) + 2s_i(\theta) \geqslant 2\mu_i(\theta).$$

Этот результат говорит о том, что при таком методе измерений за конечное число шагов невозможно установить точное положение точки. Причина этого кроется в том, что мы требуем, чтобы при любых возможных ошибках прямоугольники покрывали область $\mathcal{D}(\theta-)$. Это требование влечет то, что $\hat{Z}_i^{est}(\theta)$ всегда в пересечении имеют прямоугольник с отличной от нуля площадью.

2.3 Оценка размера рамки после N наблюдений

Предположим, что отсутствует динамика, то есть точка M неподвижна. Тогда безразлично в какой момент времени проводить наблюдения, так как между наблюдениями рамка не меняет размер. Поэтому вместо $\alpha(\tau_j)$, где τ_j — момент j-ого наблюдения, будем использовать α_j, где индекс указывает номер измерения. Аналогичные обозначения введем и для других параметров. Пусть проведено N измерений. Тогда из предыдущего пункта получаем следующую оценку:

$$\mu_{i,j} \leqslant d_i(\tau_j+) \leqslant d_i(\tau_{j-1}+)\left(\frac{\alpha_j}{2}+\beta\right).$$

Используя (1), получим после N наблюдений

$$(\beta)^N d_i(\tau_0+) \leqslant d_i(\tau_N+) \leqslant d_i(\tau_0+)\prod_{j=1}^{N}\left(\frac{\alpha_j}{2}+\beta\right).$$

Ранее было получено, что $2\beta < 1$. Значит левая часть стремится к нулю. Таким образом, получаем следующие оценки на размер $\mathcal{D}(\tau_N+)$, если положим $\alpha_j \equiv \alpha$:

$$d_i(\tau_N+) \leqslant d_i(t_0+)\min\left\{\left(\frac{\alpha}{2}+\beta\right)^N, 1\right\}.$$

Получаем, что если $\frac{\alpha}{2}+\beta < 1$, то оценка сверху сходится к нулю, при увеличении количества измерений. Для гарантированного результата d_i в случае $\alpha_j \equiv \alpha$ достаточно провести $N \geqslant \max_{i=1,2}\left\{\frac{\ln\frac{d_i}{d_i^0}}{\ln\left(\frac{\alpha}{2}+\beta\right)}\right\}$ измерений. При этом невозможно добиться размера d_i, проведя меньше чем $\max_{i=1,2}\left\{\frac{\ln\frac{d_i}{d_i^0}}{\ln 2\beta}\right\}$ измерений.

2.4 Альтернативный алгоритм измерений

Помимо указанного выше алгоритма функционирования прибора, можно указать еще один: центры измерительных прямоугольников выбираются измерительным прибором. В этом случае результатом измерения помимо множества $\mathcal{I}(\theta) = \left\{ i \,\middle|\, M \in Y_i^{est}(\theta) \right\}$ будут координаты измерительных прямоугольников $z_i^{est}(\theta) = y_i^{est}(\theta) + \xi_i^{est}(\theta)$, которые передаются с ошибкой.

В этом случае ошибки неравнозначны, в отличие от рассмотренного алгоритма. То есть существуют ошибки, при которых алгоритм наблюдений сходится за конечное время.

3 Движение вдоль кривой

Будем считать, что точка M может двигаться по гладкой кривой на плоскости, которая описывается уравнением

$$x = r_1(u), \quad y = r_2(u), \quad u \in \mathbb{R}.$$

Введем обозначение $r(u) = (r_1(u), r_2(u))$. Считаем, что $r(\cdot)$ — достаточно гладкое отображение. Положение точки M в этом случае однозначно задается параметром u. Тогда динамику точки M можно описывать в виде:

$$\dot{u} = s(t)f(t),$$

где $f(\cdot)$ неотрицательная непрерывно дифференцируемая функция, а $s(t) \in [-1, 1]$, $\forall t$ и измерима. Таким образом, мы получаем модель движения вдоль кривой, при котором в каждый момент времени известна максимальная скорость движения вдоль кривой, но неизвестно направление движения.

3.1 Описание эволюции

Введем **информационное множество** $\mathcal{U}(t)$, которое описывает множество возможных параметров u, совместимых с

текущими наблюдениями. В момент времени τ_0 начальное множество задано в виде $\mathcal{U}_0$ — начального множества параметров, которое является ограниченным с конечным числом компонент связности. Множество $\mathcal{D}(t)$ в этом случае вводится таким образом, чтобы сохранить алгоритм проведения наблюдений, то есть $\mathcal{D}(t)$ — минимальный прямоугольник содержащий множество $r\left(\mathcal{U}(t)\right)$. Для описания эволюции множества $\mathcal{U}(t)$ удобно в каждый момент времени представить его в виде

$$\mathcal{U}(t) = \bigcup_{\alpha \in V(t)} [u_1^\alpha(t), u_2^\alpha(t)], \tag{4}$$

где $|V(t)|$ — количество компонент связности в момент t. Тогда динамику каждого множества $[u_1^\alpha(t), u_2^\alpha(t)]$ легко записать в виде:

$$\dot{u}_1 = -f(t), \quad \dot{u}_2 = f(t). \tag{5}$$

3.1.1 Эволюция $\mathcal{D}(t)$

Чтобы описать эволюцию прямоугольника $\mathcal{D}(t)$, найдем его связь с множеством $\mathcal{U}(t)$ между моментами наблюдения, эволюция которого известна из (5).

$$y_i(t) = \frac{1}{2}\left(\max_{u \in \mathcal{U}(t)} r_i(u) + \min_{u \in \mathcal{U}(t)} r_i(u)\right), \quad i = 1, 2, \tag{6}$$

$$d_i(t) = \frac{1}{2}\left(\max_{u \in \mathcal{U}(t)} r_i(u) - \min_{u \in \mathcal{U}(t)} r_i(u)\right), \quad i = 1, 2.$$

Для того, чтобы найти дифференциальные уравнения на $y_i(t)$, $d_i(t)$, необходимо найти производную от выражения $\max_{u \in \mathcal{U}(t)} r_i(u)$ и $\min_{u \in \mathcal{U}(t)} r_i(u)$. Из выражения (4) и соотношения $\bigcup_{\alpha \in V(\theta)} [u_1^\alpha(t), u_2^\alpha(t)] = \bigcup_{\alpha \in V(t)} [u_1^\alpha(t), u_2^\alpha(t)]$, где $\theta < t$ и между θ и t не производится наблюдений, имеем

$$\max_{u \in \mathcal{U}(t)} r_i(u) = \max_{\substack{\alpha \in V(\tau_j) \\ u \in [u_1^\alpha(t), u_2^\alpha(t)]}} r_i(u) = \max_{\substack{\alpha \in V(\tau_j) \\ \gamma \in [0,1]}} r_i(u_1^\alpha(t)(1-\gamma) + u_2^\alpha(t)\gamma).$$

Потребуем, чтобы $V(\tau_j)$ состояло из конечного числа элементов. В этом случае легко проверить, что $r_i(u_1^\alpha(t)(1-\gamma)+u_2^\alpha(t)\gamma)$ непрерывна вместе со своей производной по t на метрическом компакте $V(\tau_j)\times[0,1]$. Значит справедлива теорема о дифференцировании максимума (в ней утверждается о производной по направлению, но так как результат будет использоваться как подынтегральное выражение, будем там, где максимизатор не единственный, понимать производную справа, а также будем предполагать, что таких точек конечное число).

$$\frac{d}{dt}\max_{u\in\mathcal{U}(t)} r_i(u) = \max_{\{\alpha^*,\gamma^*\}\in N(t)} \Big[r_i'(u_1^{\alpha^*}(t)(1-\gamma^*)+u_2^{\alpha^*}(t)\gamma^*)\times \\ \times(2\gamma^*-1)f(t)\Big], \quad (7)$$

где

$$N(t) = \operatorname*{Argmax}_{\substack{\alpha\in V(\tau_j)\\ \gamma\in[0,1]}} r_i(u_1^\alpha(t)(1-\gamma)+u_2^\alpha(t)\gamma).$$

В точках, в которых $|N(t)|>1$, в указанном равенстве нужно перейти к производным по направлению.

Рассмотрим $\{\alpha^*,\gamma^*\}\in N(t)$. Возможны несколько случаев:

1. $\gamma\in(0,1)$. Тогда выполняются необходимые условия экстремума по γ. Таким образом, $r_i'(u_1^{\alpha^*}(t)(1-\gamma^*)+u_2^{\alpha^*}(t)\gamma^*)=0$. Подставив в выражение под максимумом в (7), получим, что производная не меньше нуля.
2. $\gamma=0$. Тогда выполнено:

$$\frac{d}{d\gamma} r_i(u_1^{\alpha^*}(t)(1-\gamma^*)+u_2^{\alpha^*}(t)\gamma^*)\Big|_{\gamma^*=0} < 0$$

Значит $r_i'(u_1^{\alpha^*}(t))<0$. Подставив в выражение под максимумом в (7), получим следующее неравенство $-r_i'(u_1^{\alpha^*}(t))f(t)\geqslant 0$.

3. $\gamma = 1$. Аналогично получаем: $r_i'(u_2^{\alpha^*}(t)) > 0$. Подставив в выражение под максимумом в (7), получим $r_i'(u_2^{\alpha^*}(t))f(t) \geqslant 0$.

Окончательно имеем, что нужно проверить лишь граничные точки. Если они являются максимизаторами, то взять максимальное значение производной среди граничных точек. Если ни одна граничная точка не доставляет максимум, то нужно брать 0. Учитывая выражение (6), приходим к результату:

$$\frac{d}{dt}\max_{u\in\mathcal{U}(t)} r_i(u) = \max\Big\{\max_{\alpha\in V, j=1,2}\big[(-1)^j r_i'(u_j^\alpha(t))f(t)\big| \\ \big|r_i(u_j^\alpha(t)) = y(t) + d_i(t)\big], 0\Big\}.$$

Аналогичный результат получаем при дифференцировании минимума. Введем обозначение $b_i^*(t) = \max_{u\in\mathcal{U}(t)} r_i(u)$ и $b_{*i}(t) = \min_{u\in\mathcal{U}(t)} r_i(u)$. Дифференциальным уравнениям на положение и размер множества $\mathcal{D}(t)$ тогда легко записываются через выписанные производные b^* и b_*.

Оценим изменение размера $d_i(t) = \frac{1}{2}(b_i^*(t) - b_{*i}(t))$ при эволюции множества. Рассмотрим выражение для $b^*(t)$:

$$b_i^*(t) \leqslant b_i^*(t_0) + \int\limits_{t_0}^{t} f(\tau)\max_{\alpha\in V, j=1,2}|r_i'(u_j^\alpha(\tau))|\,d\tau \leqslant \\ \leqslant b_i^*(t_0) + \int\limits_{t_0}^{t}\max_{u\in\mathcal{U}(\tau)}|r_i'(u)|f(\tau)\,d\tau.$$

При этом, данная оценка достижима. Например, если $r_i' \equiv \mathrm{const}$. Действуя аналогично, получим оценку на b_*.

3.2 Проведение наблюдений

Рассмотрим как меняется множества $\mathcal{U}(t)$, $\mathcal{D}(t)$ при проведении измерений. Наблюдения проводятся согласно

описанному выше алгоритму. Однако, $\mathcal{D}(\theta+)$ выбирается несколько иначе. Получив $D(\theta+)$, устанавливается множество $U(\theta+) = \{u : r(u) \in D(\theta+)\}$. После этого выбираем $\mathcal{U}(\theta+) = \mathcal{U}(\theta-) \bigcap U(\theta+)$. Исходя из $\mathcal{U}(\theta+)$ выбирается соответствующая рамка $\mathcal{D}(\theta+)$. Поэтому справедлива оценка (3). В итоге получаем

$$d_i(\tau_{j+1}+) \leqslant \left(\frac{\alpha_{j+1}}{2} + \beta\right) \left[d_i(\tau_j+) + \int\limits_{\tau_j}^{\tau_{j+1}} \max_{u \in \mathcal{U}(t)} |r_i'(u)| f(t)\, dt \right]. \tag{8}$$

3.3 Наилучшее размещение моментов наблюдения

Сперва приведем полезную в дальнейшем оценку. Пусть в момент ϑ производится наблюдение. На отрезке $[\vartheta, t]$ нет отличных от ϑ моментов, когда проводится наблюдение. Оценим $\max_{u \in \mathcal{U}(t)} |r_i'(u)|$ сверху:

$$\max_{u \in \mathcal{U}(t)} |r_i'(u)| \leqslant \max_{u \in \mathcal{U}(\vartheta+)} |r_i'(u)| + \int\limits_{\vartheta}^{t} f(\tau)\, d\tau \max_{u \in \mathcal{U}(t)} |r''(u)|. \tag{9}$$

Кроме того, можно оценить $\mathcal{U}(t)$ через $\mathcal{U}(s)$, $s < t$. Введем множество

$$\hat{\mathcal{U}}(s,t) = \bigcup_{\gamma \in [-1,1]} \left\{ \gamma \int\limits_{s}^{t} f(\tau)\, d\tau + \mathcal{U}(s) \right\} = [-1,1] \int\limits_{s}^{t} f(\tau)\, d\tau + \mathcal{U}(s).$$

Для $\hat{\mathcal{U}}(\cdot, t)$ при $s_1 < s_2$ выполняется включение $\hat{\mathcal{U}}(t, s_1) \subset \hat{\mathcal{U}}(t, s_2)$. Аналогичное включение выполняется и по другой паре аргументов. В момент измерения выполнены следующие включения

$$\mathcal{U}(\vartheta+) \subset \mathcal{U}(\vartheta-) \Rightarrow \hat{\mathcal{U}}(\vartheta+, s) \subset \hat{\mathcal{U}}(\vartheta-, s).$$

Рассмотрим отрезок времени $[t_0, t_1]$. Пусть доступно N измерений в моменты $\tau_1, \ldots, \tau_N$. Обозначим $\tau_0 = t_0$, $\tau_{N+1} = t_1$. Нетрудно проверить, что

$$\max_{u \in \mathcal{U}(t)} |r_i'(u)| \leqslant \max_{u \in \mathcal{U}(t_0)} |r_i'(u)| + \int\limits_{t_0}^{t} f(\tau)\, d\tau \max_{u \in \hat{\mathcal{U}}(t_0, t_1)} |r''(u)|.$$

Тогда, используя обозначения $A = \max_{u \in \mathcal{U}(t_0)} |r_i'(u)|$, $B = \max_{u \in \hat{\mathcal{U}}(t_0, t_1)} |r''(u)|$ и $\tau_{N+1} = t_1$, получим

$$\begin{aligned} d_i(t_1) \leqslant d_i(t_0) \prod_{k=1}^{N} \left(\frac{\alpha_k}{2} + \beta\right) + \\ + A \left[\sum_{k=1}^{N+1} \int\limits_{\tau_{k-1}}^{\tau_k} f(\tau)\, d\tau \prod_{j=k}^{N} \left(\frac{\alpha_j}{2} + \beta\right)\right] + \\ + B \left[\sum_{k=1}^{N+1} \int\limits_{\tau_{k-1}}^{\tau_k} \int\limits_{t_0}^{t} f(\tau)\, d\tau f(t)\, dt \prod_{j=k}^{N} \left(\frac{\alpha_j}{2} + \beta\right)\right] = E_i^N. \end{aligned}$$

Необходимым условием экстремума является $\frac{\partial}{\partial \tau_k} E_i^N = 0$, $\forall k = 1, \ldots, N$:

$$\begin{aligned} \frac{\partial}{\partial \tau_k} E_i^N = f(\tau_k) \prod_{j=k+1}^{N} \left(\frac{\alpha_j}{2} + \beta\right) \left(\frac{\alpha_k}{2} + \beta - 1\right) \times \\ \times \left\{ A + B \int\limits_{t_0}^{\tau_k} f(t)\, dt \right\}. \end{aligned}$$

Если считать, что $f(t) > 0$ при всех t, то становится ясно, что минимум достигается, когда все наблюдения проводятся в момент времени t_1.

Это происходит в силу того, что мы не можем оценить как меняется при наблюдении множество $\mathcal{U}$. Мы можем утверждать лишь то, как изменится размер образа при отображении $r(\cdot)$. При этом объединение всех возможных состояний (то есть множеств $\mathcal{U}(\theta+)$, которые могли получится из $\mathcal{U}(\theta-)$ в результате наблюдения) после наблюдение будет составлять множество $\mathcal{U}(\theta-)$.

Пример 1. *Кривая задана уравнением*

$$r(t) = t\begin{pmatrix}\cos(t)\\ \sin(t)\end{pmatrix} + 0{,}1t\begin{pmatrix}\cos(t) & \sin(t)\\ -\sin(t) & \cos(t)\end{pmatrix}\begin{pmatrix}\cos(5t)\\ \sin(5t)\end{pmatrix}.$$

Функция $f(\cdot)$ имеет вид

$$f(t) = 2 + \cos(t).$$

Исследование происходит при $t_0 = 0$, $t_1 = 0{,}4$. При этом $\beta = 0{,}2$, $\alpha = 1{,}05(1 + 2\beta) = 1{,}47$. Тогда $\frac{\alpha}{2} + \beta < 1$. На рисунке 1-2 наблюдение происходит в момент $\theta = 0{,}01$. На рисунке 3-4 наблюдение происходит в момент $\theta = 0{,}39$. На изображениях серым пунктиром обозначено множество $r(\mathcal{U}(s_0+))$, черной сплошной линией обозначено множество увеличение множества $r(\mathcal{U}(s_1-))$ по сравнению с $r(\mathcal{U}(s_0+))$, где $[s_0, s_1]$ — отрезок, на котором происходит эволюция. Черной толстой линией обозначено движение точки M. Также на изображении приведены соответствующие прямоугольники $\mathcal{D}(s_0+)$ и $\mathcal{D}(s_1-)$. Кроме того серыми точками изображена вся кривая, на которой происходит исследование.

4 Движение на плоскости

Будем считать, что положение точки M описывается дифференциальным уравнением

$$\dot{x} = C(t)f(t),$$

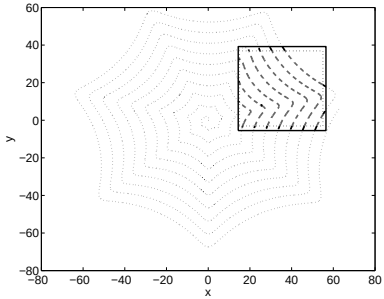


Рис. 1. Эволюция на $[0,0{,}01]$.

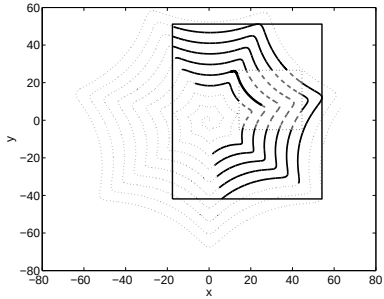


Рис. 2. Эволюция на $[0{,}01, 0{,}4]$.

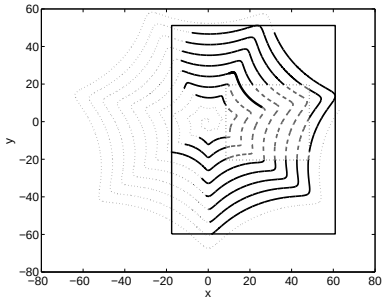


Рис. 3. Эволюция на $[0, 0{,}39]$.

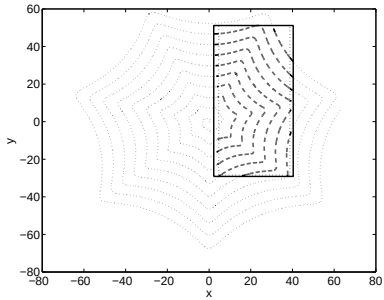


Рис. 4. Эволюция на $[0{,}39, 0{,}4]$.

где $f(t) \in Q(t)$ при всех t и измерима, а $Q(t)$ — выпуклый компакт при всех t. При этом $Q(\cdot)$ — непрерывное многозначное отображение, а $C(\cdot)\colon \mathbb{R} \to \mathbb{R}^{2\times 2}$ — непрерывно.

4.1 Описание эволюции

Введем **информационное множество** $\mathcal{R}(t)$, которое описывает множество возможных положений точки M в момент t, совместимых с полученными наблюдениями. Тогда $\mathcal{D}(t)$ — наименьший прямоугольник, содержащий $\mathcal{R}(t)$. Сначала опишем эволюцию множества $\mathcal{R}(t)$.

Пусть на $[\tau_k, t]$ не производится никаких наблюдений. Тогда $\mathcal{R}(t) = \mathcal{R}(\tau_k+) + \int_{\tau_k}^{t} C(\tau)Q(\tau)\,d\tau$. Если $\mathcal{R}(\tau_k+)$ выпуклый компакт, то и $\mathcal{R}(t)$ выпуклый компакт. Отсюда нетрудно записать эволюцию опорной функции множества $\mathcal{R}(t)$. Найдем связь $\mathcal{D}(t)$ и $\mathcal{R}(t)$:

$$d_i(t) = \frac{1}{2}\left[\max_{x\in\mathcal{R}(t)}\langle x, e_i\rangle + \max_{x\in\mathcal{R}(t)}\langle x, -e_i\rangle\right], \quad i = 1, 2,$$
$$y_i(t) = \frac{1}{2}\left[\max_{x\in\mathcal{R}(t)}\langle x, e_i\rangle - \max_{x\in\mathcal{R}(t)}\langle x, -e_i\rangle\right], \quad i = 1, 2.$$

Введем обозначение $\sigma(\ell|Q) = \frac{1}{2}\left(\rho(\ell|Q) + \rho(-\ell|Q)\right)$. Приходим к уравнениям на $d_i(t)$ и $y_i(t)$:

$$\begin{cases} \dot{d}_i = \sigma(e_i|C(t)Q(t)), \\ \dot{y}_i = \frac{1}{2}\left[\rho\left(e_i\,|C(t)Q(t)\,\right) - \rho\left(-e_i\,|C(t)Q(t)\,\right)\right], \\ d_i(\tau_k+) = \sigma\left(e_i\,|\mathcal{R}(\tau_k+)\,\right), \\ y_i(\tau_k+) = \frac{1}{2}\left[\rho\left(e_i\,|\mathcal{R}(\tau_k+)\,\right) - \rho\left(-e_i\,|\mathcal{R}(\tau_k+)\,\right)\right]. \end{cases}$$

4.2 Проведение наблюдений

Наблюдения проводятся как и ранее, то есть $\mathcal{R}(\tau_k+) = \mathcal{R}(\tau_k-)\bigcap D(\tau_k+)$, где $D(\tau_k+)$ — пересечение измерительных прямоугольников. $\mathcal{D}(\tau_k+)$ выбирается в соответствии с (10). При этом, вне зависимости от метода справедлива оценка

$$d_i(\tau_{j+1}+) \leqslant \left(\frac{\alpha_{j+1}}{2} + \beta\right)\left[d_i(\tau_j+) + \int\limits_{\tau_j}^{\tau_{j+1}} g_i(t)\,dt\right], \quad i = 1, 2,$$

где $g_i(t) = \sigma\left(e_i\,|C(t)Q(t)\,\right)$.

4.3 Наилучшее размещение моментов наблюдения

Рассмотрим отрезок времени $[t_0, t_1]$. Пусть доступно N измерений в моменты $\tau_1, \ldots, \tau_N$. Обозначим $\tau_0 = t_0$, $\tau_{N+1} = t_1$.

$$d_i(t_1) \leqslant \prod_{j=1}^{N} \left(\frac{\alpha_j}{2} + \beta\right) d_i(t_0) + \sum_{k=1}^{N} \int\limits_{\tau_{k-1}}^{\tau_k} g_i(\tau)\, d\tau \prod_{j=k}^{N} \left(\frac{\alpha_j}{2} + \beta\right) + \\ + \int\limits_{\tau_N}^{t_1} g_i(\tau)\, d\tau = E_i^N, \quad i = 1, 2. \quad (10)$$

В предположении $\frac{\alpha_j}{2} + \beta \leqslant 1$, нетрудно проверить, что $E_i^N(\tau_1, \ldots, \tau_N) \geqslant E_i^N(t_1, \ldots, t_1)$. Таким образом, оптимальным является размещение наблюдений в последний момент.

4.4 Обеспечение заданных ограничений на размер $\mathcal{D}(t)$

В некоторых приложениях такое поведение может оказаться неприемлемым, потому что положение точки M уточняется только в последний момент, а до этого информационное множество только увеличивается.

Чтобы избавиться от такого поведения, можно наложить ограничение на минимальное время между наблюдениями или ввести функцию, которая ограничивает сверху оценку. Рассмотрим второй вариант, в предположении $g_i(t) > 0$. Будем проводить исследование раздельно для каждого $i = 1, 2$.

Пусть заданы гладкие строго убывающие функции $\nu_i(\cdot)\colon [t_0, t_1] \to \mathbb{R}_+$. Требуется обеспечить $d_i(t) \leqslant \nu_i(t)$ на $[t_0, t_1]$, подходящим выбором моментов наблюдения. Тогда задачу можно охарактеризовать следующим образом: минимизировать N, при условии, что $E_i^N(t) \leqslant \nu_i(t)$ на $[t_0, t_1]$. Пусть $\tilde{E}_i(N)$ — оценка $d_i(t_0)$. Фиксируем N и будем максимизировать

$\tilde{E}_i$ по $\{\tau_1, \ldots, \tau_N\}$, при условии $E_i^N(t) \leqslant \nu_i(t)$. Тогда минимальное N, при котором $\tilde{E}_i(N) \geqslant d_i(t_0)$ и будет решением. При этом ясно, что если $E_i^N(\tau_k-) \leqslant \nu_i(\tau_k-)$, то и на всем $[t_0, t_1]$ неравенство будет выполнено. Получаем задачу

$$\tilde{E}_i(N) \to \max_{\{\tau_1,\ldots,\tau_N\}},$$

$$\psi_{1,i}(\tau_1, \ldots, \tau_N) = \tilde{E}_i(N) + \int\limits_{t_0}^{\tau_1} g_i(t)\,dt - \nu_i(\tau_1) \leqslant 0,$$

$$\psi_{k,i}(\tau_k, \ldots, \tau_N) = \left(\frac{\alpha_k}{2} + \beta\right)[\psi_{k-1,i}(\tau_{k-1}, \ldots, \tau_N) + \nu_i(\tau_{k-1})] + \\ + \int\limits_{\tau_{k-1}}^{\tau_k} g_i(t)\,dt - \nu_i(\tau_k) \leqslant 0, \quad k \geqslant 2,$$

$$\tilde{E}_i(N) = \prod_{j=1}^{N}\left(\frac{\alpha_j}{2} + \beta\right)^{-1} E_i^N(t_1) - \\ - \sum_{k=1}^{N} \int\limits_{\tau_k}^{\tau_{k+1}} g_i(\tau)\,d\tau \prod_{j=1}^{k}\left(\frac{\alpha_j}{2} + \beta\right)^{-1} - \int\limits_{t_0}^{\tau_1} g_i(\tau)\,d\tau.$$

Тогда, выписав функцию Лагранжа $L = -\tilde{E}_i(N) + \sum_{j=1}^{N} \lambda_{j,i}\psi_{j,i}$ и приравняв $\frac{\partial L}{\partial \lambda_{k,i}} = 0$, получим следующие выражения на λ_k:

$$\lambda_{k,i} = g(\tau_k)\left[1 - \left(\frac{\alpha_k}{2} + \beta\right)^{-1}\right] \times \\ \times \left\{-\prod_{j=1}^{k-1}\left(\frac{\alpha_j}{2} + \beta\right)^{-1} + \sum_{s=1}^{k-1} \lambda_{s,i} \prod_{j=s}^{k-1}\left(\frac{\alpha_j}{2} + \beta\right)^{-1}\right\} \times \\ \times \left[g(\tau_k)\left(\frac{\alpha_k}{2} + \beta\right)^{-1} - \nu_i'(\tau_k)\right]^{-1}$$

Пусть $\alpha_j \equiv \alpha$. И обозначим $b = \frac{\alpha}{2} + \beta < 1$. Можно проверить, что

$$\lambda_{k,i} = \frac{1-b}{\prod\limits_{j=1}^{k}\left(1 - b\frac{\nu_i'(\tau_j)}{g_i(\tau_j)}\right)} \prod_{j=1}^{k-1}\left(1 - \frac{\nu_i'(\tau_j)}{g_i(\tau_j)}\right)$$

является решением. Ввиду того, что $\nu_i'(t) < 0$, $g_i(t) > 0$, получаем, что $\lambda_{k,i} > 0$. Из условий дополняющей нежесткости получаем, что $\psi_{k,i} = 0$. Значит оптимальным выбором τ_j являются моменты, когда оценка размера достигает кривой $\nu_i(t)$. Поэтому разумно использовать следующий алгоритм.

Пусть на отрезке $[t_0, t_1]$ заданы гладкие строго убывающие функции $\nu_i(t)$. Так как в результате наблюдений нам известны $d_i(t)$, то следующее наблюдение будем проводить в момент времени θ, которое является минимальным из θ_i, которое является решением следующего уравнения: $d_i(t) + \int_t^{\theta_i} g_i(\tau)\,d\tau = \nu_i(\theta_i)$, $\theta = \min_{i=1,2}\theta_i$.

Оценим минимально возможный интервал между наблюдениями:

$$b\nu_i(t) + \int\limits_{t}^{t+\Delta_i(t)} g_i(\tau)\,d\tau = \nu_i(t + \Delta_i(t)) \Rightarrow$$

$$\Rightarrow \Delta_i(t) \max_{\tau\in[t,t_1]}\left[g_i(\tau) - \nu_i'(\tau)\right] \geqslant (1-b)\nu_i(t_1).$$

При этом $\Delta(t) = \min_{i=1,2}\Delta_i(t)$, так как случай, когда $d_i(t) < \nu_i(t)$ можно оценить через $d_i(t) = \nu_i(t)$. То есть ситуация, когда интервалы между наблюдениями «вызванными» i-ой компонентой чередуются с наблюдениями «вызванными» другой компонентой и время между наблюдениями будет меньше минимума из оценок Δ_i, не возникает. Отсюда же можно получить и верхнюю оценку на N — число наблюдений: $N \leqslant \frac{t_1 - t_0}{\Delta(t_0)}$.

Пример 2. *Множество $Q(t)$ задано в виде вращающегося эллипсоида:*

$$Q(t) = \{x \,|\langle T(t)x, QT(t)x\rangle \leqslant 1\}\,,$$

где $T(t)$ — матрица поворота на угол $5 \cdot t$, $Q = \mathrm{diag}\{100, 1\}$. Матрица $C(t) \equiv E$. Таким образом, $\rho\,(\ell\,|C(t)Q(t)\,) = \sqrt{\langle T(t)\ell, Q^{-1}T(t)\ell\rangle}$. Начальное множество — треугольник с вершинами в точках $(-1,1)^T$, $(-1,-1)^T$, $(1,-1)^T$. $\nu_i(t) = 2\,(1+t)^{-\frac{1}{2}}$. Исследование проводится при $t_0 = 0$, $t_1 = 4$. При этом $\beta = 0{,}2$, $\alpha = 1{,}05(1+2\beta) = 1{,}47$. На рисунке 5 изображена эволюция размера рамки.

Пример 3. *Все параметры такие же как и в предыдущем примере, за исключением функции $\nu_i(t) = 2\,(1+t)^{-\frac{5}{4}}$. На рисунке 6 изображена эволюция размера рамки.*

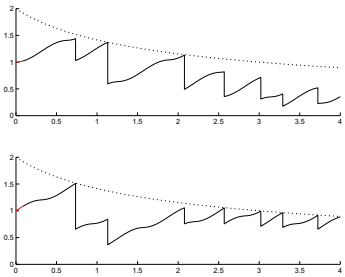


Рис. 5. Эволюция размера до момента $t = 4$. $\nu_i(t) = 2\,(1+t)^{-\frac{1}{2}}$.

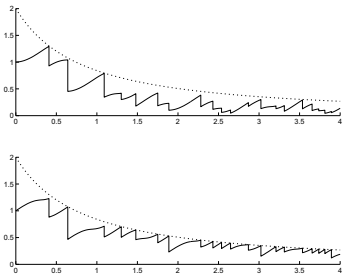


Рис. 6. Эволюция размера до момента $t = 4$. $\nu_i(t) = 2\,(1+t)^{-\frac{5}{4}}$.

4.5 Линейная динамика на плоскости

Теперь предположим, что уравнение на положение точки M имеет вид

$$\dot{x} = A(t)x + C(t)f(t),$$

где $A(t)$ — непрерывная матрица. Условия на матрицу C и множество Q сохранены. Информационное множество в этом случае удовлетворяет $\mathcal{R}(t) = X(t,t_0)\mathcal{R}(t_0) + \int_{t_0}^{t} X(t,\tau)C(\tau)Q(\tau)\,d\tau$. Рассмотрим как это повлияет на оптимальные моменты проведения наблюдений.

Пусть на $[t_0, t_1]$ доступно одно наблюдение. Найдем оценки на размер прямоугольника $\mathcal{D}(t_1)$.

$$d_i(t_1) = \sigma(e_i|\mathcal{R}(t_1)) = \sigma(X^T(t_1,\theta)e_i|\mathcal{R}(\theta+)) + \\ + \int\limits_{\theta}^{t_1} \sigma(X^T(t_1,\tau)e_i|C(\tau)Q(\tau))\,d\tau.$$

Из того, что $\mathcal{R}(\theta+) \subset D(\theta)$ можно получить:

$$d_i(t_1) \leqslant \int\limits_{\theta}^{t_1} \sigma(X^T(t_1,\tau)e_i|C(\tau)Q(\tau))\,d\tau + \\ + \left(\frac{\alpha}{2} + \beta\right) \sum_{j=1}^{2} |\langle X^T(t_1,\theta)e_i, e_j\rangle| \left[\sigma\left(X^T(\theta,t_0)e_j\,|\mathcal{R}(t_0)\right) + \right. \\ \left. + \int\limits_{t_0}^{\theta} \sigma\left(X^T(\theta,\tau)e_j\,|C(\tau)Q(\tau)\right)\,d\tau\right]. \quad (11)$$

Рассмотрим два частных случая:

1. $\theta = t_0$. Тогда (11) примет вид

$$d_i(t_1) \leqslant \left(\frac{\alpha}{2} + \beta\right) \sum_{j=1}^{2} |\langle X^T(t_1,t_0)e_i, e_j\rangle| \sigma\left(e_j\,|\mathcal{R}(t_0)\right) + \\ + \int\limits_{t_0}^{t_1} \sigma(X^T(t_1,\tau)e_i|C(\tau)Q(\tau))\,d\tau.$$

2. $\theta = t_1$. Тогда

$$d_i(t_1) \leqslant \left(\frac{\alpha}{2} + \beta\right) \left[\sigma\left(X^T(t_1, t_0)e_i \,|\mathcal{R}(t_0)\right) + \right.$$
$$\left. + \int\limits_{t_0}^{t_1} \sigma\left(X^T(\theta, \tau)e_i \,|C(\tau)Q(\tau)\right)\, d\tau \right].$$

Учитывая, что $\sigma(\ell|A) = \sigma(-\ell|A)$, положительно однородна и $\sigma(\ell_1 + \ell_2|A) \leqslant \sigma(\ell_1|A) + \sigma(\ell_2|A)$ имеем

$$\sigma\left(X^T(t_1, t_0)e_i \,|\mathcal{R}(t_0)\right) \leqslant \sum_{j=1}^{2} |\langle X^T(t_1, t_0)e_i, e_j\rangle| \sigma\left(e_j \,|\mathcal{R}(t_0)\right).$$

Таким образом, при любой матрице $A(t)$ оценка на размер прямоугольника, в случае проведения наблюдения в конце, всегда лучше, чем оценка на размер прямоугольника, в случае проведения наблюдения в начале. В частном случае матрицы $A(t) = h(t)E$, где $h(\cdot)$ — непрерывная функция, можно показать, что измерение всегда следует производить в момент t_1.

4.6 Задача идентификации

Пусть уравнение движения точки M имеет следующий вид

$$\dot{x} = A(t)x + C(t)(f + w(t)),$$

где $A(t)$, $C(t)$ — непрерывные матрицы, а $C(t)$ невырожденная. f — фиксированный элемент компакта Q_0. $w(\cdot)$ — неизвестная функция принимающая значения в эллипсоиде $\mathcal{E}(0, W)$. Путем проведения наблюдений требуется установить значение f.

Предположим, что на момент времени τ_k доступна оценка Q_k. Тогда получим соотношения для момента времени τ_{k+1}, где τ_i — моменты проведения наблюдений. Ясно, что $f \in Q_{k+1}$,

если

$$f \in Q_k, \exists x_0 \in \mathcal{R}(\tau_k+), w(\cdot):$$
$$X(\tau_{k+1}, \tau_k)x_0 + \int\limits_{\tau_k}^{\tau_{k+1}} X(\tau_{k+1}, t)C(t)(f + w(t))\,dt \in \mathcal{R}(\tau_{k+1}+).$$

Обозначим $U_k = \int_{\tau_k}^{\tau_{k+1}} X(\tau_{k+1}, t)C(t)\,dt$. В силу предположений о невырожденности матрицы $C(t)$, имеется обратная матрица U_k (если предполагать, что $C(t)$ — матрица полного ранга, то можно утверждать о существовании левой обратной матрицы, чего в целом достаточно для дальнейших рассмотрений). Тогда имеем оценку $\hat{Q}_{k+1}$:

$$\hat{Q}_{k+1} = Q_k \bigcap \left\{\mathcal{E}(0, W) + U_k^{-1}\left[\mathcal{R}(\tau_{k+1}+) - X(\tau_{k+1}, \tau_k)\mathcal{R}(\tau_k+)\right]\right\}. \tag{12}$$

Оценкой $\hat{Q}_{k+1}$ не пользуемся, так как на практике получить ее тяжело. Можно оценить $\hat{Q}_{k+1}$ с помощью Q_{k+1}, которую получаем подстановкой вместо $\mathcal{R}(\cdot)$ множеств $\mathcal{D}(\cdot)$. При этом $\hat{Q}_{k+1} \subset Q_{k+1}$. Нетрудно проверить, что сумма, линейные преобразования, пересечения выпуклых политопов остаются выпуклыми политопами. Поэтому, если $W \equiv 0$, оценку Q_k можно выбирать в классе выпуклых политопов. Если же $W \neq 0$, то сумму политопа и эллипсоида можно оценить политопом, у которого грани смещены следующим образом

$$\tilde{b} = b + \langle n, Wn\rangle^{\frac{1}{2}},$$

где $\left(n, \tilde{b}\right)$ — новая пара, определяющая гиперплоскость. Однако, дальнейшие оценки будут проводится именно для оценки из (12).

4.6.1 Оценка Q_k и $d_i(\tau_k)$

Рассмотрим как эволюционируют Q_k и $d_i(\tau_k)$. Считаем $\alpha_j \equiv \alpha$. Начнем с $d_i(\tau_k)$. Из (11) имеем

$$d_i(\tau_{k+1}) \leqslant \left(\frac{\alpha}{2}+\beta\right)\left[\sum_{j=1}^{2}|\langle X(\tau_{k+1},\tau_k)e_j,e_i\rangle|d_j(\tau_k)+ \right.$$
$$\left. +\|U_k\|(r_k+\sqrt{\|W\|})\right],$$

где $r_k = \frac{1}{2}\mathrm{diam}Q_k = \max_{e:\ \|e\|=1}\sigma(e|Q_k)$. Введем переменную $s_k = d_1(\tau_k)+d_2(\tau_k)$. Тогда будем иметь

$$s_{k+1} \leqslant \left(\frac{\alpha}{2}+\beta\right)\left[\sqrt{2}\|X(\tau_{k+1},\tau_k)\|s_k + 2\|U_k\|(r_k+\sqrt{\|W\|})\right]. \tag{13}$$

Найдем теперь оценку для r_k. Для этого заметим, что

$$U_kQ_{k+1} \subset U_k\mathcal{E}(0,W) + [\mathcal{D}(\tau_{k+1}+) - X(\tau_{k+1},\tau_k)\mathcal{D}(\tau_k+)].$$

Тогда

$$\sigma(\ell|U_kQ_{k+1}) \leqslant \left[\sqrt{2}\left(\frac{\alpha}{2}+\beta\right)+1\right](\|X(\tau_{k+1},\tau_k)\|\|\ell\|s_k+$$
$$+\sqrt{\|W\|}\|U_k\|\|\ell\|\Big) + \sqrt{2}\left(\frac{\alpha}{2}+\beta\right)\|U_k\|\|\ell\|r_k. \tag{14}$$

Выразим $\sigma(e|Q_{k+1})$ через $\sigma({U_k^{-1}}^T e|U_kQ_{k+1})$, обозначим $\kappa(U_k) = \|U_k\|\|U_k^{-1}\|$ — число обусловленности матрицы U_k и выразим r_{k+1}. Окончательно будем иметь

$$r_{k+1} \leqslant \left[\sqrt{2}\left(\frac{\alpha}{2}+\beta\right)+1\right](\|X(\tau_{k+1},\tau_k)\|\|U_k^{-1}\|s_k+$$
$$+\sqrt{\|W\|}\kappa(U_k)\Big) + \sqrt{2}\left(\frac{\alpha}{2}+\beta\right)\kappa(U_k)r_k. \tag{15}$$

Введем $\hat{r}_k$, $\hat{s}_k$, которые при всех k оценивают сверху s_k, r_k. Учитывая (13) и (15), окончательно получаем систему (обозначим $b = \frac{\alpha}{2}+\beta$)

$$\hat{s}_{k+1} = \sqrt{2}b\|X(\tau_{k+1},\tau_k)\|\hat{s}_k + 2b\|U_k\|\hat{r}_k + 2b\|U_k\|\sqrt{\|W\|},$$

$$\hat{r}_{k+1} = \min\left\{\hat{r}_k, \left(\sqrt{2}b+1\right)\left[\|X(\tau_{k+1},\tau_k)\|\|U_k^{-1}\|\hat{s}_k + \right.\right.$$
$$\left.\left. + \sqrt{\|W\|}\kappa(U_k)\right] + \sqrt{2}b\kappa(U_k)\hat{r}_k\right\}. \quad (16)$$

Тогда Q_k содержится в некотором шаре радиуса $\hat{r}_k$ при всех k, а $\mathcal{R}(\tau_k+)$ содержится в квадрате со стороной $\hat{s}_k$ при всех k. Можно доказать следующее утверждение.

Утверждение 1. *Пусть $A(t) \equiv A$, существуют C^* и C_* такие, что $\|C(t)\| \leqslant C^*$ и $\|C^{-1}(t)\| \leqslant C_*$ при всех t. Если $\left(\frac{\alpha}{2}+\beta\right) = b < (\sqrt{2}C^*C_*)^{-1}$, то для любого $\varepsilon > 0$ существуют такие $\delta > 0$, $\Delta > 0$, $N \in \mathbb{N}$, что*

$$\hat{r}_n \leqslant \frac{(\sqrt{2}b+1)C^*C_*\sqrt{\|W\|}}{1-\sqrt{2}bC^*C_*} + \varepsilon, \quad \forall n \geqslant N.$$

4.7 Примеры

Пример 4. *Пусть*

$$A(t) \equiv \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad C(t) = \begin{pmatrix} \cos(1{,}25t) & \sin(1{,}25t) \\ -\sin(1{,}25t) & \cos(1{,}25t) \end{pmatrix}.$$

Q_0 — прямоугольник размером $(30, 30)$ с центром $(-8, 0)$. Начальное множество $\mathcal{R}_0$ — четырехугольник с вершинами в $(10, 10)^T$, $(-10, 10)^T$, $(10, -10)^T$, $(-20, -10)^T$, $f = (-4, 6)^T$, $\beta = 0{,}01$, $\alpha = 1{,}07$, W — нулевая матрица. Результаты представлены на рисунке 7. В этом примере и последующих сплошной линией обозначен центр наименьшего прямоугольника, который содержит Q_k, пунктиром обозначено изменение размером этого прямоугольника, а точечной линией обозначен вектор $f + w(t)$. При этом изображается только проекция на первую координату. График, который соответствует второй координате, выглядит аналогично.

Пример 5. *Пусть*

$$A(t) \equiv \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad C(t) = \begin{pmatrix} \cos(1{,}25t) & \sin(1{,}25t) \\ -\sin(1{,}25t) & \cos(1{,}25t) \end{pmatrix}.$$

Q_0 — прямоугольник размером $(30, 30)$ с центром $(-8, 0)$. Начальное множество $\mathcal{R}_0$ — четырехугольник с вершинами в $(10, 10)^T$, $(-10, 10)^T$, $(10, -10)^T$, $(-20, -10)^T$, $f = (-4, 6)^T$, $\beta = 0{,}01$, $\alpha = 1{,}07$, W — соответствует шару радиуса $1{,}6$.

$$w(t) = 1{,}44 \begin{pmatrix} \cos(18t) \\ \sin(18t) \end{pmatrix}.$$

Результаты представлены на рисунке 8.

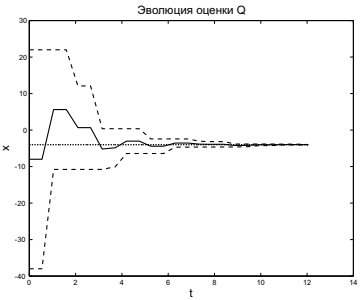


Рис. 7. Пример 4.

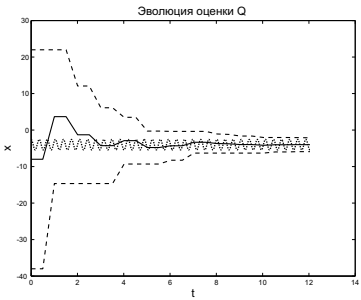


Рис. 8. Пример 5.

Пример 6. *В этом примере проиллюстрируем как влияет $w(t)$ на точность определения f.*

$$A(t) \equiv \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad C(t) = \begin{pmatrix} \cos(1{,}25t) & \sin(1{,}25t) \\ -\sin(1{,}25t) & \cos(1{,}25t) \end{pmatrix}.$$

Q_0 — прямоугольник размером $(30, 30)$ с центром $(-8, 0)$. Начальное множество $\mathcal{R}_0$ — четырехугольник с вершинами в $(10, 10)^T$, $(-10, 10)^T$, $(10, -10)^T$, $(-20, -10)^T$, $f = (-4, 6)^T$, $\beta = 0{,}01$, $\alpha = 1{,}07$, W — соответствует шару радиуса $1{,}6$.

$$w_1(t) = 7{,}96 \begin{pmatrix} \cos(t) \\ \sin(t) \end{pmatrix}, \quad w_2(t) = 7{,}2 \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

На рисунке 10 представлены результаты моделирования для $w_1(t)$ с описанным выше алгоритмом наблюдения.

Если теперь вместо $w_1(t)$ использовать постоянный шум $w_2(t)$, то даже при новом способе измерения результатом (то есть Q_N, где N — число наблюдений) является множество которое почти не отличается от множества помех $\mathcal{E}(0, W)$, если его сдвинуть в другую точку. На рисунке 9 изображен результат.

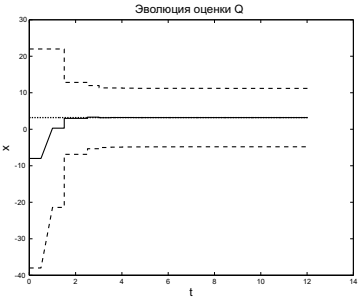


Рис. 9. Используется w_2.

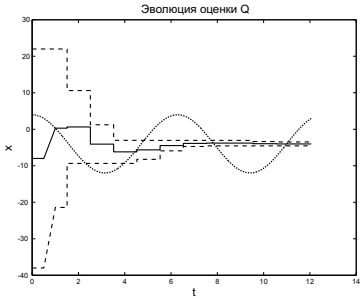


Рис. 10. Используется w_1.

5 Заключение

В работе рассмотрена задача поиска на кривой и на плоскости, даны условия на оптимальное размещение моментов наблюдения и предложен алгоритм для решение задачи идентификации с указанным механизмом наблюдений.

Список литературы

[1] *Куржанский А. Б.* Избранные труды. М., Издательство МГУ, 2009. — 756 с. — ISBN 978-5-211-05758-6

УДК 004.43

Библиотечная реализация Пролог-решателя

© 2012 г. И. В. Струков

iley@iley.ru

Кафедра алгоритмических языков

1 Введение

Язык логического программирования Пролог был разработан в 70-х годах XX века для решения задач обработки естественного языка. Со временем область успешного применения этого языка расширилась, и сейчас она включает в себя такие задачи искусственного интеллекта, как автоматическое доказательство теорем, экспертные системы, игровые программы и многое другое [2]. Ярким примером применения Пролога в современном программировании может служить компьютерная система Watson [5], известная тем, что в 2011 году выступила в качестве оппонента человеку в интеллектуальной игре Jeopardy и одержала победу. Часть её программного обеспечения написана на Прологе [3].

Несмотря на все достоинства данного языка и логической парадигмы программирования в целом, Пролог в наше время применяется при создании крупных программных систем чрезвычайно редко, даже в ситуациях, когда разработчики хорошо владеют данным языком. Основной причиной этого является, по всей видимости, тот факт, что любая крупная программа включает в себя большое число компонентов, и, как правило, лишь часть из них целесообразно выражать в терминах логического программирования. К примеру, уже упомянутая система Watson использует Пролог лишь для решения некоторых

задач, в частности обработки естественного языка, другие же части системы написаны на языках Java и C++.

Данная работа посвящена описанию экспериментальной реализации языка Пролог при помощи метода непосредственной интеграции в рамках библиотеки InteLib [1]. Эта реализация может упростить использование Пролога в качестве дополнительного языка программирования в программных проектах на C++.

2 Пролог как дополнительный язык программирования

При использовании Пролога в программном проекте, где основным является один из широко применяемых сегодня объектно-ориентированных языков программирования, возникают определённые трудности, связанные с интеграцией частей системы, написанных на разных языках. Для интеграции программного кода на Прологе в систему, написанную на объектно-ориентированном языке программирования, чаще всего применяют один из следующих двух подходов.

1. Написание двух раздельных программ, которые взаимодействуют посредством каких-либо средств межпроцессных коммуникаций.

2. Использование встраиваемого интерпретатора.

Оба эти подхода требуют написания дополнительного программного кода, служащего связующим звеном между частями системы. В первом случае этот код представляет собой реализацию некоторого протокола взаимодействия для каждого из языков системы. Во втором случае это так называемые обёртки, которые адаптируют структуры данных и операции, реализованные на базовом языке, для использования в коде на

дополнительном языке, который исполняется с помощью интерпретатора. В обоих случаях дополнительный код усложняет систему и затрудняет её разработку, отладку, тестирование и поддержку. Кроме того, каждый из этих подходов имеет свои ограничения. Например, использование встраиваемого интерпретатора не всегда допустимо из-за специфики целевой платформы или по административным причинам. Указанные проблемы в совокупности могут свести на нет преимущества, приобретаемые от применения дополнительного языка программирования.

Одним из подходов к совместному применению различных языков программирования в рамках одного проекта, призванных решить описанные проблемы, является метод непосредственной интеграции. Данный метод состоит в имитации синтаксиса и семантики дополнительного языка выразительными средствами языка базового. Примером использования метода непосредственной интеграции является библиотека InteLib, которая позволяет встраивать код на языках Лисп, Scheme [7], Рефал [6] и Плэнер [8] в программы на C++. Примечателен тот факт, что InteLib изначально спроектирована таким образом, чтобы с её помощью можно было реализовать вычислительные модели довольно широкого класса языков. Условно можно выделить в данной библиотеке две основные части. Первая часть включает в себя классы, реализующие общие для всех дополнительных языков структуры данных. Эти классы, помимо прочего, обеспечивают поддержку динамического определения типов во время исполнения программы и упрощают управление памятью с помощью техники так называемых «умных указателей». Вторая часть библиотеки включает в себя реализации вычислительных моделей альтернативных языков программирования. Благодаря тому, что все структуры данных, используемые реализациями языков в рамках InteLib, построены на основе общих базовых классов, передача данных между частями программы, написанными на различных языках програм-

мирования, максимально упрощается.

Оставшаяся часть статьи описывает InteLib Prolog, реализацию языка Пролог в рамках библиотеки InteLib. Полученная реализация не только позволяет упростить интеграцию кода на Прологе в программные проекты на C++, но также даёт возможность использовать Пролог совместно с языками программирования, уже реализованными в InteLib — Лисп, Scheme, Рефал, Плэнер. Разработанная реализация расширяет библиотеку InteLib классами, реализующими представление данных и вычислительную модель языка Пролог. Кроме того, InteLib Prolog позволяют имитировать синтаксис Пролога настолько, насколько это возможно в исходном коде на C++.

3 Представление программ и данных

Программа на Прологе представляет собой набор *правил*, которые могут быть как *фактами*, выражающими элементарные отношения между сущностями, так и сложными предложениями. Правила записываются при помощи *термов*. Терм по определению — это либо число, либо *атом*, либо список, либо *переменная*, либо *составной терм*. Составной терм состоит из *функтора*[1] и списка *аргументов*. Важен тот факт, что термы используются для представления как программ, так и данных в Прологе[2]. Можно провести аналогию между термами в Прологе и S-выражениями [4], используемыми в языке Лисп. Прологовскому атому соответствует *символ* в Лиспе, списки и числа используются аналогично в этих двух языках. Для обеспечения совместимости между программами на Лиспе и Прологе в рамках InteLib необходимо было повторно использовать указанные типы данных. Кроме того, поскольку InteLib, так же как и многие другие современные реализации языка Лисп,

[1] Функтор состоит из атома и числа, так называемой арности, т.е. числа аргументов

[2] Данное свойство языка программирования называют гомоиконностью

Таблица 1. Сравнение синтаксических элементов

Пролог	InteLib Prolog	Комментарий
`f`	`f`	атом
`X`	`X`	переменная
`f(1, 2, 3)`	`f(1, 2, 3)`	составной терм
`f(1, 2, 3).`	`*f(1, 2, 3)`	факт
`[1, 2, 3]`	`(S|1, 2, 3)`	список
`[H|T]`	`H^T`	список
`f(X) :- g(X).`	`f(X) <<= g(X)`	правило
`f(X), g(X)`	`f(X) & g(X)`	конъюнкция
`f(X); g(X)`	`f(X) | g(X)`	дизъюнкция
`f(X), !`	`f(X) & cut`	отсечение
`X + Y * Z`	`X + Y * Z`	арифм. выражение
`Z is X + Y`	`Z.is(X + Y)`	операция `is`

расширяет понятие S-выражения рядом дополнительных типов, например, таких, как строка (string), хэш-таблица, вектор и очередь, важно было обеспечить совместимость между языками и при использовании таких типов. В связи с этим, модель данных и реализация вычислителя InteLib Prolog была разработана таким образом, чтобы повторно использовать как можно большее число уже реализованных типов данных. Так, например, для представления списков в InteLib Prolog непосредственно используются ранее описанный в InteLib класс `SExpressionCons`. В тех случаях, когда непосредственное использование существующих классов было невозможным, повторное использование достигалось при помощи наследования. Примером может служить класс `PlgExpressionAtom`, представляющий атом в Прологе.

Для имитации синтаксиса Пролога в библиотеке предусмотрены перегрузки операторов. Для иллюстрации рассмотрим примеры выражений в стандартном синтаксисе Пролога и их аналоги в InteLib Prolog, представленные в таблице 1.

Как видно из таблицы, запись атомов, переменных и составных термов в InteLib Prolog полностью соответствует стандартному синтаксису. Для построения составных термов используется перегруженный оператор `()`. Для записи списков используется принятый в InteLib синтаксис, это позволяет добиться единообразия при использовании нескольких дополнительных языков программирования, например, Лиспа и Пролога.

В качестве замены оператора `:-`, который отсутствует в C++, используется оператор присваивания с побитовым сдвигом `<<=`, который визуально схож с `:-` и символом $\leftarrow$, спользуемым в математической логике. Факт в InteLib Prolog можно записать либо при помощи оператора `<<=`, как и при определении составных правил, либо при помощи оператора `*`. Второй способ позволяет визуально выделить факты и сделать запись короче.

Вместо запятой и точки с запятой, обозначающих соответственно конъюнкцию и дизъюнкцию в Прологе, используются символы `&` и `|`. Решение о выборе данных символов было принято в связи с тем, что точка с запятой в C++ имеет специальное значение: она разделяет операторы и не может быть перегружена, а запятая, хотя и может быть перегружена, обладает очень низким приоритетом, что не соответствует приоритету запятой в Прологе. Выбранные символы `&` и `|`, хотя и не схожи визуально с их эквивалентами в Прологе, обладают аналогичным смыслом в C++. В C++ эти символы обозначают побитовые конъюнкцию и дизъюнкцию, и их смысл в программе на InteLib Prolog должен быть интуитивно понятен программисту на C++.

Для записи арифметических операций используется перегузка всех необходимых операторов C++. Кроме того, для имитации бинарной операции `is` предусмотрен специальный метод в классе, представляющем переменные. Для замены символов, для которых не удалось найти адекватного представления в синтаксисе C++, используется текстовая запись. Так, напри-

мер, вместо символа отсечения `!` используется идентификатор `cut`.

Зачастую может быть удобнее писать программы при помощи оригинального синтаксиса Пролога, а не его имитации, и, кроме того, может возникнуть необходимость адаптировать программу, написанную на традиционном Прологе, для выполнения при помощи InteLib Prolog. В связи с этим была разработана программа-транслятор, которая преобразует исходный код на Прологе в синтаксис InteLib Prolog.

4 Вычислительная модель языка Пролог

Как уже упоминалось в статье, программа на Прологе состоит из набора правил. Все эти правила в совокупности составляют так называемую базу данных, а выполнение программы представляет собой вычисление некоторого запроса к базе данных. Для выполнения запроса Пролог-система производит поиск такого набора значений переменных, который сделал бы искомый запрос истинным в терминах правил, составляющих базу данных. Поиск производится при помощи обхода дерева решений в глубину.

Чтобы реализовать вычислительную модель Пролога в рамках InteLib Prolog, в библиотеку был введён класс, представляющий состояние вычислителя (`PlgContinuation`) . В ответ на любой запрос пользователь библиотеки получает экземпляр класса `PlgContinuation`, который может использоваться для последовательного получения всех решений. Экземпляр `PlgContinuation` содержит в себе три важные структуры данных: контекст, хранящий связи переменных, очередь текущих целей и стек точек возврата. Алгоритм поиска очередного решения для заданного запроса состоит из следующих шагов.

1. Проверить, есть ли в очереди текущих целей запросы, для которых ещё не найдены решения. Если нет — попытать-

ся произвести *откат*. Если откат произвести не удалось, решения не существует.

2. Извлечь из очереди целей первый элемент и произвести для него замену имён переменных на их значения в текущем контексте. Если очередь пуста — поиск успешно завершён.

3. Найти *предикат*, связанный с функтором текущей цели, и вычислить его.

4. Если вычисление предиката неуспешно, попытаться произвести откат. Если откат произвести не удалось, решения не существует. Иначе перейти к пункту 2.

Рассмотрим более подробно отдельные пункты алгоритма. Первый пункт необходим для проверки граничного условия. Благодаря нему алгоритм поиска сообщит о неудаче в случае, когда все решения уже были найдены.

Пункты с второго по четвёртый образуют основной цикл поиска. На первый взгляд может показаться, что для процесса поиска решения, который по сути является рекурсивным, использование рекурсии должно было бы привести к более простой реализации, однако, из-за различий в вычислительных моделях Пролога и C++ итеративная реализация оказывается более простой и эффективной. Этот факт подтверждает предположение, выдвинутое в [7] и состоящее в том, что вычислительные модели, основанные на рекурсии, следует реализовывать без использования рекурсии.

Используемый в третьем пункте предикат может быть реализован как на C++ (такой предикат назовём специальным), так и на Прологе. Такие операции, как конъюнкция и дизъюнкция, представляют собой специальные предикаты, поэтому их реализацию не пришлось напрямую предусматривать в алгоритме поиска. Важно отметить, что пользователь библиотеки может определять собственные специальные предикаты

при помощи того же механизма, с помощью которого реализованы конъюнкция и дизъюнкция. Этот факт даёт пользователю возможность произвольным образом расширять семантику языка.

Рассмотрим подробнее, как реализовано вычисление предикатов, написанных на Прологе. Каждый такой предикат состоит из некоторого списка правил и идентифицируется функтором. В InteLib Prolog ссылка на указанный список хранится в объекте, представляющем атом функтора. При вычислении предиката происходит поиск в списке первого подходящего правила при помощи *унификации*. Если такое правило найдено, его тело заносится в список целей. Позиция в списке, на которой был прекращён поиск, сохраняется в так называемой *точке возврата*. Если при откате потребуется найти другое решение для данного предиката, поиск будет продолжен с той же позиции.

5 Унификация

Одним из ключевых механизмов, на которых основана вычислительная модель языка Пролог, является унификация. Унификация — это операция, применяемая к двум произвольным термам. В процессе унификации Пролог-решатель производит поиск такого набора значений для несвязанных переменных, входящих в термы, что при подстановке этих значений вместо всех вхождений соответствующих переменных термы становятся тождественными. В упрощённом виде стандартный алгоритм унификации в Прологе состоит из следующих шагов.

1. Если один из термов является несвязанной переменной, задать её значение равным второму терму в текущем контексте. В таком случае унификация считается успешной.

2. Если оба терма являются списками, произвести попарную унификацию элементов списков. Если все элементы были

унифицированы, унификация списков считается успешной.

3. Если оба терма являются составными, проверить, что их функторы равны, а списки аргументов унифицируются. В таком случае унификация составного терма успешна.

4. Если оба терма являются числами или атомами, унификация сводится к простой проверке на равенство.

5. Если ни одно из условий 1-4 не выполнено, унификация считается неуспешной.

К реализации унификации в InteLib Prolog выдвигались следующие требования. Во-первых, унификация должна работать для уже существующих типов данных InteLib, причём, желательно, без написания дополнительного кода для каждого отдельного типа. Во-вторых, механизм унификации должен быть расширяемым. То есть, реализация унификации для дополнительных типов данных, которые в будущем могут быть добавлены в InteLib Prolog, не должна вызывать больших затруднений.

В результате были приняты следующие конструктивные решения. Во-первых, для унификации примитивных типов данных использовать стандартную операцию сравнения, предусмотренную InteLib. Во-вторых, для реализации унификации составных типов данных, таких как составные термы, использовать механизм виртуальных функций. Наконец, для уже существующих составных типов, таких как `SExpressionCons`, в качестве исключения выполнять унификацию специальным образом. С учётом перечисленных решений алгоритм унификации двух термов в InteLib Prolog выглядит следующим образом.

1. Если один из термов реализует виртуальный метод Unify, результатом унификации будет вычисление данного мето-

да с соответствующими параметрами. В частности, реализация данного метода в классе `PlgExpressionVariable` добавляет в текущий контекст связь данной переменной со вторым аргументом унификации, а реализация в `PlgExpressionTerm` проверяет, что второй аргумент также является составным термом и производит унификацию функторов и списков аргументов.

2. Если оба терма являются списками, итеративно произвести попарную унификацию элементов списка.

3. Если пункты 1-2 не выполнены, результатом унификации является вычисление метода `IsEqual`, реализующего стандартную операцию сравнения в InteLib. Это верно для чисел, атомов, строк и других типов данных.

Для использования каких-либо новых примитивных типов данных в программе на InteLib Prolog достаточно реализовать операцию сравнения. Для использования новых составных типов данных необходимо применить наследование и реализовать виртуальный метод `Unify`. Как правило, реализация данного метода не составляет большого труда, так как унификация для какого-либо составного типа данных в большинстве случаев тривиальным образом сводится к унификации объектов, которые экземпляр данного типа в себе содержит.

6 Откат

Другим важным механизмом, используемым в Прологе, является откат. При выполнении отката Пролог-вычислитель возвращается к состоянию[3], которое он принимал в прошлом в момент выбора пути в дереве решений. Такой выбор, в частно-

[3]Здесь под состоянием вычислителя подразумеваются значения переменных и список целей.

сти, осуществляется при обработке дизъюнкций и выборе правила из базы данных.

Для реализации отката в InteLib Prolog используются точки возврата. Каждая точка возврата при её создании сохраняет информацию о состоянии контекста переменных и очереди целей. В текущей реализации контекст, представляемый классом `PlgContext`, состоит из хэш-таблицы, предназначенной для хранения значений связанных переменных, и стека связей. Каждый элемент стека связей (будем называть его *фреймом*) — это список имён переменных, для которых были добавлены связи начиная с момента создания некоторой точки возврата. Когда создаётся очередная точка возврата, в стек связей добавляется новый пустой фрейм, а ссылка на предыдущую вершину стека сохраняется в точке возврата. Во время отката при помощи некоторой точки возврата T достаточно удалить из контекста связи для всех переменных, чьи имена встречаются в стеке связей выше фрейма, ссылка на который хранится в T. Таким образом будет восстановлено состояние контекста на момент создания T.

В работе со стеком точек возврата есть важный нюанс, связанный с использованием отсечения. Операция отсечения в Прологе удаляет все точки возврата, созданные с момента начала выполнения текущего предиката. Для того, чтобы реализовать такое поведение в InteLib Prolog, в стек точек возврата помещаются специальные метки, которые позволяют при выполнении отсечения определить, какие из точек следует удалить. В результате, алгоритм отсечения состоит в том, чтобы удалять точки возврата с вершины стека, пока не будет встречена метка начала выполнения предиката, либо стек не окажется пуст.

7 Пример программы

Ниже представлен полный пример программы, реализующей алгоритм быстрой сортировки [9] (quicksort) при помощи InteLib Prolog. Надо заметить, что данная программа отличается от своего аналога на стандартном Прологе только дополнительными объявлениями, необходимыми в C++, и деталями синтаксиса.

```
#include <stdio.h>
#include <prolog/prolog.hpp>

int main() {
  // необходимо для использования стандартных предикатов
  using namespace PlgStdLib;

  // вспомогательные объекты для конструирования списков
  SListConstructor S;
  SReference &nil = *PTheEmptyList;

  // определения всех используемых атомов и переменных
  PlgAtom qsort("qsort"), split("split");
  PlgVariable H("H"), T("T"),
    R("R"), L("L"),
    X("X"), Res("Res"),
    LS("LS"), RS("RS");

  // предикат сортировки qsort(+In, -Out)
  // In - входной список, Out - результат
  *qsort(nil, nil);

  qsort(H^T, Res) <<=
    split(H, T, L, R) &
    qsort(L, LS) &
    qsort(R, RS) &
```

```
    append(LS, H ^ RS, Res);

  // предикат для разбиения списка
  // split(+Pivot, +In, -Lower, -Upper)
  // Pivot - граничное значение, In - входной список,
  // Lower и Upper - результирующие списки
  *split(X, nil, nil, nil);

  split(X, H^T, H^LS, RS) <<=
    (H <= X) &
    split(X, T, LS, RS);

  split(X, H^T, LS, H^RS) <<=
    (H >  X) &
    split(X, T, LS, RS);

  // входной список
  SReference list = (S|3, 1, 2, 5, 4);

  // получаем объект для выполнения запроса
  PlgContinuation cont = qsort(list, X).Query();

  // вычисляем первый результат запроса
  // в нашем случае вычисление всегда успешно
  cont->Next();

  // получаем значение переменной X, с которой
  // при выполнении запроса был связан результат
  SReference sortedList = cont->GetValue(X);

  // выводим результат на экран
  puts(sortedList->TextRepresentation().c_str());
  return 0;
}
```

8 Заключение

В результате выполненной работы библиотека InteLib была дополнена реализацией языка программирования Пролог, которая на данный момент, хотя и не является полной, вполне работоспособна и может быть использована для интеграции несложных программ на Прологе в проекты на C++. Полученная реализация включает в себя библиотеку классов на C++, которая позволяет представлять и исполнять программный код на Прологе, а также транслятор для адаптации существующих программ на Прологе к данной реализации. Как показывает практика, InteLib Prolog позволяет в отдельных случаях значительно упростить интеграцию кода на Прологе в программные проекты на C++, что подтверждает перспективность метода непосредственной интеграции.

Список литературы

[1] Elena Bolshakova, Andrey Stolyarov. Building functional techniques into an object-oriented system. // *Knowledge-Based Software Engineering. Proceedings of the 4th JCKBSE*, volume 62 of *Frontiers in Artificial Intelligence and Applications*, pages 101–106, Brno, Czech Republic, 2000. IOS Press, Amsterdam.

[2] I. Bratko. *Prolog programming for artificial intelligence.* International computer science series. Addison Wesley, 2001.

[3] Adam Lall, Paul Fodor. Natural language processing with prolog in the ibm watson system. Technical report, IBM Thomas J. Watson Research Center, 2011.

[4] John McCarthy. Recursive functions of symbolic expressions and their computation by machine, part i. *Commun. ACM*, 3(4):184–195, April 1960.

[5] IBM Systems, Technology Group. Watson — a system designed for answers. Technical report, IBM Corporation, 2011.

[6] И. Е. Бронштейн, А. В. Столяров. Библиотечная поддержка объектно-ориентированной модели языка Рефал. // *Сборник статей молодых учёных факультета ВМиК МГУ*, pages 36–46, Москва, 2009. Издательский отдел факультета ВМиК МГУ.

[7] А. В. Столяров. Импорт вычислительной модели языка scheme в объектно-ориентированное окружение. // *Сборник статей молодых учёных факультета ВМиК МГУ*, pages 119–130, Москва, 2008. Издательский отдел факультета ВМиК МГУ.

[8] О. Г. Фролова. Библиотечная реализация вычислительной модели языка Плэнер. // *Сборник статей молодых учёных факультета ВМиК МГУ*, pages 24–33, Москва, 2008. Издательский отдел факультета ВМиК МГУ.

[9] Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. Глава 7. Быстрая сортировка // *Алгоритмы: построение и анализ = Introduction to Algorithms / Под ред. И. В. Красикова*, с. 198-219, Москва, 2005.

РЕФЕРАТЫ

Н. А. Андреев. Современные математические модели влияния на цену: приложение к задаче оптимального управления портфелем на рынке, движимом заявками // Сборник статей молодых ученых факультета ВМК МГУ, № 9, 2012, с. 7–25. Оптимальное управление портфелем на рынке акций является одной из основных задач финансовой инженерии. Определяющую роль при ее решении играет модель динамики рынка и влияния игрока на цену актива. В работе приведен обзор микроструктурных моделей влияния на цену на рынке, движимом заявками, отражающих наиболее значимые аспекты при управлении позицией. В работе разбираются основные предположения моделей и их взаимосвязь, а также ряд возникающих смежных вопросов.

Библиография 14 раб.

А. И. Аристов. Усовершенствованный алгоритм нечеткого поиска и особенности его реализации в облачной архитектуре // Сборник статей молодых ученых факультета ВМК МГУ, № 9, 2012, с. 26–45. В работе введено понятие расстояния от строки до шаблона. Предложен алгоритм вычисления этого расстояния, рассмотрены способы его реализации как в облачной, так и в традиционной архитектуре.

Библиография 5 раб.

М. А. Губанова. Задача оптимального управления для математической модели финансового кризиса // Сборник статей молодых ученых факультета ВМК МГУ, № 9, 2012, с. 46–63. В данной статье изучается задача оптимального управления взаимодействием фирм в условиях финансового кризиса.

Библиография 9 раб.

Д. А. Гуляев. Об одной смешанной задаче для уравнения теплопроводности, приводящей к спектральной задаче с граничными условиями третьего рода, одно из которых содержит спектральный параметр // Сборник статей молодых ученых факультета ВМК МГУ, № 9, 2012, с. 64–67. Рассматривается смешанная задача для уравнения теплопроводности, приводящая к спектральной задаче с граничными условиями третьего рода, одно из которых содержит спектральный параметр. Данная задача возникает, например, при моделировании процесса теплопереноса в однородном стержне, на одном из концов которого задается линейная взаимосвязь значений теплового потока и температуры, а на другом конце помещена сосредоточенная теплоемкость.

Библиография 4 раб.

Ю. С. Корухова, М. В. Мытрова. Об одном подходе к решению задачи поиска нот музыкальных произведений // Сборник статей молодых ученых факультета ВМК МГУ, № 9, 2012, с. 68–85. В статье предложен метод поиска нот музыкальных произведений в формате MusicXML по записи фрагмента мелодии. В основе подхода лежит использование волновых правил, позволяющих устранять различия в похожих математических выражениях. Метод позволяет находить в коллекции файлов множество похожих на поисковый запрос и оценивать их соответствие запросу.

Библиография 16 раб.

А. С. Лебедев. Оценка фона изображения для автоматического обнаружения разметки и дефектов дорожного покрытия // Сборник статей молодых ученых факультета ВМК МГУ, № 9, 2012, с. 86–101. В статье предлагается новый алгоритм оценки фона изображения, который позволяет получать более дискриминативные признаки для обнаружения разметки и дефектов дорожного покрытия.

Библиография 15 раб.

А. И. Месяц. Управление эллипсоидальным контейнером // Сборник статей молодых ученых факультета ВМК МГУ, № 9, 2012, с. 102–121. В статье рассматривается общая постановка управления эллипсоидальным контейнером в $\mathbb{R}^n$, приводится метод решения в $\mathbb{R}^n$, даются более подробные результаты для $\mathbb{R}^2$.

Библиография 10 раб.

Н. А. Моросанова. Схемы оценки уверенности в экспертных системах // Сборник статей молодых ученых факультета ВМК МГУ, № 9, 2012, с. 122–135. В работе рассматриваются различные подходы к организации схемы оценки уверенности в информации, использующиеся в экспертных системах. Изучаются трансформации различных схем друг в друга, задачи, возникающие при этом, и возможности, получаемые с их помощью.

Библиография 12 раб.

А. О. Новикова. Построение множеств достижимости некоторых управляемых систем пиксельным методом // Сборник статей молодых ученых факультета ВМК МГУ, № 9, 2012, с. 136–153. Реализован пиксельный метод построения множеств достижимости управляемых систем с использованием параллельных вычислений на графических процессорах. Проведен анализ результатов для контрольных примеров, в том числе для известной задачи Ли-Маркуса.

Библиография 7 раб.

В. Г. Саргсян. О числе множеств, k-свободных от нуля, в группах простого порядка // Сборник статей молодых ученых факультета ВМК МГУ, № 9, 2012, с. 154–172. С помощью преобразования Фурье доказывается, что логарифм числа подмножеств группы $\mathbf{Z}_p$, в которых не имеет решения уравнение $x_1 + \cdots + x_k = 0$, асимптотически равен p/k.

Библиография 17 раб.

И. С. Сорокин, В. А. Лапшин. Алгоритм оценки кривой доходности по нескольким группам облигаций с оптимальным выбором весов // Сборник статей молодых ученых факультета ВМК МГУ, № 9, 2012, с. 173–190. В статье предлагается метод для определения срочной структуры процентных ставок для группы облигаций разного кредитного качества.

Библиография 14 раб.

Д. Е. Стельмашенко. Алгоритм восстановления глобального контекста // Сборник статей молодых ученых факультета ВМК МГУ, № 9, 2012, с. 191–209. В статье предложен алгоритм слияния нескольких формальных контекстов, отображающих данные разных полевых исследований. Предложенный подход позволяет построить единую понятийную структуру исследуемой проблемной области.

Библиография 5 раб.

В. А. Степанович. Гарантированное оценивание в задаче поиска на плоскости // Сборник статей молодых ученых факультета ВМК МГУ, № 9, 2012, с. 210–234. В работе рассматривается задача поиска объекта на плоскости. Движение объекта удовлетворяет известным законам движения, возможно с неопределенностью. В частности, были рассмотрены следующие случаи: неподвижный объект; движение вдоль известной кривой с неизвестной, но ограниченной известными величинами, скоростью; движение на плоскости в соответствии с линейным дифференциальным уравнением, в которое входит неизвестная функция — помеха.

Библиография 1 раб.

И. В. Струков. Библиотечная реализация Пролог-решателя // Сборник статей молодых ученых факультета

ВМК МГУ, №9, 2012, с. 235–250. Статья посвящена описанию экспериментальной реализации языка Пролог при помощи метода непосредственной интеграции в рамках библиотеки InteLib. Эта реализация может упростить использование Пролога в качестве дополнительного языка программирования в программных проектах на C++.

Библиография 9 раб.