



Адрес редакции
670000, г. Улан-Удэ,
ул. Смолина, 24а
E-mail: univ@bsu.ru

Адрес издателя
670000, г. Улан-Удэ,
ул. Смолина, 24а
E-mail: riobsu@gmail.com

Свидетельство о регистрации
ПИ №ФС77–36152 от 06 мая
2009 г. Федеральная служба по
надзору в сфере связи, инфор-
мационных технологий и мас-
совых коммуникаций (Роском-
надзор)

Редактор *Р.В. Хабдаева*
Компьютерная верстка
Д.О. Трунина

Подписано в печать 25.06.13.
Формат 60 x 84 1/8.
Уч.-изд. л. 11,28. Усл. печ. л. 22,56.
Тираж 1000. Заказ 514.
Цена договорная.

Отпечатано в типографии
Издательства БГУ
670000, г. Улан-Удэ,
ул. Сухэ-Батора, 3а

Учредитель
ФГБОУ ВПО «Бурятский государственный университет»

ВЕСТНИК БУРЯТСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

Издается с 1997 г.

Выходит 1 раз в год

Выпуск МАТЕМАТИКА, ИНФОРМАТИКА 9 / 2013

Журнал включен Высшей аттестационной комиссией в Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученых степеней доктора и кандидата наук

Редакционный совет «Вестника БГУ»

С.В. Калмыков, чл.-кор. РАО, д-р пед. наук, проф. (председатель);
И.К. Шаранхаев, канд. физ.-мат. наук, доц. (зам. председателя);
Н.Н. Татарникова (зам. председателя, директор Издательства БГУ);
М.В. Бадмаева, д-р филос. наук, доц.; *Т.С. Базарова*, д-р пед. наук, доц.;
А.С. Булдаев, д-р физ.-мат. наук, проф.; *Д.И. Бураев*, д-р ист. наук, проф.;
А.В. Гаськов, д-р пед. наук, проф.; *Н.Ж. Дагбаева*, д-р пед. наук, проф.;
Ц.З. Доржиев, д-р биол. наук, проф.; *С.С. Имхелова*, д-р филол. наук,
проф.; *Л.П. Ковалева*, канд. филол. наук, проф.; *К.Б.-М. Митупов*, д-р ист.
наук, проф.; *И.И. Осинский*, д-р филос. наук, проф.; *М.Н. Очиров*, д-р
пед. наук, проф.; *В.В. Хахинов*, д-р хим. наук, проф.; *В.Е. Хитрихеев*, д-р
мед. наук, проф.

Редакционная коллегия выпуска

А.С. Булдаев, д-р физ.-мат. наук, проф. (главный редактор); *С.Н. Васильев*,
акад. РАН; *А.Д. Мижидон*, д-р техн. наук, проф.; *В.И. Гурман*, д-р
техн. наук, проф.; *Д.Ш. Ширапов*, д-р физ.-мат. наук, проф.; *В.Б. Цыре-
нова*, д-р пед. наук, доц.; *В.В. Кибирев*, канд. физ.-мат. наук, проф.;
Г.А. Шишкин, канд. физ.-мат. наук, проф.; *В.И. Антонов*, канд. физ.-мат.
наук, доц.; *И.Б. Юмов*, канд. физ.-мат. наук, доц.; *И.К. Шаранхаев*, канд.
физ.-мат. наук, доц.; *Т.Г. Дармаев*, канд. физ.-мат. наук, доц.



1. Управляемые системы и методы оптимизации

УДК 519.6

© А.И. Колмакова

МНОГОМЕТОДНАЯ ОПТИМИЗАЦИЯ УПРАВЛЕНИЯ В ЭКОНОМИЧЕСКОЙ МОДЕЛИ ВЫБОРА НАЛОГОВОЙ СТАВКИ ДЛЯ ПРЕДПРИЯТИЙ ЭНЕРГЕТИЧЕСКОЙ ОТРАСЛИ

В рамках нелинейной однопродуктовой модели с производственной функцией постоянной эластичности замещения факторов и логистическим уравнением Ферхюльста для описания динамики численности персонала, с внесенными поправками, связанными с непосредственным учетом налоговых отчислений, формулируется математическая постановка задачи в виде системы дифференциальных уравнений, описывающих налогообложение прибыли предприятий. Рассматривается задача поиска оптимального программного управления.

Ключевые слова: оптимальное управление, динамическое программирование, налог на прибыль, модель деятельности предприятия.

© А.И. Колмакова

MULTIMETHOD OPTIMIZATION OF CONTROL IN ECONOMIC MODEL OF TAX RATE CHOICE FOR COMPANIES OF ENERGY SECTOR

Within the nonlinear single product model with constant elasticity of substitution production function and Verhulst logistic equation for description of dynamics of personnel number with corrections related to direct accounting of tax charges, a mathematical problem is formulated in the form of a system of differential equations describing taxation of enterprise profits and a problem of optimal software control search is considered.

Keywords: optimal control, dynamic programming, profits tax, enterprise activities model.

Введение

Появление налогов было обусловлено необходимостью удовлетворять растущие потребности финансового хозяйства государства. Процесс взимания налогов осуществляется государством и является его функцией. Способ, характер и масштабы мобилизации денежных ресурсов в бюджете государства зависят от стадии экономического развития общества. При этом минимальный размер формируемых денежных фондов, состоящих в большей степени из налоговых платежей, определяется суммой расходов государства на исполнение его функций: социальной, экономической, оборонной и др.

Система налогообложения является одним из важнейших факторов, влияющих на решение об инвестировании, особенно если речь идет о прямых иностранных инвестициях. В целом на принятие решений об инвестировании оказывают влияние как налоговые, так и неналоговые факторы, при помощи которых осуществляется сравнение потенциальной прибыли от инвестиций и риска. Обычно компании при выборе того или иного проекта принимают во внимание различие в законных ставках налога, направляя доходы в страны с низким налогообложением [1].

Оптимальность налогообложения принято оценивать с точки зрения общего эффекта для благосостояния общества и с точки зрения выгод для определенного налогоплательщика (экономического агента). Именно от характера и сущности проводимой государством и его органами общей налогово-бюджетной политики зависит, определять ли налоги как «бремя», «зло» или как способ (эффективный в большей или меньшей степени) перераспределения материальных благ и финансирования общих объективных потребностей общества.

1. Многометодные алгоритмы расчета оптимального управления

Предлагается вычислительная технология расчета оптимального управления, основанная на применении нескольких процессоров для поиска численного решения задачи несколькими методами оптимизации одновременно. В соответствии с этой технологией, после нахождения очередного приближения все методы оцениваются по полученному приращению функционала, и из них выбирается

наиболее эффективный метод для продолжения оптимизации, а полученное этим методом приближение передается остальным методам в качестве начального для выполнения следующей итерации.

1.1. Расчет градиентов в задаче оптимального управления с фазовыми ограничениями

В методах градиентного типа трудоемкую операцию расчета градиента, требующую интегрирования сопряженной системы, следует выполнить только один раз, а затем использовать найденный градиент в итерационных формулах всех методов. Вычислительные затраты на выполнение одного шага многометодного алгоритма в этом случае значительно сократятся. Кроме того, реализация шага каждым из методов будет выполнена с использованием одних и тех же приближенно найденных величин.

Приведем общую постановку задачи оптимального управления. Пусть управляемый процесс описывается системой

$$f = 10\% \tag{1}$$

терминальными условиями

$$I_j(u) = \varphi^j(x(t)) = 0, \tag{2}$$

и фазовыми ограничениями

$$J_i(u, t) = g^i(x, t) = 0, \quad i = \overline{1, s}, \quad t \in T. \tag{3}$$

Управление стеснено следующими ограничениями:

$$u(t) \in U, \tag{4}$$

где U – ограниченное замкнутое множество из R^r . Вектор-функция $f(x, u, t)$ непрерывно дифференцируема по x и u и непрерывна по t ; $\varphi^j(x)$, $j = \overline{1, m}$ – непрерывно дифференцируемые функции.

Пусть на правых концах траекторий системы (1) определена непрерывно дифференцируемая выпуклая скалярная функция

$$I_0(u) = \varphi^0(x(t_1)), \tag{5}$$

которую требуется минимизировать.

Градиенты функционалов $I_j(u)$, $j = \overline{0, m}$ с помощью $H^j(\psi_j, x, u, t) = \psi_j'(t)f(x, u, t)$ и сопряженной системы

$$\dot{\psi}_j = -f_x(x, u, t)' \psi_j(t), \quad \psi_j(t_1) = -\psi_x^j(x(t_1)) \tag{6}$$

вычисляются по формулам

$$\nabla I_j(u) = -H_u^j(\psi_j, x, u, t), \quad j = \overline{1, m}. \tag{7}$$

Для каждого $t \in T$ можно аналогично вычислить градиенты $J_j(u, t)$, $j = \overline{1, s}$:

$$\nabla J_j(u, t) = -\overline{H}_u^j(\Phi_j, x, u, t, \tau), \quad t_0 \leq \tau \leq t \leq t_1, \tag{8}$$

где $\overline{H}_u^j(\Phi_j, x, u, t, \tau) = \Phi_j'(t, \tau)f(x, u, \tau)$, $\Phi_j(t, \tau)$, $j = \overline{1, s}$ – решения сопряженных систем $\frac{\partial \Phi_j(t, \tau)}{\partial \tau} = -\frac{\partial f(x, u, \tau)}{\partial x} \Phi_j(t, \tau)$, $\tau \in T$, $\Phi_j(t, t) = -\frac{\partial g^j(x(t))}{\partial x}$, $j = \overline{1, s}$.

1.2. Численные методы для решения задач с ограничениями на управление

Перейдем теперь к рассмотрению алгоритмов для решения задач с ограничениями на управление (4), но без ограничений (2)–(3). Предположим, что при некотором $u^k(t) \in U$, $t \in T$ найдено решение системы (1) $x^k(t)$, $t \in T$.

Полагая $j=0$, проинтегрируем сопряженную систему от $t = t_1$ до $t = t_0$ при $u = u^k(t)$, $x = x^k(t)$. На ее решении $\psi^k = \psi^0(t)$ вычислим управление из принципа максимума:

$$\bar{u}^k(t) = \arg \max_{u \in U} H(\psi^k, x^k, u, t), \quad t \in T,$$

и найдем значения скалярной функции

$$w_k(\bar{u}(t), t) = H(\psi^k, x^k, \bar{u}, t) - H(\psi^k, x^k, u^k, t), \quad t \in T.$$

Если для заданного u^k и найденных x^k, ψ^k, \bar{u}^k принцип максимума нарушается: $w_k(\bar{u}^k(\tau_k), \tau_k) > 0$, то можно реализовать одну итерацию метода [3] для улучшения u^k .

Множество точек, в которых нарушается принцип максимума, обозначим через

$$T_\varepsilon = \left\{ t \in T : w_k(\bar{u}^k(t), t) \geq \varepsilon w_k(\bar{u}^k(\tau_k), \tau_k) \right\}, \quad \varepsilon \in [0, 1],$$

где τ_k – точка максимума этой функции на T_ε . Варьируя ε , можно найти его оптимальное значение ε_k , при котором управление

$$u_\varepsilon^{k+1}(t) = \begin{cases} \bar{u}^k(t), & t \in T_\varepsilon, \\ u^k(t), & t \in T \setminus T_\varepsilon, \end{cases} \quad (9)$$

доставит наименьшее значение целевому функционалу. При поиске ε_k можно использовать несколько потоков для одновременного интегрирования системы (1) с управлениями (9), соответствующими разным значениям ε_k .

В силу структуры управлений, генерируемых итерационной формулой (9), релаксации алгоритма могут прекратиться еще до получения управления, удовлетворяющего принципу максимума. Тогда для продолжения процесса оптимизации необходимо применить другой алгоритм, на итерациях которого конструируются управления не только с граничными, но и с внутренними относительно множества допустимых управлений значениями. Так, например, восстановить сходимость обычно удается с помощью построения выпуклой комбинации двух управлений:

$$u^{k+1}(t) = u^k(t) + \alpha [\bar{u}^k(t) - u^k(t)], \quad \alpha \in [0, 1]. \quad (10)$$

Вычисления по формулам (9) и (10) можно вести параллельно, выбирая из полученных приближений такое u^{k+1} , которому соответствует меньшее значение функционала. Если значения функционала будут сравниваться через несколько итераций, то в качестве критерия для сравнения эффективности методов (9) и (10) следует использовать значения приращений функционала, полученных на соседних итерациях каждого из методов. На практике установлено, что применение вариаций двух типов – «горизонтальной» (9) и «вертикальной» (10) позволяет избежать проявления эффекта «прилипания управления к границам», свойственного алгоритмам, основанным на принципе максимума.

Если в итерационной формуле (10) управление $\bar{u}^k(t)$ будет вычисляться из линеаризованного принципа максимума

$$\bar{u}^k(t) = \arg \max_{u \in U} H_u(\psi^k, x^k, u, t)'u(t), \quad t \in T,$$

то получим итерации метода условного градиента. Очевидно для линейных по управлению систем это управление будет совпадать с управлением, найденным из принципа максимума.

1.3. Численное решение задач с фазовыми ограничениями

Идея излагаемого ниже алгоритма [5] состоит в том, что на каждой его итерации решается вспомогательная задача минимизации модифицированной функции Лагранжа при линеаризованных ограничениях (2), (3). Якобиан линеаризованных ограничений строится из градиентов (7), (8), для расчета которых можно так же использовать параллельные вычисления. Значения двойственных переменных, полученные в результате решения вспомогательной задачи на каждой итерации, являются новым приближением для этих переменных на следующей итерации.

Пусть $u^k(t)$ – текущее приближение управления, а $x^k(t)$ – фазовая траектория, соответствующая $u^k(t)$, $t \in T$. Используя градиенты (7), (8), линеаризуем условия (2), (3) в окрестности $u^k(t)$:

$$I_i^L(u^k, u) = I_i(u^k) + \int_{t_0}^{T_i} \nabla I_i(u^k, t)'(u(t) - u^k(t)) dt = 0, i = \overline{1, m}, \tag{11}$$

$$J_j^L(u^k, u, \tau) = J_j(u^k, \tau) + \int_{t_0}^{\tau} \nabla J_j(u^k, t)(u(t) - u^k(t)) dt = 0, \tag{12}$$

$$j = \overline{1, s}, \tau \in T$$

Построим модифицированную функцию Лагранжа для задачи (1)-(5) в следующем виде

$$L(u, u^k, \lambda^k, \mu^k) = I_0(u) - \lambda^k'(I(u) - I^L(u^k, u)) - \int_{t_0}^{t_1} \mu^k(t)'(J(u, t) - J^L(u^k, u, t)) dt + \frac{\rho}{2}(I(u) - I^L(u^k, u))'(I(u) - I^L(u^k, u)) + \frac{\rho}{2} \int_{t_0}^{t_1} (J(u, t) - J^L(u^k, u, t))'(J(u, t) - J^L(u^k, u, t)) dt, \tag{13}$$

где I, I₀ – m-векторы, J, J₀ – s-векторы, λ^k, μ^k – m и s-мерные множители Лагранжа; ρ ≥ 0 – коэффициент штрафа.

На (k+1)-й итерации рассматриваемого метода решается задача минимизации функционала (13) на решениях системы (1) при линейных ограничениях (11), (12), (4). При этом для численного решения формируется задача математического программирования с линейными ограничениями специальной структуры.

2. Математическая модель деятельности предприятия

Сформулируем постановку задачи об оптимизации ставки налога на прибыль для предприятий энергетической отрасли.

В нашем случае переменными состояниями будут затраты на производство x₁ в момент t, налоговые отчисления x₂ в момент t, количество основных производственных фондов (ОПФ) x₃ в момент t, количество трудовых ресурсов x₄ в момент t. В качестве управления возьмем налоговую ставку v(t).

Уравнение динамики затрат на производство получено из экономического смысла состава затрат на производство и прибыли, облагаемой налогом. Уравнение динамики для налоговых отчислений получено дисконтированием величины налога. Уравнение динамики основных производственных фондов получено из взаимосвязи валового продукта, производственного потребления и конечного продукта.

Простейшей моделью, характеризующей демографические процессы, является экспоненциальный рост. Но недостаток такой модели в том, что рост или уменьшение численности персонала происходит неограниченно. В энергетике же численность персонала, независимо от загрузки действующего оборудования, остается постоянной и изменяется только при вводе новых и демонтаже устаревших агрегатов. Данной ситуации позволяют избежать модели из ряда плотно-зависимого роста. Простейшей моделью данного типа является логистическое уравнение Ферхюльста. Предполагается, что удельная скорость изменения численности работников линейно уменьшается с ростом (убыванием) численности, и имеется некая предельная численность персонала (емкость среды) [7].

Система уравнений, описывающая экономическую систему будет иметь вид [8]:

$$\frac{dx_1}{dt} = \mu(F(x_3, x_4) - v(t)((F(x_3, x_4) - x_1) - x_1);$$

$$\frac{dx_2}{dt} = e^{-\delta t}(F(x_3, x_4) - x_1)v(t);$$

$$\frac{dx_3}{dt} = \frac{1}{q}(1-\alpha)(1-\nu(t))F(x_3, x_4) - \chi x_3 - cx_1 - x_3;$$

$$\frac{dx_4}{dt} = lx_4 \left(1 - \frac{x_4}{L}\right).$$

Значения параметров данной задачи выбираются из интервалов, определяемых экономическим смыслом:

$0 < \mu < 1$ – доля выручки, которая составит затраты на производство в следующем периоде;

$0 < \delta < 1$ – коэффициент дисконтирования;

$q \geq 0$ – коэффициент эффективности использования основных фондов;

$0 < \alpha < 1$ – коэффициент производственных затрат;

$\chi > 0$ – коэффициент амортизации;

$c \geq 0$ – непроизводственное потребление;

$r \geq 0$ – мальтузианский параметр;

$L > 0$ – емкость среды;

$F(x_3, x_4)$ – производственная функция.

Наибольшее применение в теоретическом и прикладном макроэкономическом анализе имеют два вида ПФ: мультипликативная (чаще именуемая функцией Кобба-Дугласа) и функция с постоянной эластичностью заменяемости ресурсов (ПЭЗ). Эти функции обладают преимуществами с нескольких точек зрения: 1) они хорошо экономически интерпретируются; 2) имеют небольшое число параметров, что облегчает их статистическую оценку; 3) соответствующие этим ПФ показатели экономического роста, эффективности, интенсификации имеют удобную аналитическую форму [9].

При использовании ПЭЗ функции удастся избежать тех недостатков, которые присущи ПФ Кобба-Дугласа, в частности, нет неправдоподобного замещения одного фактора другим, а производительность труда не растет неограниченно. Единственным препятствием для использования наиболее «хорошей» с аналитической точки зрения реальности ПЭЗ-функции является ее достаточная сложность при исследовании и оценке параметров. Функция ПЭЗ имеет вид:

$$F(t) = A(t) \left[kK(t)^{-p} + (1-k)L(t)^{-p} \right]^{\frac{\beta}{p}},$$

где $A(t)$ – научно-технический прогресс, $0 < k < 1$ – параметр степени фондоемкости, $\beta > 0$ – параметр отдачи на масштаб производства или степени однородности функции, $\sigma = \frac{1}{1+p}$ – эластичность замещения ресурсов, $\sigma > 0$, $\sigma \neq 1$ [10].

При моделировании научно-технического прогресса (НТП) будем исходить из представления об экзогенном (или автономном) техническом прогрессе, который существует также в том случае, когда основные производственные факторы не изменяются. Частным случаем такого НТП является нейтральный прогресс по Хиксу, который обычно учитывается с помощью экспоненциального множителя. Тогда с учетом НТП ПЭЗ примет вид:

$$F(t) = ae^{\lambda t} \left[kK(t)^{-p} + (1-k)L(t)^{-p} \right]^{\frac{\beta}{p}},$$

где a_0 – параметр масштабирования, λ – темп прироста за счет нейтрального НТП.

Обычно x_1 и x_3 измеряются в стоимостных единицах, x_4 – в человеко-часах или численности работников. Такая производственная функция используется для описания объектов от промышленного объединения до отрасли, характеризующейся устойчивым, стабильным функционированием. Сумма коэффициентов эластичности представляет собой показатель эффекта расширения масштаба.

Как было сказано выше, производственная сфера функционирует при заданной налоговой ставке $\nu(t)$ и своими внутренними управлениями она распоряжается так, чтобы минимизировать x_2 , а

внешняя среда заинтересована в том, чтоб налоговые отчисления были «побольше». В нашем случае задача оптимизации налоговой ставки состоит в максимизации налоговых отчислений x_2 , то есть $x_2 \rightarrow \max$.

Формализованная постановка задачи об оптимизации ставки налога на прибыль будет иметь вид:

Переменные состояния: x_1, x_2, x_3, x_4 .

Переменные управления: $v(t)$.

Уравнения процесса:

$$\begin{aligned} \frac{dx_1}{dt} &= \mu(ae^{\lambda t} [kK(t)^{-p} + (1-k)L(t)^{-p}]^{\frac{\beta}{p}} - \\ &- v(t)((ae^{\lambda t} [kK(t)^{-p} + (1-k)L(t)^{-p}]^{\frac{\beta}{p}} - x_1) - x_1; \\ \frac{dx_2}{dt} &= e^{-\delta t} (ae^{\lambda t} [kK(t)^{-p} + (1-k)L(t)^{-p}]^{\frac{\beta}{p}} - x_1)v(t); \\ \frac{dx_3}{dt} &= \frac{1}{q}(1-\alpha)(1-v(t))ae^{\lambda t} [kK(t)^{-p} + (1-k)L(t)^{-p}]^{\frac{\beta}{p}} - \\ &- \chi x_3 - cx_1 - x_3; \\ \frac{dx_4}{dt} &= lx_4(1 - \frac{x_4}{L}). \end{aligned}$$

Начальные значения:

$$x_1(0) = x_{10}, x_2(0) = x_{20}, x_3(0) = x_{30}, x_4(0) = x_{40}, t \in [0, T].$$

Ограничения на фазовые переменные:

$$x_1 > 0, x_2 \geq 0, x_3 > 0, x_4 > 0.$$

Ограничение на управление:

$$v_{\min} \leq v(t) \leq v_{\max}.$$

Целевая функция: $x_2 \rightarrow \max$.

Параметры:

$$\mu, a, \lambda, k, \rho, \beta, \delta, \chi, q, \alpha, \chi, c, l, L.$$

Значения параметров являются постоянными:

$$\mu = \mu^*, a = a^*, \lambda = \lambda^*, k = k^*, \rho = \rho^*, \beta = \beta^*, \delta = \delta^*, \chi = \chi^*, q = q^*, \alpha = \alpha^*, \chi = \chi^*, c = c^*, l = l^*, L = L^*.$$

3. Численные эксперименты

Проведем численное исследование модели на основе данных годовой отчетности, бухгалтерской отчетности по российским стандартам и ежеквартальных отчетов эмитента за период 2002-2010 гг. ОАО «Иркутскэнерго» [10].

Доля выручки, которая составит затраты на производство в следующем году μ , определяется средним значением отношения выручки, уменьшенной на сумму налоговых отчислений, к затратам на производство за рассматриваемый период. По результатам расчетов $\mu = 0,77$.

Коэффициент дисконтирования получен на основе рентабельности по прибыли. Для энергетической промышленности рентабельность должна составлять не менее 10% (для «Иркутскэнерго» среднее значение за период 10,5%). Тогда коэффициент дисконтирования $\delta = 0,91$.

Фондоотдача, рассчитанная на основе среднегодовой стоимости ОПФ, равна $q = 0,36$. В динамике данный показатель растет, и, следовательно, повышается эффективность использования ОПФ.

Коэффициент производственных затрат, который определяется как отношение суммы материальных затрат к выпуску, в среднем за изучаемый период составил $\alpha = 0,40$.

Для различных ОПФ энергетического предприятия установлена разная норма амортизации: от

0,4% на высотные здания до 20% на вентиляционные системы. Для «Иркутскэнерго» средняя норма амортизации $\chi = 2,5\%$.

Коэффициент непроизводительного потребления составил $c = 1295,01$.

Мальтузианский параметр, т.е. скорость роста (убывания) численности персонала получен с помощью линейного оценивания параметров. Он равен $l = 1,159$, что означает стремление численности к устойчивому равновесию.

Емкость среды выражается в единицах численности и носит системный характер. После анализа данных об изменении числа сотрудников ОАО «Иркутскэнерго» получено значение $L = 8000$.

Параметры производственной функции получены с использованием метода Левенберга-Марквардта и метода сопряженных градиентов: $a = 0,03$, $\lambda = 0,99$, $k = 0,79$, $\rho = 0,19$, $\beta = 0,276$.

Начальные значения для расчетов взяты из годовой отчетности: $x_1(0) = 32\,956\,118$ тыс.р., $x_2(0) = 2\,645\,999$ тыс. р., $x_3(0) = 93\,291\,660$ тыс. р., $x_4(0) = 8072$ чел.

Теперь найдем оптимальную ставку налога на прибыль для $10\% \leq v(t) \leq 35\%$ и $t \in [0,6]$ с помощью комплекса программ OPTCON [11]. Для каждого из приведенных в первом разделе класса задач в пакете OPTCON программно реализовано несколько методов оптимизации. При расчете оптимального управления в каждой конкретной задаче автоматически используется своя многометодная схема оптимизации. Рассмотрим динамику показателей деятельности ОАО «Иркутскэнерго» для полученной оптимальной ставки налога на прибыль (рис. 1):

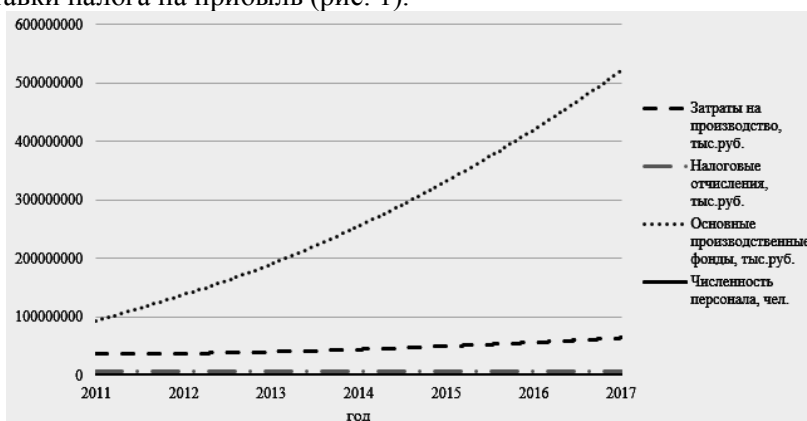


Рис. 1. Поведение модели при оптимальной ставке налога на прибыль

В ходе оптимизации ставки налогообложения прибыли предприятия было установлено, что на протяжении 6 лет наблюдается рост затрат на производство, налоговых отчислений, количества ОПФ и количества трудовых ресурсов. К окончанию рассматриваемого периода ставка налога на прибыль снижается с 35 до 20% (рис. 2), а налоговые отчисления становятся равными 7 979 170 тыс. р., вместо первоначальных 7 965 270 тыс. р.

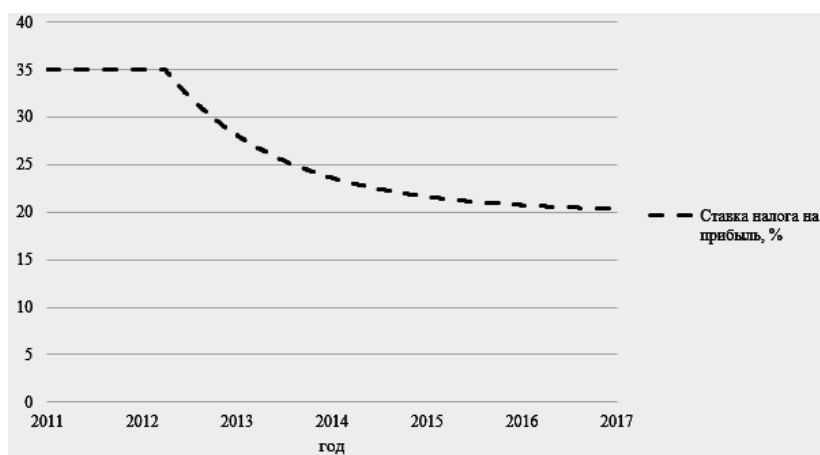


Рис. 2. Оптимальная ставка налога на прибыль предприятия

После введения оптимальной ставки налога на прибыль к концу исследуемого периода (т.е. к концу 2017 г.) затраты на производство уменьшились на 0,38%, налоговые отчисления увеличились на 0,07%, основные производственные фонды уменьшились на 5,8%, а численность персонала осталась без изменений. Такие незначительные изменения для налоговых отчислений заставляют задуматься: а есть ли смысл изменять ставку налога на прибыль предприятий, чтобы увеличить поступления в бюджет на 5310 тыс. р.? Скорее всего, нет.

Заключение

В ходе исследования и оптимизации ставки налогообложения прибыли предприятия были найдены значения параметров математической системы, моделирующей деятельность ОАО «Иркутскэнерго», ставки налога на прибыль и таких показателей, как затраты на производство, налоговые отчисления, количество основных производственных фондов, количество трудовых ресурсов.

Не стоит забывать, что трудно рассчитывать на то, что можно теоретически обосновать идеальную шкалу налогообложения. Немаловажное значение в оценке ее справедливости имеют национальные, психологические и культурные факторы. В данном случае, когда в результатах расчетов выяснено, что ставку нужно сначала поднять до максимального уровня в 35%, а затем опять вернуться к 20%, получив при этом относительно небольшие отчисления в государственный бюджет, следует оставить текущее значение ставки налога на прибыль в 20%.

Таким образом, можно сделать вывод, что для каждой задачи существует своя последовательность шагов, состоящая из итераций разных методов оптимального управления [3-6], которая адекватно учитывает особенности решаемой задачи и тем самым обеспечивает наиболее эффективный поиск оптимального управления. В многометодных алгоритмах построение такой последовательности выполняется автоматически по некоторому заданному критерию, оценивающему эффективность процесса оптимизации на всех этапах решения задачи.

На практике многометодная технология существенно повышает надежность получения правильных результатов при расчете оптимального управления реальными объектами, что в значительной мере обеспечивает эффективность и безопасность их движения по оптимальной траектории.

Литература

1. Влияние налогообложения на инвестиции [Электронный ресурс]: URL: www.investmarket.ru/Reports/show.asp?id=87.
2. НК РФ.
3. Васильев О.В., Тятюшкин А.И. Об одном методе решения задач оптимального управления, основанном на принципе максимума // Журн. вычисл. матем. и мат. физики. – Т.21, №.6. – 1981. – С. 1376-1384.
4. Тятюшкин А.И. Параллельные вычисления в задачах оптимального управления // Сиб. журн. выч. матем. / РАН. Сиб. отд-ние. – Новосибирск, 2000. – Т. 3, №. 2. – С. 181-190.
5. Тятюшкин А.И. Численные методы решения задач оптимального управления с ограничениями

на фазовые координаты // Изв. РАН. Теория и системы управления. – 1998. – №. 2. – С. 127-133.

6. Тятюшкин А.И. Многометодная технология оптимизации управляемых систем. Новосибирск: Наука, 2006. – 343 с.

7. Ризниченко Г.Ю., Рубин А.Б. Математические модели биологических продукционных процессов. – М.: Изд-во МГУ, 1993. – 301 с.

8. Москаленко А.И. Оптимальное управление моделями экономической динамики. – Новосибирск: Наука, Сибирское предприятие РАН, 1999.

9. Гуяев М.В. Эконометрика. Конспект лекций. Керченский государственный морской технологический университет. – Керчь, 2012. – 68 с.

10. Раскрытие информации ОАО «Иркутскэнерго» [Электронный ресурс]: URL: <http://www.irkutskenergo.ru/qa/reports.html>

11. Горнов А.Ю. Комплекс программ OPTCON для решения прикладных задач оптимального управления: Информационные и вычислительные технологии и системы / материалы Всеросс. конф. – Улан-Удэ, 2003. – Ч.1. – С.112-115.

Колмакова Анастасия Ивановна, аспирант Байкальского государственного университета экономики и права, e-mail: kolmakova.a.i@mail.ru.

Kolmakova Anastasiya Ivanovna, postgraduate student, Baikal State University of Economics and Law, e-mail: kolmakova.a.i@mail.ru.

УДК 519.853+519.863

© Д.С. Медвежонков

ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ АЛГОРИТМОВ ВНУТРЕННИХ ТОЧЕК НА НЕЛИНЕЙНЫХ ЗАДАЧАХ ПОТОКОРАСПРЕДЕЛЕНИЯ

Приводятся результаты экспериментальных исследований прямых и двойственных алгоритмов внутренних точек с различными способами задания весовых коэффициентов на нелинейных задачах потокораспределения. В экспериментах выявлено преимущество линейных весовых коэффициентов, деленных на множители Лагранжа, перед квадратичными. Установлено, что при использовании двойственного алгоритма требуемая точность решения достигается быстрее, чем при использовании прямого алгоритма.

Ключевые слова: прямые и двойственные алгоритмы внутренних точек, симметричная двойственность, весовые коэффициенты, задачи потокораспределения.

© D.S. Medvezhonkov

EXPERIMENTAL RESEARCHES OF INTERIOR-POINT ALGORITHMS FOR SOLUTION OF FLOW DISTRIBUTION NONLINEAR PROBLEMS

The results of experimental researches of direct and dual interior point algorithms with different methods of specifying weight coefficients in nonlinear problems of flow distribution are given. The experiments have revealed the advantage of linear weight coefficients, divided by the Lagrange multipliers, in comparison with the quadratic ones. It has been found out that the required accuracy of solution is attained faster when using the dual algorithm, than when using direct algorithm.

Keywords: direct and dual interior point algorithms, symmetric duality, weight coefficients, flow distribution problems.

Введение

Для описания функционирования различных технических и экономических транспортных систем, например, систем водо-, нефте-, газо-, электроснабжения, используют модели потокораспределения [1, 4, 7, 9, 10, 12]. При исследовании свойств этих моделей важную роль играет теория двойственности в оптимизации. Двойственные задачи применяются, в частности, для интерпретации моделей, для теоретического обоснования и разработки новых алгоритмов решения задач математического программирования.

В работах [9, 10, 12] формулируется двойственность для класса задач оптимизации с выпуклой сепарабельной целевой функцией и линейными ограничениями на базе преобразования Лежандра-Фенхеля. Для данного класса задач двойственная задача к двойственной задаче оптимизации совпадает с исходной. Такой случай назван симметричной двойственностью.

К эффективным методам решения задач оптимизации с ограничениями-неравенствами относятся алгоритмы метода внутренних точек. Приводятся результаты численных экспериментальных исследований алгоритмов внутренних точек специального типа. Пионерами в разработке алгоритмов этого типа были отечественные ученые С.М. Анциз, И.И. Дикин, Ю.Г. Евтушенко, В.Г. Жадан, В.И. Зоркальцев. Метод, который лежит в основе исследуемых здесь вариантов прямых и двойственных алгоритмов внутренних точек, предложен в [3].

В работах [5, 6, 11] обсуждаются правила задания весовых коэффициентов функций штрафа, при которых можно доказать сходимость алгоритмов внутренних точек. Наиболее известными являются квадратичные весовые коэффициенты. Алгоритм Дикина [2] использует данное правило задания весовых коэффициентов. Недостатком квадратичных весовых коэффициентов является то, что они очень чувствительны к неизбежным погрешностям в решении вспомогательной задачи.

В [5, 6] был введен способ определения весовых коэффициентов с использованием множителей Лагранжа, вычисляемых на предыдущей итерации. Алгоритмы с такими весовыми коэффициентами более устойчивы к погрешностям решения вспомогательной задачи. В работах [5, 6, 8] была экспе-

риментально подтверждена эффективность таких коэффициентов при решении задач линейного программирования.

Имеет место недостаток сравнительных экспериментальных исследований вариантов алгоритмов внутренних точек на задачах нелинейной оптимизации. Работа посвящена устранению этого пробела. Кроме выявления эффективных способов выбора весовых коэффициентов исследуются двойственные алгоритмы, которые могут обладать преимуществом в скорости перед прямыми алгоритмами, в них сразу со стартовой точки начинается процесс оптимизации в области допустимых решений.

1. Взаимно-двойственные задачи оптимизации на базе преобразования Лежандра-Фенхеля

Пусть задана матрица A размера $m \times n$ и множество индексов $J = \{1, \dots, n\}$. Множества L, H являются некоторыми подмножествами J . Также заданы: вектор $b \in R^m$, вектор $s \in R^n$, величины $\underline{x}_j, j \in L$ и $\bar{x}_j, j \in H$, при этом $\underline{x}_j < \bar{x}_j, j \in L \cap H$.

Обозначим Z множество функций из R в R , которые равны нулю в нуле и имеют непрерывные возрастающие производные. Дополнительно предполагаем, что производные этих функций равны нулю в нуле и изменяются от $-\infty$ до $+\infty$ при изменении аргумента соответственно от $-\infty$ до $+\infty$.

Пусть задан набор функций $F_j(x_j), j \in J$, принадлежащих множеству Z . Введем обозначение: $F(x) = \sum_{j \in J} F_j(x_j)$. Рассмотрим задачу выпуклой оптимизации с выпуклой сепарабельной целевой функцией и линейными ограничениями, переменными которой являются компоненты вектора $x \in R^n$:

$$F(x) + s^T x \rightarrow \min, \quad (1)$$

$$Ax = b, \quad (2)$$

$$\underline{x}_j \leq x_j, \quad j \in L, \quad (3)$$

$$x_j \leq \bar{x}_j, \quad j \in H. \quad (4)$$

Назовем задачу (1)–(4) *исходной задачей* оптимизации. *Расширенным решением* исходной задачи назовем набор, состоящий из вектора исходных переменных x , вектора $u \in R^m$ множителей Лагранжа ограничений (2), векторов $l \in R^n$ и $h \in R^n$, содержащих множители Лагранжа ограничений (3) и (4), причем $l_j = 0, j \in J \setminus L$ и $h_j = 0, j \in J \setminus H$, а также вектора $y \in R^n$ с компонентами $y_j = f_j(x_j)$, где f_j – производная $F_j, j \in J$.

Обозначим Φ_j функцию из Z , связанную с функцией $F_j \in Z, j \in J$ преобразованием Лежандра-Фенхеля, известным из выпуклого анализа. Введем обозначение: $\Phi(y) = \sum_{j \in J} \Phi_j(y_j)$. В [12] установлено, что *двойственную* задачу оптимизации к задаче (1)–(4) можно записать в виде:

$$\Phi(y) - b^T u - \sum_{j \in L} \underline{x}_j l_j + \sum_{j \in H} \bar{x}_j h_j \rightarrow \min, \quad (5)$$

$$y + s - l + h = A^T u, \quad (6)$$

$$l_j \geq 0, \quad j \in L, \quad h_j \geq 0, \quad j \in H, \quad (7)$$

$$l_j = 0, \quad j \in J, \quad j \notin L, \quad h_j = 0, \quad j \in J, \quad j \notin H. \quad (8)$$

Здесь переменными являются векторы $y \in R^n, u \in R^m$ и компоненты векторов $l \in R^n$ и $h \in R^n$ соответственно при $j \in L$ и $j \in H$.

Расширенным решением двойственной задачи назовем набор, состоящий из векторов ее переменных y, u, l, h , а также вектора $x \in R^n$ множителей Лагранжа ограничений (6).

Рассмотрим систему уравнений и неравенств с условиями (2)–(4), (8) и:

$$y_j = f_j(x_j), \quad j \in J, \quad (9)$$

$$y + s - l + h = A^T u, \quad (10)$$

$$l_j = (f_j(x_j) + s_j - [A^T u]_j)_+, \quad j \in J^L, \quad (11)$$

$$h_j = ([A^T u]_j - f_j(\bar{x}_j) - s_j)_+, \quad j \in J^H. \quad (12)$$

Здесь f_j – производная функции F_j , $j \in J$, а $(\alpha)_+ = \max\{0, \alpha\}$.

В [12] обосновывается справедливость утверждений следующей теоремы, которая говорит о возможности восстановить решение исходной задачи по решению двойственной. Это дает право пользоваться двойственными алгоритмами внутренних точек для решения задач потокораспределения.

Теорема. Если ограничения исходной задачи (1)–(4) совместны, то решение этой задачи существует и единственно, решение двойственной задачи (5)–(8) существует и единственно по вектору u , решение системы уравнений и неравенств (2)–(4), (8), (9)–(12) существует, единственно по векторам x , u и совпадает с расширенными решениями исходной и двойственной задач. В противном случае решения исходной и двойственной задач не существуют, система уравнений и неравенств (2)–(4), (8), (9)–(12) несовместна.

Замечание. Условия (11), (12) заменяют известные в оптимизации ограничения дополняющей нежесткости. Эти условия лучше в вычислительном отношении, чем билинейные ограничения дополняющей нежесткости.

2. Прямой и двойственный алгоритмы внутренних точек

Итерационный процесс прямых алгоритмов внутренних точек решения исходной задачи (1)–(4) заключается в последовательном построении нового приближения: $x^{k+1} = x^k + \lambda \Delta x$, где Δx – направление корректировки текущего приближения, λ – величина шага вдоль этого направления. Процесс начинается из точки x^0 , которая удовлетворяет ограничениям-неравенствам в строгой форме. В этом алгоритме выделяются два этапа вычислений. Сначала осуществляется ввод в область допустимых решений, в процессе которого уменьшаются невязки ограничений-равенств (2). На втором этапе осуществляется оптимизация в области допустимых решений.

Для двойственной задачи (5)–(8) несложно априори сформировать допустимое по ее ограничениям решение, для которого все ограничения-неравенства выполняются в строгой форме. Поэтому одним из преимуществ рассматриваемых двойственных алгоритмов внутренних точек является то, что в них со стартовой точки начинается процесс оптимизации в области допустимых решений.

Для выбора направления корректировки на каждой итерации как прямого, так и двойственного алгоритма решается вспомогательная задача минимизации выпуклой сепарабельной квадратичной функции при линейных ограничениях-равенствах. Целевая функция этой задачи содержит квадратичную аппроксимацию целевой функции решаемой задачи, а также функцию штрафа для учета ограничений-неравенств, представляемую для прямых алгоритмов в виде $\sum_{j \in J} (\Delta x_j^k)^2 / d_j^k$, где d_j^k – весо-

вые коэффициенты, меняющиеся по итерациям по заданным правилам. Рассматривались два вида весовых коэффициентов: квадратичные коэффициенты и линейные коэффициенты, деленные на приближения к множителям Лагранжа. Для прямого алгоритма в случае двусторонних ограничений-неравенств квадратичные коэффициенты представлялись в виде $d_j^k = (\min(x_j^k - \underline{x}_j, \bar{x}_j - x_j^k))^2$, а линейные – в виде

$$d_j^k = \min(x_j^k - \underline{x}_j, \bar{x}_j - x_j^k) / \max(\delta_2, l_j^{k-1}, h_j^{k-1}), \text{ где } \delta_2 \text{ – малая константа.}$$

Величина шага по направлению корректировки выбирается таким образом, чтобы новое приближение всегда находилось внутри области, задаваемой ограничениями-неравенствами. Для этого вычислялась величина $\lambda = \min\{\gamma \lambda_1, \lambda_2\}$, где γ – константа из интервала $(0, 1)$, λ_1 – величина шага вдоль направления корректировки до ближайшей границы ограничений-неравенств, λ_2 – величина шага вдоль направления корректировки до минимума оптимизируемой целевой функции.

3. Экспериментальные исследования алгоритмов внутренних точек

Была выполнена программная реализация прямого и двойственного алгоритмов внутренних точек на языке C++. Проведены экспериментальные исследования четырех вариантов прямого и двойственного алгоритмов внутренних точек на задачах потокораспределения. Использовались два вида весовых коэффициентов: квадратичные коэффициенты и линейные коэффициенты, деленные на при-

ближения к множителям Лагранжа. В таблице приводятся характеристики решенных в ходе эксперимента задач и результаты расчетов для них. Критерием остановки алгоритмов было выполнение условий (2)–(4), (8), (9)–(12) с точностью 0,1.

Таблица
Результаты расчетов задач потокораспределения с использованием четырех вариантов алгоритмов внутренних точек

Характеристики задач					Кол-во итераций для вар-тов алгоритма			
узлов	ветвей	решен. задач	Среднее число		Прямой	Двойств.	Прямой	Двойств.
			двустор. огранич.	активн. огранич.	Квадратичные весовые коэффиц.		Линейные весовые коэффиценты	
25	39	2	29,6	5,2	43,5	44,4	30,2	17,9
25	48	2	40,0	6,6	52,9	33,2	30,0	17,3
50	60	2	35,4	7,5	62,5	37,9	52,3	23,5
50	136	2	93,8	27,2	90,9	46,6	25,0	25,5
100	116	2	40,2	8,4	66,9	24,2	31,7	23,0
100	195	2	84,3	23,6	53,5	83,7	32,0	26,5
200	300	2	150,0	16,1	82,3	34,8	31,8	20,3
338	712	2	500,0	13,3	101,8	82,4	32,4	26,7
среднее геометрическое:					66,7	44,4	32,5	22,3

Расчеты показывают, что прямой и двойственный алгоритмы с линейными весовыми коэффициентами в среднем в два раза быстрее (по числу итераций) своих аналогов с квадратичными коэффициентами. Двойственные алгоритмы в среднем в полтора раза быстрее своих прямых аналогов.

Заключение

На основе экспериментальных исследований вариантов прямых и двойственных алгоритмов внутренних точек для решения нелинейных задач потокораспределения выявлены наиболее эффективные способы выбора весовых коэффициентов. Показано преимущество линейных весовых коэффициентов, деленных на множители Лагранжа, над квадратичными. Установлено, что при использовании двойственного алгоритма требуемая точность решения достигается быстрее, чем при использовании прямого алгоритма.

Литература

1. Деннис Дж.Б. Математическое программирование и электрические цепи. – М.: Изд-во иностранной литературы, 1961. – 216 с.
2. Дикин И.И. Итеративное решение задач линейного программирования // Доклады АН СССР. – Т. 174. – 1967.
3. Дикин И.И., Зоркальцев В.И. Итеративное решение задач математического программирования (алгоритмы метода внутренних точек). – Новосибирск: Наука, 1980. – 144 с.
4. Rockafellar R. Tyrrell: Network flows and monotropic optimization. – Pure and Applied Mathematics. – New York: Wiley-Interscience, 1984. – 616 p.
5. Зоркальцев В.И. Методы прогнозирования и анализа эффективности функционирования системы топливоснабжения. – М.: Наука, 1988.
6. Зоркальцев В.И. Проективные алгоритмы оптимизации, использующие множители предыдущей итерации // Журнал вычислительной математики и математической физики. – 1994. – Т. 34, №7. – С. 943–950.
7. Bertsekas D.P. Network Optimization: Continuous and Discrete Models. – Athena Scientific, Belmont, Massachusetts, 1998. – 608 p.

-
8. Войтов О.Н., Зоркальцев В.И., Филатов А.Ю. Определение допустимых режимов электроэнергетических систем алгоритмами внутренних точек // Сибирский журнал индустриальной математики, Т. 3, №1(5), 2000. – С. 57-65.
 9. Зоркальцев В.И. Симметричная двойственность в оптимизации и ее приложения // Известия высших учебных заведений. Математика, 2006. – №2. – С. 53-59.
 10. Зоркальцев В.И., Хамисов О.В. Равновесные модели в экономике и энергетике. – Новосибирск: Наука, 2006. – 221 с.
 11. Зоркальцев В.И. Класс алгоритмов внутренних точек // Журнал вычислительной математики и математической физики. – 2009. – № 12. – С. 3-28.
 12. Епифанов С.П., Зоркальцев В.И., Медвежонков Д.С. Модель гидравлической сети с регуляторами расхода // Управление большими системами. Спец. вып. 30.1 «Сетевые модели в управлении». – М.: Изд-во ИПУ РАН, 2010. – С.286-299.

Медвежонков Дмитрий Сергеевич, младший научный сотрудник, Институт систем энергетики им. Л.А. Мелентьева СО РАН, тел. (3952) 500646 доп. 255, e-mail: dmitry@isem.sei.irk.ru

Medvezhonkov Dmitry Sergeevich, junior researcher, Melentiev Energy Systems Institute SB RAS, ph. (3952) 500646 (255), e-mail: dmitry@isem.sei.irk.ru

УДК 517.98

© А.Д. Мижидон, К.А. Мижидон

ЗАДАЧА ОПТИМАЛЬНОГО УПРАВЛЕНИЯ ЛИНЕЙНОЙ СИСТЕМОЙ ПРИ ФАЗОВЫХ И СМЕШАННЫХ ОГРАНИЧЕНИЯХ¹

Статья посвящена построению алгоритмического обеспечения решения задачи управления линейной динамической системой, в которой основной целью управления является удержание системы в фазовых и смешанных ограничениях при заданном детерминированном возмущении при минимуме энергетических затрат.

Ключевые слова: линейная система, оптимальное управление, фазовые ограничения, смешанные ограничения, фундаментальная матрица, матричный ряд.

© A.D. Mizhidon, K.A. Mizhidon

THE PROBLEM OF OPTIMAL CONTROL FOR LINEAR SYSTEM WITH STATE AND MIXED CONSTRAINTS

The article is devoted to the construction of algorithmic support of solution the problem of control for linear dynamic system which main purpose is to keep the system in the state and mixed control-state constraints under deterministic given perturbation with a minimum expenditure of energy.

Keywords: linear system, optimal control, state constraints, mixed constraints, fundamental matrix, matrix series.

Введение

В статье рассматривается задача управления линейной динамической системой, в которой основной целью управления является удержание системы в фазовых и смешанных ограничениях при заданном детерминированном возмущении при минимуме энергетических затрат. Для ее решения рассматривается вспомогательная задача оптимального управления с квадратичным критерием качества, матрицы которого зависят от некоторых весовых коэффициентов, выбор которых в конечном итоге обеспечивает при оптимальном управлении выполнение фазовых ограничений. Предложен подход к построению алгоритмического обеспечения решения задачи, основанный на представлении фундаментальной матрицы системы в виде матричной экспоненты, разложенной в матричный ряд.

1. Постановка задачи

Рассмотрим линейную управляемую систему с постоянными коэффициентами

$$\dot{x} = Ax(t) + Bu(t) + f(t), \quad x(0) = x^0, \quad (1)$$

где x – n -мерный вектор фазовых координат системы; u – r -мерный вектор управления; A – постоянная $(n \times n)$ -матрица; B – постоянная $(n \times r)$ -матрица, $f(t)$ – n -мерный вектор внешних возмущений. Вектор-функция $f(t)$ является ограниченной при $t \geq 0$.

Система управляется на временном отрезке $0, T$. При этом допустимыми управлениями $u(t)$ являются непрерывно дифференцируемые функции.

Нормальное функционирование системы предполагает выполнение в каждый момент времени $t \in 0, T$ фазовых ограничений

$$x'(t)Q_i x(t) \leq 1, \quad (i = 1, 2, \dots, s) \quad (2)$$

и смешанных ограничений

$$x'(t)C_j x(t) + x'(t)P_j u(t) + u'(t)D_j u(t) \leq 1, \quad (j = 1, 2, \dots, p), \quad (3)$$

¹ Исследование выполнено при финансовой поддержке РФФИ в рамках научных проектов (№ 12-08-00309 а, № 12-01-00914-а)

где P_j – матрицы размерности $(n \times r)$; Q_i, C_j – неотрицательно-определенные, симметричные $(n \times n)$ -матрицы; D_j – неотрицательно-определенные, симметричные $(r \times r)$ -матрицы; штрих $()'$ – здесь и ниже операция транспонирования. Матрицы P_j, C_j, D_j таковы, что матрицы G_j , составленные из них

$$G_j = \begin{pmatrix} C_j & \frac{1}{2}P_j \\ \frac{1}{2}P_j' & D_j \end{pmatrix},$$

положительно-определенные матрицы.

Требуется при заданном возмущении $f(t)$ найти управление $u^*(t)$, при котором на траекториях системы (1) выполняются в каждый момент времени $t \in 0, T$ фазовые (2) и смешанные ограничения (3) и обеспечивается минимум функционала

$$J(u) = \int_0^T u'(t)Ru(t)dt, \tag{4}$$

где R – положительно-определенная, симметричная $r \times r$ -матрица.

Функционал (4) характеризует энергетические затраты.

Отметим, необходимость рассмотрения поставленной задачи возникла в связи с исследованиями, связанными с разработкой алгоритмического обеспечения решения задач автоматизации проектирования виброзащитных систем [1].

2. Вспомогательная задача

Рассмотрим задачу оптимального управления:

$$\begin{cases} \dot{x} = Ax(t) + Bu(t) + f(t), & x(0) = x^0, \\ x(t), f(t) \in E^n, & u(t) \in E^r, t \in [0, T], \\ J(u(\cdot)) = \frac{1}{2} \int_0^T (x'Q(\alpha, \beta)x + x'P(\beta)u + u'R(\beta)u)dt \rightarrow \min, \end{cases} \tag{5}$$

где $Q(\alpha, \beta)$, $P(\beta)$ и $R(\beta)$ – матрицы соответственно размерности $(n \times n)$, $(n \times r)$ и $(r \times r)$, определяемые следующим образом

$$Q(\alpha, \beta) = \alpha_1 Q_1 + \alpha_2 Q_2 + \dots + \alpha_s Q_s + \beta_1 C_1 + \beta_2 C_2 + \dots + \beta_p C_p, \\ P(\beta) = \beta_1 P_1 + \beta_2 P_2 + \dots + \beta_p P_p, \quad R(\beta) = R + \beta_1 D_1 + \beta_2 D_2 + \dots + \beta_p D_p.$$

Здесь $\alpha_i \geq 0$ и $\beta_i \geq 0$ – весовые коэффициенты.

Отметим, что рассматриваемая задача как задача аналитического конструирования оптимального регулятора детально была исследована в [2].

В задаче оптимального управления (5) в силу того, что на управление не наложены ограничения, удается выразить из условия максимума

$$\frac{\partial H}{\partial u} = B'\psi - \frac{1}{2}P'(\beta)x - R(\beta)u = 0$$

функции Понтрягина

$$H = \psi'(Ax + Bu + f(t)) - \frac{1}{2} x'Q(\alpha, \beta)x + x'P(\beta)u + u'R(\beta)u$$

управление:

$$\hat{u} = R^{-1}(\beta) \left[B'\psi - \frac{1}{2}P'(\beta)x \right]. \tag{6}$$

Подставив найденное управление (6) в исходную систему (1) и в сопряженную систему

$$\dot{\psi} = -A'\psi + Q(\alpha, \beta)x + \frac{1}{2}P(\beta)u, \quad \psi(T) = 0,$$

объединив их, получим краевую задачу принципа максимума:

$$\begin{cases} \dot{x} = Ax + BR^{-1} \left[B'\psi - \frac{1}{2}P'(\beta)x \right] + f(t), & x(0) = x^0, \\ \dot{\psi} = -A'\psi + Q(\alpha, \beta)x + \frac{1}{2}P(\beta)R^{-1}(\beta) \left[B'\psi - \frac{1}{2}P'(\beta)x \right], & \psi(T) = 0. \end{cases} \quad (7)$$

Таким образом, если при некоторых весовых коэффициентах α, β найдено решение краевой задачи (7), то оптимальное управление и оптимальную фазовую траекторию соответственно можно представить в виде

$$\begin{cases} u^*(\alpha, \beta, t) = R^{-1}(\beta) \left[B'\psi(\alpha, \beta, t) - \frac{1}{2}P'(\beta)x(\alpha, \beta, t) \right], \\ x^*(\alpha, \beta, t) = x(\alpha, \beta, t), \end{cases} \quad (8)$$

где $x(\alpha, \beta, t)$ и $\psi(\alpha, \beta, t)$ решение краевой задачи (7).

В общем случае для решения краевых задач не существует общих методов решения. Наиболее полно исследована краевая задача для систем линейных дифференциальных уравнений (в частности, метод Абрамова), к которым относится система (7). Но тем не менее представляет интерес разработка новых эффективных методов решения линейных краевых задач. В нашем случае интерес определяется не только возможностью эффективного решения с их помощью краевых задач, но и их проблемной ориентированностью на применение при разработке алгоритма решения задачи выбора весовых коэффициентов α, β , обеспечивающих при оптимальном управлении $u^*(\alpha, \beta, t)$ выполнение ограничений (2) и (3) и при этом обеспечивающих минимум функционалу (4).

3. Краевая задача

Рассмотрим краевую задачу

$$\begin{cases} \dot{x} = A_{11}x + A_{12}\psi + f(t), & x(0) = x^0, \\ \dot{\psi} = A_{21}x + A_{22}\psi, & \psi(T) = 0, \end{cases} \quad (9)$$

где матрицы $A_{11}, A_{12}, A_{21}, A_{22}$ определяются следующим образом

$$A_{11} = A - \frac{1}{2}BR^{-1}P'(\beta), \quad A_{12} = BR^{-1}B',$$

$$A_{21} = Q(\alpha, \beta) - \frac{1}{4}P(\beta)R^{-1}(\beta)P'(\beta), \quad A_{22} = -A' + \frac{1}{2}P(\beta)R^{-1}B'.$$

Введя обозначения:

$$\tilde{A} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad y = \begin{pmatrix} x \\ \psi \end{pmatrix}, \quad \tilde{f}(t) = \begin{pmatrix} f(t) \\ 0 \end{pmatrix},$$

запишем систему (9) в виде

$$\dot{y} = \tilde{A}y + \tilde{f}(t). \quad (10)$$

Фундаментальную матрицу $F(t, \tau)$ однородной системы, соответствующую системе (10), представим в виде блоков

$$F(t, \tau) = \begin{pmatrix} F_{11}(t, \tau) & F_{12}(t, \tau) \\ F_{21}(t, \tau) & F_{22}(t, \tau) \end{pmatrix}.$$

Решение задачи Коши системы (10) при начальном условии $x(0) = x^0$ и при некотором $\psi(0) = \psi^0$ согласно формуле Коши запишется в виде

$$\begin{cases} x(t) = F_{11}(t,0)x^0 + F_{12}(t,0)\psi^0 + \int_0^t F_{11}(t,\tau)f(\tau)d\tau, \\ \psi(t) = F_{21}(t,0)x^0 + F_{22}(t,0)\psi^0 + \int_0^t F_{21}(t,\tau)f(\tau)d\tau. \end{cases} \quad (11)$$

Используя условие на правом конце $\psi(T) = 0$, из второго выражения из (11) получим

$$\psi(T) = F_{21}(T,0)x^0 + F_{22}(T,0)\psi^0 + \int_0^T F_{21}(T,\tau)f(\tau)d\tau = 0.$$

Отсюда, начальное условие ψ^0 можем определить в виде

$$\psi^0 = -F_{22}^{-1}(T,0) \left[F_{21}(T,0)x^0 + \int_0^T F_{21}(T,\tau)f(\tau)d\tau \right]. \quad (12)$$

Таким образом, при некоторых весовых коэффициентах α, β , можем записать решение краевой задачи (9) в виде

$$\begin{cases} x(\alpha, \beta, t) = F_{11}(t,0)x^0 + \int_0^t F_{11}(t,\tau)f(\tau)d\tau + F_{12}(t,0)\psi^0, \\ \psi(\alpha, \beta, t) = F_{21}(t,0)x^0 + \int_0^t F_{21}(t,\tau)f(\tau)d\tau + F_{22}(t,0)\psi^0, \end{cases} \quad (13)$$

где ψ^0 определяется выражением (12).

Для нахождения начального значения ψ^0 сопряженной переменной согласно (12), а также решения краевой задачи $x(\alpha, \beta, t)$ и $\psi(\alpha, \beta, t)$ в соответствии с (13) возникает необходимость вычисления фундаментальной матрицы $F(t, \tau)$. Для этого воспользуемся представлением фундаментальной матрицы в виде матричной экспоненты $F(t, \tau) = e^{\tilde{A}(t-\tau)}$, которая в свою очередь представима в виде матричного ряда

$$F(t, \tau) = e^{\tilde{A}(t-\tau)} = E + \tilde{A}(t-\tau) + \frac{1}{2!} \tilde{A}^2(t-\tau)^2 + \dots + \frac{1}{k!} \tilde{A}^k(t-\tau)^k + \dots \quad (14)$$

Представление фундаментальной матрицы в виде матричного ряда (14) можно использовать для построения алгоритмов вычисления фундаментальной матрицы для любой линейной системы дифференциальных уравнений с постоянными коэффициентами [3].

В соответствии с представлением матрицы \tilde{A} из (14) можем получить соотношения для вычисления блоков матрицы $F(t, \tau)$:

$$\begin{aligned} F_{11}(t, \tau) &= E + \sum_{k=0}^{\infty} \frac{S_{11}^k(t-\tau)^{k+1}}{(k+1)!}, & F_{12}(t, \tau) &= \sum_{k=0}^{\infty} \frac{S_{12}^k(t-\tau)^{k+1}}{(k+1)!}, \\ F_{21}(t, \tau) &= \sum_{k=0}^{\infty} \frac{S_{21}^k(t-\tau)^{k+1}}{(k+1)!}, & F_{22}(t, \tau) &= E + \sum_{k=0}^{\infty} \frac{S_{22}^k(t-\tau)^{k+1}}{(k+1)!}, \end{aligned} \quad (15)$$

где $(n \times n)$ -матрицы $S_{11}^k, S_{12}^k, S_{21}^k, S_{22}^k$ определяются рекуррентными соотношениями

$$\begin{cases} \begin{pmatrix} S_{11}^0 & S_{12}^0 \\ S_{21}^0 & S_{22}^0 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \\ \begin{pmatrix} S_{11}^{k+1} & S_{12}^{k+1} \\ S_{21}^{k+1} & S_{22}^{k+1} \end{pmatrix} = \begin{pmatrix} S_{11}^k A_{11} + S_{12}^k A_{21} & S_{11}^k A_{12} + S_{12}^k A_{22} \\ S_{21}^k A_{11} + S_{22}^k A_{21} & S_{21}^k A_{12} + S_{22}^k A_{22} \end{pmatrix}. \end{cases} \quad (k=0,1,2,\dots). \quad (16)$$

Для приближенного вычисления блоков матрицы $F(t, \tau)$ согласно (16) суммы ряда будем ограничивать конечным числом членов ряда, когда добавление нового члена ряда изменяет каждый элемент частичной суммы меньше, чем на заданную точность.

4. Задача поиска оптимальных весовых коэффициентов

Рассмотрим задачу поиска весовых коэффициентов $\alpha^* > 0$, $\beta^* > 0$, обеспечивающих выполнение неравенств

$$\begin{cases} x^{*'}(\alpha, \beta, t)Q_i x^*(\alpha, \beta, t) \leq 1, \\ x^{*'}(\alpha, \beta, t)C_j x^*(\alpha, \beta, t) + x^{*'}(\alpha, \beta, t)P_j u^*(\alpha, \beta, t) + u^{*'}(\alpha, \beta, t)D_j u^*(\alpha, \beta, t) \leq 1, \\ (i=1, \dots, s), \quad (j=1, 2, \dots, p), \quad t \in 0, T, \end{cases} \quad (17)$$

и доставляющих минимум функции

$$\varphi(\alpha, \beta) = J(u^*(\alpha, \beta, \cdot)) = \int_0^T u^{*'}(\alpha, \beta, t) R u^*(\alpha, \beta, t) dt, \quad (18)$$

где $x^*(\alpha, \beta, t)$ и $u^*(\alpha, \beta, t)$ соответственно оптимальная траектория и оптимальное управление во вспомогательной задаче (5), определяемые при некоторых весовых коэффициентах α, β соотношениями (8).

Для решения этой задачи предлагается, разбив отрезок $0, T$ точками $t_0 = 0, t_1, \dots, t_{N-1}, t_N = T$, рассмотреть следующую аппроксимирующую задачу: найти весовые коэффициенты $\alpha^* > 0$, $\beta^* > 0$, доставляющие минимум функции (18), на множестве, определяемом неравенствами

$$\begin{cases} x^{*'}(t_k, \alpha, \beta)Q_i x^*(t_k, \alpha, \beta) \leq 1, \\ x^{*'}(\alpha, \beta, t_k)C_j x^*(\alpha, \beta, t_k) + x^{*'}(\alpha, \beta, t_k)P_j u^*(\alpha, \beta, t_k) + u^{*'}(\alpha, \beta, t_k)D_j u^*(\alpha, \beta, t_k) \leq 1, \\ (i=1, \dots, s), \quad (j=1, 2, \dots, p), \end{cases} \quad (19)$$

где $k = 1, 2, \dots, N$.

Замечание 1. При построении конкретных алгоритмов при вычислении целевой функции (18) возможно представление интеграла в виде простейших кубатурных формул (прямоугольника, трапеций, Симпсона), соответствующих приведенному выше разбиению отрезка $0, T$. Для решения полученной задачи математического программирования разрабатываются специальные проблемно-ориентированные итерационные процедуры, учитывающие специфику конкретных возмущений $f(t)$.

При реализации итерационных процедур возникает необходимость выбора начального приближения на множестве допустимых весовых коэффициентов. Здесь допустимыми весовыми коэффициентами называем коэффициенты α, β , удовлетворяющие неравенствам (17).

Пусть найдено при некоторых весовых коэффициентах α^0, β^0 решение краевой задачи (7) $x(\alpha^0, \beta^0, t)$ и $\psi(\alpha^0, \beta^0, t)$, и в соответствии с (8) решение вспомогательной задачи оптимального управления (5). Если при этом оптимальное управление $u^*(\alpha^0, \beta^0, t)$ и оптимальная траектория $x^*(\alpha^0, \beta^0, t)$ соответственно удовлетворяют ограничениям (2) и (3), то α^0, β^0 можно взять в качестве начального приближения. В противном случае рассмотрим следующую задачу выбора весовых коэффициентов, обеспечивающих выполнение неравенств

$$\varphi_l(\gamma) \leq 1, \quad l=1, 2, \dots, (s+p)N, \quad (20)$$

где $\varphi_l(\gamma)$ – функция задающая l -е неравенство в (19); $\gamma = (\alpha, \beta)$.

Пусть при некотором $\gamma^0 = (\alpha^0, \beta^0)$ некоторые из неравенств (19) нарушаются. И при этом выполняется $\varphi_l(\gamma^0) < 1 - \delta$ при $l \in I$ и $\varphi_l(\gamma^0) \geq 1 - \delta$ при $l \in J$, где δ – малое положительное число; I и J –

множества индексов, при которых соответствующие ограничения выполняются или не выполняются. Очевидно $I \cup J = 1, 2, \dots, (s + p)N$.

Для поиска весовых коэффициентов, обеспечивающих выполнение неравенств (19) строится итерационный процесс попадания в область определяемую неравенствами (19). Не обсуждая деталей, имеющих место при конкретной реализации, приведем процедуру реализации одного шага итерационного процесса.

1. Направление спуска h построим следующим образом

$$h = - \sum_{l \in J} \lambda_l \nabla \varphi_l(\gamma^0), \tag{21}$$

где $\nabla \varphi_l(\gamma^0)$ – градиент функции $\varphi_l(\gamma)$, вычисленный в точке γ^0 ; λ_l – неотрицательные числа. Коэффициенты λ_l выберем так, чтобы выполнялись неравенства

$$\begin{aligned} \nabla \varphi_i(\gamma^0), h &= - \left(\nabla \varphi_i(\gamma^0), \sum_{l \in J} \lambda_l \nabla \varphi_l(\gamma^0) \right) = \\ &= \sum_{l \in J} \lambda_l \nabla \varphi_i(\gamma^0), \nabla \varphi_l(\gamma^0) < 0 \end{aligned} \tag{22}$$

при всех индексах i , при которых нарушаются неравенства (19).

Отметим, что в этом случае направление спуска (21) определяет направление убывания функций в точке γ^0 всех функций, при которых нарушились неравенства (19).

2. Следующую точку итерационной последовательности определим, сделав шаг в направлении вектора h

$$\gamma^1 = \gamma^0 + \chi h.$$

При этом шаг $\chi > 0$ выберем так, чтобы выполнилось условие

$$\max_{l \in J} \varphi_l(\gamma^1) < \max_{l \in J} \varphi_l(\gamma^0)$$

при выполнении условий

$$\varphi_l(\gamma^0) < 1 \quad \text{при всех } l \in I.$$

5. К вычислению частных производных по весовым коэффициентам от решений краевой задачи

Отметим, что при реализации итерационных процедур возникает необходимость вычисления частных производных по весовым коэффициентам γ_i от функций $x(\gamma, t)$ и $\psi(\gamma, t)$ решений краевой задачи (9). В соответствии с представлением (13) решение краевой задачи (9), учитывая, что фундаментальная матрица зависит от выбора весовых коэффициентов, запишем в виде

$$y(\gamma, t) = F(\gamma, t, 0)y^0 + \int_0^t F(\gamma, t, \tau) \tilde{f}(\tau) d\tau, \tag{23}$$

где

$$y(\gamma, t) = \begin{pmatrix} x(\gamma, t) \\ \psi(\gamma, t) \end{pmatrix}, \quad y^0 = \begin{pmatrix} x^0 \\ \psi^0 \end{pmatrix}, \quad F(\gamma, t, \tau) = \begin{pmatrix} F_{11}(\gamma, t, \tau) & F_{12}(\gamma, t, \tau) \\ F_{21}(\gamma, t, \tau) & F_{22}(\gamma, t, \tau) \end{pmatrix}.$$

В силу используемого представления (23) при разработке алгоритмического обеспечения, реализующего процедуру выбора весовых коэффициентов, вычисление частных производных функций $y(\gamma, t)$ по весовым коэффициентам γ_i сводится к нахождению частных производных фундаментальной матрицы $F(\gamma, t, \tau)$ по весовым коэффициентам γ_i .

Для вычисления фундаментальной матрицы выше было использовано ее представление в виде матричного ряда

$$F(\gamma, t, \tau) = e^{\tilde{A}(\gamma)(t-\tau)} = E + \sum_{k=1}^{\infty} \frac{1}{k!} \tilde{A}^k(\gamma)(t-\tau)^k. \tag{24}$$

Ряд (24) является при фиксированных значениях t и τ функциональным матричным рядом переменной γ , сходящимся при любом значении γ . Продифференцировав по γ_i данный ряд, получим матричный ряд, составленный из производных

$$\sum_{k=1}^{\infty} \frac{1}{k!} (t-\tau)^k \left[\sum_{s=0}^{k-1} \tilde{A}^s(\gamma) \frac{\partial \tilde{A}}{\partial \gamma_i}(\gamma) \tilde{A}^{k-s-1}(\gamma) \right]. \quad (25)$$

Теорема. При фиксированных значениях t и τ функциональный матричный ряд (25) переменной γ , составленный из производных ряда (24), равномерно сходится и, следовательно, для частных производных фундаментальной матрицы $F(\gamma, t, \tau)$ по весовым коэффициентам γ_i справедливо

$$\frac{\partial F(\gamma, t, \tau)}{\partial \gamma_i} = \sum_{k=1}^{\infty} \frac{1}{k!} (t-\tau)^k \left[\sum_{s=0}^{k-1} \tilde{A}^s(\gamma) \frac{\partial \tilde{A}}{\partial \gamma_i}(\gamma) \tilde{A}^{k-s-1}(\gamma) \right]. \quad (26)$$

Доказательство. Для сходимости матричных рядов достаточно, чтобы сходился ряд, составленный из норм слагаемых [4]. Рассмотрим функциональный ряд переменной γ , составленный из норм слагаемых ряда (25)

$$\sum_{k=1}^{\infty} \frac{1}{k!} |t-\tau|^k \left\| \sum_{s=0}^{k-1} \tilde{A}^s(\gamma) \frac{\partial \tilde{A}}{\partial \gamma_i}(\gamma) \tilde{A}^{k-s-1}(\gamma) \right\|. \quad (27)$$

Имеем

$$\frac{1}{k!} |t-\tau|^k \left\| \sum_{s=0}^{k-1} \tilde{A}^s(\gamma) \frac{\partial \tilde{A}}{\partial \gamma_i}(\gamma) \tilde{A}^{k-s-1}(\gamma) \right\| \leq |t-\tau|^k \left\| \frac{\partial \tilde{A}}{\partial \gamma_i}(\gamma) \right\| \frac{1}{(k-1)!} \|\tilde{A}(\gamma)\|^{k-1}.$$

Таким образом, ряд (27) мажорируется рядом

$$\sum_{k=1}^{\infty} |t-\tau|^k \left\| \frac{\partial \tilde{A}}{\partial \gamma_i}(\gamma) \right\| \frac{1}{(k-1)!} \|\tilde{A}(\gamma)\|^{k-1}.$$

Данный ряд является равномерно сходящимся рядом. Действительно, представим его в виде

$$\sum_{k=1}^{\infty} |t-\tau|^k \left\| \frac{\partial \tilde{A}}{\partial \gamma_i}(\gamma) \right\| \frac{1}{(k-1)!} \|\tilde{A}(\gamma)\|^{k-1} = |t-\tau| \left\| \frac{\partial \tilde{A}}{\partial \gamma_i}(\gamma) \right\| \left(1 + \sum_{m=1}^{\infty} \frac{\|\tilde{A}(\gamma)\|^m}{m!} |t-\tau|^m \right).$$

Выражение

$$1 + \sum_{m=1}^{\infty} \frac{\|\tilde{A}(\gamma)\|^m}{m!} |t-\tau|^m$$

является разложением в ряд экспоненты $e^{\|\tilde{A}(\gamma)\|(t-\tau)}$, который равномерно сходится.

Отсюда следует, что функциональный ряд (27) переменной γ мажорируется равномерно сходящимся рядом и, следовательно, сам равномерно сходится. Таким образом, матричный ряд (25), составленный из производных функционального матричного ряда (24), равномерно сходится, что и доказывает справедливость (26). *Теорема доказана.*

Замечание 2. При разработке алгоритмического обеспечения, реализующего процедуру выбора весовых коэффициентов для приближенного вычисления частных производных функций $y(\gamma, t)$ по весовым коэффициентам γ_i , используется при нахождении частных производных фундаментальной матрицы $F(\gamma, t, \tau)$ по весовым коэффициентам γ_i их представление в виде матричного ряда (26).

Заключение

В работе предложен подход к построению алгоритмического обеспечения решения задачи управления линейной динамической системой, в которой основной целью управления является удержание системы в фазовых и смешанных ограничениях при заданном детерминированном возмущении при минимуме энергетических затрат. Для ее решения предлагается рассмотреть вспомогательную задачу оптимального управления с квадратичным критерием качества, матрицы которого зависят от некото-

рых весовых коэффициентов, выбором которых в конечном итоге обеспечивается при оптимальном управлении выполнение фазовых и смешанных ограничений при минимуме исходного критерия качества. Построение алгоритмического обеспечения решения краевой задачи принципа максимума, возникающей при решении вспомогательной задачи, и итерационной процедуры выбора оптимальных весовых коэффициентов, основано на представлении фундаментальной матрицы в виде экспоненты матрицы системы, разложенной в матричный ряд.

Литература

1. Мижидон А.Д., Елтошкина Е.В., Имыхелова М.Б. Типовые задачи автоматизации проектирования и их алгоритмическое обеспечение // Вестник ВСГУТУ. – 2012. – № 4. – С. 6-12.
2. Мижидон А.Д. Об одной задаче аналитического конструирования оптимального регулятора // Автоматика и телемеханика. – 2011. – № 11. – С. 102-116.
3. Квакернаак Х., Сиван Р. Линейные оптимальные системы управления. – М.: Мир, 1977. – 638 с.
4. Беллман Р. Введение в теорию матриц. – М.: Наука, 1976. – 351 с.

Мижидон Арсалан Дугарович, доктор технических наук, профессор, заведующий кафедрой Восточно-Сибирского государственного университета технологий и управления, тел.: (902-5) 633204. E-mail: miarsdu@esstu.ru

Мижидон Клара Арсалановна, аспирант Бурятского государственного университета, тел.: (951-6) 398322. E-mail: mizhidon@gmail.com

Mizhidon Arsalan Dugarovich, doctor of technical sciences, professor, head of applied mathematics department, East-Siberian State University of Technology and Management. E-mail: miarsdu@esstu.ru

Mizhidon Klara Arsalanovna, postgraduate student, applied mathematics department, Buryat State University. E-mail: mizhidon@gmail.com

УДК 519.853.4

© А.В. Орлов

ГИБРИДНЫЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ГЛОБАЛЬНОГО ПОИСКА ОПТИМИСТИЧЕСКИХ РЕШЕНИЙ В ЗАДАЧАХ ДВУХУРОВНЕВОЙ ОПТИМИЗАЦИИ¹

Рассматривается разработка гибридного подхода к решению квадратично-линейных задач двухуровневой оптимизации в оптимистической постановке. Эффективность предложенного подхода демонстрируется вычислительным экспериментом.

Ключевые слова: двухуровневая оптимизация, оптимистическое решение, теория глобального поиска, локальный поиск, построение аппроксимации поверхности уровня, генетический алгоритм.

© A.V. Orlov

HYBRID GENETIC ALGORITHM OF GLOBAL SEARCH FOR OPTIMISTIC SOLUTIONS IN BILEVEL OPTIMIZATION PROBLEMS

The article is devoted to elaboration of a hybrid approach to solving quadratic-linear bilevel optimization problems in optimistic formulation. The computational experiment shows the efficiency of the proposed approach.

Keywords: bilevel optimization, optimistic solution, global search theory, local search, constructing of the level surface approximation, genetic algorithm.

Введение

Как известно, разработка эффективных методов поиска решений в задачах с иерархической структурой представляет собой актуальную задачу современной математической оптимизации [1, 2], что обусловлено в первую очередь широким полем приложений иерархических задач. Аппарат двухуровневой оптимизации весьма удобен при моделировании конфликтов с неравноправным положением сторон, возникающих в практических задачах управления, экономики, транспорта и других областей [3, 4].

Например, при исследовании транспортно-логистических систем с помощью двухуровневой оптимизации моделируются задача назначения тарифов (Toll Setting Problem) [5] и задача развития транспортной сети (Highway Network Design Problem) [6]. На железнодорожном транспорте двухуровневой структурой обладает задача формирования поездов с целью перевозки грузов (Train Set Organization) [7] и ряд других.

Отметим, что практические приложения зачастую требуют решения задач достаточно высокой размерности. Ранее в нашей группе были разработаны методы локального и глобального поиска оптимистических решений в линейных [8, 9] и квадратично-линейных [8, 10, 11] задачах двухуровневой оптимизации, базирующиеся на теории глобального поиска в невыпуклых задачах с (d.c.) функциями А.Д. Александрова (т.е. представимыми в виде разности двух выпуклых функций) [12, 13]. Численное тестирование разработанных методов на большом спектре тестовых задач различной сложности и размерности продемонстрировало эффективность предлагаемого подхода по решению линейных задач с суммарной размерностью переменных на обоих уровнях до 1000 [9] и по решению квадратично-линейных задач – с размерностью до 300 [8, 11].

Тем не менее проблема разработки новых алгоритмов двухуровневой оптимизации, позволяющих оперировать с задачами еще большей размерности, в настоящее время остается актуальной. В данной работе с этой целью на примере квадратично-линейной двухуровневой задачи предлагается гибридный подход к построению метода глобального поиска, который, с одной стороны, базируется на теории глобального поиска [12], а с другой стороны – для реализации одного из этапов глобального поиска использует блоки генетических алгоритмов, такие как кроссовер и мутация [14].

¹ Работа выполнена при финансовой поддержке РФФИ, проекты 11-01-00270-а, 12-07-33045-мол_а_вед, 12-07-13116-офи_м_РЖД.

При этом разрабатываемый гибридный алгоритм включает в себя специальные методы локального поиска, использующие билинейную структуру исследуемой задачи [13].

1. Элементы теории глобального поиска

При решении невыпуклых задач математической оптимизации (математического программирования) с помощью подхода, основанного на теории глобального поиска [12], одним из ключевых этапов является построение аппроксимации поверхности уровня выпуклой функции, задающей базовую невыпуклость в исследуемой задаче [13]. Успешное построение аппроксимации позволяет «выскочить» из стационарной (критической) точки, полученной с помощью локального поиска, что, как известно, является одной из главных целей глобального поиска.

Напомним кратко этапы глобального поиска в задаче d.c. минимизации [12]:

$$\Phi(x) = g(x) - f(x) \downarrow \min, \quad x \in D, \quad (DC)$$

где $g(\cdot)$, $f(\cdot)$ – выпуклые функции, D – выпуклое множество.

Пусть известна некоторая приближенно критическая (стационарная) в задаче (DC) точка z^k со значением целевой функции $\zeta_k := \Phi(z^k)$, построенная с помощью какого-либо метода локального поиска. Тогда производится следующая цепочка операций:

1) выбирается число $\gamma \in [\gamma_-, \gamma_+]$, где $\gamma_- := \inf(g, D)$, $\gamma_+ := \sup(g, D)$ и строится некоторая конечная аппроксимация

$$A_k = \{v^1, \dots, v^N \mid f(v^i) = \gamma - \zeta_k, i = 1, \dots, N\}$$

поверхности уровня $U(\zeta_k) = \{y \mid f(y) = \gamma - \zeta_k\}$ выпуклой функции $f(\cdot)$;

2) для всех точек аппроксимации A_k проверяется неравенство $g(v^i) \leq \gamma$, $i = 1, 2, \dots, N$, следующее из условий глобальной оптимальности для задачи (DC) [12];

3) исходя из точек v^i , выбранных на втором этапе, осуществляется локальный поиск, доставляющий приближенно критические точки \hat{x}^i , $i \in \{1, \dots, N\}$ в задаче (DC);

4) значение целевой функции в каждой точке \hat{x}^i сравнивается со значением целевой функции в текущей критической точке z . Если какая-то из точек \hat{x}^i оказывается лучше текущей, происходит обновление последней, и весь процесс повторяется до исчерпания аппроксимации поверхности уровня.

Отметим, что построенная на этапе 1 аппроксимация должна быть достаточно репрезентативной, чтобы можно было судить о том, является ли текущая критическая точка z^k глобальным решением [12].

В настоящее время не разработано методов построения репрезентативной аппроксимации для задачи (DC) в общем виде. При решении конкретных невыпуклых задач эта проблема решается на основе накопленного опыта построения аппроксимаций [8, 9, 11-13]. Только для нескольких простейших невыпуклых задач (например, для задачи максимизации нормы на параллелепипеде) удалось построить аппроксимации, которые гарантируют достижение глобального решения [12], поэтому разработка новых подходов к решению этой проблемы является актуальной задачей невыпуклой оптимизации.

В данной работе эту задачу предлагается решать с использованием элементов генетического алгоритма.

2. Основные этапы генетического алгоритма

Напомним далее основные этапы генетического алгоритма [14], представив их для простоты на примере самой общей задачи оптимизации следующего вида:

$$F(x) \downarrow \min, \quad x \in D. \quad (P_0)$$

Сначала каким-либо образом, например, случайно, задается набор допустимых в задаче (P_0) точек, называемый популяцией:

$$Pop = \{x^i \mid x^i \in D, i = 1, \dots, N\}.$$

Затем выбирается функция пригодности, с помощью которой можно оценить эти точки (чаще всего для этого используется целевая функция задачи $F(\cdot)$).

Далее, с помощью двух случайно выбранных точек (особей) из популяции, которые называются родителями, определенным образом строятся две новые особи (потомки): $x^i, x^j \in Pop \Rightarrow y^1 \in D, y^2 \in D$. Эта процедура носит название кроссовер (кроссинговер, скрещивание). Существующие варианты осуществления кроссовера весьма разнообразны. Наиболее популярными являются одноточечный, двухточечный и однородный кроссовер [14]. Главной сложностью здесь является необходимость сохранения допустимости в исходной задаче построенных потомков.

Следующий этап – случайная мутация некоторых компонент построенных потомков, осуществляемая с определенной вероятностью: $y^1 \Rightarrow w^1 \in D, y^2 \Rightarrow w^2 \in D$. Здесь также необходимо следить за допустимостью получающихся особей [14].

После этого две получившиеся особи сравниваются друг с другом, и если лучшая из них с точки зрения функции пригодности оказывается лучше худшей особи из популяции, производится обновление последней: $w := \arg \min\{F(w^1), F(w^2)\}, j: F(x^j) \geq F(x^i) \forall x^i \in Pop$; если $F(w) < F(x^j)$, то $x^j := w$.

Процесс обычно заканчивается, когда произведено определенное заранее заданное количество итераций (поколений) описанной процедуры [14].

Разумеется, здесь описан только один из многих вариантов генетических алгоритмов. Количество публикаций по этой теме для различного сорта задач очень и очень велико (см., например, литературу в [14]). При этом неизменным остается главная идея подобного сорта подходов – аналогия с эволюцией живых организмов в ходе естественного отбора.

Далее, на примере квадратично-линейной двухуровневой задачи представим один из способов «встраивания» схемы генетического алгоритма в процедуру глобального поиска, описанную в первом разделе статьи.

3. Базовый алгоритм глобального поиска в квадратично-линейной двухуровневой задаче

Рассмотрим следующую квадратично-линейную задачу двухуровневой оптимизации в оптимистической постановке (на верхнем уровне производится минимизация по переменным обоих уровней):

$$\left. \begin{aligned} F(x, y) &= \frac{\Delta}{2} \langle x, Cx \rangle + \langle c, x \rangle + \frac{1}{2} \langle y, C_1 y \rangle + \langle c_1, y \rangle \downarrow \min_{x, y}, \\ (x, y) &\in X \stackrel{\Delta}{=} \{(x, y) \in R^m \times R^n \mid Ax + By \leq a, x \geq 0\}, \\ y &\in Y_*(x) \stackrel{\Delta}{=} \text{Arg min}_y \{\langle d, y \rangle \mid y \in Y(x)\}, \\ Y(x) &\stackrel{\Delta}{=} \{y \in R^n \mid A_1 x + B_1 y \leq b, y \geq 0\}, \end{aligned} \right\} (BP)$$

где $A \in R^{p \times m}, B \in R^{p \times n}, A_1 \in R^{q \times m}, B_1 \in R^{q \times n}, C \in R^{m \times m}, C_1 \in R^{n \times n}, c \in R^m, c_1, d \in R^n, a \in R^p, b \in R^q$ кроме того, $C = C^T \geq 0, C_1 = C_1^T \geq 0$.

Отметим, что целевая функция верхнего уровня $F(x, y)$ является выпуклой квадратичной с разделенными переменными. Целевая функция нижнего уровня является линейной. Пусть функция $F(x, y)$ ограничена снизу на множестве X , а функция $\langle d, y \rangle$ ограничена снизу на множестве $Y(x)$

для всех $x \in Pr(X) \stackrel{\Delta}{=} \{x \in R^m \mid \exists y: (x, y) \in X\}$, так что

$$\inf_x \inf_y \{\langle d, y \rangle \mid y \in Y(x), x \in Pr(X)\} > -\infty.$$

Обозначим через $h(x, y, v) := \langle d, y \rangle - \langle A_1 x - b, v \rangle$ невязку двойственности для задачи нижнего уровня, где $v \in R^q$ – вектор множителей Лагранжа в задаче нижнего уровня. В [8, 10] была осуществлена редукция задачи (BP) к следующему μ -параметрическому семейству задач:

$$\left. \begin{aligned} \Phi(x, y, v) = F(x, y) + \mu h(x, y, v) \downarrow \min_{x, y, v}, \\ (x, y, v) \in D = \{(x, y, v) \mid Ax + By \leq a, \quad A_1 x + B_1 v \leq b, \\ d + v B_1 \geq 0, \quad x \geq 0, \quad y \geq 0, \quad v \geq 0\}, \end{aligned} \right\} (P(\mu))$$

где $\mu > 0$ – штрафной множитель.

Заметим, что ограничения в задаче $(P(\mu))$ накладываются на пару (x, y) и на переменную v раздельно. Основываясь на этом факте, для осуществления локального поиска в задаче $(P(\mu))$ в [8, 10] были предложены две процедуры (XY-процедура и V-процедура), доставляющие точки, которые являются частично глобальными решениями исходной задачи по несвязанным группам переменных (критические точки). Эти процедуры заключаются в последовательном решении вспомогательных задач выпуклого квадратичного и линейного программирования (ЛП).

Для построения процедуры глобального поиска, согласно методологии из [12], прежде всего необходимо получить явное д.с. разложение целевой функции задачи. Нетрудно видеть, что целевая функция $\Phi(x, y, v)$ задачи $(P(\mu))$ может быть представлена в виде разности двух выпуклых функций, например, следующим образом:

$$\Phi(x, y, v) = g(x, y, v) - f(x, y, v), \tag{1}$$

где

$$g(x, y, v) = \frac{1}{2} \langle x, Cx \rangle + \langle c, x \rangle + \frac{1}{2} \langle y, C_1 y \rangle + \langle c_1, y \rangle + \mu (\langle v, b \rangle + \frac{1}{4} P v - A_1 x P^2),$$

$$f(x, y, v) = \mu (\frac{1}{4} P v + A_1 x P^2 - \langle y, d \rangle) - \text{выпуклые функции.}$$

Представим далее алгоритм глобального поиска в задаче $(P(\mu))$, использующий д.с. разложение (1). Пусть дана некоторая точка $(x_0, y_0, v_0) \in R^{m+n+q}$, числовые последовательности $\{\tau_k\}, \{\delta_k\}$, $\tau_k, \delta_k > 0, k = 0, 1, \dots, \tau_k \downarrow 0, \delta_k \downarrow 0 (k \rightarrow \infty)$, множество направлений

$$Dir = \{(a^l, b^l, c^l) \in R^{m+n+q} \mid (a^l, b^l, c^l) \neq 0, l = 0, 1, \dots, N\},$$

числа $\gamma_- := \inf_{(x, y, v)} (g, D)$ и $\gamma_+ := \sup_{(x, y, v)} (g, D)$ и параметры алгоритма M и ν .

Базовый алгоритм глобального поиска

Шаг 0. Положить $k := 0$, $(\overset{-k}{x}, \overset{-k}{y}, \overset{-k}{v}) := (x_0, y_0, v_0)$, $l := 1$, $\gamma := \gamma_-$, $\Delta\gamma := (\gamma_+ - \gamma_-) / M$.

Шаг 1. Начиная с точки $(\overset{-k}{x}, \overset{-k}{y}, \overset{-k}{v})$, с помощью XY- или V-процедуры локального поиска построить τ_k – критическую точку $(x^k, y^k, v^k) \in D$ в задаче $(P(\mu))$.

Положить $\zeta_k := \Phi(x^k, y^k, v^k)$.

Шаг 2. По точке $(a^l, b^l, c^l) \in Dir$ построить точку (z^l, u^l, w^l) из аппроксимации A_k поверхности уровня функции $f(\cdot)$:

$$A_k = \{(z^l, u^l, w^l) \mid f(z^l, u^l, w^l) = \gamma - \zeta_k\}.$$

Шаг 3. Если $g(z^l, u^l, w^l) > \gamma + \nu\gamma$, то положить $l := l + 1$ и перейти на шаг 2.

Шаг 4. Начиная с точки (z^l, u^l, w^l) , с помощью процедуры локального поиска построить δ_k – критическую точку $(\hat{x}^l, \hat{y}^l, \hat{v}^l) \in D$ в задаче $(P(\mu))$.

Шаг 5. Если $\Phi(\hat{x}^l, \hat{y}^l, \hat{v}^l) < \Phi(x^k, y^k, v^k)$, то положить

$(\hat{x}^{k+1}, \hat{y}^{k+1}, \hat{v}^{k+1}) := (\hat{x}^l, \hat{y}^l, \hat{v}^l)$, $k := k + 1$, $l := 1$, $\gamma := \gamma_-$ и перейти на шаг 1.

Шаг 6. Если $\Phi(\hat{x}^l, \hat{y}^l, \hat{v}^l) \geq \Phi(x^k, y^k, v^k)$, $l < N$, то положить $l := l + 1$ и вернуться на шаг 2.

Шаг 7. Если $\Phi(\hat{x}^l, \hat{y}^l, \hat{v}^l) \geq \Phi(x^k, y^k, v^k)$, $l = N$ и $\gamma < \gamma_+$, то положить $\gamma := \gamma + \Delta\gamma$, $l := 1$ и вернуться на шаг 2.

Шаг 8. Если $l = N$, $\Phi(\hat{x}^l, \hat{y}^l, \hat{v}^l) \geq \Phi(x^k, y^k, v^k) \quad \forall \gamma \in [\gamma_-, \gamma_+]$ (т.е. одномерный поиск по γ на отрезке $[\gamma_-, \gamma_+]$ закончен), то стоп; (x^k, y^k, v^k) – полученное решение задачи.

Шаги 1-8 представляют собой алгоритмизованную запись этапов глобального поиска, приведенных в разделе 1, которая осуществлена в терминах задачи $(P(\mu))$. При этом отметим, что на шаге 3 алгоритма производится проверка точки на пригодность к дальнейшим исследованиям с помощью неравенства, вытекающего из условий оптимальности для задач d.c. минимизации [12]. При численной реализации необходимо использовать параметр ν , варьированием которого можно изменять точность выполнения этого неравенства (с целью уменьшить влияние машинных ошибок округления [8, 11, 13]). Различные же значения параметра M отвечают за разбиение отрезка $[\gamma_-, \gamma_+]$ на соответствующее количество частей для реализации пассивного одномерного поиска по γ .

Обратим также внимание на шаг 2, где аппроксимация поверхности уровня функции f , которая задает базовую невыпуклость в исследуемой задаче, строится с помощью заданного множества направлений Dir . Это множество было выбрано экспериментально для задач с билинейной структурой (к которым относится и задача $(P(\mu))$) [8, 11, 13] и включает в себя стандартные векторы евклидова базиса и компоненты текущей критической точки. Недостатком такого подхода к построению аппроксимации является, в частности, то, что оно не меняется от итерации к итерации алгоритма и содержит достаточно большое количество точек (в некоторых случаях порядка $(m+n)q$). Последний факт в значительной степени влияет на эффективность работы алгоритма глобального поиска при повышении размерности задачи.

Гибридный алгоритм глобального поиска, описанию которого посвящен следующий раздел, лишен указанных недостатков.

4. Гибридный алгоритм глобального поиска и его тестирование

Прежде всего опишем выбранные способы реализации генетических блоков для гибридного алгоритма решения задачи $(P(\mu))$.

В качестве популяции на каждой итерации глобального поиска выбираются точки, составляющие текущую аппроксимацию поверхности уровня:

$$Pop_k = \{(r^s, z^s, w^s) \mid (r^s, z^s, w^s) \in A_k, s = 1, \dots, N\}.$$

Функция пригодности $Loc(\cdot)$, оценивающая точки аппроксимации, представляет собой значение целевой функции в критической точке, построенной исходя из соответствующей точки аппроксимации: $Loc(r^s, z^s, w^s) := \Phi(\hat{x}^s, \hat{y}^s, \hat{v}^s)$. Здесь отметим, что поскольку точки аппроксимации могут быть недопустимыми, важным представляется свойство используемых процедур локального поиска, сохраняющих работоспособность при старте с недопустимой точки [8, 10].

Для осуществления скрещивания была реализована процедура так называемого однородного кроссовера [14]: выберем два произвольных индекса $s_1, s_2 \in \{1, \dots, N\}$ ($s_1 \neq s_2$) и положим $l := Random[0,1]$.

Далее $\forall j = 1, \dots, m+n+q$, если $l < 0.5$, то $(\tilde{r}, \tilde{z}, \tilde{w})_j := (r^{s_1}, z^{s_1}, w^{s_1})_j$, $(\tilde{r}, \tilde{z}, \tilde{w})_j := (r^{s_2}, z^{s_2}, w^{s_2})_j$, иначе $(\tilde{r}, \tilde{z}, \tilde{w})_j := (r^{s_1}, z^{s_1}, w^{s_1})_j$, $(\tilde{r}, \tilde{z}, \tilde{w})_j := (r^{s_2}, z^{s_2}, w^{s_2})_j$. Таким образом, каждая компонента одного из векторов-потомков с вероятностью $1/2$ представляет собой компоненту одного из родителей.

В качестве процедуры мутации была выбрана следующая: пусть $P_m := 0.01$, $K = const$. Положим $l_1 := Random[0,1]$. Если $l_1 < P_m$, то $(\tilde{r}, \tilde{z}, \tilde{w})_j := Random[-K, K] \quad \forall j = 1, \dots, m+n+q$. Положим

$l_2 := \text{Random}[0,1]$. Если $l_2 < P_m$, то $(\tilde{r}, \tilde{z}, \tilde{w})_j := \text{Random}[-K, K] \quad \forall j = 1, \dots, m+n+q$. Подобного сорта процедура называется равномерной мутацией [14].

В результате был построен следующий гибридный алгоритм глобального поиска, сочетающий этапы базового алгоритма и блоки генетического алгоритма.

Пусть даны точка $(x_0, y_0, v_0) \in D$, множество направлений

$Dir = \{(a^1, b^1, c^1), \dots, (a^N, b^N, c^N) \in R^{m+n+q} \mid (a^s, b^s, c^s) \neq 0, s = 1, \dots, N\}$, числа $\gamma_- := \inf(g, D)$ и $\gamma_+ := \sup(g, D)$, вероятность мутации P_m и количество поколений G_{max} .

Гибридный алгоритм глобального поиска

Шаг 0. Положить $k := 0$, $(\bar{x}^k, \bar{y}^k, \bar{v}^k) := (x^0, y^0, v^0)$, $s := 1$, $\gamma := \gamma_-$, $\Delta\gamma = (\gamma_+ - \gamma_-) / N$.

Шаг 1. Начиная с точки $(x_0, y_0, v_0) \in D$ с помощью XY- или V-процедуры локального поиска найти приближенно критическую точку $(\hat{x}^0, \hat{y}^0, \hat{v}^0) \in D$ в задаче $(P(\mu))$. Положить $\zeta_s := \Phi(\hat{x}^0, \hat{y}^0, \hat{v}^0)$.

Шаг 2. По точкам $(a^s, b^s, c^s) \in Dir$ построить точки (r^s, z^s, w^s) аппроксимации поверхности уровня функции f , $s = 1, \dots, N$, такие что $f(r^s, z^s, w^s) = \gamma + \Delta\gamma * (s-1) - \zeta_s$, т.е. построить популяцию точек поверхности уровня Pop_k . Для каждой точки популяции вычислить значение функции пригодности: $\zeta_s := Loc(r^s, z^s, w^s)$, $s = 1, \dots, N$.

Шаг 3. $s_1 := \text{Random}\{1, \dots, N\}$, $s_2 := \text{Random}\{1, \dots, N\}$. С помощью процедуры кроссовера по точкам $(r^{s_1}, z^{s_1}, w^{s_1})$ и $(r^{s_2}, z^{s_2}, w^{s_2})$ построить точки $(\tilde{r}, \tilde{z}, \tilde{w})$ и $(\bar{r}, \bar{z}, \bar{w})$.

Шаг 4. С вероятностью P_m реализовать процедуру мутации для точек $(\tilde{r}, \tilde{z}, \tilde{w})$ и $(\bar{r}, \bar{z}, \bar{w})$.

Шаг 5. Вычислить $\tilde{\zeta} := \Phi(\tilde{r}, \tilde{z}, \tilde{w})$, $\bar{\zeta} := \Phi(\bar{r}, \bar{z}, \bar{w})$, $\tilde{\gamma} := g(\tilde{r}, \tilde{z}, \tilde{w})$, $\bar{\gamma} := g(\bar{r}, \bar{z}, \bar{w})$ и построить две новые точки лежащие на поверхностях уровня: $f(\bar{r}^{s_1}, \bar{z}^{s_1}, \bar{w}^{s_1}) = \tilde{\gamma} - \tilde{\zeta}$ и $f(\bar{r}^{s_2}, \bar{z}^{s_2}, \bar{w}^{s_2}) = \bar{\gamma} - \bar{\zeta}$.

Шаг 6. Вычислить значения $Loc(\bar{r}^{s_1}, \bar{z}^{s_1}, \bar{w}^{s_1})$ и $Loc(\bar{r}^{s_2}, \bar{z}^{s_2}, \bar{w}^{s_2})$. Пусть $(\bar{r}^s, \bar{z}^s, \bar{w}^s) := \arg \min\{Loc(\bar{r}^{s_1}, \bar{z}^{s_1}, \bar{w}^{s_1}), Loc(\bar{r}^{s_2}, \bar{z}^{s_2}, \bar{w}^{s_2})\}$.

Шаг 7. Вычислить номер $j: Loc(r^j, z^j, w^j) \geq Loc(r^s, z^s, w^s) \quad \forall (r^s, z^s, w^s) \in Pop_k$; если $Loc(\bar{r}^s, \bar{z}^s, \bar{w}^s) < Loc(r^j, z^j, w^j)$, то $(r^j, z^j, w^j) := (\bar{r}^s, \bar{z}^s, \bar{w}^s)$.

Шаг 8. Если $k < G_{max}$, то $k := k + 1$ и перейти на шаг 3, иначе стоп. $(x^*, y^*, v^*) := \arg \min\{Loc(r^s, z^s, w^s) \mid (r^s, z^s, w^s) \in Pop_k, s = \overline{1, N}\}$ – полученное решение задачи.

Отметим, что процедуру построения точек, лежащих на поверхности уровня, на шаге 5 алгоритма можно реализовать для любой точки пространства (кроме 0), так что проблем с обеспечением допустимости точек, полученных в результате применения блоков генетического алгоритма, здесь не возникает.

Для сравнения эффективности базового и гибридного алгоритмов глобального поиска использовались тестовые задачи, сгенерированные с помощью специального метода, предложенного в [15], который основан на построении задач произвольной размерности с известными локальными и глобальными решениями из задач-ядер небольшой размерности (см. также [8-11, 13]). При этом для простоты полагаем $C = 0$ и $C_1 = 0$.

Результаты сравнения приведены в таблице со следующими обозначениями: *Name* – имя тестового примера в формате $n + m_N$, где n – число переменных на верхнем уровне, m – число переменных на нижнем уровне, N – номер примера данной размерности; LP_b и LP_g – количество вспомогательных задач ЛП, которое потребовалось решить для достижения глобального решения базовым и гибридным алгоритмами соответственно; T_b и T_g – время решения примера в формате мин:сек.доли с помощью базового и гибридного алгоритма соответственно; *Pop* и *MaxG* –

параметры гибридного алгоритма, обозначающие размер популяции и количество поколений соответственно.

Таблица

Сравнение базового и гибридного алгоритмов глобального поиска

Name	LP_b	T_b	Pop	MaxG	LP_g	T_g
1+1_1	118	0:00.02	3	5	56	0:00.00
3+3_1	118	0:00.02	3	5	55	0:00.00
5+5_1	18	0:00.02	3	5	60	0:00.02
5+5_2	512	0:00.09	3	5	62	0:00.02
10+10_1	854	0:00.72	3	5	62	0:00.05
15+15_1	1210	0:02.92	3	10	104	0:00.26
20+20_1	1610	0:08.67	3	20	193	0:01.09
25+25_1	2010	0:20.33	3	20	200	0:02.26
30+30_1	2410	0:39.27	5	20	208	0:04.00
35+35_1	2810	1:10.64	5	20	208	0:06.36
35+35_2	2810	1:11.12	5	20	208	0:06.38
40+40_1	3728	2:21.38	5	50	470	0:21.03
40+40_2	3210	2:02.36	5	50	469	0:21.00
50+50_1	4650	5:39.44	10	50	492	0:42.03
50+50_2	4010	4:53.03	10	50	492	0:43.44
75+75_1	6970	32:14.69	10	50	497	2:53.38
75+75_2	6982	32:42.48	10	50	497	2:54.31
100+100_1	9290	102:09.97	10	50	519	7:21.72

По результатам, представленным в таблице, можно видеть, что во всех случаях, кроме одного (для примера 5+5_1, который отмечен в таблице жирным шрифтом), удалось подобрать такой размер популяции и количество поколений, что гибридный алгоритм оказался существенно эффективнее и по времени, и по количеству вспомогательных задач ЛП, которые потребовалось решить для достижения глобального решения. Последнюю величину можно считать некоторой мерой сложности решаемой задачи для используемого алгоритма. В некоторых случаях превосходство гибридного алгоритма над базовым составило 10-15 раз.

Заключение

В работе предложен новый гибридный подход к разработке алгоритмов поиска оптимистических решений в квадратично-линейных двухуровневых задачах оптимизации, сочетающий использование теории глобального поиска [12] и элементы генетических алгоритмов [14].

На основании проведенного вычислительного эксперимента по сравнению базового и гибридного алгоритмов глобального поиска можно сделать вывод о том, что использование для одного из самых сложных этапов глобального поиска – построения аппроксимации поверхности уровня – блоков генетического алгоритма оказалось оправданным. Имеющийся численный и алгоритмический опыт по решению различного сорта задач исследования операций с билинейной структурой [8-11, 13] позволяет также утверждать, что предложенный гибридный подход будет эффективен и при решении более сложных двухуровневых задач, в том числе, возникающих на автомобильном и железнодорожном транспорте [5-7].

Литература

1. Pang J.-S. Three modelling paradigms in mathematical programming // *Mathematical programming, Ser.B.* 2010. V. 125, No. 2. P. 297-323.
2. Dempe S. *Foundations of Bilevel Programming.* – Dordrecht: Kluwer Academic Publishers, 2002.
3. Bard J.F. *Practical Bilevel Optimization.* – Dordrecht: Kluwer Academic Publishers, 1998.
4. Colson B., Marcotte P., Savard G. An overview of bilevel optimization // *Annals of operations research.* 2007. V. 153. P. 235-256.
5. Brotcorne L., Labbe M., Marcotte P., Savard G. A Bilevel Model for Toll Optimization on a Multicommodity Transportation Network // *Transportation Science.* 2001. V. 35, No. 4. P. 345–358.
6. Ben-Ayed O., Blair C.E., Boyce D.E., LeBlanc L.J. Construction of a real-world bilevel linear programming model of the highway network design problem // *Annals of Operations Research.* 1992. V. 34, No. 1-4. P. 219-254.
7. Gao Y., Lu J., Zhang G., Gao S. A Bilevel Model for Railway Train Set Organizing Optimization // *Proceedings of the International Conference on Intelligent Systems and Knowledge Engineering (ISKE 2007) (October 15-16, 2007, Chengdu, China).* Atlantis Press, 2007, doi:10.2991/iske.2007.239.
8. Strekalovsky A.S., Orlov A.V., Malyshev A.V. On computational search for optimistic solution in bilevel problems // *Journal of Global Optimization.* 2010. V. 48, No. 1. P. 159-172.
9. Груздева Т.В., Петрова Е.Г. Численное решение линейной двухуровневой задачи // *Журнал вычислительной математики и математической физики.* 2010. Т. 50, № 10. С. 1715-1726.
10. Стрекаловский А.С., Орлов А.В., Малышев А.В. Локальный поиск в квадратично-линейной задаче двухуровневого программирования // *Сиб. журн. вычисл. математики.* 2010. Т. 13, № 1. С. 75-88.
11. Стрекаловский А.С., Орлов А.В., Малышев А.В. Численное решение одного класса задач двухуровневого программирования // *Сиб. журн. вычисл. математики.* 2010. Т. 13, № 2. С. 201-212.
12. Стрекаловский А.С. *Элементы невыпуклой оптимизации.* Новосибирск: Наука, 2003.
13. Стрекаловский А.С., Орлов А.В. *Биматричные игры и билинейное программирование.* М.: ФИЗМАТЛИТ, 2007.
14. Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs.* New York: Springer-Verlag, 1994.
15. Calamai P., Vicente L. Generating Linear and Linear-Quadratic Bilevel Programming Problems // *SIAM Journal on Scientific Computing.* 1993. V. 14, No. 4. P. 770-782.

Орлов Андрей Васильевич, кандидат физико-математических наук, доцент, старший научный сотрудник Федерального государственного бюджетного учреждения науки Институт динамики систем и теории управления Сибирского отделения Российской академии наук, 664033, Иркутск, ул. Лермонтова, 134, тел. +7(3952) 45-30-63, e-mail: anor@icc.ru.

Orlov Andrey Vasilievich, candidate of physical and mathematical sciences, associate professor, senior researcher, Institute for System Dynamics and Control Theory SB RAS, 664033, Irkutsk, Lermontov str., 134, ph. +7(3952) 45-30-63, e-mail: anor@icc.ru.

УДК 517.977

© *И.В. Расина, О.В. Усенко*

ДОСТАТОЧНЫЕ УСЛОВИЯ ОПТИМАЛЬНОСТИ В ДИСКРЕТНОЙ ИЕРАРХИЧЕСКОЙ МОДЕЛИ¹

Рассматривается двухуровневая модель сетевой структуры, верхний уровень которой представляет собой абстрактную сеть операторов, а нижний уровень – дискретные динамические модели. Для указанной сети ставится задача оптимизации и выводятся достаточные условия оптимальности аналогично условиям для дискретно-непрерывных динамических систем.

Ключевые слова: двухуровневая модель, задача оптимизации, достаточные условия оптимальности.

© *I.V. Rasina, O.V. Usenko*

SUFFICIENT CONDITIONS OF OPTIMALITY IN A DISCRETE HIERARCHICAL MODEL

In the article a two-level model of network structure is considered, which top level is an abstract network of operators, and the lower level - discrete dynamical models. For this network the task of optimization is put forward and sufficient conditions of optimality are derived analogically as for the discrete-continuous dynamic systems.

Keywords: two-level model, optimization problem, sufficient optimal conditions.

Введение

Истоком исследований неоднородных систем можно считать работы Е.А. Барбашина и С.В. Емельянова, Ю.В. Долгоненко, Ю.И. Неймарка, I. Flugge-Lots J. 50-х гг. XX в., а их активное систематическое изучение начато достаточно давно [1-5]. К настоящему времени исследователи разных школ и направлений проявляют к ним все возрастающий интерес в связи с потребностями практики, о чем свидетельствует увеличивающийся поток публикаций [6-10]. Исследования главным образом посвящены системам дифференциальных уравнений переменной структуры (так называемым гибридным). Подобные системы получаются путем различных обобщений самого понятия классической дифференциальной системы. Весьма полное представление об этих исследованиях дают недавние обстоятельные обзоры [11, 12].

Цель данной статьи – продолжение исследований неоднородных систем в рамках альтернативного направления, берущего начало от основополагающей работы В.Ф. Кротова [13], исходя из трактовки неоднородной управляемой системы как конкретизации абстрактной динамической многошаговой системы, для которой в [13] получены общие достаточные условия оптимальности.

На этом пути в [3] была предложена двухуровневая модель дискретно-непрерывной системы (ДНС), на нижнем уровне которой происходят непрерывные однородные процессы, а на верхнем осуществляется дискретное управление их переключениями. Эта модель позволяет систематически учесть изменение не только структуры, но и порядка и даже самой природы однородных подсистем и процессов.

Такая трактовка позволяет эффективно декомпозировать соответствующие задачи управления, прежде всего оптимального управления, на «однородные» подзадачи так, чтобы применить нетрадиционные методы, изложенные в работах В.Ф. Кротова и В.И. Гурмана [14-16] для однородных непрерывных и дискретных систем. Кроме того, как показывает практика [17], нижний уровень может быть представлен и дискретными однородными процессами. В [18] введено понятие абстрактной сети операторов и для нее получены достаточные условия оптимальности как обобщение дискретной

¹ Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект 12-01-00256-а) и Российского гуманитарного научного фонда (проект 11-02-00171-а).

динамической модели и достаточных условий [14, 16]. При этом дискретную модель [14, 16] можно рассматривать как частный случай сети – цепочку операторов.

В предлагаемой работе строится новая двухуровневая модель, верхний уровень которой составляет не цепочка, а сеть операторов, а нижний – однородные дискретные управляемые системы. Для этой модели получаются достаточные условия оптимальности. Предварительно напомним понятие сети операторов согласно [18].

1. Сеть операторов и задача оптимизации

Пусть имеется N операторов произвольной природы

$$f_k : X_k \times U_k \rightarrow Y_k \quad (y_k = f(k, x_k, u_k)).$$

Вводятся подмножества X_{kq} , такие что

$$\prod_{q=1}^{n_k} X_{kq} = X_k.$$

Будем говорить, что выход оператора l подается на вход оператора k , если для некоторого q имеет место равенство $\chi(k, q, x_k) = y_l$, где $\chi(k, q, x_k)$ – оператор проектирования на подмножество X_{kq} .

Пусть рассматриваемые операторы соединены указанным образом по некоторой схеме, представленной ориентированным графом (рис.1). Предполагается, что для данного k между номерами q и l имеет место взаимно-однозначное соответствие. Иными словами, X_k олицетворяет множество входов k -го оператора, занятых в соединениях, а U_k – множество свободных входов. Такую систему будем называть *сетью операторов*.

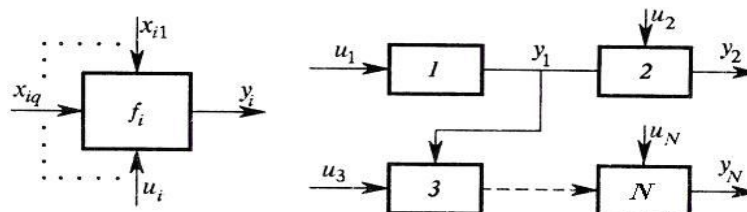


Рис. 1

Описанную сеть можно рассматривать как некоторый оператор

$$F : X \rightarrow Y, \quad X = \prod_{i \in I} U_i, \quad Y = \prod_{i \in I} Y_i.$$

Будем называть его *оператором следующего верхнего уровня* (при сравнении в обратном порядке будем говорить об *операторах следующего нижнего уровня*). Такое представление дает возможность отобразить иерархическое строение моделируемого объекта. Специальный случай сети – цепочка произвольных операторов — может рассматриваться как общая модель динамической системы с переменной структурой.

Рассматривается задача о минимуме функционала

$$I = \sum_1^N I_k(y_k) = \sum_1^N I_k(f(k, x_k, u_k)) = \sum_1^N f^0(k, x_k, u_k) \tag{1}$$

на множестве \mathbf{D} наборов $m = \{(x_k, u_k)\}$, $k = 1, \dots, N$, связанных указанными соотношениями сети и возможными дополнительными ограничениями вида $(x_k, u_k) \in B(k)$, где $B(k)$ – заданное при каждом k множество.

Требуется найти минимизирующую последовательность $\{m_s\} \subset D$, т.е. такую, что $I(m_s) \rightarrow \inf_D I$.

2. Дискретная двухуровневая модель

Предлагается следующая конкретизация указанной абстрактной дискретной модели. Представим условие $(x_k, u_k) \in B(k)$ в форме $x_k \in X(k)$, $u_k \in U(k)$, где $X(k)$ – проекция на X_k , $U(k)$ – сечение $B(k)$ при данных k, x_k . Пусть на некотором подмножестве $K' \subset K = \{1, \dots, N\}$ имеем $u = (u^d, m^d)$, где u^d – произвольной природы, а m^c – некоторый дискретный управляемый процесс, так что сечение множества $U(k, x)$ при фиксированных x, u^d есть допустимое множество $D^c(k, x, u^d)$ с соответствующей рекуррентной цепочкой

$$\begin{aligned} x^c(t+1) &= f^c(z, t, x^c(t), u^c(t)), \\ t &\in T(z), \quad T(z) = t_I, t_I + 1, \dots, t_F \\ x^c &\in X^c(z, t) \subset R^{n(k)}, \quad u^c \in U^c(z, t, x^c) \subset R^{p(k)}, \quad z = (k, x, u^d). \end{aligned} \quad (2)$$

Оператор правой части (1) сводится к следующему:

$$\begin{aligned} y_k &= f(k, x, u) = \theta(z, \gamma^c), \\ \gamma^c &= (t_I, x_I^c, t_F, x_F^c) \in \{\gamma^c : t_I = \tau(z), x_I^c = \xi(z), t_F = \vartheta(z), x_F^c \in \Gamma^c(z)\}. \end{aligned}$$

Решением этой комбинированной системы будем считать набор $m = (x(k), u(k)) \in D$, где при $k \in K' : u(k) = (u^d(k), m^c(k))$, $m^c \in D^c(t, x(k), u^d(k))$.

Эту модель можно рассматривать как двухуровневую управляемую систему. Нижний уровень для соответствующего оператора представляет собой описание дискретного управляемого процесса, играющего роль управления для верхнего уровня (рис. 2).

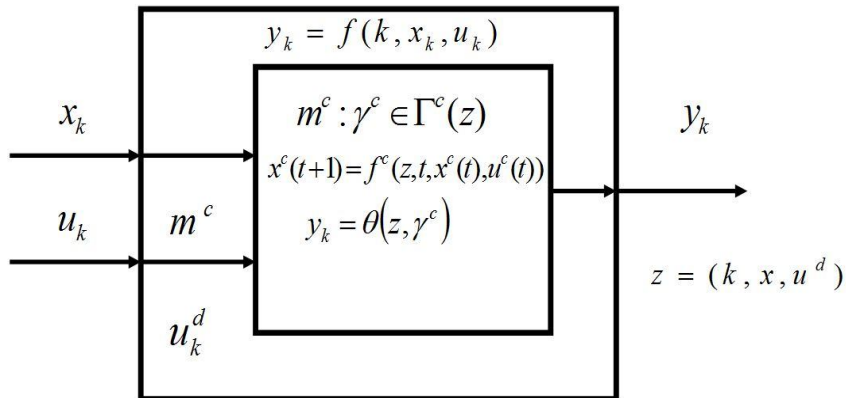


Рис. 2

Задача оптимизации формулируется для верхнего уровня. Достаточные условия оптимальности для нее получаются по аналогии с дискретно-непрерывными процессами [19, 20].

Для номеров $k \in K'$ вводится дополнительно параметрическое семейство (с параметром z) функций $\varphi^c : R^{m(k)+1} \rightarrow R$. Строится соответствующая модификация обобщенного лагранжиана [19, 20]:

$$L = - \sum_{K \setminus K'} R_k - \sum_{K'} R'_k, \text{ где}$$

$$R_k(x, u) = \sum_{l=1}^N (\varphi(k, l, f(k, x, u)) - \varphi(l, k, \chi(k, l, x))) - f^0(k, x, u),$$

$$R'_k = G(z, \gamma^c) + \sum_{l=1}^N (R^c(z, t, x^c(z, t), u^c(z, t)) - \mu^c(z)),$$

$$G(z, \gamma^c) = \sum_{l=1}^N (\varphi(k, l, y_k) - \varphi(l, k, \chi(k, j, x_k))) + \varphi^c(z, t_l, x_l^c) -$$

$$- \varphi^c(z, t_F, x_F^c) + \sum_{l=1}^N \mu^c(z, t) - I_k(\theta(z, \gamma^c))$$

$$R^c(z, t, x^c, u^c) = \varphi^c(t+1, f^c(z, t, x^c, u^c)) - \varphi^c(z, t, x^c),$$

$$\mu^c(z, t) = \sup \{R^c : x^c \in X^c(z, t), u \in U^c(z, t, x^c)\},$$

где $y_k = \theta(z, \gamma^c)$ при $K \in K'$, $y_k = f(k, x, u)$ при $k \in K \setminus K'$. Обозначим

$$\mu'(k) = \sup \{G(z, \gamma^c) : \gamma^c \in \Gamma^c(z), x_l^c \in X^c(z, t_l)\}.$$

Легко убедиться, что $L(m) = I(m)$ при $m \in D$, т.е. при выполнении отброшенных связей. Для этого сначала рассмотрим выражение для функции R^c . При выполнении рекуррентной цепочки в (2)

$$R^c(z, t, x^c, u^c) = \varphi^c(z, t+1, x^c(t+1)) - \varphi^c(z, t, x^c(t)). \text{ Тогда}$$

$$- \sum_{t_l}^{t_F-1} R^c(z, t, x^c, u^c) + \varphi^c(z, t_l, x_l^c) - \varphi^c(z, t_F, x_F^c) =$$

$$= - \sum_{t_l}^{t_F-1} (\varphi^c(t+1, x(t+1)) - \varphi^c(z, t, x^c(t))) + \varphi^c(z, t_l, x_l^c) -$$

$$- \varphi^c(z, t_F, x_F^c) = - \sum_{t_l}^{t_F} (\varphi^c(z, t, x^c(t)) - \varphi^c(z, t, x^c(t))),$$

$$\sum_{K'} R'_k = \sum_{K'} \sum_{l=1}^N (\varphi(k, l, y_k) - \varphi(k, l, y_l)) + \sum_{t_l}^{t_F-1} \mu^c(z, t) - I_k(y_k) - \sum_{t_l}^{t_F-1} \mu^c(z, t).$$

Тогда

$$\sum_{K'} R'_k = \sum_{K'} \sum_{l=1}^N (\varphi(k, l, y_k) - \varphi(k, l, y_l)) - I_k(y_k).$$

Окончательно имеем

$$L = - \sum_{K \setminus K'} R_k - \sum_{K'} R'_k = - \sum_{k,l=1}^N \left(\varphi(k, l, x(l)) - \varphi(k, l, x(k)) + \sum_{k=1}^N I_k(y_k) \right) = I.$$

Отсюда непосредственно вытекает следующая теорема.

Теорема 1. Пусть имеются последовательность элементов $\{m_s\} \subset D$ и пара (φ, φ^c) такие, что:

1. $R(k, x_s(k), u_s(k)) \rightarrow \mu(k);$

2. $\sum_{t_i}^{t_F-1} (R^c(z_s(k), t, x_s^c(k, t), u_s(k, t)) - \mu^c(z_s(k), t)) \rightarrow 0, \quad k \in K'$

3. $G(z_s(k), \gamma_s^c(k)) - \mu'(k) \rightarrow 0, \quad k \in K'.$

Тогда последовательность $\{m_s\}$ минимизирующая для **I** на **D**.

Полученные общие достаточные условия могут быть использованы для разработки конкретных методов и алгоритмов оптимизации управления в сложных системах, моделируемых как сети операторов. При этом может быть использован опыт исследования моделей, рассматриваемых как частные случаи сети операторов: динамической ДНС, дерева операторов [21], моделирующего, например, дискретизованное распределение гидрохимических компонентов в бассейне реки.

Заключение

Предложенная модель на основе абстрактной сети операторов является переходом от чисто динамических неоднородных систем к более общему классу систем сетевой структуры.

Заметим, что рассматриваемая на нижнем уровне дискретная однородная подсистема может быть получена в результате естественной дискретизации непрерывной системы на этапе вычислений. Такая схема для динамических неоднородных систем реализована в [22]. Это существенно расширяет область приложений иерархического представления.

Изложенный подход к исследованию и оптимизации неоднородных процессов позволяет декомпозировать соответствующие задачи управления на задачи для однородных подсистем, что позволяет рассматривать широкий круг сложных прикладных задач.

Литература

1. Теория систем с переменной структурой / под ред. С.В. Емельянова. М.: Наука, 1970.
2. Цыпкин Я.З. Попков Б.С. Теория нелинейных импульсных систем. М.: Наука, 1973.
3. Гурман В.И. К теории оптимальных дискретных процессов // *АиТ*. 1973. № 6. С. 53 - 58.
4. Габелко К.Н. Последовательное улучшение многоэтапных процессов // *АиТ*. 1974. № 12. С. 72-80.
5. Орлов А.Г., Расина И.В. Сложные процессы и достаточные условия относительной оптимальности // *Управляемые системы*. 1979. Вып. 18. Новосибирск: Изд-во ИМ СО АН СССР. С. 39-46.
6. Бортаковский А.С, Пантелеев А.В. Достаточные условия оптимальности управления непрерывно-дискретными системами // *АиТ*. 1987. №7. С. 57-66.
7. Branicky M., Borkar V., Mitter S. A Unified Framework for Hybrid Control: Model and Optimal Control Theory // *IEEE TAC*. 1998. V. 43. P. 31-45.
8. Aubin J.-P. Impulse Differential Equations and Hybrid Systems: A Viability Approach // *Lecture Notes*. University of California at Berkeley, 2000.
9. Vassilyev S.N. Homomorphisms of Impulsive Differential Equations with Impulses at Unfixed Times and Comparison Method // *Int. J. Hybrid Syst*. 2002. V. 2. No 3. P. 289-296.
10. Куржанский А.Б., Точилин П.А. Импульсные управления в моделях гибридных систем // *Дифференциальные уравнения*. 2009. Т. 45. № 5. С. 716-727.
11. Васильев С.Н., Маликов А.И. О некоторых результатах по устойчивости переключаемых гибридных систем // *Актуальные проблемы механики сплошной среды. К 20-летию ИММ КазНЦ РАН*. 2011. Т.1. С.23-81. Казань: Фолиант.
12. Миллер Б.М., Рубинович Е.Я. Разрывные решения в задачах оптимального управления и их представления с помощью сингулярных пространственновременных преобразований // *АиТ*. 2013. (В печати)
13. Кротов В.Ф. Достаточные условия оптимальности для дискретных управляемых систем. *ДАН СССР*. 1967. Т. 172. № 1. С. 18-21.

14. Кротов В.Ф., Гурман В.И. Методы и задачи оптимального управления. М.: Наука, 1973.
15. Krotov V.F. Global Methods in Optimal Control Theory. New York: Marcel Dekker, 1996.
16. Гурман В.И. Принцип расширения в задачах управления. М.: Наука, 1997.
17. Расина И.В. Сложные дискретные процессы. // Методы оптимизации и исследование операций (прикладная математика). Иркутск, СЭИ СО АН СССР, 1976. С. 64-70.
18. Гурман В.И. Оптимизация дискретных систем (уч. пос.). Иркутск: Изд-во ИГУ, 1976.
19. Расина И.В. Итерационные алгоритмы оптимизации дискретно-непрерывных процессов // АиТ. 2012. № 10, С. 3-15.
20. Расина И.В. Дискретно-непрерывные модели и оптимизация управляемых процессов // Программные системы: теория и приложения. Переславль-Залесский: Институт программных систем имени А.К. Айламазяна Российской академии наук. Электрон. научн. журн. 2011. № 5(9). С. 49-72. URL: http://psta.psiras.ru/read/psta2011_5_49-72.pdf.
21. Константинов Г.Н. Нормирование воздействий на динамические системы. Иркутск: Изд-во Иркут. ун-та, 1983.
22. Гурман В.И., Расина И.В., Трушкова Е.А., Усенко О.В. Иерархическая модель неоднородной дискретной системы и ее приложения // Управление большими системами. Вып. 41. М.: Изд-во ИПУ РАН, 2013. С. 249-269.

Расина Ирина Викторовна, кандидат физико-математических наук, зав. кафедрой математических и естественнонаучных дисциплин Сибирской академии права, экономики и управления; тел. (3952)422869, e-mail: irinarasina@gmail.com

Усенко Олег Валерьевич, старший преподаватель кафедры информатики и программной инженерии Сибирской академии права, экономики и управления; тел. (3952)332521, e-mail: o.v.usenko@gmail.com

Rasina Irina Victorovna, candidate of physical and mathematical sciences, head of department of mathematical and natural sciences, Siberian Academy of Law, Economics and Management, e-mail: irinarasina@gmail.com

Usenko Oleg Valeryevich, senior lecturer, department of computer science and software engineering, Siberian Academy of Law, Economics and Management, e-mail: o.v.usenko@gmail.com

УДК 517.97

© И.-Х.Д. Хишектыева

АЛГОРИТМ ОПТИМИЗАЦИИ ПАРАМЕТРОВ НЕЛИНЕЙНЫХ ДИНАМИЧЕСКИХ СИСТЕМ¹

Приводятся результаты численных экспериментов по анализу эффективности метода оптимизации параметров нелинейных динамических систем, основанного на поиске неподвижных точек специального оператора проектирования.

Ключевые слова: параметрическая оптимизация, задача о неподвижной точке.

© I.-Kh.D. Khishektueva

ALGORITHM FOR OPTIMIZATION OF PARAMETERS OF NONLINEAR DYNAMIC SYSTEMS

The results of numerical experiments on the analysis of efficiency of the method for optimization of parameters of nonlinear dynamic systems, based on the search of fixed points of special projection operator are given.

Keywords: parametric optimization, problem of a fixed point.

1. Метод решения

Рассматривается задача оптимизации управляющих параметров

$$\Phi(u) = \varphi(x(t_1)) + \int_T F(x(t), u, t) dt \rightarrow \min_{u \in U}, \quad (1)$$

$$\dot{x}(t) = f(x(t), u, t), \quad x(t_0) = x^0, \quad t \in T = [t_0, t_1], \quad (2)$$

в которой $x(t) = (x_1(t), \dots, x_n(t))$ – вектор состояния, $u = (u_1, \dots, u_m)$ – вектор управляющих параметров со значениями в выпуклом множестве $U \subset R^m$. Начальное состояние x^0 и промежуток управления T заданы.

Предполагаются выполненными следующие условия:

- 1) функция $\varphi(x)$ непрерывно-дифференцируема на R^n , вектор-функция $F(x, u, t)$, векторная функция $f(x, u, t)$ и их производные $F_x(x, u, t)$, $F_u(x, u, t)$, $f_x(x, u, t)$, $f_u(x, u, t)$ непрерывны по совокупности аргументов (x, u, t) на множестве $R^n \times U \times T$;
- 2) функция $f(x, u, t)$ удовлетворяет условию Липшица по x в $R^n \times U \times T$ с константой $L > 0$

$$\|f(x, u, t) - f(y, u, t)\| \leq L \|x - y\|.$$

Строится релаксационная (улучшающаяся) последовательность управлений, определяемых на каждой итерации в ходе решения задач улучшения, формулируемых в следующей форме: для заданного управления $u \in U$ требуется найти управление $v \in U$ с условием $\Delta_v \Phi(u) \leq 0$.

Для реализации задачи улучшения предлагается решить задачу о неподвижной точке [1]:

$$v = P_U(u + \alpha(\int_T H_u(p(t, u, v), x(t, v), u, t) dt + s)), \quad u \in U, v \in U, s \in R^m, \quad (3)$$

где P_U – оператор проектирования на множество U в евклидовой норме, $\alpha > 0$ – параметр проектирования, $H(p, x, u, t)$ – функция Понтрягина, $x(t, v)$ – решение системы (2) при $u = v$, $p(t, u, v)$ – решение дифференциально-алгебраической сопряженной системы вида

$$\begin{aligned} \dot{p}(t) &= -H_x(p(t), x(t), w, t) - r(t), \\ \langle H_x(p(t), x(t), w, t), y(t) - x(t) \rangle + \langle r(t), y(t) - x(t) \rangle &= \Delta_{y(t)} H(p(t), x(t), w, t) \end{aligned}$$

¹ Работа выполнена при финансовой поддержке РФФИ (проекты № 12-01-00914, 12-01-98011, 13-01-92200)

с краевыми условиями

$$p(t_1) = -\varphi_x(x(t_1)) - q, \\ \langle \varphi_x(x(t_1)), y(t_1) - x(t_1) \rangle + \langle q, y(t_1) - x(t_1) \rangle = \Delta_{y(t_1)} \varphi(x(t_1)).$$

Величина s определяется соотношением

$$\int_T \Delta_v H(p(t, u, v), x(t, v), u, t) dt = \left\langle \int_T H_u(p(t, u, v), x(t, v), u, t) dt, v - u \right\rangle + \langle s, v - u \rangle \quad (4)$$

Для решения задачи о неподвижной точке (3) используется итерационный алгоритм при $k \geq 0$:

$$v^{k+1} = P_U(u + \alpha(\int_T H_u(p(t, u, v^k), x(t, v^k), u, t) dt + s^k)), \quad v^0 \in U,$$

Величина s^k удовлетворяет соотношению (4) при $v = v^k$.

2. Численные эксперименты

Численный расчет задачи о неподвижной точке (3) проводился до первого строгого улучшения управления $u \in U$. Далее строилась новая задача о неподвижной точке и итерационный алгоритм повторялся. В качестве критерия остановки итерационного процесса выбиралось условие $\|v^{k+1} - v^k\| \leq \varepsilon \|v^k\|$ в евклидовой норме, где $\varepsilon > 0$ – заданная точность. Численное интегрирование дифференциальных систем реализовывалось методом Рунге-Кутты-Вернера переменного (5-6) порядка и шага с помощью программы DIVPRK библиотеки IMSL Фортран PowerStation 4.0. Значения вычисленных фазовых и сопряженных переменных запоминались в узлах заданной равномерной сетки с шагом дискретизации $d = 10^{-3}$.

Пример 1.

$$\Phi(u) = \frac{1}{2} \int_0^1 (x^2(t) + u^2) dt \rightarrow \min,$$

$$\dot{x}(t) = u, \quad x(0) = 1, \quad u \in U = [-1; 1], \quad t \in T = [0, 1].$$

Рассматривается начальное управление $u = 0$ с соответствующей фазовой траекторией $x(t, u) = 1, t \in T$ и значением целевой функции $\Phi(u) = \frac{1}{2}$. Дифференциально-алгебраическая сопряженная система принимает вид

$$\dot{p}(t) = x(t, u) - r(t),$$

$$-x(t, u)(vt + 1 - x(t, u)) + r(t)(vt + 1 - x(t, u)) = \frac{1}{2}(x(t, u)^2 - (vt + 1)^2),$$

$$p(1) = -q,$$

$$q(v + 1 - x(t_1, u)) = 0.$$

После преобразований она сводится к дифференциальной системе

$$\dot{p}(t) = \frac{1}{2}(x(t, u) + vt + 1), \quad p(1) = 0.$$

Задача о неподвижной точке и соответствующий итерационный процесс имеют вид

$$v = P_U(u + \alpha(\int_0^1 p(t, u, v) dt - \frac{1}{2}(u + v))),$$

$$v^{k+1} = P_U(u + \alpha(\int_0^1 p(t, u, v^k) dt - \frac{1}{2}(u + v^k))).$$

Результаты численных расчетов для разных расчетных параметров представлены в таблице 1.

Таблица 1

Результаты численных расчетов (пример 1)

v^0	α	ε	v^*	$\Phi(v^*)$	Число итераций
0	10^{-2}	10^{-4}	-0,3750	0,40625	150
		10^{-6}	-0,3750	0,40625	150
	10^{-1}	10^{-4}	-0,3750	0,40625	25
		10^{-6}	-0,3750	0,40625	27
	0,2	10^{-4}	-0,3750	0,40625	15
		10^{-6}	-0,3750	0,40625	17
	0,3	10^{-4}	-0,3750	0,40625	10
		10^{-6}	-0,3750	0,40625	16
	0,4	10^{-4}	-0,3750	0,40625	10
		10^{-6}	-0,3750	0,40625	14
	0,5	10^{-4}	-0,3750	0,40625	20
		10^{-6}	0,3750	0,40625	25
	0,8	10^{-4}	-0,3750	0,40625	86
		10^{-6}	-0,3750	0,40625	118
	1	10^{-4}	-0,3750	0,40625	43
		10^{-6}	-0,3750	0,40625	62
-1	10^{-2}	10^{-4}	-0,3750	0,40625	184
		10^{-6}	-0,3750	0,40625	184
	10^{-1}	10^{-4}	-0,3750	0,40625	25
		10^{-6}	-0,3750	0,40625	33
	0,2	10^{-4}	-0,3750	0,40625	11
		10^{-6}	-0,3750	0,40625	16
	0,4	10^{-4}	-0,3750	0,40625	11
		10^{-6}	-0,3750	0,40625	15
	0,8	10^{-4}	-0,3750	0,40625	14
		10^{-6}	-0,3750	0,40625	19
	1	10^{-4}	-0,3750	0,40625	47
		10^{-6}	-0,3750	0,40625	68
1	10^{-2}	10^{-4}	-0,3750	0,40625	123
		10^{-6}	-0,3750	0,40625	183
	10^{-1}	10^{-4}	-0,3750	0,40625	20
		10^{-6}	-0,3750	0,40625	26
	0,4	10^{-4}	-0,3750	0,40625	14
		10^{-6}	-0,3750	0,40625	18
	1	10^{-4}	-0,3750	0,40625	47
		10^{-6}	-0,3750	0,40625	68

Улучшаемое управление $u = 0$. В качестве начальных приближений для итерационного процесса рассматривались управления $v^0 = 0, -1, 1$. Параметр α варьировался от 10^{-2} до 1,2. Наилучшие результаты по точности достигнуты при $\alpha = 0,4$, расчетное значение параметра $v^* = -0,3750$, соответствующее значение целевой функции $\Phi(v^*) = 0,40625$. При $\alpha > 1,1$ наблюдалась расходимость итерационного процесса. В данном примере легко аналитически определяется оптимальное управление $u^* = -\frac{3}{8}$.

Пример 2. «Самолет с автопилотом» [2]

$$\begin{aligned} \dot{z}_1 &= z_2, \dot{z}_2 = z_3, \dot{z}_3 = z_4, \dot{z}_4 = z_5, \dot{z}_5 = z_6, \\ \dot{z}_6 &= -291.1z_1 - 771.2z_2 - 1146.5z_3 - 364.2z_4 - 107.4z_5 - 16.4z_6, \\ z_1(0) &= 0, z_2(0) = 20, z_3(0) = \dots = z_6(0) = 0. \end{aligned}$$

Аппроксимация исходной системы задавалась в виде системы второго порядка [3]

$$\begin{aligned} \dot{x}_1 &= x_2, \quad \dot{x}_2 = u_2x_1 + u_1x_2, \\ x_1(0) &= z_1(0) = 0, \quad x_2(0) = z_2(0) = 20. \end{aligned}$$

Мерой близости решений служил интеграл:

$$I = \int_0^7 \sum_{i=1}^2 \lambda_i [x_i(t) - z_i(t)]^2 dt.$$

В данном случае дифференциально-алгебраическая сопряженная система имеет вид

$$\begin{cases} \dot{p}_1(t) = -p_2(t)u_2 + 2\lambda_1x_1(t,u) - 2\lambda_1z_1(t) - r_1(t), \\ \dot{p}_2(t) = -p_1(t) - p_2(t)u_1 + 2\lambda_2x_2(t,u) - 2\lambda_2z_2(t) - r_2(t), \\ (p_2(t)u_2 - 2\lambda_1x_1(t,u) + 2\lambda_1z_1(t))(x_1(t,v) - x_1(t,u)) + r_1(t)(x_1(t,v) - x_1(t,u)) = \\ (p_2(t)u_2 + 2\lambda_1z_1(t))(x_1(t,v) - x_1(t,u)) - \lambda_1(x_1(t,v)^2 - x_1(t,u)^2), \\ (p_1(t) + p_2(t)u_1 - 2\lambda_2x_2(t,u) + 2\lambda_2z_2(t))(x_2(t,v) - x_2(t,u)) + r_2(t)(x_2(t,v) - x_2(t,u)) = \\ (p_1(t) + p_2(t)u_1 + 2\lambda_2z_2(t))(x_2(t,v) - x_2(t,u)) - \lambda_2(x_2(t,v)^2 - x_2(t,u)^2) \end{cases}$$

$$\begin{aligned} p_1(1) &= -q_1, \\ p_2(1) &= -q_2, \\ q_1(x_1(t_1,v) - x_1(t_1,u)) &= 0, \\ q_2(x_2(t_1,v) - x_2(t_1,u)) &= 0. \end{aligned}$$

После преобразований система сводится к дифференциальной системе

$$\begin{cases} \dot{p}_1(t) = -p_2(t)u_2 + \lambda_1x_1(t,u) - 2\lambda_1z_1(t) + \lambda_1x_1(t,v), & p_1(1) = 0, \\ \dot{p}_2(t) = -p_1(t) - p_2(t)u_1 + \lambda_2x_2(t,u) - 2\lambda_2z_2(t) + \lambda_2x_2(t,v), & p_2(1) = 0. \end{cases}$$

Задача о неподвижной точке принимает форму

$$\begin{aligned} v_1 &= P_U(u_1 + \alpha(\int_0^7 p_2(t,u,v)x_2(t,v)dt + s_1)), \\ v_2 &= P_U(u_2 + \alpha(\int_0^7 p_2(t,u,v)x_1(t,v)dt + s_2)), \\ \int_0^7 p_2(t,u,v)x_2(t,v)(v_1 - u_1)dt &= \int_0^7 p_2(t,u,v)x_2(t,v)dt \cdot (v_1 - u_1) + s_1(v_1 - u_1), \\ \int_0^7 p_2(t,u,v)x_1(t,v)(v_2 - u_2)dt &= \int_0^7 p_2(t,u,v)x_1(t,v)dt \cdot (v_2 - u_2) + s_2(v_2 - u_2). \end{aligned}$$

Отсюда получаем уравнения для неподвижных точек в виде

$$\begin{aligned} v_1 &= u_1 + \alpha \int_0^7 p_2(t,u,v)x_2(t,v)dt, \\ v_2 &= u_2 + \alpha \int_0^7 p_2(t,u,v)x_1(t,v)dt, \end{aligned}$$

и соответствующий итерационный процесс

$$v_1^{k+1} = u_1 + \alpha \int_0^7 p_2(t, u, v^k) x_2(t, v^k) dt,$$

$$v_2^{k+1} = u_2 + \alpha \int_0^7 p_2(t, u, v^k) x_1(t, v^k) dt.$$

Задача решалась для фиксированных значений: а) $\lambda_1 = 0, \lambda_2 = 1$; б) $\lambda_1 = 1, \lambda_2 = 1$. Результаты расчетов представлены соответственно в таблицах 2 и 3.

Таблица 2

Результаты численных расчетов (пример 2, а)

v^0	α	ε	v^*	$\Phi(v^*)$	Число итераций
-0,6 -0,25	10^{-5}	10^{-4}	-0,5573 -0,3536	15,089	709
	10^{-3}	10^{-4}	-0,5573 -0,3536	15,089	171
-1 -0,5	10^{-5}	10^{-4}	-0,5573 -0,3536	15,089	710
	10^{-3}	10^{-4}	-0,5573 -0,3536	15,089	172
-0,5453 -0,2899	10^{-5}	10^{-4}	-0,5573 -0,3536	15,089	751
	10^{-3}	10^{-4}	-0,5573 -0,3536	15,089	168

В качестве начального улучшаемого управления рассматривались параметры $u_1 = -0.6, u_2 = -0.25$, указанные в [2] как оптимальные. За начальные приближения для итерационного процесса принимались значения: $v_1^0 = -0.6, v_2^0 = -0.25, v_1^0 = -1, v_2^0 = -0.5$ и $v_1^0 = -0.5453, v_2^0 = -0.2899$. Параметр α варьировался от 10^{-5} до 10^{-2} . Наилучшие результаты достигнуты при $\alpha = 10^{-3}$, расчетные значения параметров $v_1^* = -0.5573, v_2^* = -0.3536$, соответствующее значение целевой функции $\Phi(v^*) = 15.0789$. При $\alpha > 0.001$ итерационный процесс не сходится. В [2] указано оптимальное значение целевой функции 25,8030.

Таблица 3

Результаты численных расчетов (пример 2, б)

v^0	α	ε	v^*	$\Phi(v^*)$	Число итераций
-0,6 -0,25	10^{-5}	10^{-4}	-0,5486 -0,2927	28,526	71
-0,3 -0,1	10^{-5}	10^{-4}	-0,5486 -0,2927	28,526	68
-0,5589 -0,2940	10^{-5}	10^{-4}	-0,5486 -0,2927	28,526	144

Во втором случае аналогично в качестве начального улучшаемого управления рассматривались параметры $u_1 = -0.6, u_2 = -0.25$. За начальные приближения для итерационного процесса были взяты значения: $v_1^0 = -0.6, v_2^0 = -0.25, v_1^0 = -0.3, v_2^0 = -0.1$ и $v_1^0 = -0.5589, v_2^0 = -0.2940$. Расчетные значения параметров $v_1^* = -0.5486, v_2^* = -0.2927$, соответствующее значение целевой функции $\Phi(v^*) = 28.526$. При $\alpha > 10^{-5}$ итерационный процесс не сходится. В [2] указано оптимальное значение

ние целевой функции 36,1646.

Заключение

Проделанные численные расчеты и полученные результаты позволяют сделать вывод о достаточной эффективности предлагаемого метода в задачах параметрической оптимизации нелинейных систем. В задаче управления самолета с автопилотом получено меньшее значение целевой функции по сравнению с известными результатами. Расчеты показывают, что скорость сходимости алгоритма существенно регулируется выбором проекционного параметра α .

Литература

1. Булдаев А.С., Очирбат Б., Хишектуева И.-Х.Д. Поиск неподвижных точек операторов проектирования в задачах параметрической оптимизации систем // Вестн. Бурят. гос. ун-та. Математика, информатика. - Улан-Удэ: Изд-во Бурят. гос. ун-та, 2012. – № 2. – С. 4-14.
2. Каляев А.В. Расчет переходного процесса в линейных системах методом понижения порядка дифференциального уравнения // АиТ. – 1959. – Т. 20, № 9. – С. 1171-1179.
3. Тятюшкин А.И. Численные методы и программные средства оптимизации управляемых систем. – Новосибирск: Наука. Сиб. отд-ние, 1992. – 193 с.

Хишектуева Ишин-Хорло Дамбадоржиевна, инженер-программист Научно-образовательного и инновационного центра системных исследований и автоматизации (НОИЦ СИА) Бурятского государственного университета, тел. (914) 8363885, e-mail: ishin@ulanovka.ru

Khishektueva Ishin-Khorlo Dambadorzhievna, software engineer, Scientific-Educational and Innovative Center for Systems Study and Automation (SEIC SSA), Buryat State University, e-mail: ishin@ulanovka.ru

2. Информационные системы и технологии

УДК 519.693+519.688+519.71+519.1

© *Н.В. Абасов, М.Ю. Чернышов, А.В. Каверзина, Е.Н. Осипчук, И.А. Чернышова*

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ СИНТЕЗА ПРИКЛАДНЫХ ПРОГРАММНЫХ МОДЕЛЕЙ И МЕТАМОДЕЛЕЙ НА ОСНОВЕ ПРИНЦИПОВ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Описывается программный комплекс (ПК), разработанный авторами на основе языка LMPL и представляющий собой средство синтеза прикладных программных моделей и метамodelей, строящихся на основе принципов математического программирования (МП). Реализованы прикладные программные версии ПК.

Ключевые слова: язык LMPL, моделирование, метамоделирование, математическое программирование, автоматическое построение и преобразование моделей.

© *N.V. Abasov, M.Yu. Chernyshov, A.V. Kaverzina, E.N. Osipchuk, I.A. Chernyshova*

A SOFTWARE COMPLEX FOR SYNTHESIS OF APPLIED SOFTWARE MODELS AND META-MODELS ON THE BASIS OF MATHEMATICAL PROGRAMMING PRINCIPLES

A software complex (SC) elaborated by the authors on the basis of the language LMPL and representing a software tool for synthesis of applied software models and meta-models constructed on the basis of mathematical programming (MP) principles is described. The software versions of the SC have been implemented.

Keywords: language LMPL, modeling, meta-modeling, mathematical programming, automatic construction and transformation of models.

Оригинальный программный комплекс (ПК) LMPL-Software разработан авторами на основе языка, получившего название Light Mathematical Programming Language (ниже LMPL). LMPL построен на основе прототипов (языка MPL [11] и его расширенного (augmented) варианта AMPL [8]). Идея LMPL основана на принципах декларативного описания логики программ языка ОЛФИС. LMPL предназначен для компактного наглядного моделирования, построения логического вывода при принятии решений и решения задач математического программирования (МП). Технически он обеспечивает упрощенный синтез моделей и метамodelей (ММ). Под ММ понимается обобщенная модель, пригодная для описания множества простых моделей [1, 4]. ММ включает в себя обобщенную базу знаний и конструкторы для синтеза прикладных моделей. LMPL характеризуется компактной, наглядной формой декларативного представления МП-моделей (в т. ч. уравнений, ограничений, критерий оптимизации) в виде набора блоков (табл. 1) с определением индексных переменных и упрощенный синтаксис (нет операторов, характерных для императивных языков программирования). Характеристики LMPL подробно описаны в [7]. Табл. 1 представляет перечень базовых блоков языка LMPL с кратким описанием их функций, табл. 2 – синтаксис языка LMPL в расширенной форме Бэкуса-Наура. Заметим, что любая прикладная модель, записанная в терминах LMPL, после преобразований автоматически приводится к классической форме соответствующей исходной МП-задаче, т. е. к форме, во-первых, удобной для исследователя, а, во-вторых, – для автоматической передачи различным решателям задач (Ip_solve, COPL_QP, GAMS [15], AMPL и др.).

ПК LMPL-Software, построенный на LMPL, обеспечивает синтез, модификацию МП-моделей и метамodelей, позволяет моделировать процессы принятия решений. ПК представляет моделируемый объект на языке LMPL и обеспечивает выполнение следующих операций: 1) построение метамodelи из базовых блоков (рис. 1); 2) построение прикладной модели на ее основе; 3) сравнение модели с прототипом (если есть), выявление структурных отличий, выделение общих частей; 4) построение графов связей между объектами модели (рис. 2); 5) построение логического вывода на модели и для

этого поддержку проведения многоитерационных расчетов, включая стохастическую оптимизацию; б) поддержку автоматических преобразований моделей (эта функция может быть использована, например, в исследовании алгоритмов). Компактная (портативная) и кроссплатформенная реализация ПК (в виде библиотек на языке Lua с использованием пакета Ip_solve в качестве базового решателя) допускает возможность подключения почти любых универсальных внешних ПК, пригодных для решения МП-задач.

Таблица 1

Базовый набор основных функциональных блоков языка LMPL и их описание

блок	описание блока
TITLE	идентификатор модели, класс оптимизационной задачи; опционально указывается предполагаемый решатель
IN	входные переменные и их типы: дискретные (целочисленные INT, логические и бинарные BIN); свободные (FREE); с ненулевыми показателями (SOS)
OUT	выходные переменные
CONST	константы
OPT	целевая функция
EQ	система уравнений, часть уравнений определяется в блоке VAR
VAR	промежуточные переменные из уравнений блока EQ
LIMIT[N]	различные классы ограничений
INDEX	пределы для индексных переменных

Синтаксис языка LMPL в расширенной форме Бэкуса-Наура представлен в табл. 2.

Таблица 2

Синтаксис языка LMPL в расширенной форме Бэкуса-Наура

базовые блоки	
<pre>block = bnam ":" {bbody [";"]} bnam = "TITLE" "IN" "OUT" "CONST" "OPT" "EQ" "VAR" "LIMIT"[nat] "INDEX" bbody = btitle binp bout bconst bopt beq bvar blimit bind</pre>	
блок TITLE: идентификатор модели, классы задач, решатели	
<pre>btitle = idmod "," optc ["," slv] idmod = name optc = "lp" "nlp" "mip" "lfp" "rmip" "minlp" "rminlp" "mcp" "cns" slv = "lp_solve" "GAMS" "AMPL" "CPLEX" "MINOS"</pre>	
блок IN: входные переменные	блок OUT: выходные переменные
<pre>binp= vtype var var="exp name="seq var = name name "[" exp "]" vtype = "SOS" "BIN" "INT" "FREE"</pre>	<pre>bout = varv varv = {var ","} var</pre>
блок CONST: константы	блок VAR: промежуточные переменные
<pre>bconst = var "=" exp name "=" seq</pre>	<pre>bvar = var "=" exp</pre>
блок OPT: критерий оптимизации	блок EQ: система уравнений
<pre>bopt = exp "->" optm optm = "min" "max"</pre>	<pre>beq = exp "=" exp</pre>
блок LIMIT: ограничения	блок INDEX: индексы
<pre>blimit = exp comop exp exp "*=" ["[" "("] exp "," exp ["]" ")"]</pre>	<pre>bindex = name "*=" exp "," exp</pre>

выражения	последовательности
$exp = term \mid [unop]\{exp\} \ binop \ exp \mid$ $"(" \ exp \ ")"$ $fnam \ "(" \ exp \ ")"$ $term = digit \mid var$	$seq = "{" \ expv \ "}"$ $expv = \{exp \ ", \ " \} \ exp$
операции	функции
$binop = "+" \mid "-" \mid "*" \mid "/" \mid "^"$ $unop = "-"$ $comop = "<" \mid ">" \mid "<=" \mid ">="$	$fnam = "sum" \mid "prod" \mid "abs"$
где digit - десятичное число, name - идентификатор	

Функция метамоделирования как синтеза предполагает следующие этапы: 1) задание шаблона метамодели; 2) объявление блоков, входящих в метамодель (например, включение блока базовых уравнений моделей, констант); 3) объявление заменяемых в процессе генерации модели промежуточных переменных (метаварiable); 4) задание параметров метамодели (например, названия объекта, режима работы, начального состояния) и выбор предполагаемого решателя задач (его характеристик); 5) генерация модели на языке LMPL с помощью разработанного модуля-конвертера; 6) приведение LMPL-модели к формату решателя. В случае необходимости может быть выполнен анализ результатов, полученных решателем, и генерация отчетов (в текстовом и графическом видах).

Процесс моделирования предполагает: 1) объявление набора случайных параметров и их характеристик; 2) задание параметров управления блоком стохастической оптимизации (количество итераций, шаг дискретизации); 3) генерацию множества детерминированных моделей (генерации случайных чисел); 4) решение оптимизационной задачи с сохранением текущих оптимальных показателей по каждой итерации; 5) обработку накопленной статистики; 6) формирование итоговых таблиц с вероятностными распределениями.

Структура исходного текста конкретной прикладной модели строится из функциональных блоков, которым сопоставляются класс задачи, идентификатор типа модели, тип решателя задачи, целевая функция, функция сравнения, ограничения, индексные переменные, переменные и константы, соответствующие задачам и условиям моделирования. На рис. 1 приведена универсальная структура прикладной модели на LMPL (базовый набор основных функциональных блоков языка LMPL, функции, переменные и константы). Очевидно, что LMPL позволяет описать модель в форме, являющейся в некотором смысле отображением матлогической формы, поскольку предполагает использование множества индексных переменных, констант, операторов и кванторов. Представления о переменных, множествах, граничных условиях позволяют обращаться к ним, используя соответствующие индексы. Опционально может быть указан предполагаемый решатель МП-задач. При синтезе моделей важной является возможность их представления в декларативной форме. В случае императивной формы, в терминах которой строится большинство программ и моделей, логика синтеза существенно осложнена необходимостью последовательного процесса синтеза модели. Декларативная форма дает возможность эффективно описывать модели, а также создавать гибкие средства для последующего исследования алгоритмов, их развития и сопровождения.

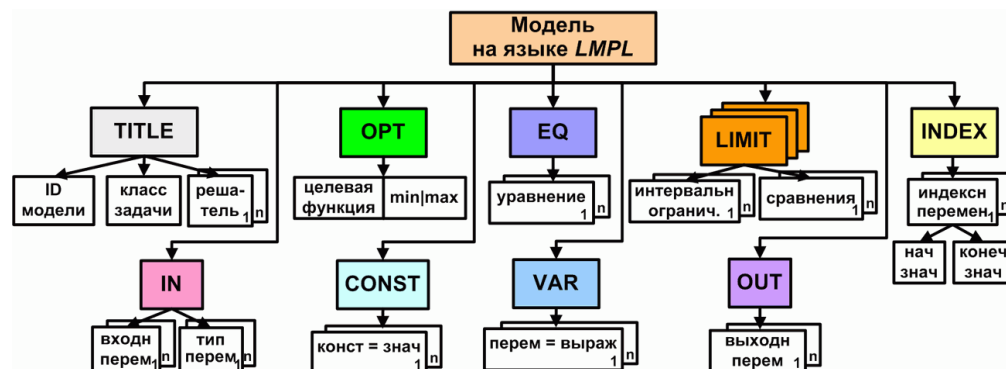


Рис. 1. Универсальная структура прикладной модели на языке LMPL (функциональные блоки, функции, переменные и константы)

ПК LMPL-Software позволяет выполнять следующие операции синтеза: а) построение модели из частей; б) сравнение модели с прототипом и выявление структурных отличий; в) построение графов связей между объектами модели (рис. 2), где в качестве объектов присутствуют различные переменные или множества переменных.

ПК LMPL-Software позволяет выполнять следующие операции анализа моделей: 1) анализ графов связей между объектами модели (рис. 2) (т. е. между различными переменными и множествами переменных); 2) выявление структурных отличий модели от прототипа; 3) выделение общих частей модели и прототипа. Она позволяет строить метамодель объекта прикладной модели на ее основе, а также выполнять преобразования такой модели, связанные с построением новых моделей из блоков, выделенных при анализе.

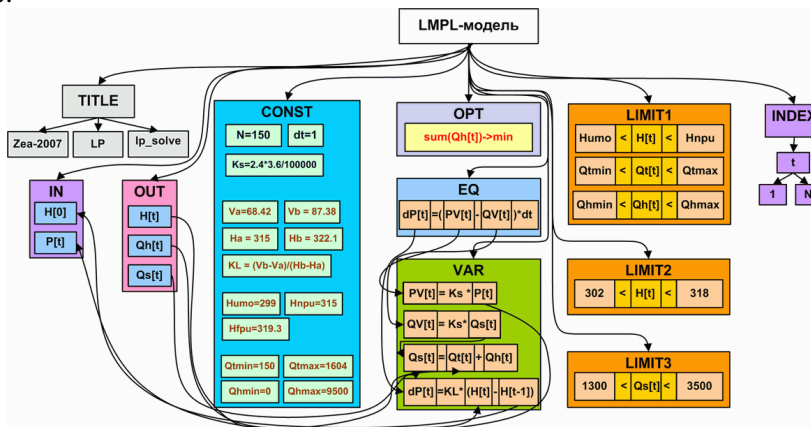


Рис. 2. Пример графа связей между объектами модели

Рассмотрим лишь два примера метамоделирования: 1) пример метамоделирования режимов ГЭС и 2) пример метамоделирования дискурсивных формаций (ДФ) в метамодели дискурса, связанного с редакционно-издательской деятельностью (РИД).

Пример метамоделирования режимов ГЭС. При рассмотрении и сравнении различных моделей режимов ГЭС у специалистов обычно появляется необходимость объединить их в единую систему, построив целостную модель. Это можно сделать на основе ММ и описания связей между моделями.

ММ предполагает получение модельного представления процесса выбора оптимальных режимов функционирования ГЭС. БЗ в ММ содержит фрагменты различных типов модельного представления режимов, для которых можно выделить, например, основные уравнения, переменные, константы, критерии оптимизации и ограничения. Это позволяет использовать средства метамоделирования ПК LMPL-Software для автоматической генерации конкретных моделей режимов и выбора лучших. Пример модельного представления процесса выбора оптимальных режимов ГЭС, полученного на основе принципа метамоделирования, представлен на рис. 3.



Рис. 3. Пример модельного представления процесса выбора оптимальных режимов функционирования ГЭС, полученного на основе принципа метамоделирования

Метамодель может быть построена в детерминированном и стохастическом вариантах. Она может быть стохастической в том смысле, что в ней один или несколько параметров описываются некоторой функцией распределения.

Процесс метамоделирования в детерминированном варианте, предполагающем получение модельного представления процесса выбора оптимальных режимов ГЭС, делится на следующие этапы: 1) задание шаблона данной метамодели; 2) объявление блоков, входящих в метамодель (в том числе, включение блока основных уравнений моделей и констант); 3) объявление метaperеменных, заменяемых в процессе генерации данной прикладной модели; 4) задание параметров метамодели (объект, режим работы ГЭС, параметры, отражающие начальные состояния, т.е. значения переменных и констант); 5) выбор предполагаемого решателя задач; 6) генерация модели режима ГЭС на языке LMPL с помощью модуля-конвертера; 6) приведение построенной LMPL-модели к формату решателя задач; 7) анализ результатов, полученных решателем; 8) формирование отчетов в текстовом и графическом виде.

Процесс моделирования в стохастическом режиме предполагает: 1) объявление набора случайных параметров и их характеристик; 2) задание параметров управления стохастическим блоком (количество итераций, шаг дискретизации); 3) создание множества детерминированных моделей с помощью генератора случайных чисел (ГСЧ) произвольного распределения; 4) решение оптимизационной задачи с сохранением текущих оптимальных показателей на каждой итерации; 5) обработку накопленной статистики, формирование итоговой таблицы с вероятностными распределениями.

Пример метамоделирования функциональной структуры ДФ с помощью семантических сетей. Описание исследования дискурса с помощью семантических сетей можно встретить еще в работах начала 1970-х гг. [14]. В отношении уровня такого описания необходимо сделать скидку, учитывая прогресс представлений ко второму десятилетию XXI в.

Моделирование дискурсивных отношений предполагает определение предусловий (множества условий, представленных, например, логическими формулами, которые истинны до того, как моделирование выполняется) и построение логической (или сетевой) модели вывода, связанной, например, с принятием решений. Для представления знаний даже о небольшом сегменте дискурса в РИД требуется построить систему моделей, последовательно представляющих достижение целей РИД. В самом простом случае это могут быть, например: 1) объектная целевая модель РИД (см. рис. 4) и 2) упрощенная функционально-деятельностная модель (ФДМ) дискурсивного принятия решений, связанных с РИД (рис. 5).



Рис. 4. Объектная целевая модель РИД (О – рукопись, представляемая для опубликования; Е – конечный продукт в форме публикации)

Такая ФДМ должна представлять собой модель принятия редактором издательства решения об опубликовании рукописи автора. Анализ возможных основ для построения модели показал, что весьма эффективным средством представления структуры функциональных отношений в ДФ может быть сеть Петри типа «предикат–переход» с n -местными предикатами, допускающими переход по некоторому условию. На основе такой сети можно строить имитационные модели, отражающие динамику функциональных отношений между объектами и, возможно, даже смысловых отношений между процессами.

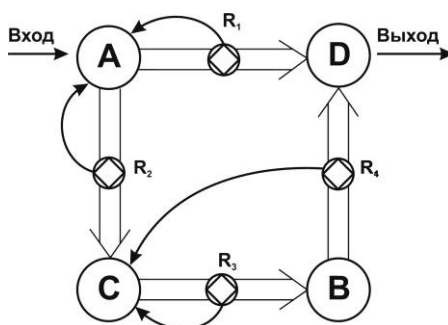


Рис. 5. Упрощенная ФДМ дискурсивного принятия решений, связанных с РИД на основе сети Петри

Поясним использование сети Петри применительно к задаче моделирования отношений деятельности, связанной с РИД, опосредованных дискурсом. В ФДМ на рис. 5: *A* – деятельность автора, связанная с опубликованием рукописи; *R*₁ – совокупность требований к автору, связанных с опубликованием рукописи; *C* – деятельность редактора, связанная с опубликованием рукописи; *R*₂ – совокупность требований, предъявляемых к процессу редактирования рукописи редактором; *R*₃ – совокупность требований, предъявляемых к процессу научной экспертизы рукописи рецензентом; *B* – деятельность рецензента, связанная с опубликованием рукописи; *R*₄ – совокупность правовых требований, предъявляемых к процессу опубликования; *D* – деятельность, связанная с выдачей дирекцией разрешения на опубликование.

Описание логики отношений внутри такой ФДМ возможно в терминах логики предикатов первого порядка.

Заметим, что переход от *A* к *D* осложнен предикатом $R_1: \forall x \forall t [A(x) \& R_1(x, t) \rightarrow D(t)]$, переход от *A* к *C* – предикатом $R_2: \forall x \forall y [B(x) \& R_4(x, y) \rightarrow D(y)]$.

Для переходов от *C* к *B* и от *B* к *D* имеем, соответственно: $\forall y \forall z [C(y) \& R_3(y, z) \rightarrow B(z)]$ и $\forall z \forall t [B(z) \& R_2(z, t) \rightarrow D(t)]$.

Разомкнутая функциональная модель представляет собой модель «вход–выход». В нашем случае ФДМ на основе сети Петри, построенная, чтобы отразить упрощенную совокупность отношений, связанных с РИД, позволяет сделать вывод о возможности принятия решения об опубликовании рукописи. На вход такой ФДМ поступают численные значения основных переменных и констант процесса, основные начальные и граничные условия процесса; на ее выходе возможно получение лишь двух типов сигналов-решений: 0 или 1, т. е. «да» или «нет». Применительно к РИД это означает, что разрешение на публикацию конечного продукта может быть получено или нет. Решение о возможности опубликования может быть получено, если все агенты такой модели (намеренно предельно упрощенной в нашем случае) дискурсивного принятия решений (автор, редактор и рецензент и др.) выполняют все требования, связанные с их функциями и сферой полномочий в системе РИД.

Основными условиями принятия решения являются: 1) оценка качества рукописи автором (критерий качества рукописи (*C*_{au}) выражается количественным показателем $C_{au} = \{1, 10\}$ при допустимой оценке, например, $C_{au} = \{9, 10\}$); 2) оценка качества рукописи редактором (критерий качества рукописи (*C*_{ed}) выражается количественным показателем $C_{ed} = \{1, 10\}$ при допустимой оценке, например, $C_{ed} = \{8, 10\}$, что и является, по сути, планкой импакт-фактора данного журнала); 3) оценка качества рукописи рецензентом (критерий качества рукописи (*C*_{rev}) выражается количественным показателем $C_{rev} = \{1, 10\}$ при допустимой оценке, например, $C_{rev} = \{7, 10\}$); 4) выполнение всех правовых требований (критерий качества рукописи (*C*_{law}) выражается количественным показателем $C_{law} = \{0, 1\}$ при допустимой оценке, например, $C_{law} = 1$).

Наличие модели ФДМ и определенность начальных и граничных условий позволяют построить модель или даже метамодель в форме компьютерной программы принятия решений о допущении рукописи к публикации в данном издании.

Система, предполагающая построение компьютерной модели ФДМ средствами метамоделирования. Попытки исследования смысловой связности дискурса предпринимались еще в 1970–80-х гг. [9; 10; 12; 13]. В статье Дж. Левина и Дж. Мура дискурс имитировался как «диалоговые игры», причем авторами была предпринята попытка представить мета-коммуникативные структуры дискурсивного взаимодействия [9]. Позднее убедительное обсуждение проблем металингвистики межкультурного делового общения было предложено в работах Н.К. Рябцевой [5]. Потребовались десятилетия отшлифовки идей, прежде чем концепция метамоделирования дискурса нашла воплощение в практически работающих компьютерных программных комплексах [7].

ПК LMPL-Software как средство моделирования процессов построения дискурса и ДФ. Достоинством ПК является наглядность процесса моделирования [3]. Базовый набор основных функциональных блоков, представленных на рис. 1, может репрезентировать метамодель ДФ. В таком случае ПК позволит определить: 1) тип модели ДФ; 2) класс ставящейся задачи (построение модели ДФ, ее анализ, выполнение сравнения [6], логический вывод и т. д.); 3) класс решателя задачи; 4) тип целевой функции ДФ; 5) логико-семантические зависимости внутри ДФ. Такой ПК обеспечивает: 1) декларативную форму записи моделей в виде набора блоков с определением индексных переменных, и, как следствие, упрощенный синтаксис; 2) проведение многоитерационных расчетов на модели; 3) автоматические преобразования моделей; 4) построение метамodelей. Модель строится из функциональных блоков, которым сопоставляются класс задачи, идентификатор типа модели, тип решателя, целевая функция, функция сравнения, ограничения, индексные переменные, переменные и константы, соответствующие конкретным условиям моделирования. При синтезе компьютерной модели ДФ важным является возможность ее представления в декларативной форме. Это дает возможность эффективно построить модель ДФ, генерировать средства последующего исследования ДФ.

О метамоделировании логико-смысловых отношений в дискурсе. LMPL-Software обеспечивает построение метамodelей (т. е. обобщенных моделей) дискурса как взаимосвязанной системы из множества ДФ (см. напр. рис. 6).



Рис. 6. Граф логико-смысловых связей между моделями множества ДФ в метамодели дискурса

Итак, разработанный ПК LMPL-Software обеспечивает: 1) анализ ДФ, предполагающий в итоге построение графа связей составляющими ДФ; 2) анализ дискурса с построением графа связей между множеством моделей ДФ; 3) синтез моделей ДФ и дискурса; 4) сравнение моделей. Попытки моделирования дискурсивных формаций средствами разработанных нами метода и программного комплекса показали, что они эффективны и обеспечивают построение функционирующих моделей, связанных с редакционно-издательской деятельностью. Кроме того, эти средства обеспечивают возможность анализа дискурса как целостной логико-смысловой конструкции, состоящей из множества ДФ.

Практическое опробование технологии метамоделирования с использованием ПК LMPL-Software на ряде разнотипных прикладных задач (в том числе, моделирование режимов ГЭС; моделирование семантически связанных дискурсивных формаций; моделирование оптимальных отношений в дискурсе [7]) показало ее эффективность. Можно утверждать, что LMPL, реализованный в форме технологии метамоделирования, позволяет эффективно и комплексно решать ряд оптимизационных задач математического программирования, относящихся к различным прикладным областям.

Литература

1. Абасов Н.В., Осипчук Е.Н. Язык описания метамodelей задач математического программирования и его применение в гидроэнергетике // Вестник ИрГТУ. 2012. №5(64).
2. Исследование структуры дискурса средствами математического программирования и формальный анализ его логической семантики / А.В. Каверзина, М.Ю. Чернышов, Н.В. Абасов и др. // Вестник ИГЛУ. – 2013. – №2(23). – С. 180-187.
3. Компьютерная технология построения и исследования дискурса / Н.В. Абасов, А.В. Каверзина, М.Ю. Чернышов и др. // Вестник ИрГТУ. – 2013. – №4. – С. 83–90.

4. Лядова Л.Н. Метамоделирование и многоуровневые метаданные как основа технологии создания адаптируемых информационных систем // *Advanced Studies in Software and Knowledge Engineering International Book Series «Information Science & Computing»*. – Varna, Bulgaria, 2008. – № 4. – P. 125–132.
5. Рябцева Н.К. Металингвистика межкультурного делового общения: от стиля к жанру // *Жанр и культура* / отв. ред. В.В. Дементьев. – Саратов, 2007.
6. Чернышов М.Ю. К построению интеллектуальной аналитической системы. Ч. 1. Основы принципа сравнения операторных формул как функциональных моделей элементарных смыслов // *Вестник ТОГУ*. – 2013. – № 1(28). – С. 83–92.
7. Язык LMPL как средство синтеза прикладных программных моделей и метамоделей на основе принципов математического программирования / Н.В. Абасов, А.В. Каверзина, М.Ю. Чернышов и др. // *Вестник ИрГТУ*. – 2013. – №3. – С. 75–80.
8. Fourer R., Gay D.M., Kernighan B.W. *AMPL: A Modeling Language for Mathematical Programming*. – New York: Thomson, Brooks and Cole, 2002. – 517 p.
9. Levin J.A. *Dialogue-games: Meta-communication structures for natural language interaction* // *Cognitive Sci.* – 1977. – Vol. 1. – P. 395–420.
10. Moore R.C. *Reasoning about knowledge and action*. – PhD dissertation. – MIT, Cambridge Univ., Cambridge, MA, 1980. – 223 p.
11. *MPL Modeling System*, Maximal Software. – URL: <http://www.maximal-usa.com> (дата обращения: 20.01.2013).
12. Reichman R. *Plain-speaking : A theory and grammar of spontaneous discourse*. – PhD dissertation. – Dept. Comput. Sci., Harvard Univ., 1981. – 181 p.
13. Reichmann R. *Conversational coherency* // *Cognitive Sci.* – 1978. – Vol. 2. – No. 4. – P. 283–328.
14. Simmons R., Slocum J. *Generating English discourse from semantic networks* // *Commun. ACM.* – 1972. – Vol. 15, No. 10. – P. 891–905.
15. Tikhonova O., McKinney Daene C., Savitsky A. *GAMS Manual*. URL: http://www.gams.com/docs/contributed/gamsman_russian.pdf (дата обращения: 18.01.2013).

Абасов Николай Викторович, кандидат технических наук, ведущий научный сотрудник, Институт систем энергетики им. Л.А.Мелентьева СО РАН, г. Иркутск, тел.: 83952500646, e-mail: nva@isem.sei.irk.ru

Чернышов Михаил Юрьевич, кандидат филологических наук, зав. научно-методической частью Президиума ИИЦ СО РАН, тел.: 89149327490, e-mail: Michael_Yu_Chernyshov@mail.ru

Каверзина Анастасия Владимировна, аспирант, Иркутский государственный лингвистический университет, тел.: 89246067987, e-mail: anastasiakaver@mail.ru

Осипчук Евгений Николаевич, младший научный сотрудник, Институт систем энергетики им. Л.А.Мелентьева СО РАН, г. Иркутск, тел.: 89500694090, e-mail: evgeny-osipchuk@yandex.ru

Чернышова Инна Александровна, аспирант, Иркутский государственный лингвистический университет, тел.: 89179941225, e-mail: evsot93@rambler.ru

Abasov Nikolay Viktorovich, candidate of engineering sciences (computational technologies), leading researcher, Melentyev Energy Systems Institute SB RAS, Irkutsk; tel.: 83952500646, e-mail: nva@isem.sei.irk.ru

Chernyshov Mikhail Yurievich, candidate of philological sciences (mathematical linguistics, artificial intelligence), head of the research-methodological division, Presidium of Irkutsk Scientific Center SB RAS, Irkutsk; tel.: 89149327490, e-mail: Michael_Yu_Chernyshov@mail.ru

Kaverzina Anastasia Vladimirovna, postgraduate student, Irkutsk State Linguistic University; tel.: 89246067987, e-mail: anastasiakaver@mail.ru

Osipchuk Evgeny Nikolayevich, junior researcher, Melentiev Energy Systems Institute SB RAS, Irkutsk; tel.: 89500694090, e-mail: evgeny-osipchuk@yandex.ru

Chernyshova Inna Alexandrovna, postgraduate student, Irkutsk State Linguistic University; tel.: 89179941225, e-mail: evsot93@rambler.ru

УДК 004.822

© Ж.Б. Бименова, Р.А. Джумамуратов, Е.А. Сидорова

ПОДХОД К ПОСТРОЕНИЮ РУССКО–КАЗАХСКОГО ТЕЗАУРУСА ПО ИНФОРМАТИКЕ¹

В работе описывается подход к построению тезауруса на русском и казахском языках. Применяются методы корпусного исследования текстов на разных уровнях языковой действительности: морфемном, морфологическом и семантическом. Предложены методы автоматического извлечения терминов и родовидовых отношений между ними. Рассмотрена архитектура системы автоматизированного построения тезауруса, ориентированного на решение задач информационного поиска.

Ключевые слова: тезаурус, разметка, выделение терминов, лексические шаблоны.

© Zh.B. Bimenova, R.A. Dzhumamuratov, E.A. Sidorova

THE APPROACH TO DEVELOPMENT OF THE RUSSIAN-KAZAKH THESAURUS ON COMPUTER SCIENCE

In the work the approach to development of the thesaurus in Russian and Kazakh languages is described. Methods of corpus research of texts at different levels of language reality are applied: morphemic, morphological and semantic. Methods of automatic extraction of terms and generic relations between them have been proposed. An architecture of system of automated development of the thesaurus focused on solutions of problems of information search is considered.

Keywords: thesaurus, marking, term extraction, lexical patterns.

Введение

Тезаурус – многозначный термин. В современной лингвистике и информатике тезаурус представляет собой некоторое, особым образом оформленное накопление [1]. В данной работе под термином «тезаурус» будем понимать словарь, в котором слова и словосочетания с близкими по смыслу значениями сгруппированы в единицы, называемые терминами или дескрипторами, и в котором явно указываются семантические отношения между этими терминами (дескрипторами) [2].

Построения тезауруса на казахском языке – непростая задача. Несмотря на то, что существуют определенные стандарты построения тезаурусов, не всегда возможно напрямую воспроизвести существующие методики. Причинами этого являются, во-первых, специфика казахского языка (агглютинативного языка с богатой морфологией, в котором морфология языка тесно связана с его семантикой), во-вторых, отсутствие источников лексической информации (например, больших корпусов текстов), в-третьих, отсутствие доступных морфологических словарей казахского языка. Все это предполагает создание инструментов на основе подхода «чистой доски».

Высококачественные тезаурусы в большинстве своем создаются вручную. Процесс поддержания тезаурусов в актуальном состоянии довольно трудоемок, особенно в быстроразвивающихся областях. Таким образом, построение тезаурусов становится «узким местом» для практической реализации проектов, использующих их для решения своих задач (например, информационный поиск), требуются методы автоматизации их наполнения и поддержки. Одной из проблем, возникающей при автоматическом наполнении тезаурусов, является большое количество «шума», который надо эффективно отсеивать. В связи с этим наряду с автоматическими методами используют последующую ручную обработку полученного материала для получения данных большей точности (такие методы называются автоматизированными).

Методы автоматического построения тезаурусов не могут работать без разнообразных словарей. Для построения тезауруса на казахском языке, ориентированного на поддержку информационного поиска, необходимо начинать с создания морфологического словаря. Данный словарь должен поддерживать функцию морфологического анализа и извлечения из текстов на казахском языке терми-

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 12-07-31216)

нов предметной области. Для отделения общезначимых слов от терминов предметной области используется статистическая информация и создается специальный словарь предметной лексики [3].

Методы автоматического установления связей между терминами можно условно разделить на три основные группы в зависимости от области заимствования основного подхода: методы, основанные на подходах из области искусственного интеллекта, статистические методы и методы, использующие лингвистические подходы [4]. Исследования Марти Хест [5] показали, что для идентификации отношения гипонимии «достаточно адекватный» результат показывает метод лексических шаблонов.

Корпусная лингвистика – удобный инструмент для исследования языка и создания лексикографических ресурсов [6,7]. Разметка корпуса может осуществляться по разным основаниям (разным лингвистическим уровням), в частности, на морфемном, морфологическом, терминологическом, семантическом уровне и др. Использование размеченных корпусов позволяет автоматизировать наполнение необходимых словарей.

В данной работе предложен подход к автоматизированному построению тезаурусов на казахском языке, который демонстрируется в предметной области «Информатика». Реализуется полная цепочка разработки тезауруса – от этапа корпусного исследования предметной области и особенностей подязыка, создания лексикографических ресурсов до методов автоматизированного наполнения и поддержки тезауруса.

1. Тезаурус

Тезаурус описывает терминологию предметных областей как терминосистему в виде словаря терминов и словокомплексов с фиксированными семантическими связями между ними, поддерживая возможность их редактирования в процессе функционирования.

Для построения тезауруса предметной области на казахском языке необходимо решить следующие задачи:

- 1) определить структуру словарной статьи тезауруса;
- 2) сформировать параллельный корпус текстов по тематике тезауруса;
- 3) определить способ выделения терминов из корпуса текстов и сформировать список терминов;
- 4) установить отношения между терминами.

1.1. Структура словарной статьи тезауруса

В каждой словарной статье тезауруса описывается один термин. Это знак специальной семиотической системы, обладающий номинативно-дефинитивной функцией, или другое определение: термин – это устоявшееся понятие [8]. Все термины в тезаурусе являются дискрипторами.

Согласно ГОСТ 7.25-2001, в словарную статью включаются следующие типы лексических единиц: одиночные слова (существительные, прилагательные, глаголы, наречия), именные словосочетания, лексически значимые компоненты сложных слов и сокращения слов и словосочетаний [9].

Структуру словарной статьи тезауруса можно представить в виде тройки:

$T = \langle A, B, C \rangle$, где

A – символьное имя термина, соответствующее названию представляемого им понятия предметной области,

B – множество бинарных связей термина, включая иерархические, ассоциативные отношения, отношения синонимии и перевода,

C – множество идентификаторов термина (язык, морфологический класс, тип словосочетания и т.п.).

Между терминами устанавливается два вида иерархических отношений: «Класс-Подкласс» и «Часть -Целое».

Отношение «Класс-Подкласс» в зависимости от терминологических традиций в области использования ресурса может носить разное название: таксономическое отношение, родовидовое отношение, IS-а отношение, отношение гипонимии и гиперонимии. Оно устанавливается, если объем одного понятия входит в объем другого понятия.

Отношение «Часть-Целое» устанавливается в тех случаях, когда одно понятие включено в другое понятие независимо от контекста [2].

Отношение «Ассоциируется с» устанавливается между понятиями, которые находятся в связях, отличных от синонимии и иерархических отношений.

Отношение «Перевод» устанавливается между эквивалентными терминами на разных языках. Если термин не может быть выражен на другом языке одним дескриптором, тогда для него в соответствии с ГОСТом 7.24-2007 указывается в качестве эквивалента комбинация нескольких дескрипторов [10].

1.2. Архитектура системы построения тезауруса

На рис. 1 представлена общая архитектура системы разработки тезаурусов.

Система включает два основных модуля автоматизации наполнения тезауруса:

- подсистема извлечения терминов,
- подсистема извлечения отношений.

Пользователь имеет доступ к тезаурусу через редактор, который позволяет просматривать и редактировать содержание тезауруса, запускать модули автоматического пополнения тезауруса на новых корпусах текстов, а также осуществлять контроль качества работы данных методов. Обучающий корпус текстов, лексические шаблоны и словарь терминов должны подготавливаться лингвистами совместно с экспертами данной предметной области.

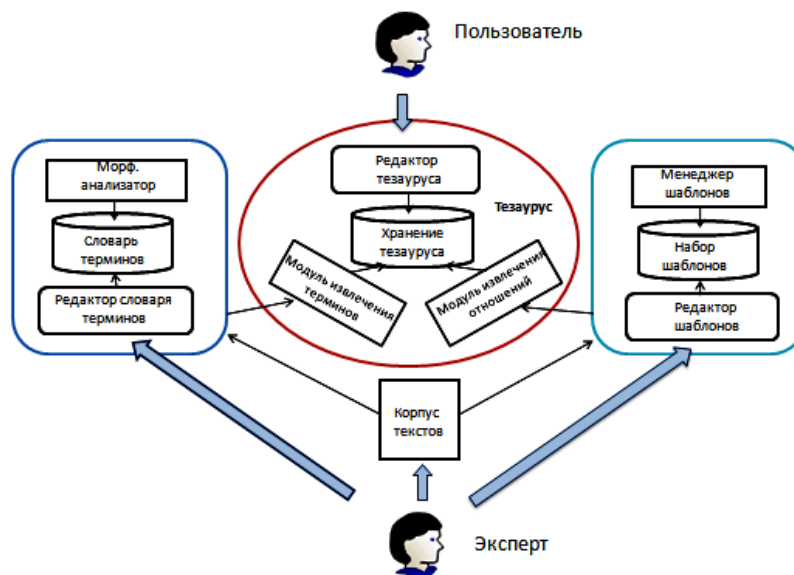


Рис. 1. Архитектура системы

2. Извлечение терминов предметной области

В процессе извлечения терминов из документа исходный текст подвергается графематическому (разбиение на слова), морфологическому (определение нормальной формы и набора параметров) и поверхностно-синтаксическому (сборка словосочетаний) анализу (рис. 2).

На этапе графематического анализа, после разбиения текста на слова, происходит поиск составных слов, которые должны рассматриваться как одно (с точки зрения морфологического анализатора). Морфологический анализ работает на уровне отдельных слов (в том числе составных) и возвращает морфологическую норму и атрибуты данного слова. При этом может оказаться, что одной словоформе может быть сопоставлено несколько возможных вариантов слов. Синтаксический анализатор может осуществлять поиск словосочетаний на основе синтаксических шаблонов сборки именных групп аналогично [11, 12]. В результате анализа приведенные к нормальному виду слова и словосочетания помещаются в предварительный словарь терминов.

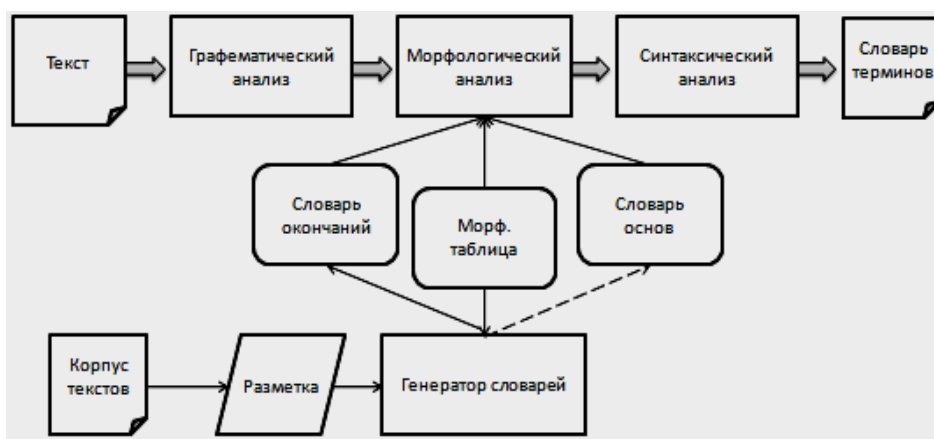


Рис. 2. Общая схема выделения терминов

На текущий момент недоступны программные инструменты, проводящие морфологический анализ текстов на казахском языке. Поэтому нами были разработаны специализированный модуль и морфологическая модель казахского языка для системы Klap [12], предназначенной для автоматизированного создания терминологических словарей. Эта же система использовалась для морфологического синтаксического анализа текстов на русском языке.

Поиск терминов-словосочетаний осуществляется на основе правил, разработанных в рамках системы Klap для русского языка и спроецированных на морфологическую таблицу казахского языка. Учет дополнительных особенностей языка в плане образования устойчивых словосочетаний требует привлечения казахских специалистов и является одной из ближайших целей проекта.

2.1. Корпусное исследование подъязыка предметной области

Казахский язык относится к группе агглютинативных языков, в которых концепция слова значительно шире, чем просто набор элементов лексики, и отличается относительной регулярностью, позиционной и грамматической стабильностью морфологической структуры различных словоформ. Слова в нем образуются присоединением к корню или основе слова грамматических частиц – аффиксов [13].

В целях построения модели морфологии казахского языка была проведена морфемно-морфологическая разметка (ММР) корпуса казахских текстов (рис. 3).

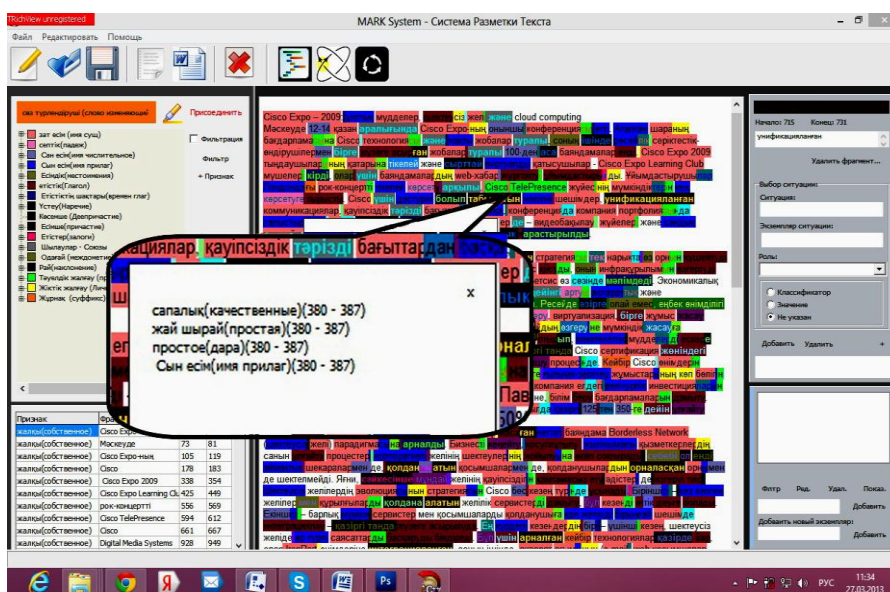


Рис. 3. Морфемно-морфологическая разметка казахского текста

Разметка была создана с помощью системы MarkSystem [14]. Помимо основного функционала, связанного с непосредственным просмотром текста и его «раскраской», в системе реализуется возможность динамического создания и пополнения системы признаков (признаками помечаются фрагменты текста) и отношений, редактирования цветовой схемы «раскраски» (посредством сопоставления признаков цветовой схемы) и управление визуализацией (фильтрация, конкорданс). Каждому фрагменту может быть сопоставлено множество признаков и связей.

Система MarkSystem ориентирована как на лингвистов, осуществляющих лингвистическое исследование корпуса текстов, так и на экспертов, отмечающих терминологию, характерную для заданной предметной области. Параллельно ММР на другом уровне создавалась семантическая разметка (СР) текстов, которая ориентирована на заданную предметную область и включает терминологическую разметку, разметку тезаурусных отношений и индикаторов.

В процессе создания ММР и СР были разработаны иерархии признаков (рис. 4).

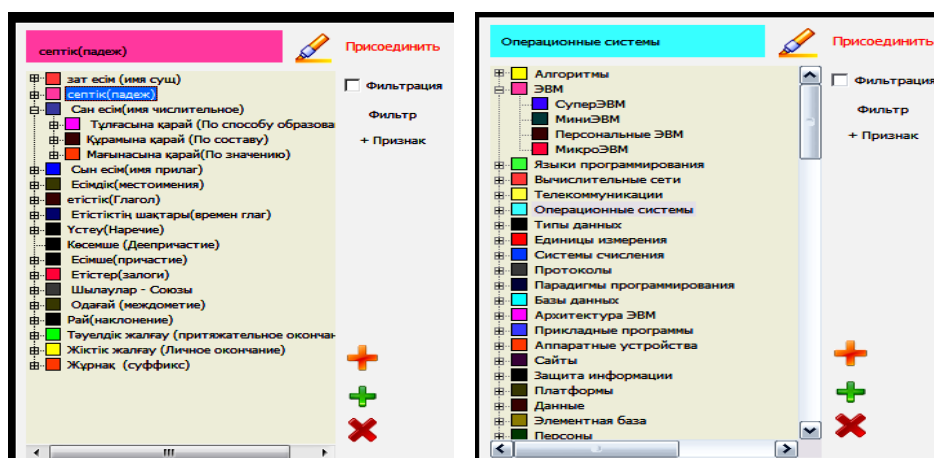


Рис. 4. Иерархия признаков для морфемно-морфологической и семантической разметки текстов

На основе ММР была создана морфологическая таблица, описывающая модель казахского языка. Иерархия признаков ММР содержит 117 вершин, на основе которых было сформировано 17 морфологических атрибутов (в том числе и часть речи) и выделено 36 морфологических классов терминов-лексем. По типу словоизменения классы сгруппированы в 16 групп, для каждой из которых формируется свой список окончаний и функция, сопоставляющая окончанию значение одного из атрибутов. Для сравнения: русская таблица содержит 17 атрибутов, 42 класса и 15 групп. Данная таблица легла в основу создаваемого морфологического словаря казахского языка.

Иерархия признаков СР соответствует верхнему уровню иерархии терминов тезауруса по информатике и на текущий момент включает 29 вершин. В процессе СР любой термин, выделенный экспертом, либо соотносится с признаком (что означает, что эксперт считает данный термин обобщающим классом предметной области), либо связывается с одним из ранее введенных признаков. Выделенные фрагменты также могут связываться специальным отношением Род-вид (подробнее о СР будет сказано ниже). Таким образом, назначение СР – выделить основные классы терминов предметной области, которые впоследствии будут служить основой автоматизированных методов поиска новых терминов и отношений между терминами.

2.2. Морфологический анализ текста на казахском языке

На вход морфологического анализатора подается упорядоченный список словоформ (с учетом знаков), полученный в результате графематического анализа. Для каждой словоформы на первом этапе осуществляется нормализация, т.е. поиск основы – начальной формы слова. Затем, в зависимости от части речи и найденных аффиксов, вычисляются морфологические характеристики слова.

Модуль нормализации в процессе своей работы осуществляет следующую последовательность шагов:

1 шаг: Выполняется поиск слова в словаре начальных форм. Если слово в словаре найдено, то шаг 5.

2 шаг: Слово считывается посимвольно в обратном порядке (начиная с конца слова). Если слово закончилось, то работа алгоритма завершается. На основе текущего списка аффиксов формируется список гипотетических аффиксов.

3 шаг: Выполняется поиск всех гипотетических аффиксов в словаре аффиксов. Все найденные аффиксы добавляются в список аффиксов. Если ни один новый аффикс не найден, то переходим к шагу 2.

4 шаг: Выполняется поиск начальной части слова в словаре начальных форм. Если слово не найдено, то переходим к шагу 2.

5 шаг: В результат добавляется найденная основа и сопутствующий набор аффиксов. Переход к шагу 2.

После нормализации для каждого найденного слова осуществляется вычисление его морфологических характеристик на основе его аффиксов и морфологического класса основы.

Продемонстрируем результат морфологического анализа на следующем примере (рис. 5).

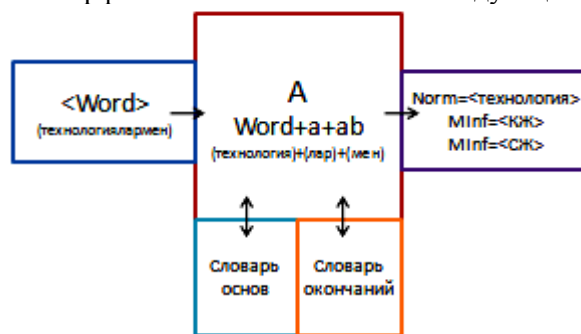


Рис. 5. Пример морфологического анализа слова

На вход анализатору подается словоформа *технологиялармен*, происходит поиск в словарях аффиксов *мен*, *лар* и основы *технология*. На основе морфологического класса основы (зат есім) и аффиксов вычисляем морфологическую информацию: *лар* <мн.число>, *мен* <род.падеж>.

Казахский язык характеризуется строгой последовательностью присоединения аффиксов к корню: вначале присоединяется словообразовательный аффикс, потом словоизменяющие аффиксы: принадлежности, падежей, лица и числа. Для имен существительных к основе слова вначале добавляется окончание множественного числа, затем притяжательное окончание, далее следует падежное окончание и последним – окончание формы спряжения [13]. Окончания прибавляются по правилу, которое можно представить следующим образом:

$$C = OC + KЖ + TЖ + CЖ + ЖЖ, \tag{1}$$

где C – словоформа; OC – основа слова; KЖ – окончание множественного числа; TЖ – притяжательное окончание; CЖ – падежное окончание; ЖЖ – окончание формы спряжения [15].

К особенностям казахского языка относится необходимость определения некоторых морфологических характеристик слова по контексту, т.е. словам, находящимся до или после данного слова. Например, время глагола или деепричастия может быть определено по слову, стоящему соответственно до или после, а превосходную степень прилагательного определяют по специальным словам, таким как *ен*, *ете*, *аса*, *тым* и т.д.

Нормальная форма слова – это одна из форм слова (строка), принятая для обозначения понятия, связанного с данным словом. Словоформа – это форма слова, связанная с нормальной формой слова и указывающая на особенности употребления данного слова.

Формально словоформа характеризуется пятеркой вида:

$W_{form} = \langle form, PS, norm, PSN, Minf \rangle$, где

form – написание словоформы (строка),

PS – часть речи,

norm – нормальная форма, от которой была образована данная словоформа,

PSN – часть речи нормальной формы,

Minf – набор морфологических параметров, приписываемых к данной словоформе.

Результат работы морфоанализатора казахского текста представлен множеством таких пятерок с дополнительным параметром позиции в тексте.

3. Установление семантических отношений между терминами

В работах, направленных на автоматический поиск отношений между терминами, можно выделить два основных подхода. Первый подход основан на распределительной гипотезе [16], где утверждается, что семантически близкие термины встречаются в схожих контекстах. Например, можно предполагать, что контексты слов *жолдас* (друг) и *дос* (друг) содержат много общих слов.

При втором подходе используются лексические шаблоны [5, 17], которые сигнализируют о возможных семантических отношениях между терминами. Лексические шаблоны представляют собой характерные выражения (словосочетания и обороты), конструкции из определенных элементов языка. Такие шаблоны позволяют построить семантическую модель, соответствующую фрагменту текста, к которому они применяются [18].

Рассмотрим отношение гипонимии. Традиционно установление этого отношения базируется на известном критерии Лайонза [17]: для пары терминов X и Y (в предположении, что Y является гиперонимом для X) строятся предложения вида «Это есть X» и «Это есть Y», в которых данные термины являются логическими предикатами и проверяется, что из истинности первого следует истинность второго. Анализируя корпус текстов, можно выявить в нем пары терминов-кандидатов, подозреваемых на гипонимическую связанность. Для этого необходимо «отловить» в текстах некоторые гипонимические конструкции, например, такие как:

- 1) "X и другие/прочие/остальные Y-и"
- 2) "Все Y-и, кроме/помимо/за исключением X-а"
- 3) "Все Y-и, включая/в том числе X"
- 4) "Некоторые Y-и, например/в частности, X"
- 5) "Одни Y-и, например/в частности, X" и т.п.

Но набор понятий и терминов и соответственно их иерархия зависят от языка и культуры. Например, в русском языке отсутствует однозначный аналог «индикатора» гиперонима «is a».

Для разработки лексических шаблонов было просмотрено и размечено несколько десятков научно-технических текстов, в которых выделялись фразы, используемые для определения или пояснения термина. После предварительного анализа полученные фразы были формализованы в виде лексических шаблонов. Для разработки шаблонов использовался язык Diglex DSL [11], который позволяет определять произвольные символьные выражения, указывать альтернативные выражения, образуемые при использовании сокращений, аббревиатур, синонимов, пропусков и перестановок в лексическом составе конструкции, задавать условия, опциональные подвыражения, дистантный контекст и т.д.

Разработка шаблонов состоит из следующих этапов.

Этап 1. В тексте ищутся предложения, содержащие термины, связанные семантическими отношениями. Найденные термины размечаются (рис. 6).

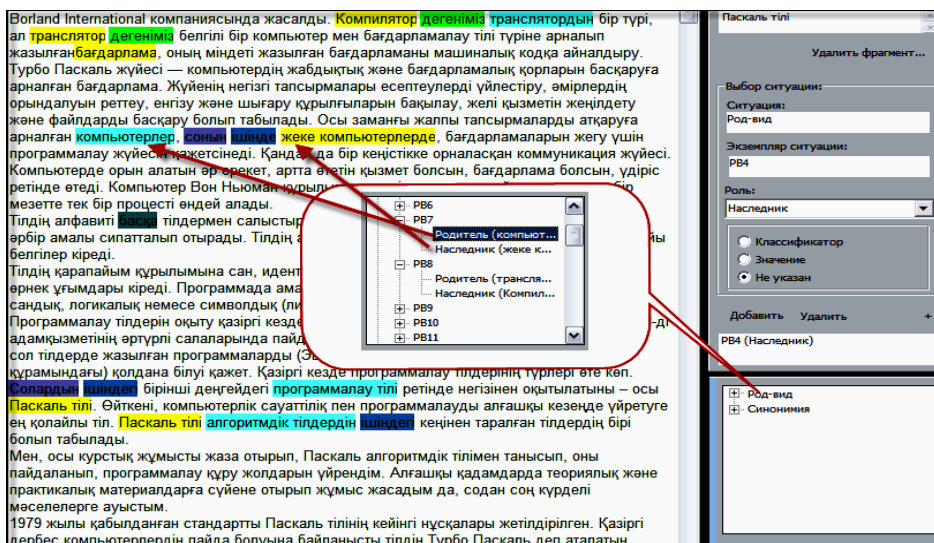


Рис. 6. Разметка пар иерархически-связанных терминов

Этап 2: Фиксируется окружение пар связанных терминов. Найденные контексты анализируются и наиболее часто встречающиеся фразы из контекстов помечаются как «индикаторы» связи (рис. 7).

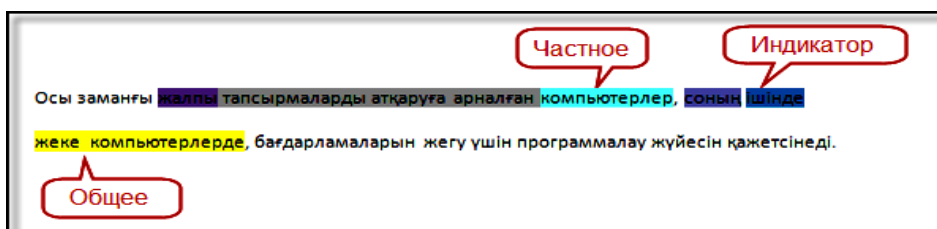


Рис. 7. Разметка пар терминов, индикаторов связи и окружения терминов

Были обнаружены такие «индикаторы», как басқа (другие), басқадай (прочие), бөлек (кроме), бөтен (помимо), қоспағанда (за исключением), қоса есептегенде (включая), соның ішінде (в том числе), мысалы (например), атап айтқанда (в частности).

Этап 3: На основе найденных индикаторов формируются лексические шаблоны специального вида (рис. 8), предназначенные для извлечения родовидовых связей между терминами. Сформированные шаблонные конструкции записываются на языке Diglex DSL.

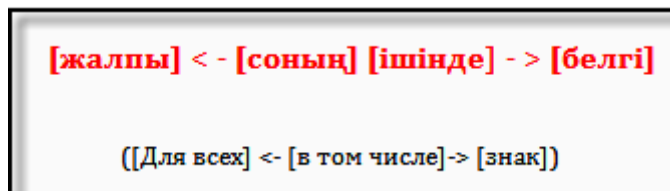


Рис. 8. Пример шаблона на языке DSL Diglex

Таким образом, проблема разработки лексических шаблонов связана с поиском индикаторов и формализацией шаблонных конструкций, а их применение – с проблемой шума на выходе.

На рис. 9 представлена общая схема извлечения отношений между терминами.

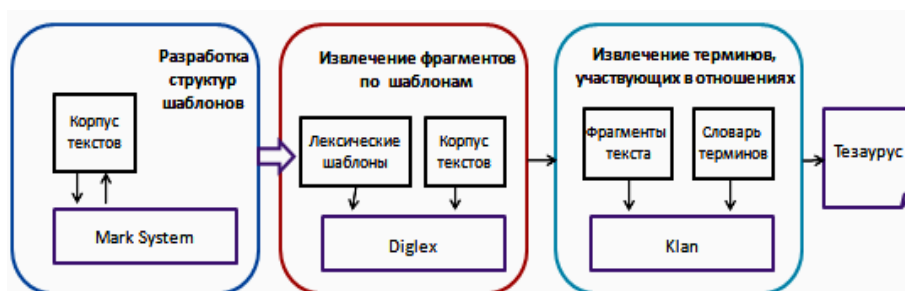


Рис. 9. Схема извлечения отношений между терминами

Ядерным компонентом в схеме является система Diglex. Лексические шаблоны, разработанные на основе размеченного корпуса текстов, записываются на языке Diglex DSL и образуют словарь гипонимических конструкций. Процессор Diglex принимает на вход словарь и на выходе формирует множество вхождений шаблонов, которые сигнализируют о возможной связи между терминами, покрытыми данным вхождением. Далее полученные на выходе системы Diglex фрагменты текстов подаются на вход системы Klan для разбиения текста на лексические единицы и поиска в терминологическом словаре. На основе найденной информации делаются выводы об установлении отношений между терминами.

Заключение

В работе предложен подход, позволяющий автоматизировать процесс построения двуязычного тезауруса предметной области. Для апробации подхода был создан параллельный корпус текстов из предметной области «Информатика». Интересным направлением дальнейшего исследования является автоматизация поиска слов-индикаторов для разработки лексических шаблонов, а также сравнение результативности применения лексических шаблонов для русского и казахского языков.

Литература

1. Луков Вал.А., Луков Вл.А. Концепция курса «Мировая культура»: тезаурологический подход // Педагогическое образование. – 1992, № 5. – С. 8–14.
2. Лукашевич Н.В. Тезаурусы в задачах информационного поиска. – М.: Изд-во МГУ, 2011. – 495 с.
3. Сидорова Е.А. Многоцелевая словарная подсистема извлечения предметной лексики // Компьютерная лингвистика и интеллектуальные технологии: материалы ежегодной междунар. конф. «Диалог – 2008». – 2008. – Вып. 7 (14). – С. 475-481.
4. Мозжерина Е. С. Автоматическое построение онтологии по коллекции текстовых документов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: тр. XIII Всерос. науч. конф. – Воронеж, 2011.
5. Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In In Proceedings of the 14th International Conference on Computational Linguistics. – 1992. – P. 539–545.
6. Захаров В.П. Корпусная лингвистика: учеб.-метод. пособие. – СПб., 2005. – 48 с.
7. Крижановский А.А. Автоматизированное построение списков семантически близких слов на основе рейтинга текстов в корпусе с гиперссылками и категориями // Компьютерная лингвистика и интеллектуальные технологии: тр. междунар. конф. «Компьютерная лингвистика и интеллектуальные технологии». – М.: Изд-во РГГУ, 2006. – С. 297-302.
8. Никитина С.Е. Семантический анализ языка науки. – М.: Наука, 1987. – 141 с.
9. ГОСТ 7.24-2007. Система стандартов по информации, библиотечному и издательскому делу. Тезаурус информационно-поисковый многоязычный. Состав, структура и основные требования к построению (введен в действие с 1 июля 2008 г.)
10. Загорюлько Ю.А., Боровикова О.И., Кононенко И.С., Соколова Е.Г. Подход к разработке русско-английского тезауруса по компьютерной лингвистике // Электронные библиотеки: перспектив-

ные методы и технологии, электронные коллекции: тр. XIII Всерос. науч. конф. – Воронеж: Изд-во Воронеж. гос. ун-та, 2011. – С.27-34.

11. Большаков И.А. Какие словосочетания следует хранить в словарях? // Труды международного семинара «Диалог – 2002» по компьютерной лингвистике и ее приложениям. – Протвино, 2002. – Т.2. – С.61–69.

12. Загорулько Ю.А., Сидорова Е.А. Система извлечения предметной терминологии из текста на основе лексико-синтаксических шаблонов // Проблемы управления и моделирования в сложных системах: тр. XIII Междунар. конф. / под ред. Е.А. Федосова, Н.А. Кузнецова, В.А. Виттиха. – Самара: Изд-во Самар. науч. центра РАН, 2011. – С.506-511.

13. Сопоставительная грамматика русского и казахского языков. Морфология. – Алма-Ата: Наука, 1966. – 459 с.

14. Сидорова Е.А., Загорулько М.Ю. Программный инструментарий разработки лингвистических ресурсов // Открытые семантические технологии проектирования интеллектуальных систем: тр. III Междунар. науч.-техн. конф. – Минск: Изд-во БГУИР, 2013. – С.159-164.

15. Шарипбаев А.А., Беманова Г.Т. Построение логической семантики слов казахского языка // Знания-Онтологии-Теории: материалы всерос. конф. с междунар. участием, 3-5 октября 2011 г. – Новосибирск, 2011.

16. Harris Z.S. Mathematical Structures of Language. – Wiley, New York, 1968.

17. Лайонз Дж. Введение в теоретическую лингвистику. – М.: Прогресс, 1978.

18. Рабчевский Е.А. Автоматическое построение онтологий на основе лексико-синтаксических шаблонов для информационного поиска // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: тр. XI Всерос. науч. конф. – Петрозаводск, 2009. – С. 69-77.

Бименова Жанат Батырбековна, магистрант Новосибирского государственного университета, тел. 8(965) 8297533. E-mail: zhanka777@gmail.com

Джумамуратов Руслан Азатович, магистрант Новосибирского государственного университета, тел. 8(903) 0764191. E-mail: Dzhumamuratov.rus@gmail.com

Сидорова Елена Анатольевна, кандидат физико-математических наук, научный сотрудник Института систем информатики им. А.П. Ершова СО РАН, Новосибирск, тел. 8-(383)- 3307991. E-mail: lena@iis.nsk.su

Bimenova Zhanat Batyrbekovna, graduate student, Novosibirsk State University. E-mail: zhanka777@gmail.com

Dzhumamuratov Ruslan Azatovich, graduate student, Novosibirsk State University. E-mail: Dzhumamuratov.rus@gmail.com

Sidorova Elena Anatolievna, candidate of physical and mathematical sciences, research associate, A.P.Ershov Institute of Informatics Systems SB RAS, Novosibirsk. E-mail: lena@iis.nsk.su

МОДЕЛЬ ПРЕДМЕТНОЙ ОБЛАСТИ НА ЯЗЫКЕ ОПИСАНИЯ ОНТОЛОГИЙ

В статье рассматриваются два средства моделирования предметной области – ER-модель и онтология. Также приведен пример записи данных на языке описания онтологий RDF.

Ключевые слова: ER-модель, концептуальная схема, база знаний, онтология, языки описания онтологий, RDF, OWL.

© Yu.S. Verkhoturova

MODEL OF THE OBJECT DOMAIN IN THE LANGUAGE OF ONTOLOGY DESCRIPTION

The article considers two tools of the object domain modeling - ER-model and ontology. There is also an example of data records in the language of ontology description RDF.

Keywords: ER-model, conceptual scheme, knowledge base, ontology, RDF, OWL.

1. Концептуальная схема и ER- модель как истоки онтологии

Технологии БД сформировались как значимая ветвь информационных технологий в конце 60-х – начале 70-х гг. XX века. Начали производиться системы управления базами данных (СУБД) общего назначения, стали востребованными методологии проектирования БД и реализующий их инструментарий.

В соответствии со сложившимися методологиями проектирования БД начальным этапом этого процесса является формирование спецификации абстрактного представления предметной области (ПО) с помощью подходящих выразительных средств. Этот этап называется концептуальным моделированием ПО, а его результат – концептуальной схемой предметной области (КС). *Концептуальная модель* – абстрактное описание предметной области, независимое от аспектов реализации систем, в рамках которых оно используется, определяющее концептуальную структуру и поведение сущностей в предметной области [4].

В процессе эволюции технологий баз данных был разработан целый ряд концептуальных моделей данных, самой распространенной из которых стала модель «сущностей-связей», или ER-модель данных.

В рамках ER-модели окружающий мир представляется как набор сущностей(*n*-арных отношений). Полученная семантическая модель вводит термины для описания предметной области и определяет их значения путем задания взаимосвязей и ограничений. Из семантической модели создается логическая схема, описывающая структуры данных в БД, типы данных, взаимосвязи и ограничения. Несмотря на то, что ER-модель обеспечивает независимость модели данных от средств ее реализации, она позволяет лишь однозначное интерпретирование данных. А создаваемые на ее основе модели ПО являются моделями со «слабой» семантикой.

2. Онтологический инжиниринг

В середине 90-х гг. в крупных корпорациях, для которых проблемы обработки информации стали критическими, появилось понятие «управление знаниями».

Оно подразумевает создание базы знаний, которая содержит объединенные знания всех сотрудников, необходимые, например, для поддержания процесса принятия решений. При этом стало очевидным, что основным узким местом является работа со знаниями (сохранение, поиск, интеграция и пр.), накопленными специалистами компании, так как именно знания обеспечивают преимущество перед конкурентами [5].

Появление знаний как информационных объектов для обработки на компьютерах определило переход от БД к БЗ.

Раздел искусственного интеллекта, изучающий базы знаний и методы работы со знаниями, называется *инженерией знаний*.

Онтология – это спецификация на концептуальном уровне [2]. Данный термин произошел от названия раздела философии, где изучаются проблемы бытия. В контексте информационных технологий этот термин приобрел другой смысл, здесь онтология – это попытка всеобъемлющей и детальной формализации некоторой области знаний с помощью концептуальной схемы. Онтология включает в себя словарь указателей на термины ПО и логические выражения, описывающие соотношение между ними, т.е. онтология – это сеть, содержащая термины и понятия и показывающая взаимосвязь между ними.

Все современные онтологии, независимо от спецификации языка и его программной реализации, строятся примерно одинаково. Основными компонентами онтологий являются концепты (классы, понятия, сущности, категории), свойства концептов (слоты, атрибуты роли), отношения между концептами и некоторые ограничения. Экземпляр концепта служит для представления элемента предметной области, группу элементов которого описывает концепт. Например, студент – концепт, а Петров Петр – это уже экземпляр концепта. Сама онтология и множество экземпляров составляют базу знаний.

Концепт – это шаблон, содержащий множество правил, определяющих форму экземпляра, т.е. то, каким образом может быть построен экземпляр. Концепты могут иметь *атрибуты* – имена или структуры полей записи. Атрибуты характеризуют размер или тип информации, содержащейся в поле. Отношениями между концептами называются зависимости между экземплярами онтологии [3]. Обычно отношением является атрибут, ссылающийся на другой экземпляр.

3. Языки описания онтологий

Одним из важных моментов в создании онтологической модели ПО является выбор конкретного языка описания. Цель таких языков – сделать машинное представление данных более похожим на положение вещей в реальном мире, существенно повысить выразительные возможности концептуального моделирования слабо структурированных данных, дать возможность указывать дополнительную машинно-интерпретируемую семантику ресурсов.

Существуют традиционные языки спецификации онтологий: Ontolingua, CycL, языки, основанные на дескриптивных логиках (такие как LOOM), языки, основанные на фреймах (OKBC, OCML, F-Logic). Специально для обмена онтологиями через Web были созданы языки RDF, RDFS, DAML+OIL, OWL.

Язык RDF

Спецификация RDF (Resource Description Framework) состоит из двух частей: модели представления информации в WWW и синтаксиса для представления этой информации в конкретном цифровом виде для дальнейшего хранения, передачи и обработки. Базовыми понятиями RDF являются ресурсы, отношения, также называемые предикатами и свойствами, и утверждения, представляющие собой триплет «субъект-предикат-объект».

Субъект – это описываемый ресурс; предикат – это тот аспект ресурса, который описывается (другими словами, свойство ресурса); объект – это уже конкретное значение отношения (значение свойства ресурса).

Все ресурсы и отношения (свойства) в RDF идентифицируются с помощью URI, что позволяет описывать с помощью RDF как интернет-ресурсы, так и ресурсы, которые не могут быть получены через Интернет (человек, автомобиль, город и т.д.). Например, если информация о том, что И.Иванов является автором ресурса <http://www.example.org/>, расположена по адресу <http://www.example.org/creator#>, то в терминах RDF в нотации N-Triple это описание быть выражено следующим образом:

<<http://www.example.org/creator#person>>

<<http://www.example.org/creator#/name>>

<<http://www.example.org/creator#И.Иванов>>.

Использованная нотация N-Triple не является стандартной формой записи RDF-описаний – рекомендованный синтаксис для записи RDF является RDF/XML, который более удобен для автоматической обработки, хотя и более сложен для восприятия человеком.

Запись троек в XML-документах производится иначе, чем в нотации N-Triple. Приведем пример записи XML-документа, который используется для передачи данных из таблицы «Сведения о количестве земельных участков за 2011 г.» (Табл.1).

Таблица 1

Сведения о количестве земельных участков за 2011 г.

№ п/п	Наименование кадастрового района	Количество земельных участков в ГКН	Количество земельных участков поставлено на учет	Количество земельных участков снято с учета	Количество земельных участков, по которым проведен учет изменений
1.	Баргузинский	11486	845	35	7747
2.	Баунтовский	6289	137	987	4194
3.	Бичурский	18917	1455	140	18634
4.	Джидинский	13208	1506	317	23426
5.	Еравнинский	7304	512	108	12629

XML-документ для передачи данных таблицы будет выглядеть следующим образом:

```
<?xml version="1.0" encoding="windows-1251" standalone="no" ?>
<!-- File Name: regions.xml -->
<?xml-stylesheet type="text/css" href="ST01.css" ?>
<!DOCTYPE Region-list
[
<!ELEMENT Region-list (REGION*)>
<!ELEMENT REGION (ID,NAME_CR,GKN,REGISTERED,REMOVAL,CHANGES) >
<!ELEMENT ID ANY>
<!ELEMENT NAME_CR ANY>
<!ELEMENT GKN ANY>
<!ELEMENT REGISTERED ANY>
<!ELEMENT REMOVAL ANY>
<!ELEMENT CHANGES ANY>
]>
<Region-list>
<REGION >
<ID>1</ID>
<NAME_CR>Баргузинский</NAME_CR>
<GKN>11486</GKN>
<REGISTERED>845</REGISTERED>
<REMOVAL>35</REMOVAL>
<CHANGES>7747</CHANGES>
</REGION>
<REGION >
<ID>2</ID>
<NAME_CR>Баунтовский</NAME_CR>
<GKN>6289</GKN>
<REGISTERED>137</REGISTERED>
<REMOVAL>987</REMOVAL>
<CHANGES>4194</CHANGES>
</REGION>
</REGION >
```

```

<ID>3</ID>
  <NAME_CR>Бичурский</NAME_CR>
  <GKN>18917</GKN>
  <REGISTERED>1455</REGISTERED>
  <REMOVAL>140</REMOVAL>
  <CHANGES>18634</CHANGES>
</REGION>

```

```

.....
</Region-list>

```

Теперь рассмотрим, как будет записан RDF-граф, приведенный в примере таблицы.

Субъектом в данном случае будет «кадастровый район» (тэг <REGION> в XML-документе). Предикатами являются уникальный номер региона в таблице (тэг <ID>), наименование (тэг <NAME_CR>), количество земельных участков в ГКН (тэг <GKN>), количество земельных участков, поставленных на учет (тэг <REGISTERED>), количество земельных участков, снятых с учета (тэг <REMOVAL>) и количество участков, по которым произведен учет изменений (<CHANGES>). Соответствующие этим предикатам объекты будут являться конкретными значениями, указанными в таблице (или значениями перечисленных тэгов XML-документа).

Так как субъектов несколько, то в RDF-графе каждый из них будет записан в специальном тэге rdf:about. Тройки с каждым из этих субъектов записываются в виде дочерних элементов этого тэга. Каждый тэг дочернего элемента – это имя предиката, а его содержимое представляет собой объект. Все определения должны быть заданы в корневом тэге rdf:RDF.

Задача RDF – это описание ресурсов в Сети, поэтому предположим, что наша таблица находится по адресу <http://www.example.org/cadastral-regions#>.

```

<rdf:RDF>
  xmlns:regions="http://www.example.org/cadastral-regions#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <regions:region rdf:about="http://www.example.org/cadastral-regions#region1">
    <regions:id>1</regions:id>
    <regions:name_cr>Баргузинский</regions:name_cr>
    <regions:gkn>11486</regions:gkn>
    <regions:registered>845</regions:registered>
    <regions:removal>35</regions:removal>
    <regions:changes>7747</regions:changes>
  </regions:region>
  <regions:region rdf:about="http://www.example.org/cadastral-regions#region2">
    <regions:id>2</regions:id>
    <regions:name_cr>Баунтовский</regions:name_cr>
    <regions:gkn>6289</regions:gkn>
    <regions:registered>137</regions:registered>
    <regions:removal>987</regions:removal>
    <regions:changes>4194</regions:changes>
  </regions:region>
  <regions:region rdf:about="http://www.example.org/cadastral-regions#region3">
    <regions:id>3</regions:id>
    <regions:name_cr>Бичурский</regions:name_cr>
    <regions:gkn>18917</regions:gkn>
    <regions:registered>1455</regions:registered>
    <regions:removal>140</regions:removal>
    <regions:changes>18634</regions:changes>
  </regions:region>
  .....
</rdf:RDF>

```

Язык OWL

Другая спецификация – OWL (Web Ontology Language – язык онтологии Web), это язык, созданный для описания онтологии Web. Он совместим с языком RDF, т.е. каждый OWL-документ является также и RDF-документом.

Базовыми понятиями OWL являются класс, свойство, экземпляр класса, ограничение на свойство, мощность свойства, инверсное свойство и многие другие, с помощью которых можно явно описывать различные предметные области. OWL является фактически мета-онтологией, т.к. на основе OWL-терминов создаются другие онтологии.

Базовым элементом языка OWL является класс всех классов, определяемый как `owl: Class`, следовательно, любой OWL класс должен быть задан как экземпляр класса OWL класс. Например, если мы хотим определить класс Student (Студент), то должны определить тройку

`Student rdf: type owl: Class,`

которая в XML синтаксисе будет выглядеть следующим образом: `<owl:Class rdf:ID= "Student"/>`.

В языке OWL также существует два predefined класса:

- класс `owl:Thing` (сущность), который обозначает множество всех индивидов;
- класс `owl:Nothing` (ничто), обозначающий пустое множество.

Наследование классов в языке OWL задается с помощью конструкции `rdfs:subClassOf`, а тот факт, что один класс является дочерним классом другого, означает, что все экземпляры дочернего класса являются экземплярами родительского класса.

Свойства в OWL делятся на два класса:

- объектные свойства, используемые для связывания индивидов друг с другом. Объектные свойства – это экземпляры класса `owl:ObjectProperty`.
- свойства типов данных связывают индивидов с так называемыми значениями типов данных. Свойства типов данных – это экземпляры класса `owl:DatatypeProperty`.

Язык OWL позволяет описывать различные характеристики классов и свойств, которые обычно задаются как разного рода ограничения на структуру связей между своими экземплярами. Эти ограничения выражаются в виде predefined соотношений, называемых в OWL аксиомами [3].

Заключение

Концептуальную схему БД часто называют онтологией ПО, однако это неверно. Хотя онтология и КС являются результатами концептуального моделирования предметной области, они представляют собой различные по характеру представления структуры предметной области.

Сфера применения онтологий шире, чем КС баз данных. Онтологии используются не только в области систем БД, но и инженерии знаний, искусственном интеллекте, в проектах Семантического Веба, таких как семантический поиск и интеграция баз знаний.

Основное преимущество моделирования ПО с помощью онтологии – это более адекватное и декларативное описание предметной области, чем при использовании реляционных баз данных.

Языки описания онтологий RDF и OWL позволяют описывать семантику различных предметных областей в таком виде, который может использоваться неоднократно различными приложениями для дальнейшей обработки и, в частности, для реализации логического вывода и проверки (доказательства) утверждений.

Литература

1. Добров Б.В., Иванов В.В., Лукашевич Н.В., Соловьев В.Д. Онтологии и тезаурусы: модели, инструменты, приложения. – М.: БИНОМ, 2009. – 173 с.
2. Gruber Thomas.R. A Translation Approach to Portable Ontology Specifications //Knowledge Acquisition. 1993. № 5/2. – P. 199-220.
3. Лапшин В.А. Онтологии в компьютерных системах. – М.: Научный мир, 2010. – 224 с.
4. URL: http://ontology.ipi.ac.ru/index.php/Концептуальная_модель_предметной_области
5. Гаврилова Т.А. Онтологический подход к управлению знаниями при разработке корпоративных систем автоматизации // Новости искусственного интеллекта. 2003. №2. – С. 24-30.

Верхотурова Юлия Сергеевна, аспирант кафедры геометрии Института математики и информатики Бурятского государственного университета. E-mail: bgsha_julia@mail.ru

Verkhoturova Yulia Sergeevna, postgraduate student, Buryat State University, Institute of Mathematics and Computer Science, department of geometry. E-mail: bgsha_julia@mail.ru

УДК 681.3.06+37.01:014.544+629.7.066

© Г.С. Курганская

ОБЛАЧНЫЕ ТЕХНОЛОГИИ ИНТЕРНЕТ-ОБРАЗОВАНИЯ НА ОСНОВЕ KFS МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Статья посвящена организации интернет-обучения в облачных технологиях на основе авторской модели представления знаний. Приводится описание KFS модели представления знаний, которая обеспечивает дифференцированное и адаптированное обучение в условиях постоянной динамики объема, структуры и качества знаний. Обоснована возможность эффективной организации учебного процесса на основе облачных технологий.

Ключевые слова: представление знаний, информационные технологии, интернет-обучение, облачные технологии.

© G.S. Kurganskaya

CLOUD TECHNOLOGIES OF THE INTERNET EDUCATION BASED ON KFS KNOWLEDGE REPRESENTATION MODEL

The article is devoted to organization of the Internet education with cloud technologies on the basis of the author's model of knowledge representation. There is a description of KFS knowledge representation model, that provides differentiated and adapted education under constant dynamics of growing volume, structure and quality of knowledge. The possibility to organize on its basis the effective learning process with cloud-based technologies is substantiated.

Keywords: knowledge representation, information technologies, the Internet education, cloud-based technologies.

Сегодня можно утверждать, что современные инфокоммуникационные технологии, особенно Интернет, стали основой радикальных перемен в нашем обществе:

- экономика – переход к экономике знаний
- социум – виртуальные сообщества, социальные сети,...
- образование – переход к непрерывному образованию («long life learning»)

Поскольку, по существу, образование есть передача знаний, а новые технологии позволяют осуществлять эту передачу независимо от природы, местоположения и времени субъектов информационного взаимодействия, то очевидно, что традиционная организация и структура образования уже не соответствуют новым возможностям, следовательно, требованиям современного общества. Система образования должна обеспечить пространство, в котором свободно и естественно происходит формирование и обмен знаниями. Облачные технологии в Интернете могут обеспечить оптимальную на сегодня инфраструктуру нового образовательного пространства.

На сегодняшний день существует множество определений облачных технологий («облачных» вычислений), здесь мы будем придерживаться следующего: "Облако является новой моделью для предоставления и получения информационных услуг. Эта модель обещает снизить оперативные и капитальные затраты. Она позволяет организациям сосредоточиться на стратегических проектах, а не на рутинных задачах управления собственным центром обработки данных» [1]. Рассмотрим, как можно использовать облачные технологии в образовании. Мы рассматриваем образование как систематизированный процесс передачи структурированных знаний. Формализация процесса обучения базируется на авторской KFS модели представления знаний [2]. Следует отметить, что при разработке этой модели автор не предполагал построения на его основе «облака» обучения.

1. KFS МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

В этой модели базовым элементом является учебный блок e , при изучении которого у студента должно сформироваться некое целевое знание (умение, навык) B , для изучения блока могут потребоваться входные знания A_i

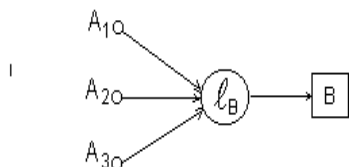


Рис.1. Учебный блок

Поскольку входные знания A_i обычно являются выходами некоторых других учебных блоков, а целевое знание блока может, в свою очередь, использоваться в других блоках, то естественным образом учебные l -блоки связываются в граф KN (Knowledge Network)

Мы можем детализировать любой учебный блок, разбив его на более мелкие блоки и установив между ними связи по знаниям, и наоборот, мы можем в графе KN выделить некий кластер с одним выходом, который дальше рассматривается как обобщенный учебный блок.

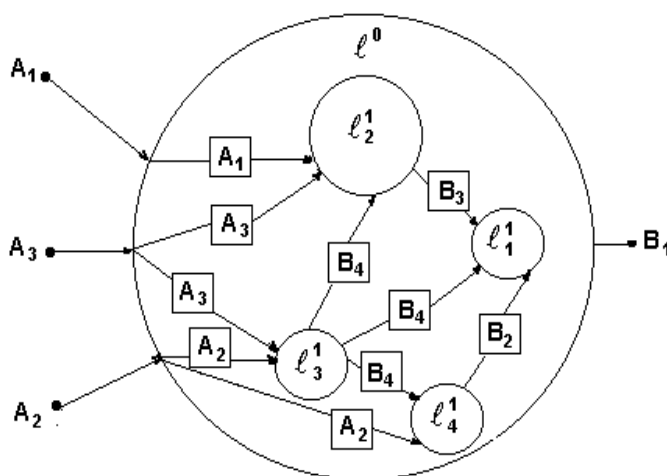


Рис. 2. Кластер

Таким образом, учебный блок на разных уровнях может обозначать отдельную тему или курс, или специализацию, программу, и т.д. Мы можем применять единый механизм организации учебного процесса независимо от его уровня.

Для того чтобы реализовать этот механизм в «облаках» Интернета, нужно по построенному KN графу автоматически формировать возможные алгоритмы обучения как для отдельного студента, так и для разного рода сообществ: групп, потоков. Мы будем использовать классический подход построения ярусно-параллельных форм, предложенный Д.А. Поспеловым [3].

2. Ярусно-параллельная форма представления графа KN

Наряду со связностью по знаниям, явно представленной в графе KN, при планировании учебного процесса важным является необходимость учитывать и логическую независимость (несвязность)

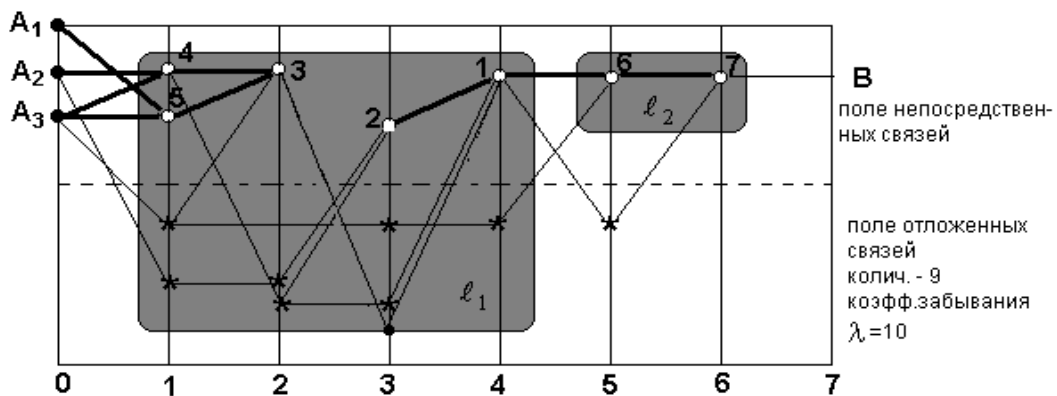
знаний. Независимость ℓ -блоков в графе KN позволяет строить различные варианты последовательностей изучения учебного материала и выбирать из них оптимальные в некотором смысле.

Определение. Ярусно-параллельной формой (ЯПФ) графа KN называется частичное упорядочение вершин по уровням, на которых расположены независимые по логическим связям ℓ -блоки так, что на 0-м уровне расположены входные знания, а на последнем – целевое знание.

На рис. 3 показаны две различные ЯПФ для графа KN, который состоит из двух ℓ -блоков ℓ_1 и ℓ_2 , где ℓ_1 соответствует приведенному на рис.2, а структура ℓ_2 непосредственно задана на 3. ℓ_1 и ℓ_2 являются кластерами, соответственно их объединение тоже является кластером. Как видно из рисунков, ЯПФ состоит из 8 уровней (ярусов), на 0-м уровне – входные знания (A_1, A_2, A_3), на последнем, 7-м уровне, – целевое знание. На каждом уровне расположены независимые знания, но разложение знаний по ярусам можно сделать различным образом, например, варианты а) и б) отличаются друг от друга различным расположением независимых вершин по уровням.

Можно образно представить, что по ЯПФ идет фронт обучения, сначала изучаются входные знания, затем знания 1-го уровня и т.д. до целевых знаний на последнем уровне, причем последовательность изложения знаний на каждом из уровней произвольна. Совокупность независимых знаний на каждом уровне ЯПФ названа логическим уровнем. Таким образом, фронт обучения пробегает последовательность логических уровней обучения.

а)



б)

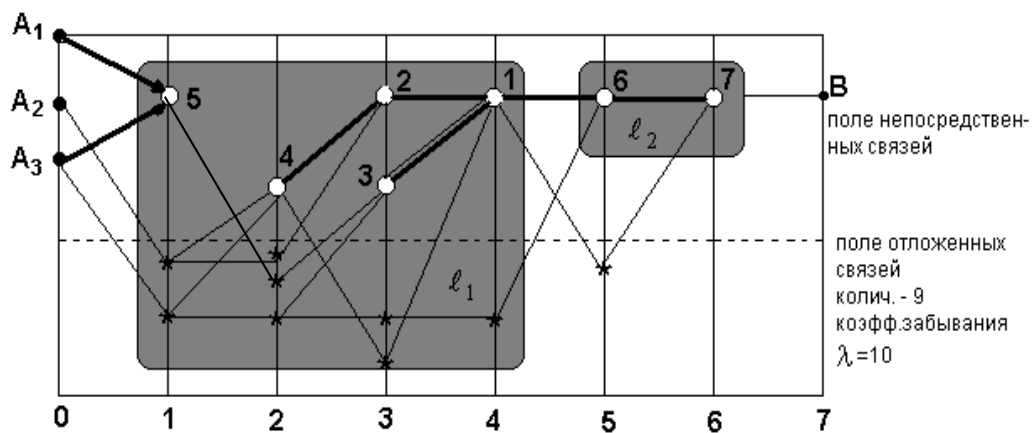


Рис.3. Ярусно-параллельная форма графа KN

3. Планирование учебного процесса для облачных технологий

ЯПФ графа знаний уже дает план учебного процесса, т.е. последовательность прохождения учебного материала фронтально по логическим уровням, при этом порядок изучения блоков одного уровня не важен.

Помимо кратко изложенного выше, в KFS модель входят еще математические модели формирования оценки структурного знания кластера по оценкам знаний входящих в него блоков. Таким образом, если мы имеем KFS модель представления знаний, то планирование учебного процесса в рамках кластера сводится к следующей процедуре:

1. Автоматическая генерация ЯПФ для всех кластеров сверху вниз сразу или по мере необходимости.

2. Формирование алгоритма изучения кластера по соответствующим ЯПФ, при этом порядок изучения блоков одного яруса будет планироваться в реальном времени, например в зависимости от загрузки серверов в «облаке».

3. Процедуры проверки знаний синхронизируются по ярусам, но могут выполняться независимо на одном ярусе, если не заданы дополнительные ограничения, например по времени.

4. Если процедура проверки целевых знаний учебного блока является комплексной и состоит из нескольких более простых процедур, то они считаются независимыми и могут быть распределены между любыми подходящими серверами «Облака».

5. Если учебный блок, в свою очередь, является кластером, то для него повторяется вся процедура планирования.

4. Система интернет-обучения ГЕКАДЕМ 5.0 на основе облачных технологий

Система интернет-обучения ГЕКАДЕМ [4] с самого начала была построена на основе KFS- модели представления знаний, поэтому сейчас есть возможность перевести всю работу в ГЕКАДЕМ «облака», а всем учебным заведениям, использующим эту систему, во-первых, самим пользоваться этими технологиями, а во-вторых, стать серверами «облачного» обучения и оказывать образовательные услуги как отдельным лицам, так и организациям, фирмам, компаниям.

Система дифференцированного интернет-обучения ГЕКАДЕМ обеспечивает надежную работу следующих групп пользователей:

- руководители и администрация образовательного учреждения;
- преподаватели - разработчики интернет-курсов;
- студенты;
- преподаватели, сопровождающие курс;
- системный администратор.

Для каждого пользователя система поддерживает авторизованный доступ к своим ресурсам в соответствии с полномочиями участника учебного процесса.

Управление учебным процессом

Система ГЕКАДЕМ дает возможность руководству образовательного учреждения реализовывать свою политику в области интернет-обучения:

- готовить учебные программы для специальностей и специализаций;
- определять перечень требований к учебным курсам;
- составлять учебные планы для каждой программы;
- вести работу с разработчиками курсов;
- организовывать процесс обучения в соответствии с учебными планами.

Индивидуальное обучение

Наряду с традиционной организацией учебного процесса (факультеты, программы, сессии, группы) в систему заложена опция, позволяющая организовать индивидуальное интернет-обучение, при котором студент имеет возможность самостоятельного выбора интернет-курсов.

Формирование учебного курса

В системе ГЕКАДЕМ преподаватель - разработчик курса – формирует структуру учебного курса из учебных блоков, размещает туда учебный материал в подходящей форме: текст, графика, звук, видео, гипертекст, игры и т.п. Новым типом является WIKI материал, который позволяет привлечь студентов для работы над ним, вместе формировать новые знания.

Личная интернет-библиотека преподавателя, где он может размещать и хранить любую информацию, существенно облегчает процесс разработки курса. Созданная версия является слоем курса, преподаватель может на его основе легко создать новую версию (слой) курса.

Проверка знаний

Для закрепления и проверки знаний, формирования практических навыков по темам конструктор курса может использовать разные процедуры, разработанные в системе ГЕКАДЕМ.

Подсистема формирования online тестов, которая позволяет компоновать комплексные тесты из вопросов типа:

- да- нет;
- выбрать правильные варианты;
- упорядочить варианты;
- вставить пропущенный текст;
- свободный ответ

При формировании вопросов можно использовать иллюстрации любого формата. Все разработанные преподавателем тесты хранятся в системе, их легко пополнять и модифицировать, составлять на их основе новые версии тестов.

Центр семинаров обеспечивает коллективное обсуждение студентами вопросов, предложенных преподавателем. Реализован в виде отдельной подсистемы, обеспечивающей автоматическую авторизацию участника семинара и организацию их работы в соответствии с его полномочиями. Разработчик учебного курса может многократно использовать созданные семинары в разных курсах или слоях одного курса, как в качестве самостоятельного раздела учебного блока (темы), так и в качестве формы контроля знаний.

Групповой проект позволяет организовать совместную работу студентов над общим проектом. Разработчик учебного курса задает темы проектов в виде краткого описания, по каждой теме автоматически формируется библиотека проекта, куда его участники могут помещать материалы для общего пользования, и интернет-семинар проекта для обмена мнениями. При обучении студент выбирает для работы проект и работает над темой в составе команды. Тьютор в завершение проекта оценивает общий результат и вклад каждого участника.

Принципиально новой формой группового проекта является *WIKI проект*, где команда готовит совместную статью на тему, заданную преподавателем. Тьютор может направлять работу, а потом оценить готовую статью в целом и степень участия в ее формировании отдельных соавторов, поскольку ведется журнализация изменений.

Для работы со студентами преподаватель может также использовать *индивидуальные задания*.

Выполнение заданий может быть ограничено по времени.

Кроме того, авторы могут постоянно развивать курс, обновлять учебный материал, готовить на его основе версии курса для определенных условий.

Обучение в системе

В системе ГЕКАДЕМ студент выбирает для себя наиболее подходящий путь изучения курса и работает в индивидуальном режиме в удобное время, следуя собственному темпу работы. Он выполняет индивидуальные задания, тесты, участвует в работе семинаров, работает в групповых и WIKI проектах, может получить консультацию преподавателя и общаться с сокурсниками. Вся работа студента протоколируется в системе, что позволяет преподавателю проводить ее анализ.

Сопровождение курса

Преподаватель, сопровождающий курс (тьютор), имеет возможность контролировать процесс изучения курса каждым студентом, оценивать выполненные им индивидуальные задания, его работу в семинарах и при необходимости оказывать помощь или давать совет. Тьютор также может проводить общий анализ состояния дел в изучении конкретного курса.

Организация, мониторинг и анализ учебного процесса

Система ГЕКАДЕМ позволяет администрации полностью организовать работу учебного заведения, анализировать учебный процесс в целом, по курсу, по циклу курсов, программе, оценивать динамику, выявлять возможные проблемы. Руководство может контролировать работу преподавателей и каждого студента.

Вышеприведенный функционал полностью реализован в 4-й версии системы ГЕКАДЕМ, при этом активно использовались средства **Веб 2.0**. Переход к «облачным» технологиям мы реализуем в 5-й версии системы ГЕКАДЕМ,

Хотя, как уже отмечалось, KFS модель представления знаний разрабатывалась до «облачных» технологий, оказалось, что она может служить хорошей теоретической базой для перехода к «облаку» образования. Покажем это на примере системы ГЕКАДЕМ.

KFS модель представления знаний на уровне программ, где блоками являются предметы, формируется руководством. При этом указываются междисциплинарные связи, задаются статусы: обязательный, по выбору, факультатив и т.д.; вес курса в программе в баллах (кредиты), формы контроля знаний и предмета, возможно, ресурс времени. Граф формируется автоматически по характеристикам предметов, затем по графу генерируется ЯПФ. Предполагается, что уровни в ЯПФ учебной программы должны соответствовать семестру. Система предлагает варианты разбивки предметов на семестры с учетом дополнительных ограничений на загрузку студентов и ограниченность других ресурсов.

Преподаватель – разработчик курсов – строит KFS модель знаний соответствующего предмета, разбивая его на учебные блоки-темы, задает связи между темами и определяет процедуры проверки целевых и/или входных знаний для сформированных блоков, при этом он может ввести ограничения по времени. У него есть также возможность детализировать отдельные темы, которые в дальнейшем могут рассматриваться как кластеры.

Формируется KN граф и строится ЯПФ курса, по которой генерируются возможные траектории (пути) изучения этого курса, из которых студент потом выберет наиболее подходящий.

После заполнения курса учебными материалами и конкретными процедурами проверки знаний разработчик передает курс руководству для использования в учебном процессе, сохраняя за собой право на этот интеллектуальный продукт.

Деканат после проверки на соответствие всем требованиям может оставить этот курс для монопольного использования в организации, а может передать его как ресурс в «облако» ГЕКАДЕМ.

Таким образом, сейчас основным ресурсом «облака» интернет-обучения ГЕКАДЕМ являются интернет-курсы и/или скомпонованные из них учебные программы, использовать их можно как в рамках учебного процесса в учебных заведениях, так и в индивидуальном обучении.

Организованное обучение

Открывая сформированную учебную программу для обучения, руководство может назначить для предмета курс, разработанный в этой организации (как делается сейчас), а может передать право назначения в «облако», где накапливаются курсы разных организаций. Но контроль за учебным процессом образовательное учреждение оставляет за собой.

Учебное заведение или организация может взять в «облаке» полностью готовую программу и по ней вести обучение групп студентов или сотрудников.

Индивидуальное обучение

Любой желающий может обратиться в ГЕКАДЕМ – «облако» с запросом получить сертифицированное образование, и/или индивидуальное самообучение.

Сертифицированное образование предполагает, что после успешного завершения выпускник получает сертификат или диплом. При наличии подходящих программ студент может быть на нее назначен, он выбирает один из вариантов, сформированных по ЯПФ этой программы, и обучается. Он может сам компоновать индивидуальный учебный план в рамках ЯПФ этой программы

Нет никаких ограничений для самообучения в «облаке», можно выбирать любой курс или программу.

Таким образом, при переходе к «облачному» образованию, как ни странно, в выигрыше остаются все:

- Обладатели интеллектуальных ресурсов: курсов или программ, освобождаются от рутинной организационной работы.
- Организации, в которых необходимо вести обучение, могут пользоваться имеющимися в ГЕКАДЕМ «облаке» образовательными ресурсами.
- Все, желающие получить образование, повысить его уровень или просто больше знать, могут легко это сделать в «облаке» обучения.

При этом важно, что происходит постоянное накопление образовательных ресурсов и опыта работы с ним. Это в дальнейшем может создать основу для интеллектуализации самого «облака». Также представляется многообещающим перейти к учебной теме внутри курса или даже к отдельному материалу, процедуре проверки знаний как минимальной единице работы в «облаке», теоретических проблем тут не просматривается, но организационные, например согласование ответственности, автору пока кажутся трудно преодолимыми.

Литература

1. Клементьев И.П., Устинов В.А. Введение в облачные вычисления. URL: <http://www.intuit.ru/>
2. Курганская Г.С. Дифференцированная система обучения через Интернет. – Иркутск: Изд-во Иркутского университета, 2000. – 103 с.
3. Поспелов Д.А. Введение в теорию вычислительных систем. – М.: Советское радио, 1972. – 280 с.
4. Система ГЕКАДЕМ. URL: <http://www.hecadem.irk.ru/>

Курганская Галина Сергеевна, доктор физико-математических наук, зав. кафедрой ИТ в управлении Иркутского государственного университета, тел. 89148833079. E-mail: galina@buk.irk.ru

Kurganskaya Galina Sergeevna, doctor of physical and mathematical sciences, head of department of IT in management, Irkutsk State University. E-mail: galina@buk.irk.ru

УДК 004.75

© Г.М. Ружников, А.Е. Хмельнов, А.С. Гаченко

СОЗДАНИЕ И РАЗВИТИЕ ГИС «ИНВЕСТОРА» г. ИРКУТСКА

Статья посвящена актуальным проблемам создания и развития геоинформационных систем в сфере управления городскими муниципалитетами.

Ключевые слова: ГИС «Инвестора», метаданные, информационные системы, географические электронные карты.

© G.M. Ruzhnikov, A.E. Khmel'nov, A.S. Gachenko

WORKING OUT AND DEVELOPMENT OF GIS «INVESTORA» OF IRKUTSK

This article is devoted to the actual problems of working out and development of geoinformation systems in the area governance of city municipalities.

Keywords: GIS «Investor», metadata, information systems, geographical information cards.

С целью повышения инвестиционной привлекательности территорий муниципалитетов и увеличения бюджета городов в России на протяжении ряда лет активно разрабатываются и внедряются инвестиционные карты, или геоинформационные системы (ГИС) «Инвестора» [3-5]. Создание таких систем требует интеграции информации о городской территории, содержащейся в базах данных (БД) и на электронных картах существующих муниципальных информационных систем (МИС). Геоинформационные системы «Инвестора» являются уникальными для каждого города и не могут тиражироваться, поскольку содержащие исходные данные МИС являются специфичными для каждого города, а их структура определяется сложившимися особенностями муниципального управления и длительной историей разработки. В данной работе рассматривается опыт создания ГИС «Инвестора» г. Иркутска на основе оригинальных технологий и технических решений.

Цель разработки ГИС «Инвестора» г. Иркутска – создание для потенциальных инвесторов благоприятных условий путем обеспечения открытости и доступности информации, необходимой для выбора на территории города объектов, представляющих возможный интерес. Система содержит актуальные пространственные и тематические данные об объектах территориальной инфраструктуры города, в состав которых входит общедоступная информация о границах земельных участков, инфраструктуре, населении города и т.д.



Рис. 1. Структура ГИС «Инвестора»

Основным элементом ГИС «Инвестора» является информационный WEB-ресурс, который разработан на основе оригинальных технологий публикации пространственных и тематических данных с использованием механизмов декларативных спецификаций. Остальные составляющие системы предназначены для администрирования и организации взаимодействия с муниципальными информационными системами, являющимися источниками исходной информации (рис. 1).

В качестве базовой топоосновы используется цифровой адресный план г. Иркутска, который регулярно автоматически обновляется по дежурному цифровому топографическому плану города [7]. Информация для ГИС «Инвестора» автоматически формируется из тематических слоев муниципальной геоинформационной системы г. Иркутска (МГИС). Кроме того, у системы имеется собственная БД, которая предназначена для хранения служебной информации, а также быстрого добавления табличных данных, получаемых из внешних источников.

Большинство электронных карт, используемых в г. Иркутске, представлено в формате ГИС «КАРТА». Нарращивание информационной компоненты происходит за счет загрузки новых таблиц и привязки информации из них к объектам на карте с использованием механизма геокодирования. Также существует возможность обновления данных в режиме добавления на карту новых слоев с уже существующей в них семантической информацией.

Структурно ГИС «Инвестора» состоит из подсистемы web-публикации (Web-сервер) под управлением IIS, подсистемы «Администрирование», специальных конверторов, набора тематических и графических данных. Функционально ГИС «Инвестора» обеспечивает визуализацию всех доступных слоев МГИС с возможностью их включения/отключения, поиска объектов, выбора объекта или группы объектов, отображения семантической информации, интегрированной по всем включённым в выборку слоям карты. Существует возможность печати и вывода в файл информации в соответствии с пользовательским запросом, что удобно для принятия инвестиционных решений.

Модуль Web-публикации – интернет-ресурс, предоставляющий информацию для анализа и оценки реализации проектов перспективного развития территории. Интерфейс представления максимально прост и доступен для восприятия пользователей, визуально компактно организован. Основным элементом пользовательского интерфейса является цифровой адресный план города.

Панель управления содержит элементы управления ГИС функциональностью (навигация, масштабирование, измерение расстояния между объектами), выбора объектов карты, в том числе по произвольному контуру, отображения легенды слоев карты, отображения семантической информации в области отчетов, поиска объектов на карте по адресу и другим атрибутам.

В ГИС «Инвестора» для web-публикации картографических данных используется формат SMD (Static Map Data) [1]. Это оригинальный формат (разработка ИДСТУ СО РАН) представления векторных данных. Он позволяет эффективно (по времени отображения и времени загрузки информации в память) визуализировать пространственные объекты. Перед публикацией данных подсистема «Администрирование» автоматически проводит конвертацию картографических материалов в этот формат.

Наиболее ресурсоемким и информативным инструментом, реализованным в интерфейсе пользователя, является блок построения запросов, который позволяет создавать отчетные выборки. Для вывода информации об объектах из базы данных используется технология декларативных спецификаций [2]. Одной из разновидностей запросной системы является выборка по произвольной (ограничивающий многоугольник) области (рис. 2). При выполнении такого запроса формируется отчет по объектам карты указанной области из отмеченных в дереве информационных слоёв. По каждому из выбранных слоёв вычисляются итоговые значения, а также может выводиться список объектов данного слоя, если выборка выполняется по ограниченной области, содержащей не слишком много таких объектов. При выводе информации об объектах слоя в отчет включаются карточки записей из связанной со слоем таблицы БД. Для формирования таких карточек также используются декларативные спецификации структуры БД [1, 2].

В целях исключения длительного формирования отчетных данных в связи с выбором большого количества объектов разработан блок прогнозирования времени формирования отчетов и оптимизации запросов с предоставлением пользователю информации о количестве объектов, включенных в его запрос.



Рис. 2. Возможности запросной системы ГИС «Инвестора»

Подсистема «Администрирование» предназначена для конфигурирования модуля web-публикации и подготовки информации из различных тематических и графических баз данных. К основным возможностям данной компоненты относятся:

- управление логической и физической структурой графической и связанной с ней информацией тематических баз данных для публикации и управления конфигурацией созданного интернет-ресурса. Здесь также используются механизмы декларативных описаний спецификаций;
- автоматическая загрузка данных в БД из таблиц Excel;
- просмотр слоёв в подготовленном к публикации виде и фиксация определенной информации от ответственного лица, подтверждающей проведенный контроль слоя и его готовность к публикации (дата, время, реквизиты ответственного лица).

Для того чтобы можно было в определенном виде публиковать информацию из слоев с ограниченным доступом об инженерной инфраструктуре г. Иркутска и возможности подключения к объектам инженерной инфраструктуры, при помощи специального конвертора генерируется бинарный файл в формате OMR (в нем хранится лишь наличие объектов в квадрате, этот файл не хранит в себе метрику и семантику, а только наличие объектов или их отсутствие), данные из этого файла отображаются при формировании отчета по заданной области. Для этого карта разбивается на квадраты с регулярной сеткой шагом в 200 метров. При запросе для формирования сводок получаем текущий номер квадрата, по которому формируется отчет, и пользователь получает информацию в виде максимального и минимального расстояния до объектов инженерной инфраструктуры в данном квадрате. Сконвертированные слои инженерной инфраструктуры невидимы для пользователей и могут использоваться в качестве носителей информации для формирования отчетов. Данная информация предоставляется в составе отчетов только в том случае, если пользователь активировал слой инженерной инфраструктуры в списке слоев ГИС «Инвестора» подсистемы web-публикации.

Пользователи в системе имеют возможность дополнительно просматривать необходимые фрагменты на космоснимках. Разработана технология мониторинга данных на картографических сервисах, которая позволяет просматривать выбранный фрагмент карты на сторонних картографических онлайн-сервисах:

- Карты Google;
- Карты Яндекс;
- OpenStreetMap Свободная вики-карта мира;
- Публичная кадастровая карта.

При выборе необходимого участка карты в системе пользователь может сделать выбор в пользу нужного картографического сервиса, ему будут предоставлены данные со сторонних карт.



Рис. 3. Пример отображения выбранного фрагмента на космоснимках Google (слева оригинал, справа картографический сервис Google)

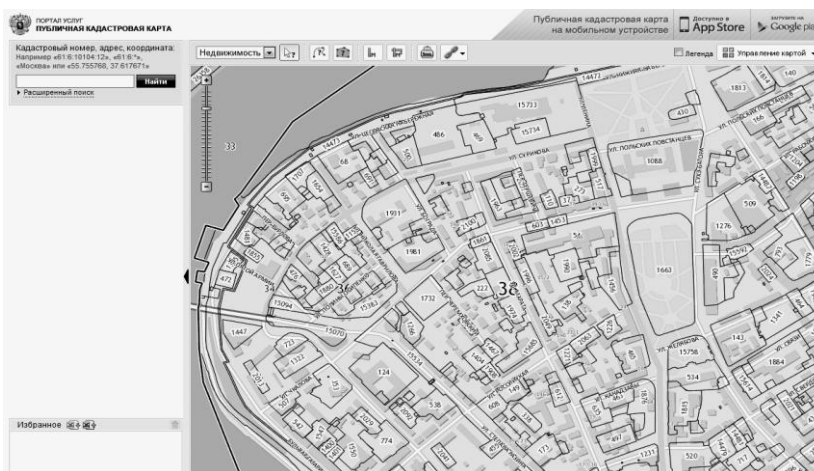


Рис. 4. Пример отображения выбранного фрагмента на публичной кадастровой карте Росреестра

ГИС «Инвестора» г. Иркутска находится в опытной эксплуатации в администрации города и позволяет совершать аналитические запросы потенциальным инвесторам, обеспечивает простое и доступное наращивание информационной составляющей для публикации путём настройки файлов конфигурации. Системы такого рода позволяют получать новые инвестиции, что обеспечит развитие муниципалитетов.

Литература

1. Интеграция информационно-аналитических ресурсов и обработка пространственных данных в задачах управления территориальным развитием / И. В. Бычков [и др.]; под ред. И. В. Бычкова; Рос. акад. наук, Сиб. отд-ние, Ин-т динамики систем и теории управления. – Новосибирск: Изд-во СО РАН, 2011. – 369 с.
2. Ветров А.А., Фереферов Е.С., Хмельнов А.Е. Технология использования метаописаний для формирования хранилищ данных и анализа многомерных данных // Вестник Бурятского государственного университета. – 2011. – № 9. – С. 83-86.
3. ГИС «Инвестора» г. Санкт-Петербург. URL: <http://map.investinfo.spb.ru/>.
4. ГИС «Инвестора» г. Киров. URL: <http://investorkirov.ru/map/map.php>.
5. ГИС «Инвестора» г. Томск. URL: <http://invest-tomsk.ru/>.
6. Корнев В.И., Сидоренко С.В. Интернет-ресурс «ГИС-инвестор» администрации Томска как инструмент поддержки принятия инвестиционного решения // Управление развитием территории. – 2011. – № 1. – С. 36-38.

7. Технология создания и ведения информационной системы «Адресный план» с использованием крупномасштабных электронных карт / А.О. Шигаров и др. // Горный информационно-аналитический бюллетень. – 2009. – Т. 18, № 12. – С. 176-180.

Ружников Геннадий Михайлович, кандидат технических наук, заместитель директора ИДСТУ СО РАН, тел. (395-2) 453006, e-mail: ruginov@icc.ru

Хмельнов Алексей Евгеньевич, кандидат технических наук, заведующий лабораторией ИДСТУ СО РАН, тел. (395-2) 453071, e-mail: hmelnov@icc.ru

Гаченко Андрей Сергеевич, кандидат технических наук, научный сотрудник ИДСТУ СО РАН, тел. (395-2) 453103, e-mail: gachenko@icc.ru

Ruzhnikov Gennady Mikhailovich, candidate of technical sciences, deputy director, Institute for System Dynamics and Control Theory SB RAS, e-mail: ruginov@icc.ru

Khmelnov Alexey Evgenievich, candidate of technical sciences, head of laboratory, Institute for System Dynamics and Control Theory SB RAS, e-mail: hmelnov@icc.ru

Gachenko Andrey Sergeevich, candidate of technical sciences, Institute for System Dynamics and Control Theory SB RAS, e-mail: gachenko@icc.ru

УДК 004.73

© *Р.Р. Фаткиева*

РАЗРАБОТКА МЕТРИК ДЛЯ ОБНАРУЖЕНИЯ АТАК НА ОСНОВЕ АНАЛИЗА СЕТЕВОГО ТРАФИКА¹

В статье представлен подход к детектированию DDoS-атак, базирующийся на анализе изменения метрик, являющихся функциями от трафика, измеряемого на сетевом интерфейсе сервера. В процессе анализа выявлено влияние атак на статистические параметры временных рядов данных метрик.

Ключевые слова: информационная безопасность, распределенный отказ в обслуживании, сетевой трафик, временные ряды.

© *R.R. Fatkueva*

DEVELOPMENT OF METRICS FOR ATTACK DETECTION ON THE BASIS OF NETWORK TRAFFIC ANALYSIS

In the article the approach to DDoS attacks detection is presented. It is based on analysis of metrics values variations, which are functions of traffic measurements at network server interface. The analysis has confirmed the influence of attacks on statistical parameters of time series, generated by the metrics.

Keywords: information security, distributed denial of service, network traffic, time series.

Введение

DDoS-атаки, существуя со времени начала массового использования глобальной сети, по-прежнему остаются одной из самых серьёзных угроз для Web-ресурсов, что свидетельствует о необходимости развития средств защиты от этих атак. Грамотно организованная масштабная DDoS-атака в большинстве случаев приводит к значительным финансовым потерям со стороны жертвы. При отсутствии средств обнаружения вторжений ресурс информационной системы тратится на обслуживание DDoS запросов, при этом стоимость использования ресурса многократно возрастает. Такие нападения отличаются простотой организации и высокой эффективностью. Именно эти особенности привлекают к DDoS внимание как специалистов по сетевой безопасности, так и злоумышленников, что обуславливает актуальность исследования DDoS-атак.

1. Организация измерений

Исследования проводились специализированной системой сбора и управления трафика, позволяющей контролировать характеристики трафика в реальном времени [1]. В качестве исходных данных для проведения исследования выбран сетевой трафик Web-сервера. Трафик снят в двух режимах: в режиме регулярного взаимодействия с клиентами по сети и в режиме DDoS-атаки классов SYN-flood, UDP-flood, HTTP-flood. При этом фиксировались характеристики сетевого трафика и системных характеристик (объем оперативной памяти, время загрузки процессора и т.п.). Наблюдаемое изменение указанных характеристик не обязательно свидетельствует об атаке, но показывает изменение закона распределения.

2. Исследуемые метрики

Изменение объёма сетевого трафика не обязательно свидетельствует о его аномальности, для анализа состояния системы необходимо оценить детально структуру трафика и определить наиболее информативные метрики [2, 3]. Рассмотрим их более подробно.

1. Отношение входящего и исходящего трафика

$$R_{ip} = \frac{T_i}{T_o},$$

¹ Работа выполнена при финансовой поддержке «ИнфоТеКс Академия 2012»

где T_i и T_o – объём соответственно входящего и исходящего IP-трафика в единицу времени. Целесообразность выбора данной величины объясняется тем, что при наличии DDoS-атаки сервер теряет способность отвечать на запросы. Повышение скорости входящего трафика без соразмерного повышения скорости исходящего трафика ведёт к росту величины R_{ip} , что означает более высокую вероятность наличия атаки. На рис. 1 показан этот эффект.

2. Число потоков критических приложений можно использовать для обнаружения атак прикладного уровня. Так как для Web-сервера специфической является атака класса HTTP-flood, целесообразно измерять число N_{web} потоков к Web серверу. N_{web} . Если Web-сервер обращается к другим приложениям (например, к базе данных или системе инженерных вычислений), следует измерить количество потоков и этих приложений. Возрастание N_{web} при возникновении атаки показано на рис. 2.

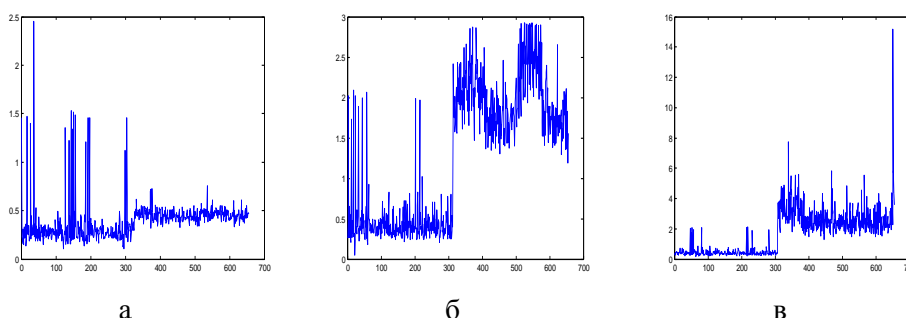


Рис. 1. Изменение параметра R_{ip} при возникновении атаки (а – HTTP-flood, б – SYN-flood, в – UDP-flood)

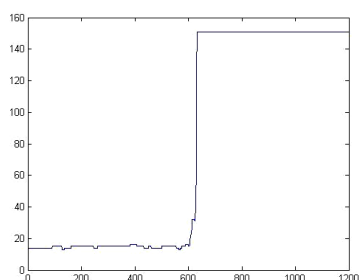


Рис. 2. Изменение числа потоков Apache от времени при начале атаки HTTP-flood

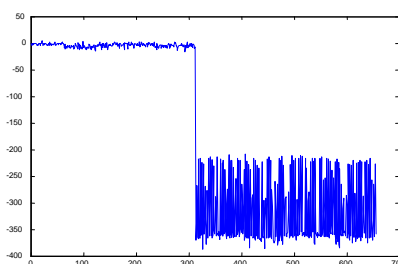


Рис. 3. Изменение D_{ack} при включении SYN-flood

3. Разность

$$D_{ack} = N_{acko} - N_{acki},$$

где N_{acko} – число исходящих ACK-флагов в TCP-трафике в единицу времени, N_{acki} – число входящих флагов, может свидетельствовать о том, как часто сервер отказывает клиенту из-за перегрузки. Такая величина имеет ценность для всех видов атак, в качестве примера показано, как она изменяется при начале атаки SYN-flood (рис. 3). Отрицательное значение D_{ack} показывает, что сервер теряет возможность отвечать на клиентские запросы ACK-пакетами.

4. Отношение

$$R_{udp} = \frac{T_{udp}}{T_{tcp}},$$

где T_{udp} – объём входящего UDP-трафика, T_{tcp} – объём входящего TCP-трафика, может характеризовать наличие атаки класса UDP-flood. UDP – протокол односторонней передачи данных. Хотя в трафике Web-сервера присутствует небольшое количество пакетов, принадлежащих этому протоколу

(рис. 4, а), в целом UDP для HTTP-соединений нехарактерен, поэтому многократное превышение UDP-трафика над TCP-трафиком позволяет выявить UDP-flood (рис. 4, б).

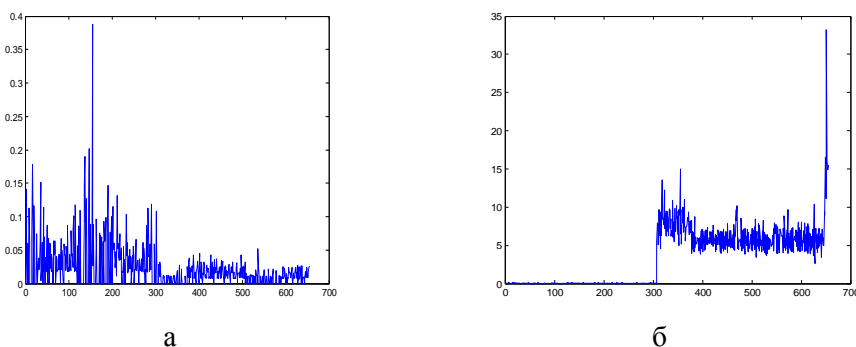


Рис. 4. Значение R_{udp} при начале атаки HTTP-flood (а) и UDP-flood (б)

5. Частоты флагов SYN и PSH во входящих пакетах позволяют определить эффективность передачи данных:

$$R_{syn} = \frac{N_{syn}}{N},$$

$$R_{psh} = \frac{N_{psh}}{N},$$

где R_{syn} – частота флагов SYN, R_{psh} – частота флагов PSH, N_{syn} – число SYN-флагов во входящих пакетах, N_{psh} – число PSH-флагов во входящих пакетах, N – число входящих TCP-пакетов. Пакеты с флагом SYN пересылаются между клиентом и сервером в ходе установления TCP-соединения, после чего начинается обмен данными с помощью пакетов без SYN-флага. Таким образом, число SYN-флагов, пришедших на сервер, равно числу запросов на соединение, а частота SYN-флагов определяет долю служебных пакетов этого типа в TCP-трафике.

Установленный флаг PSH означает, что данные, содержащиеся в пакете, должны быть переданы программе прикладного уровня. В случае Web-сервера эти данные представляют собой HTTP-запросы и HTTP-ответы, содержащие Web-страницы. Поэтому частота PSH-флагов, напротив, характеризует полезную загрузку канала.

При атаке класса SYN-flood субъект атаки не намерен передавать какие-либо данные серверу и пытается перегрузить его очередь соединений с помощью служебных пакетов. Поэтому частота флагов SYN и PSH изменяется, как показано на рис. 5.

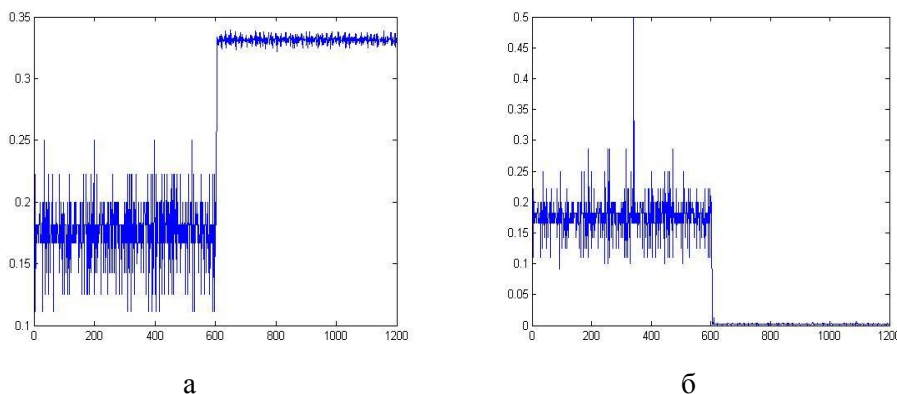


Рис. 5. Частоты флагов SYN (а) и PSH (б) при атаке SYN-flood

3. Статистический анализ

Появление DDoS-атаки приводит к изменению статистических распределений ряда метрик, что можно наблюдать непосредственно. К примеру, гистограмма R_p при появлении SYN-flood изменяет

вид следующим образом (рис. 6). Законы распределения большинства метрик в установившихся режимах (как в штатном, так и в режиме атаки) близки к гамма-закону [4]:

$$p(x) = x^{k-1} \frac{e^{-x/\theta}}{\theta^k \Gamma(k)} \bullet 1(x \geq 0),$$

где k и θ – параметры формы и масштаба. При появлении атаки происходит сдвиг значений, что сказывается прежде всего на параметре формы.

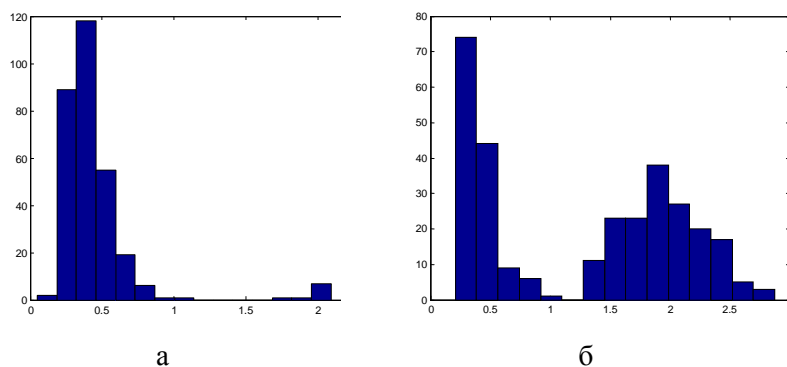


Рис. 6. Гистограмма R_{ip} (а – до атаки, б – во время атаки)

Например, для R_{ip} оценки параметров изменяются следующим образом (табл. 1):

Таблица 1

Изменение оценок параметров гамма-распределения при атаке SYN-flood

Режим \ Параметры	k	θ
Без атаки	4,2	0,1
SYN-flood	22	0,08

Чувствительность метрик к различным типам атак определяется влиянием атаки на параметры статистического распределения метрики. Данные о чувствительности приведены в табл. 2.

Таблица 2

Влияние атак на метрики

Метрика \ Тип	HTTP-flood	SYN-flood	UDP-flood
R_{ip}	+	+	+
N_{web}	+	-	-
D_{ack}	+	+	-
R_{udp}	+	+	+
R_{syn}	+	+	-
R_{psh}	+	+	-

Рассмотрим влияние DDoS-атак на основные параметры распределения метрик, к которым относятся среднее значение и среднеквадратичное отклонение. Для этого построим графики скользящих средних и стандартов интересующих нас метрик. Метод скользящих средних и стандартов позволит, во-первых, провести низкочастотную фильтрацию временных рядов, что уменьшит влияние шума, во-вторых, отследить динамику параметров распределения и влияние атаки. Например, построим графики для атаки SYN-flood (рис. 6, 7):

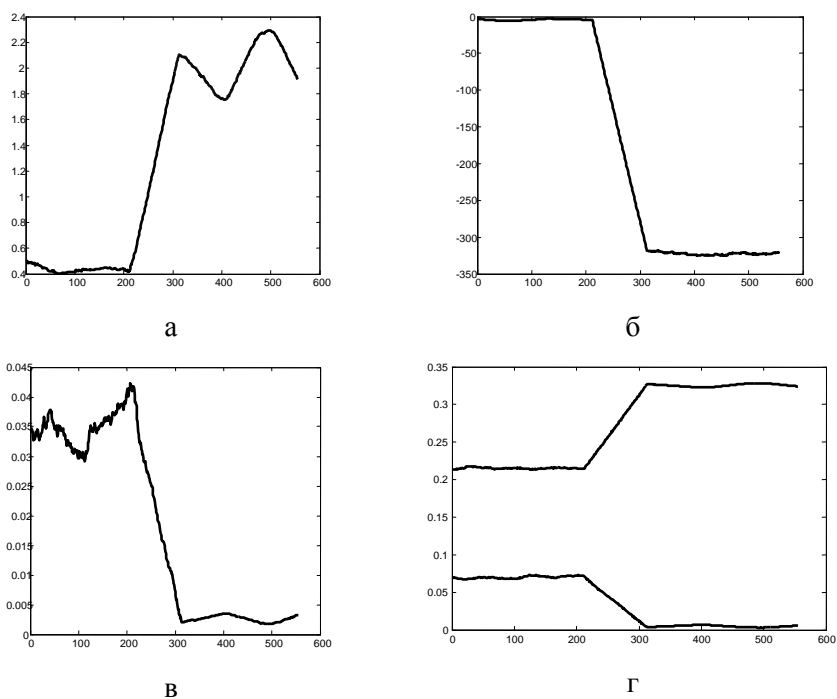


Рис. 6. Влияние SYN-flood на скользящие средние значения метрик:
 а – R_{ip} , б – D_{ack} , в – R_{udp} , г – R_{syn} и R_{psh}

При детектировании атак в автоматическом режиме можно опираться на следующие параметры:
 -пороговые значения приведённых метрик – некоторые из них могут являться точным показателем наличия атаки (например, на Web-сервере выполнение $R_{udp} > 1$ в течение продолжительного времени свидетельствует об атаке UDP-flood с большой вероятностью);

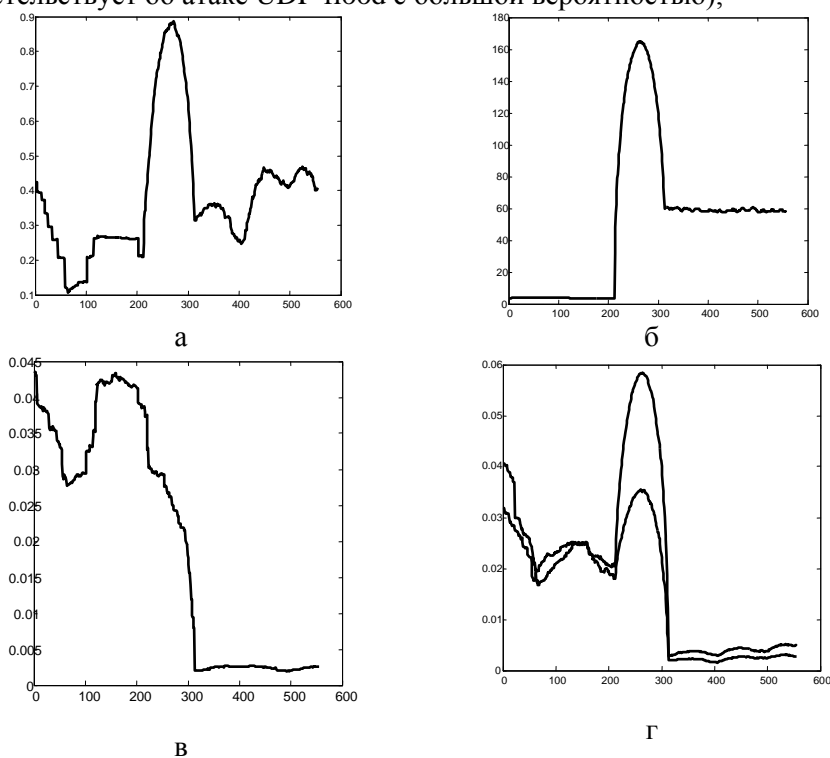


Рис. 7. Изменения среднеквадратичного отклонения при SYN-flood атаке на: а – R_{ip} , б – D_{ack} , в – R_{udp} , г – R_{syn} и R_{psh}

-скорость изменения скользящего среднего, которая достигает максимальных значений в момент начала или прекращения атаки; определение пороговых значений, которые считаются признаком атаки, может быть выполнено с помощью построения шаблона штатного функционирования системы;

-величины скользящих стандартов и скорость их изменения.

Улучшение точности обнаружения атаки достигается также рассмотрением всех приведённых выше параметров в совокупности. Так как характер влияния каждого вида атаки на параметры известен и отличается от влияния других классов атак, представляется возможным не только констатировать факт атаки, но и определить её тип.

Заключение

Параметры распределения величин, основанных на измерениях сетевого трафика, представленные в данной статье, позволяют сформировать подход к разработке системы детектирования атак. Использование библиотек операционных систем, с помощью которых возможно перехватывать сетевой трафик, позволяет измерить трафик и рассчитать в реальном времени значения рассмотренных в этой статье метрик, моменты их распределений и скорость изменения моментов. Полученные данные можно использовать для принятия решения о наличии атаки и вызове подсистемы блокировки.

Литература

1. Воробьев В.И., Евневич Е.Л., Фаткиева Р.Р. Моделирование сетевого трафика методом Монте-Карло // Вестник Бурятского государственного университета. – 2010. – Вып. 9. – С. 258–262.
2. Фаткиева Р.Р. Модель обнаружения атак на основе анализа временных рядов // Труды СПИИРАН. – 2012. – № 2. – С. 71-79.
3. Фаткиева Р.Р. Корреляционный анализ аномального сетевого трафика // Труды СПИИРАН. – 2012. – Вып. 23. – С. 93-100.
4. Бахарева Н.Ф., Ушаков Ю.А. Программная система распознавания трафика и прогнозирования характеристик мультисервисной сети // Инфокоммуникационные технологии. – 2008. – Т.6, №4. – С. 42-50.

Фаткиева Роза Равильевна, кандидат технических наук, старший научный сотрудник лаборатории информационно-вычислительных систем, Санкт-Петербургский институт информатики и автоматизации РАН (СПИИРАН). Р.т. +7(812)328-4369, факс +7(812)328-4450. E-mail: rikki2@yandex.ru

Fatkiewa Rosa Ravilevna, candidate of technical sciences, senior researcher, laboratory of computer and information systems, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). E-mail: rikki2@yandex.ru

УДК 681.51

© Л.Н. Федорченко

СИНТАКСИЧЕСКИ УПРАВЛЯЕМАЯ ОБРАБОТКА ДАННЫХ ДЛЯ ПРАКТИЧЕСКИХ ЗАДАЧ

В статье рассматриваются различные определения синтаксиса и семантики КС-языка на основе атрибутного подхода. Сравнивается синтаксическое описание атрибутов в VW и аффиксных грамматиках Костера (affix grammars) для задания трансляций. Представлен метод описания синтаксиса КС-языка с помощью синтаксической граф-схемы (СГС), построенной по регулярным выражениям. Показан принцип регуляризации грамматики для её приведения в более простой класс грамматик с помощью эквивалентных преобразований СГС. Дан алгоритм одного эквивалентного преобразования КС-грамматики в регулярной форме.

Ключевые слова: эквивалентные преобразования грамматик, аффиксная грамматика, КСР грамматика, атрибутный подход, статическая семантика.

© L.N. Fedorchenko

SYNTAX-DRIVEN DATA PROCESSING FOR PRACTICE

The article is devoted to the different definitions of syntax and semantics of programming languages on the basis of attribute approach. The syntactic description of attributes in VW and Koster affix grammars are compared for tasks of translations. The method of syntax definition of CS language is presented by means of the syntactic graph-scheme (SGS) constructed on regular expressions. To optimize a parser into simpler class of grammar the principle of grammar regularization has been shown, it uses the SGS equivalent transformations. The algorithm of one of equivalent transformations for the CS-grammar in a regular form has been given.

Keywords: equivalent transformations of grammar, affix grammar, CSR grammar, attribute approach, static semantics.

Введение

При построении синтаксического анализатора как составной части транслятора необходимо проводить эквивалентные преобразования грамматики реализуемого языка. Сейчас языковые технологии активно включаются в различные сферы нашей жизни, что привело к развитию современных транслирующих систем, ориентированных на разнообразный ассортимент вычислительных устройств во многих предметных областях. При этом проявилась проблема быстрой настройки (преобразования) синтаксического определения языка на ту форму, которая допускает автоматическую или ручную реализацию, а также проблема учёта ограничений выбранного метода синтаксического анализа. Первая проблема обусловлена разнообразием спецификаций реализуемых языков, диапазон которых простирается от обычной формы Бэкуса-Наура (БНФ) и языка разметки HTML до двухуровневых и других видов грамматик. Вторая – ведёт либо к языковой неоднозначности, либо к недетерминированности распознающего автомата. Решение этих проблем связано с корректным отображением транслируемого языка во внутреннее машинное представление. Для этого следует эффективно использовать информацию о языке, определяя его синтаксис и статическую семантику специальной КС грамматикой (трансляционной), которая помимо основной своей функции порождения цепочек языка позволяет задавать трансляции, необходимые разработчикам.

Существующие подходы к реализации языков и обширный набор средств автоматического построения трансляторов, как правило, используют встроенные в инструментальные системы разработки эквивалентные преобразования, часть из которых выполняется в ходе построения анализатора, а часть – вручную, что существенно замедляет процесс разработки и приводит к большому числу ошибочных итераций. Стало очевидным, что необходимо выполнять предварительную обработку (пре-процессинг) с целью приведения синтаксических определений языка к нужной форме с помощью автономного инструментария, позволяющего заранее сформировать синтаксис языка и оптимизировать

анализатор с применением процедуры регуляризации исходной трансляционной грамматики с помощью эквивалентных преобразований. Такой инструмент должен иметь удобный интерфейс и обладать достаточным набором функций для выполнения преобразований. Исследование этих преобразований и их применение в технологиях разработки трансляторов ведёт к разрешению проблем языковой неоднозначности и недетерминированности при автоматическом построении транслятора.

Формальные методы программной инженерии опираются на фундаментальные результаты, полученные в работах по анализу алгоритмов, в теории вычислений и теории трансляций языков. С теоретической точки зрения была описана иерархия из четырех классов грамматик (по Хомскому), вложенных последовательно один в другой, – от регулярных до рекурсивно-перечислимых. Контекстно-свободные языки в этой иерархии занимают промежуточное место после регулярных языков. Для большинства языков, используемых на практике, контекстно-свободные грамматики достаточны для описания их синтаксиса.

Формальный язык называется языком типа X (по иерархии Хомского), если существует хотя бы одна грамматика данного типа, которая порождает данный язык. Реализаторы языка стремятся выделить составляющую, определяющую язык, с помощью такой грамматики типа Y , которая наиболее проста для обработки, и таким образом перевести реализуемый язык из класса языков типа X в класс языков типа Y , для которого уже существуют отработанные алгоритмы и изученные свойства, например, из класса КС-языков в класс регулярных языков. Теоретики языков, наоборот, стремятся описать язык более выразительными и мощными формализмами, стремясь вместить в формализацию как можно больше свойств языка, например, формализовать семантику, в частности статическую, которую в отличие от динамической семантики можно обработать на фазе синтаксического анализа.

1. Определение статической семантики в языках программирования на основе атрибутивного подхода

Каждый программист-разработчик имеет интуитивное представление о том, что такое синтаксис и что такое семантика языка программирования. Однако трудно найти точное общепринятое определение этих понятий.

С прагматической точки зрения синтаксис – это множество правил, благодаря которым компилятор может проверить правильность записи программы, а семантика является понятием, свойства которого могут быть выявлены только во время выполнения программы.

С другой стороны, в некоторых теоретических работах под синтаксисом понимают множество правил контекстно-свободной грамматики (КС-грамматики), а все остальное в определении языка относят к семантике. В этом случае семантику делят на две части: те свойства языка, которые можно проверить в процессе трансляции (компиляции), называют статической семантикой, а остальные свойства языка – динамической семантикой.

Оставляя за рамками статьи споры о терминологии, будем считать, что КС-грамматика, определяющая тексты языка, называется синтаксисом, некоторые правила, относящиеся к правильному использованию слов языка, – статической семантикой, тогда как под динамической семантикой понимается то, каким образом может быть установлено input/output отображение, создаваемое программой в процессе исполнения.

Если считать, что программа состоит из слов, то терминальные символы КС-грамматики будут словами языка. В сравнении с естественными языками большинство языков программирования отличается тем, что некоторые слова не имеют заранее определенного значения или их значение определено частично. Необходимые атрибуты таких слов фиксируются только в программе. Например, идентификатор может обозначать либо тип константы или тип переменной, либо метку или операцию. В языке BASIC буква, за которой следует открывающая скобка, может обозначать либо одномерный, либо двумерный массив и т.д. Поэтому большинство языков программирования имеет встроенные механизмы проверки синтаксико-семантических свойств языка для всех допустимых слов в программе.

В реальных компиляторах все синтаксические проверки осуществляются, как правило, с помощью таблиц, где регистрируются текущие атрибуты проанализированных слов.

С теоретической точки зрения к настоящему времени создано несколько абстрактных формализмов для описания языков. Работа по созданию таких формализмов велась в трех направлениях:

- 1) замена формализма КС-грамматики на более мощную грамматику (с целью формального определения статической семантики языка);
- 2) встраивание атрибутов и предикатов (проверок) в правила КС-грамматики.
- 3) введение и пошаговая модификация декларативного состояния преобразователя, построенного по грамматике.

В первом направлении было сделано несколько безуспешных попыток до того момента, как была придумана грамматика ван Вейнгаардена (VW), которая использовалась в определении языка ALGOL 68. Отдельные семантические свойства языка были описаны и включены в «Сообщение о языке ALGOL 68» [7].

Во втором направлении важной работой стала статья Д. Кнута (D. Knuth) [9], суть которой заключается в том, что каждая синтаксическая конструкция имеет наследуемые и синтезированные атрибуты. Затем появилась аффиксная грамматика Костера [8], в которой идея с атрибутами Кнута получила дальнейшее развитие. Работы Костера по аффиксным грамматикам стали теоретической основой при определении языка CDL(Compiler Definition Language). К этому направлению относятся и многочисленные функциональные грамматики. Помимо Кнута похожий метод значительно позже был предложен Гриффитсом (M.Griffits) [12].

Третье направление, которое основывается на работе Ледгарда (H.F.Ledgard)[10], было реализовано в работах Вильямса (M.H.Williams)[11] и других авторов. В этом случае имеется таблица (или что-то подобное ей), содержащая текущие имена с текущими атрибутами, и есть множество функций, которые связаны с синтаксическими конструкциями для модификации состояния таблицы. В таком методе терминальные символы КС-грамматики имеют функции перехода из состояния в состояние. Атрибуты для синтаксических единиц более высокого уровня вводятся так же, как в грамматике (VW) ван Вейнгаардена [7,16]. Такой метод дает более ясное разделение между синтаксисом и статической семантикой. Ниже приводится пример применения данного метода для фрагмента реального языка.

2. Грамматика ван Вейнгаардена (VW) и порождаемый ею язык

Граматику ван Вейнгаардена можно рассматривать как КС-грамматику с бесконечным числом порождающих правил. Это бесконечное множество правил задаётся конструктивно. Символы грамматики обозначаются длинными буквенными строками и называются *протопонятиями*. Терминальные символы заканчиваются словом “-symbol”. Все символы грамматики отделяются друг от друга запятыми.

Для построения порождающих правил КС-грамматики даны *схемы*. Это так называемые *гиперправила*, левые и правые части которых состоят из понятий (символов грамматики), составленных из прописных слов (*протопонятий*) и из слов заглавного алфавита (*метапонятий*). Эти метапонятия являются *параметрами* гиперправила.

Для каждого метапонятия дана КС-грамматика, порождающая, возможно, бесконечное множество протопонятий.

Порождающее правило генерируется из гиперправила таким образом, что одно и то же метапонятие должно быть заменено на одно и то же протопонятие. Такой способ замены называется *согласованной подстановкой* (или *синхронной* заменой).

Далее из начального символа грамматики VW можно вывести строку (слово языка) так, что все нетерминальные символы в результате согласованных подстановок будут последовательно заменяться до тех пор, пока не получим строку, состоящую только из терминальных символов.

Используя метапонятия, мы можем генерировать порождающие правила алгоритмом типа Маркова. Некоторые символы в гиперправилах играют особую роль, а именно: из них можно вывести пустую строку. Такие символы грамматики называются предикатами. Они могут выразить отношение между параметрами-метапонятиями. Если такое отношение имеет значение «истина», значит, мы можем вывести пустую строку. Если отношение ложно, вывод останавливается и нетерминальный символ остаётся в текущей полувыведенной строке. В этом случае такой вывод считается неверным, а ветка вывода – тупиковой.

```

Метаправила:
'ALPHA': A; B;...X; Y; Z.
'LETTER': letter 'ALPHA'.
'NAME': 'LETTER'; 'LETTER' NAME'.
'DEF': 'NAME' has 'MODE'.
'TABLE': 'DEF'; 'TABLE' 'DEF'.
'DEFSETY': 'TABLE'; 'EMPTY'.
'EMPTY': .

Гиперправила:
Program:
    Begin symbol, Declare of 'TABLE', 'TABLE'
    restrictions,
    'TABLE' statement train, end symbol.
'DEFSETY' 'NAME' has 'MODE' restrictions:
    where 'NAME' is not in 'DEFSETY',
    'DEFSETY' restrictions;
    where 'DEFSETY' is 'EMPTY'.
Where 'NAME1' is not in "NAME2" has 'MODE' 'DEFSETY':
    where 'NAME1' differs from 'NAME2',
    where 'NAME1' is not in 'DEFSETY';
    where 'NAME1' differs from 'NAME2',
    where 'DEFSETY' is 'EMPTY'.
    Where 'EMPTY' is 'EMPTY':
    'EMPTY'.
    Where 'NAME1' letter 'ALPHA1'differs from 'NAME2' letter
'ALPHA2':
    where 'NAME1' differs from 'NAME2';
    'ALPHA1 is not 'ALPHA2'.
Where 'NAME' letter 'ALPHA1' differs from letter 'ALPHA2':
    'EMPTY'.
Where letter 'ALPHA1' differs from 'NAME' letter 'ALPHA2':
    'EMPTY'.
A is not B: 'EMPTY'.
A is not C: 'EMPTY'.
A is not D: 'EMPTY'.
и т.д.

```

Рис.1. Пример 1 грамматики ван Вейнгаардена

Примечание. Двухуровневые грамматики более выразительны, чем грамматики типа 0 по иерархии Хомского. При некоторых допущениях двухуровневые грамматики могут порождать языки с неограниченным набором терминальных символов. Грамматики типа 0 не могут. Доказательство этого факта можно найти в работах Дика Груне (Dick Grune) [14, 15, 17].

В примере 1 покажем фрагмент двухуровневой грамматики, с помощью которой можно проверить свойство, что каждая переменная в программе объявлена только один раз. (Заметим, что альтернативы в правилах разделяются точкой с запятой, левая и правая части КС-правила – двоеточием, правило заканчивается точкой).

В этом фрагменте 'TABLE' restrictions является предикатом, который проверяет единственность описания. Из примера видно, что такой тип определения является понятным и исчерпывающим. С другой стороны, мы можем видеть, что такое определение является скорее избыточным, не математическим, но на практике реализуемым, поскольку используются очень простые функции и трюки.

3. Аффиксные грамматики и язык описания компилятора CDL

Другое решение было предложено Д. Кнутом (D.E. Knuth). Оно состояло в том, что каждая грамматическая единица в КС-грамматике (каждый грамматический символ) имеет набор атрибутов. Атрибут называется «восходящим» или синтезируемым (ascendent), если он выведен из атрибутов грамматических единиц более низкого уровня. Атрибут называется «наследуемым» (descendent), если он выведен из атрибутов грамматической единицы более высокого уровня.

Такая концепция вместе с концепцией VW грамматики породили концепцию аффиксной грамматики, где рассматривается 3 типа объектов. Нетерминальные символы обозначают грамматические единицы, терминальные символы, которые являются словами программы (или языка, порождаемого грамматикой), и предикаты, осуществляющие проверки. Эти объекты имеют конечный набор атрибутов. Терминальные символы не имеют атрибутов. Кроме этих объектов в аффиксной грамматике есть конечное множество порождающих правил, подобных правилам КС грамматики. Каждый нетерминальный символ может быть заменён строкой, состоящей из терминальных, нетерминальных символов и предикатов (проверок). Атрибуты могут располагаться в обеих частях правила (левой и правой)

и связаны с грамматическими объектами. Атрибут может иметь символьное или константное значение.

Если нетерминальный символ в левой части правила имеет атрибуты с определённым значением, то мы можем заменить его на строку в правой части правила.

Пример 2. Ниже приводим пример аффиксной грамматики, аналогичный грамматике VW из предыдущего примера 1, однако детально рассмотрим иную часть вывода.

```
Program: Begin,  
        Declaration+'TABLE',  
        Restrictions+'TABLE',  
        Statement train+'TABLE',  
        End.  
  
Declaration+'TABLE': Declare+'MODE'+ 'TABLE';  
                    Declare+'MODE'+ 'SUBTABLE1',  
                    Declaration+'SUBTABLE2',  
                    Union+'TABLE'+ 'SUBTABLE1'+ 'SUBTABLE2'.  
  
Declare+'MODE'+ 'TABLE': Declare+'MODE',  
                          Idlist+'TABLE'+ 'MODE'.  
  
Idlist+'TABLE'+ 'MODE': Identifier+'NAME',  
                        Include+'TABLE'+ 'NAME'+ 'MODE', Semicolon symbol;;  
                        Identifier+'NAME',  
                        Include+'TABLE'+ 'NAME'+ 'MODE', Comma symbol,  
                        Idlist+'TABLE'+ 'MODE'.
```

Рис. 2. Пример аффиксной грамматики

В примере 2 ограничения Restrictions являются предикатом над областью таблиц 'TABLE', который проверяет единственность имени в таблицах. Предикат объединение 'Union' проверяет, что таблица 'TABLE' является объединением двух подтаблиц 'SUBTABLE'. Предикат Include проверяет наличие имени 'NAME' типа 'MODE' в таблице 'TABLE'.

Наиболее известным в применении аффиксных грамматик стал язык CDL (Compiler Description Language), созданный Костером (Cornelis H. A. Koster) в университете Nijmegen.

В первоначальной редакции CDL – это аффиксная грамматика, слегка модифицированная. “CDL-программа” является синтактико-семантическим определением, которое транслируется в код, выполняющий синтаксический анализ программы. Трансляция CDL-программы организована так, что нетерминальные символы (грамматические понятия) транслируются в рекурсивные процедуры, а терминальные символы и предикаты – в макросы, которые должен писать (разрабатывать) сам пользователь (разработчик транслятора исходного языка). Телом каждой такой процедуры является последовательность вызовов процедур и макросов в соответствии со структурой объектов аффиксного правила. Синтаксический разбор идёт по типу «сверху-вниз» и «слева-направо». Поэтому в CDL леворекурсивные правила исключаются.

Примечание. Расширить класс аффиксных грамматик, используемых в CDL или других системах подобного рода можно за счёт эквивалентного преобразования, исключающего леворекурсивные правила из грамматики [18–19, 27, 30–32], при котором синтактико-семантическое значение грамматических конструкций языка остаётся прежним (каким оно планировалось до преобразования).

В CDL атрибуты нетерминальных символов транслируются в параметры процедур и макросов. Наследуемые атрибуты – во входные, а синтезируемые – в выходные параметры. В аффиксной грамматике значениями атрибутов являются строки, порождаемые КС-грамматикой.

Последняя, 3-я версия, названная CDL3, нашла практическое применение при построении современных небольших коммерческих компиляторов для различных вычислительных платформ. По данным Википедии, язык CDL3 используется не только для написания компиляторов, но и для создания графических интерфейсов.

Из примеров 1, 2 видно, что некоторые семантические свойства языков программирования можно задавать с помощью атрибутов как в грамматиках VW, так и в аффиксных грамматиках, используя

при этом естественные базовые структуры данных, такие как целые числа, массивы чисел и символьные строки.

Если сравнивать VW-грамматику с аффиксной, то концепции и решения в той и другой грамматиках похожи, но механизм реализации атрибутов разный, в аффиксной грамматике более ясный и более простой.

К недостаткам атрибутного подхода следует отнести следующее: большинство атрибутов создаётся на уровне терминальных символов, но атрибутами обладают обычно синтаксические конструкции более высокого уровня, поэтому при использовании атрибутов необходимо обрабатывать слишком много параметров.

3. Метод переходных состояний

Как уже упоминалось выше, грамматика ван Вейнгаардена является синхронной моделью. Такая модель в качестве порождающей для определения языка работает достаточно хорошо. Под порождающей моделью здесь имеется в виду то, что сначала решается, какие слова и атрибуты требуются, а затем генерируется текст программы с требуемыми словами и атрибутами. Такая модель удобна для пользователя, который хочет сгенерировать текст программы.

Аффиксная грамматика может рассматриваться как синхронная модель. Она по направлениям атрибутов определяет, как строить дерево разбора для программы. В некотором смысле эта модель более удобна. На практике аффиксная грамматика создаёт проблемы при разборе (конфликтные ситуации). Отсюда возникают ограничения либо на вид грамматических правил, либо на алгоритм разбора, либо на них обоих вместе.

Для большинства языков программирования характерно то, что конструкции языка и атрибуты должны сначала создаваться, а затем использоваться. CDL, созданный исключительно для разработчиков трансляторов, реализует именно такую модель, поэтому его нельзя применять для языков, в которых описания идентификаторов встречаются в любом месте в программе.

Иной подход был предложен Ледгардом (H.F.Ledgard), который был использован для определения статической семантики в PL/1. В этой модели атрибуты не включаются в контекстно-свободную грамматику. Они встраиваются в синтезированные по грамматике синтаксические таблицы. По мере продвижения по тексту программы состояние такой таблицы меняется на каждом шаге перехода. Модификация состояния выполняется тогда, когда завершается распознавание синтаксической конструкции, то есть продукционное правило грамматики, определяющее данную конструкцию, завершено. В этом случае соответствующее состояние в таблице переходов выполняется и включается модифицирующая и проверяющая функции, связанные с этим правилом. Такую модель M.H.Williams назвал диахронной (в отличие от предыдущих синхронных моделей).

В последующих системах синтаксически управляемой обработки данных такой метод привязки семантических процедур к правилам грамматики стал применяться активно, то есть при выполнении разбора по определённому правилу грамматики выполняются связанные с ним семантические процедуры. Они зависят друг от друга по данным, при этом последовательность их вызова определяется синтаксисом, задаваемым грамматикой, а способы передачи данных реализуются исходя из модели, принятой разработчиками-реализаторами языка.

Дальнейшее развитие теории и практики построения транслирующих систем позволило формулировать и другие способы описания трансляций из одного языка в другой, к которым применяются следующие требования:

- возможность автоматической генерации транслятора;
- возможность автоматического анализа свойств транслируемого (входного) языка;
- наглядность описания синтаксиса и семантики входного языка для пользователя транслятора.

Использование формализованной семантики языков обеспечит высокий уровень информационной безопасности в задачах автоматизации проектирования современного программного обеспечения.

4. О регуляризации контекстно-свободных грамматик

Термин «регуляризация» для грамматик отличен от общепринятого понятия регуляризации функций в математике. Под регуляризацией контекстно-свободной грамматики (КСГ) понимается опреде-

лённая схема её эквивалентных преобразований с целью определения языка с помощью регулярных множеств слов, порождаемых данной грамматикой. В этом случае достигается максимальная эффективность синтезируемого распознавателя (анализатора) языка, а исходная грамматика может быть переведена в класс регулярных грамматик. В результате регуляризации получают КСГ в регулярной форме (КСР-грамматику) [18, 21, 29–33], эквивалентную исходной грамматике. Это означает, что языки, порождаемые исходной и преобразованной грамматиками, совпадают. Правила КСР-грамматики определяют терминальные порождения нетерминалов через регулярные выражения относительно символов всех алфавитов грамматики. В этом случае соответствующий языковой процессор подобен множеству конечных автоматов, каждый из которых распознает свой фрагмент входной цепочки языка. В процессе преобразований из правил грамматики исключаются разного рода рекурсии (лево и правосторонние), с помощью подстановок удаляются вхождения нетерминалов (где это возможно) и непродуктивные правила. Количество правил КСР-грамматики резко уменьшается, а их правые части представляют собой сложные регулярные выражения. Часть таких преобразований реализована в программной системе SynGT (Syntax Graph Transformations) [27–30]. В конце статьи приводится алгоритм исключения одновременно и лево-, и праворекурсивных нетерминалов из КСР-грамматики, регулярные выражения которой содержат операцию обобщенной итерации.

Регулярным выражениям (РВ) как мощному средству обработки текстов посвящено огромное количество исследований. РВ поддерживаются многими редакторами, утилитами, системами построения трансляторов и обработки данных. Они занимают центральное место во многих программах, написанных на языках Java, Jscript, Visual Basic, VBScript, Java Script, C, C++, C#, Perl, Python Tcl, Ruby, Awk и других. Поддержка РВ в столь разнородных приложениях объясняется тем, что они обладают исключительно богатыми возможностями.

Регулярное выражение – это абстракция, которая по-разному реализуется в различных программах. Термин «регулярные выражения» заимствован из формальной алгебры. Истоком регулярных выражений явилась работа двух нейрофизиологов - Уоррена Мак-Каллох (Warren McCulloch) и Уолтера Питтса (Walter Pitts), которые занимались моделированием деятельности нервной системы. Их статья “A logical calculus of the ideas immanent in nervous activity” была впервые опубликована в биологическом бюллетене в 1943 году, а позднее перепечатана MIT Press в 1965 г. Через несколько лет математик Стивен Клини (Stephen Kleene) формально описал эти модели на языке алгебры, которую он назвал алгеброй регулярных множеств (regular sets) [4]. Для них он разработал простую математическую запись, которую назвал регулярными выражениями.

Они строятся из элементарных выражений, которые просты, но в сочетании друг с другом образуют бесконечное множество комбинаций. В разных программах регулярные выражения выполняют различные функции, поэтому наборы метасимволов и другие возможности, поддерживаемые программами, также различаются. Совокупность этих второстепенных различий в реализации обозначают обычно термином “диалект”. В конце 1990-х годов диалект регулярных выражений языка Perl прославился особой мощью и выразительностью. Появились другие примеры диалектов регулярных выражений в таких языках, как Python, пакеты РВ для Java, Microsoft.NET Framework, Tcl, различные библиотеки языка Си и т.д. Диалекты развивались, в итоге общая ситуация становилась все более разнообразной.

В работах [1–3] описан еще один вариант определения регулярных выражений, который позволяет частично решить задачу минимизации регулярных выражений и упростить их обработку за счёт введения обобщённой бинарной операции итерации.

Далее напомним определения регулярного множества и регулярного выражения и дадим их по Клини [5], обобщения которых были введены и подробно обоснованы в [1–3].

Пусть дан алфавит символов $V = \{a_1, a_2, \dots, a_n\}$. V^* – множество всех слов в алфавите V .

Определение 1. Регулярным множеством слов (регулярным языком) над алфавитом V называют следующие множества:

- 1) $\{\epsilon\}$ – (пустое слово);
- 2) \emptyset – (пустое множество слов);
- 3) $\{a_i\}$ – элементарные множества слов над алфавитом V для любого $a_i \in V$;

4) множества $P \cup Q, PQ, P^*$ где P и Q – регулярные множества в алфавите V .

В соответствии с данным определением множество в алфавите V регулярно тогда и только тогда, когда оно либо \emptyset , либо $\{\xi\}$, либо $\{a\}$ для некоторого $a \in V$, либо его можно получить из этих множеств, применяя конечное число раз операции объединения, конкатенации и унарной итерации. Ничто другое не является регулярным множеством в алфавите V .

В работах [1–3] вводится бинарная операция обобщенной итерации ($\#$) над множествами слов в алфавите V . Иногда её называют итерацией с разделителем [2].

Если первый или второй операнд обобщенной итерации представляет собой множество $e = \{\varepsilon\}$ и, если позволяет однозначность представления, то он может быть опущен. Пусть A – регулярное множество слов, тогда справедливы следующие равенства: $A\# = A\#e = AA^* = A^+, \#A = e\#A = A^*$.

Для представления регулярного множества Клини в [4] ввел понятие регулярного выражения в алфавите V . Дадим определение регулярного выражения для случая введенных операций над множествами слов.

Определение 2. Регулярным выражением множества A называется слово $r(A)$ над расширенным алфавитом W , где

$$W = V \cup \{\#, *, ^+, |, (,), \varepsilon, \emptyset\},$$

причем:

1. если $A = \emptyset$, то $r(A) = \emptyset$;
2. если $A = e$, то $r(A) = \varepsilon$;
3. если $A = \{a\}$, где $a \in V$, то $r(A) = a$;
4. если $r(P) = p$ и $r(Q) = q$ – регулярные выражения для множеств P и Q то:
 - a $r(A) = (p | q)$, для множества $P \cup Q$;
 - b $r(A) = (pq)$, для множества PQ ;
 - c $r(A) = (p^*)$, для множества P^* ;
 - d $r(A) = (p^+)$, для множества P^+ ;
 - e $r(A) = (p\#q)$, для множества $P\#Q$

Ничто другое не является регулярным выражением.

Эти определения аналогичны определениям из [6] и являются общепринятыми. Операция усеченной итерации (p^+) обозначает $(p(p^*))$, а операция итерации с разделителем $(p\#q)$ обозначает $p((qp)^*)$. Введем операции $\{;\}$ и $\{,\}$ для объединения и конкатенации. Кроме этого, будем удалять из выражений лишние скобки. Для этого опишем приоритеты операций:

- 1) унарные операции $\{*\}$ и $\{^+\}$ обладают наивысшим приоритетом,
- 2) затем идет итерация с разделителем или обобщенная итерация $\{\#\}$,
- 3) затем конкатенация $\{,\}$,
- 4) и далее объединение $\{;\}$.

Так, регулярное выражение $p^*;q, p\#q$ с учетом приоритетов означает $((p^*);(q(p\#q)))$.

Будем говорить, что два регулярных выражения *эквивалентны*, если они обозначают одно и то же регулярное множество слов. Регулярные выражения, соответствующие одному и тому же регулярному множеству слов, могут быть получены друг из друга с помощью эквивалентных преобразований. Пусть регулярное множество слов A над алфавитом V представляется соответствующим регулярным выражением r , тогда множество A будет называться *языком*, порождаемым данным регулярным выражением r , и обозначаться $L(r)$.

В литературе для определения синтаксиса языка в виде правил грамматики с регулярными частями обычно используется нотация, называемая расширенной формой Бэкуса-Наура (EBNF), где используются операции Клини – обычная $\{*\}$ и усеченная $\{^+\}$ (см. определение языка Паскаль, Ада, С и другие). В этой статье мы используем определение КСР-грамматики, в которой в регуляр-

ных выражениях присутствует как обобщенная операция итерации $\{\#\}$ (по Г.С. Цейтину), так и операции Клини.

Следуя общепринятой терминологии [5,6], грамматики, содержащие правила вида (A, r) , называются КС-грамматиками в регулярной форме. Каждое правило такой грамматики может интерпретироваться как множество правил вида $\{A \rightarrow x \mid x \in L(r) : L(r) \text{ – регулярный язык над алфавитом } V\}$, где V – объединение алфавитов (словарей) терминальных и нетерминальных символов, $V = N \cup T$.

Опыт применения грамматик данного класса диктует дополнить определение грамматики [5] как четверки множеств $G = (N, T, P, S)$ множеством семантик, служащих для вставки необходимых обрабатываемых кодов в компилируемый по данной грамматике текст. Таким образом дадим определение КСР-грамматики, как ее понимаем в системе SynGT.

Определение 3. КСР-грамматикой называется пятерка множеств $G_R = (N, T, \Sigma, P, S)$, где N – множество нетерминалов, T – множество терминалов, Σ – множество семантик, P – множество КСР-правил, $P = \{A : R_A \mid A \in N, R_A \text{ – регулярное выражение над алфавитом } N \cup T \cup \Sigma\}$, S – начальный нетерминал грамматики.

5. Преобразования КСР-грамматики

Главной целью эквивалентных преобразований КСР-грамматики является ее максимальная регуляризация и выполнение определенных условий, гарантирующих построение детерминированного анализатора (распознавателя).

Рассмотрим преобразования регулярных выражений, не изменяющие регулярное множество, которые реализованы в SynGT. Большинство из них описано в монографии Ахо и Ульман [5] и называются регулярными тождествами. Существуют и нетривиальные тождества, например, $(A\#(B\#C)) \equiv A, (\epsilon; B\#(A; C), A)$.

Каждое правило в КСР грамматике записывается в следующей форме:

<нетерминал>:<регулярное выражение>.

Без потери общности будем считать, что для каждого нетерминала существует только одно правило в КСР-грамматике, которое назовем <нетерминал>-правило. Кроме терминалов и нетерминалов в грамматиках, обрабатываемых системой SynGT, используются семантики и другие вспомогательные символы, но в любом случае файл с правилами обрабатывается редактором в SynGT, который проверяет корректность записи регулярных выражений и выполняет простейшие эквивалентные преобразования (удаляет пустые выводимости и бесконечные циклы).

В общем случае правые части КСР-правил задают бесконечное множество цепочек слов. С практической точки зрения их удобно представлять в виде конечных ориентированных графов с помеченными вершинами и дугами. Такие графы, представляющие КСР-грамматику, легко преобразуются в детерминированную конечно-автоматную схему, в которой каждая вершина соответствует состоянию автомата, а ее метка специфицирует порождаемый символ. В SynGT набор графов, представляющих правила КСР-грамматики, снабженный дополнительной информацией на дугах, называется *синтаксической граф-схемой* КСР-грамматики [1–3]. Синтез граф-схемы рекурсивен и подробно описан в брошюре [2]. В системе SynGT используется графический интерфейс, описание которого дано в [23–25]. Набор базовых функций, реализованных в SynGT, служит основной цели этой системы – автоматизированное преобразование КС-грамматики с целью ее максимальной регуляризации, то есть превращение синтаксической граф-схемы грамматики в минимальный набор конечных автоматов для построения анализатора языка. Известно, что если контекстно-свободная грамматика порождает регулярный язык, то синтаксическая граф-схема вырождается в один граф, представляющий собой эквивалентное регулярное выражение.

Ограниченные рамки данной статьи не позволяют описать всё функциональное наполнение системы эквивалентных преобразований SynGT. Далее рассмотрим преобразование, доставляющее много хлопот программистам, удаляющим рекурсивные нетерминалы из грамматик.

Дадим формальное определение крайней рекурсии в КСР-грамматике так, как оно дается в [5].

Определение 4. Нетерминал A называется *самовложенным*, если в грамматике существует вывод вида $A \Rightarrow^* \alpha A \beta$, где α, β – непустые строки в алфавите нетерминалов, терминалов и семантик.

Иными словами, нетерминал A является самовложенным, если он выводим сам из себя с непустым левым и правым контекстом. В работе такой нетерминал ещё называют «самовставленным» (self embedded [5, 6]).

Определение 5. Нетерминал A называется *леворекурсивным*, если существует вывод $A \Rightarrow^* A \beta$, где β – непустая строка в алфавите нетерминалов, терминалов и семантик.

Определение 6. Нетерминал A называется *праворекурсивным*, если существует вывод $A \Rightarrow^* \alpha A$, где α – непустая строка в алфавите нетерминалов, терминалов и семантик.

Легко показать, что если A не является самовложенным нетерминалом, то он не может быть одновременно лево- и праворекурсивным. Иначе мы могли бы построить вывод $A \Rightarrow^* \alpha A \beta$. Из теории формальных грамматик [5, 6] известно, что язык регулярен, если любая грамматика, которая его порождает, не содержит самовложенных нетерминалов. Любой регулярный язык распознается конечным автоматом. В этом случае существует регулярное выражение над алфавитом терминалов, представляющее тот же самый язык. Оно может быть получено с помощью эквивалентных преобразований над КСР-правилами. Если при преобразовании достигается цель минимизации этого регулярного выражения (относительно числа вхождений символов алфавита грамматики), то и распознающий автомат по данному выражению будет минимальным.

Подобные преобразования выполняются в системе эквивалентных преобразований КСР-грамматики SynGT. Рассмотрим алгоритм исключения (лево/право) рекурсивных нетерминалов в КСР-грамматике G без самовложений для случая, когда регулярное выражение в правиле для рекурсивного нетерминала содержит обобщенную итерацию. Заметим, что в работах [1,3,5] этот случай не рассматривался.

Алгоритм исключения лево- и праворекурсивных нетерминалов из КСР-грамматики с обобщенной итерацией

Для простоты изложения рассмотрим A -правило в КСР-грамматике, когда нетерминал A является одновременно и лево-, и праворекурсивным, рекурсия прямая. Хорошо известно, что косвенная рекурсия методом подстановок сводится к прямой. В отличие от всех известных автору алгоритмов извлечения крайних рекурсий в системе SynGT (Syntax Graph Transformation), разработанной в СПИИРАН, реализован алгоритм прямого эквивалентного преобразования лево(право-)рекурсивного нетерминала с использованием операции бинарной итерации.

Рассмотрим A -правило для нетерминала A вида

$$A : A, r_{11}, A; A, r_{12}; r_{21}, A; r_{22}, \dots \tag{1}$$

где $r_{11}, r_{12}, r_{21}, r_{22}$ – регулярные выражения над алфавитом $N \cup T \cup \Sigma$.

Данное A -правило (1) состоит из четырёх частей (A_i -фрагментов), $i = 1, 2, 3, 4$:

$$A, r_{11}, A ; A, r_{12} ; r_{21}, A ; r_{22} ,$$

где

- A_1 -фрагмент содержит одновременно и лево, и праворекурсивное вхождение нетерминала A ;
- A_2 -фрагмент содержит левоорекурсивное вхождение нетерминала A ;
- A_3 -фрагмент содержит праворекурсивное вхождение нетерминала A ;
- A_4 -фрагмент не содержит рекурсивных вхождений нетерминала A .

Рассмотрим, какие строки могут порождаться с помощью A -правила (1).

Шаг 1. Рассмотрим A_1 -фрагмент A -правила: A, r_{11}, A . Используя данный фрагмент A -правила, мы можем вывести строки

$$A, Ar_{11}A, Ar_{11}Ar_{11}A, \dots ,$$

Из определения операции $\#$ это множество строк совпадает с множеством строк, порождаемым регулярным выражением

$$A\#r_{11} . \tag{2}$$

Шаг 2. Рассмотрим A_2 – фрагмент A – правила: A, r_{12} . Здесь мы можем вывести строки

$$A r_{12}, A r_{12} r_{12}, A r_{12} r_{12} r_{12}, \dots \quad (2.1)$$

Из определения операции $*$ следует, что множество (2.1) порождается регулярным выражением

$$A, r_{12}^* \quad (2.2)$$

Подставив вместо A в выражении (2) A_2 -фрагмент A -правила (2.2), получим регулярное выражение

$$(A, r_{12}^*) \# r_{11} \quad (3)$$

Шаг 3. Аналогично для праворекурсивного вхождения нетерминала A в A_3 -фрагменте. Здесь мы выводим множество строк $r_{21}A, r_{21}r_{21}A, r_{21}r_{21}r_{21}A, \dots$, которые порождаются регулярным выражением r_{21}^*, A . Сделав подстановку вместо вхождения нетерминала A в (3) выражения r_{21}^*, A , получим регулярное выражение

$$(r_{21}^*, A, r_{12}^*) \# r_{11} \quad (4)$$

Шаг 4. Окончательно, подставляя вместо вхождения нетерминала A в регулярное выражение (4) A_4 -фрагмент r_{22} , получим эквивалентное регулярное выражение

$$(r_{21}^*, r_{22}, r_{12}^*) \# r_{11} \quad (5)$$

Таким образом, правую часть A – правила (1) можно заменить регулярным выражением (5).

Если предварительно воспользоваться преобразованиями регулярных выражений и привести правило к такому виду, что множество $L(r_{12})$ не содержит цепочек вида αA , множество $L(r_{21})$ не содержит цепочек вида $A\alpha$, а множество $L(r_{22})$ не содержит цепочек вида $A\alpha$ и вида αA , где α произвольная цепочка, тогда прямая левая и правая рекурсии для нетерминала A отсутствуют.

Перенумеровав все нетерминалы в грамматике G и последовательно применив данное преобразование для каждого нетерминала, подставляя результаты преобразований в оставшиеся правила, получим грамматику G' , эквивалентную исходной G без крайних рекурсий.

Если $r_{11}, r_{12}, r_{21}, r_{22}$ – регулярные выражения, не содержащие вхождений нетерминала A , то нетерминал A и правило для него можно удалить из грамматики G' , а все его вхождения заменить выражением $(r_{21}^*, r_{22}, r_{12}^*) \# r_{11}$.

Заключение

КСР-грамматики, построенные в процессе подобного рода преобразований (регуляризации), гарантируют высокую эффективность соответствующего управляющего механизма распознавания (анализа) исходного языка и могут применены в различных приложениях производственной сферы на практике [34-36].

Литература

1. Федорченко Л.Н. Об алгоритме синтаксического анализа детерминированных языков // Computer Networks Leningrad Computing Center of the Academy of Sciences and Computer and Automation Institute of Hungarian Academy of Sciences: материалы совместного с Венгерской академией наук семинара. – Budapest, 1978. – С.73–78.

2. Федорченко Л.Н. Об одном алгоритме синтаксического анализа языков, порождаемых R-грамматиками // Алгоритмы и системы автоматизации исследований и проектирования: сб. науч. тр. / ЛНИВЦ АН СССР. – М.: Наука, 1980. – С. 146–155.

3. Федорченко Л.Н., Мартыненко Б.К. Эквивалентные преобразования КСР грамматик в регулярной форме в практике построения языковых процессоров. Ч. 1. Определение и распознавание КСР-языков посредством синтаксических граф-схем. – Л., 1983. – С. 55.

4. Клини С. Представление событий в нервных сетях // Автоматы. – М.:ИЛ,1956. – С. 15–67.

5. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. – М.: Мир, 1978. – Т.1. Синтаксический анализ. 612 с.; Т.2. Компиляция. 487 с.

6. Ginsburg S. The mathematical theory of context-free languages. Mc Graw-Hill Inc., 1966.
7. Revised report on the algorithmic language Algol 68 // ACTA Informatica 5. Пересмотренное сообщение о языке ALGOL 68. – 1974. – P. 1–236.
8. Koster Affix grammars // Algol 68 implementation. – Noth-Holland, 1971.
9. Knuth D.E. Semantics of context-free languages // Mathematical system theory. – 1968 (2). – P. 127–145.
10. Ledgard H.F. Production system or can we do better than BNF? // CACM. – 1974, N2. – P. 94–102.
11. Williams V.H. Static semantics features of Algol 60, and BASIC // The Computer Journal. – Vol. 21, №. 3. – P. 234–242.
12. Griffiths M. Relationship between definition and implementation of language // Advanced courses on software engineering. Lecture Notes in Economics and Math Syst. Springer-Verlag 1973.
13. Bochmann G.V. Semantic evaluation from left to right // CACM. – 1976, №2. – P. 55–62.
14. Grune Dick. Two Level grammars are more expressive than Type 0 grammars or are they? // Dept. of Mathematics and Computer Science Vrije Universiteit, De Boelelaan 1081, HV Amsterdam. dick@cs.vu.nl
15. Sintzoff M. Existence of a van Wijngaarden syntax for every recursively enumerable set // Annales de la Societe Scientifique de Bruxelles. – 1967. № 81(II). – P. 115–118.
16. van Wijngaarden A. The generative power of two-level grammars // Automata, Languages and Programming: Lecture Notes in Computer Science #14, ed. J. Loecks. Springer-Verlag. – Berlin, 1974. – P. 9–16.
17. Grune Dick, Jacobs Cerial. Parsing Techniques. A Practical Guide // Ellis Horwood, Chichester, 1990. – P. 322.
18. Федорченко Л.Н. О регуляризации контекстно-свободных грамматик // Изв. вузов. Приборостроение. – 2006. – Т. 49, №11. – С. 50–54.
19. Федорченко Л.Н., Заболотский В.П. Лингвистический инструментарий в задачах обеспечения информационной безопасности // Проблемы информационной безопасности. Компьютерные системы. – 2008. – Вып. 4. – С.60–68.
20. Федорченко Л.Н. О числе состояний распознавателя, порождаемого с помощью КС-грамматик в регулярной форме // Информационные и вычислительные проблемы в научных исследованиях: сб. науч. тр. – М.: Наука, 1983. – С. 69–73.
21. Инструментальная система для решения задач планирования и оптимизации вычислений в распределенной среде на основе графовых трансформаций / Л.Н. Федорченко и др. – М., 1998. – 70 с.
22. Федорченко Л.Н., Баранов С.Н., Кутузов М. Грамматика языка Си // ОСС – открытый компилятор языка Си. – М., 1998. – С. 5.
23. Усовершенствованный алгоритм экранного размещения синтаксической граф-схемы в системе SynGT (Syntax Graph Transformation) версии 3.1. / Л.Н. Федорченко и др. // ОСС – открытый компилятор языка Си для цифровых сигнальных процессоров. – М., 1999. – С. 45–46.
24. Средства визуализации синтаксических граф-схем – СПИИРАН / Федорченко Л.Н. и др. // Региональная информатика–98 (РИ–98): материалы VI Междунар. конф., Санкт-Петербург, 2–4 июня 1998. – СПб., 1999. – С. 55–56.
25. Федорченко Л.Н. Специализированный редактор для обработки контекстно-свободных грамматик в регулярной форме. – СПИИРАН, тр. VI Междунар. конф. «Региональная информатика–98» (RI–98), СПб., 1999. – С. 56–57.
26. Федорченко Л.Н. Извлечение крайней рекурсии из КСР грамматики в системе SynGT. // Тр. СПИИРАН. – Вып.1. – Т.1. – СПб.: СПИИРАН, 2002. – С. 350–359.
27. Федорченко Л.Н., Соловьев С.В. Синтаксические преобразования в системе SynGT и их новые приложения // Региональная информатика–2002 (РИ–2002): материалы VIII Междунар. конф. в 2 ч., Санкт-Петербург, 26 – 28 ноября 2002 г. 2002. – С. 50.
28. Федорченко Л.Н., Маньков Е.В. Современное состояние и классификация систем построения компиляторов // Региональная информатика–2004 (РИ–2004): материалы IX Междунар. конф. (Санкт-Петербург, 22–24 июня 2004 г.). 2004. – С. 66.
29. Fedorchenko L., Naumov I. Syntax Graph Transformations in The System SynGT and Their New Applications, Procs of 10-th Multi-conference on Advanced Computer Systems (ACS–AIBITS 2003), October 22–24, Poland.

30. Fedorchenko L. Syntax Graph Transformations in the System SynGT and Regularization of Grammars, Proc of Intern Multi-Conference on Advanced Computer Systems (ACS–CISIM 2004), 14–16 June, Elk, Poland [электронный ресурс] <http://acs.wi.ps.pl/info.php>

31. Федорченко Л.Н. Метод регуляризации грамматик в системах трансляции языков // Инновации в науке и образовании–2008: материалы VI Юбил. междунар. науч. конф., посвящ. 50-летию пребывания КГТУ на Калининградской земле: в 3 ч. – Калининград, 2008. – Ч. 2. – С. 305.

32. Федорченко Л.Н. Регуляризация контекстно-свободных грамматик на основе эквивалентных преобразований синтаксических граф-схем // Тр. СПИИРАН. – 2010. – Вып. 4 (15). – С. 213–230.

33. Федорченко Л.Н. Регуляризация контекстно-свободных грамматик / LAP LAMBERT Academic Publishing GmbH & Co. KG Dudweiler Landstr. 99, 66123 Saarbrücken, Germany. – 2011. – P. 180.

34. Лукьянова Л.М., Федорченко Л.Н. Средства формализации целей и проблем сложных систем производственной сферы // Вестн. БГУ. – 2012. – № 9. – С. 42–48.

35. Лукьянова Л.М., Федорченко Л.Н. Формализация систем целей в производственной сфере // Региональная информатика-2012: материалы XIII Междунар. конф. (Санкт-Петербург, 24–26 октября 2012 г.). – 2012. – С.172–173.

36. Atiskov A., Fedorchenko L., Moldovyan N., Novikov F., Vorobiev V. Ontology-Based Analysis of Cryptography Standards and Capabilities for Their Harmonization. // Theory and Practice of Cryptography Solutions for Secure Information Systems // IGI Global, 701 E. Chocolate Ave. Hershey, PA 17033, USA. 2013.

Федорченко Людмила Николаевна, кандидат технических наук, старший научный сотрудник лаборатории прикладной информатики, Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН). Тел. +7 (812) 328-1919. E-mail: LNF@ias.spb.su

Fedorchenko Lyudmila Nikolaevna, senior researcher, candidate of technical sciences, St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Ph. +7 (812) 328-1919. E-mail: LNF@ias.spb.su

УДК 004.4'244

© *Е.С. Фереферов, А.Е. Хмельнов*

АВТОМАТИЗАЦИЯ СОЗДАНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА НА ОСНОВЕ МОДЕЛИ ПРИЛОЖЕНИЯ БАЗ ДАННЫХ¹

Статья посвящена проблемам автоматизации создания пользовательского интерфейса для работы с базами данных. Для решения данной задачи авторы предлагают оригинальную модель приложения БД, декларативный язык для ее представления и инструментальную систему, позволяющую создавать такие модели приложений БД.

Ключевые слова: базы данных, модели информационных систем, декларативные языки, инструментальные системы, спецификации.

© *E.S. Fereferov, A.E. Khmelnov*

AUTOMATIZATION OF USER INTERFACE CREATION BASED ON MODELS OF DATABASE APPLICATION

The article is devoted to the problems of automated creation of user interface for work with databases. To solve this problem the authors propose an original model of database applications, models presentation declarative language and instrumental system to create such models of database application.

Keywords: databases, models of information systems, declarative languages, instrumental systems, specifications.

Введение

Современные автоматизированные информационные системы (АИС) представляют собой сложные программные комплексы, функции которых состоят в поддержке сбора, хранения, выполнении специфических для определенной предметной области преобразований информации, вычислений, поддержки анализа и предоставления информации пользователям. Современные АИС работают с большими объемами данных со сложной структурой, поэтому часто основой информационных систем являются базы данных (БД). Создание пользовательского интерфейса для работы с базами данных является одной из основных и важных задач при разработке АИС. Кроме того, при наличии в БД пространственной информации также возникает задача разработки интерфейса для ее визуализации и анализа, для чего применяют геоинформационные технологии.

Современные технологии разработки информационных систем, основанные на императивном программировании и использовании развитых библиотек визуальных компонентов (например, Delphi и Visual Component Library [1]), предоставляют достаточно общие инструментальные средства и подходят для решения большинства задач. В их арсенале имеется большое количество компонентов, реализующих как части визуального интерфейса, так и бизнес-логики для обеспечения доступа и модификации БД. Как правило, приложения создаются для работы с базами данных, имеющими определенную структуру, при этом доступ к индивидуальной таблице реализуется при помощи одних и тех же (или сходных по функциям) подпрограмм. При изменении структуры БД, например, добавлении новых таблиц или полей, необходимо соответственно вносить изменения в программный код, заново реализуя функции для доступа к новым элементам БД. Низкий уровень автоматизации реализации однотипных функций для работы с таблицами БД приводит к большим временным и трудовым затратам при создании (модернизации) пользовательского интерфейса для работы с БД.

В настоящее время ведутся активные работы по созданию новых подходов к разработке как пользовательских интерфейсов (например, MBUIDE [2]), так и к разработке АИС в целом (например, MDA [3], порождающее программирование [4]), позволяющие ускорить, сделать дешевле, качествен-

¹ Работа выполнена при частичной финансовой поддержке РФФИ проекты 11-07-00426-а, 11-07-92204-Монг-а, междисциплинарного проекта № 131 СО РАН, ОНИТ РАН (проект № 4.1)

но повысить процесс создания приложений. Основной тенденцией современных подходов является попытка структурировать метаинформацию (данные о структуре) об АИС в виде модели системы различными средствами (например, надстройки над моделями классов UML или построение онтологий предметной области [5]). Формализация знаний о структуре АИС в модели позволяет выделять схожие структуры данных и присущие им бизнес-процессы в отдельные компоненты, что позволяет генерировать соответствующие им сценарии создания структур в СУБД, алгоритмы обработки бизнес-процессов, экранные формы единой формы и распространить их на все подобные компоненты. Как показывает практика, сгенерированный код практически всегда требуется дорабатывать программисту, при этом полученные изменения не отражаются в исходных абстрактных моделях системы.

Для решения задач обработки, представления и анализа пространственных данных информационные системы должны обладать ГИС-функциональностью. Все современные геоинформационные системы предлагают разработчикам API, позволяющие реализовать как пользовательский интерфейс для работы с пространственными данными, так и функции для их обработки. Несмотря на развитость этих API, реализация ГИС-функциональности является сложной и трудоёмкой задачей, требующей от программистов знаний в области геоинформационных технологий. Часто программисту приходится повторять реализацию существующих функций целевой ГИС в своем проекте. Кроме того, при необходимости развития существующего приложения за счет ГИС-функциональности необходимо наличие исходных кодов этих АИС, а их отсутствие вынуждает разрабатывать систему заново.

Технология создания приложений баз данных

Для автоматизации создания информационных систем авторами разработана технология создания пользовательского интерфейса на основе моделей приложений БД. Данная технология предназначена для автоматизации процессов, выполняемых на этапах проектирования и реализации программного обеспечения для работы с БД. Отличием данной технологии от широко известных является отсутствие процессов компиляции и генерации программного кода.

При создании модели приложения в качестве входных данных используются метаданные о структуре уже созданной БД (схема БД), хранящиеся в СУБД. Предполагается, что для создания схемы БД используются зарекомендовавшее себя ПО от ведущих компаний-разработчиков, позволяющее строить ER-модели данных и генерировать схемы БД (например, интегрированные утилиты СУБД MS SQL Server, Oracle или CASE-средства, такие, как Sybase Power Designer, IBM Rational Rose). Схема данных является уже структурированными знаниями о сущностях и связях между ними, которые необходимо расширить знаниями о бизнес-процессах и способах представления информации пользователю для создания качественного пользовательского интерфейса.

В качестве способа представления и хранения моделей применяются декларативные спецификации, в которых все структуры модели приложения БД описаны на специально разработанном языке ЯПБД [6]. Декларативные спецификации удобны своей компактностью и значительно меньше подвержены ошибкам по сравнению с программами на императивных языках, при этом обладают предметной ориентированностью и выразительностью, а также широкой возможностью интерпретации различными трансформационными и другими процедурами.

Для создания и модификации моделей приложений БД разработана инструментальная система ГеоАРМ [7]. Кроме того, при помощи спецификаций инструментальная система автоматически формирует пользовательский интерфейс для работы с предметной базой данных. Интеграция функций создания спецификаций и их интерпретации в одной системе дает возможность разработчику сразу оценить адекватность создаваемой модели приложения и повысить скорость разработки в целом. Встроенный в ГеоАРМ модуль для работы с пространственными данными обеспечивает пользователям работу с пространственными данными.

Модель приложения баз данных

Предлагаемая модель приложения БД имеет следующий вид:

$$M = \langle \text{Schema}, \text{Display}, \text{Rules}, \text{Plugins} \rangle,$$

где **Schema** – структура, описывающая схему БД, расширенную информацией о связях, способах организации данных для пользователя и взаимодействия с пространственными данными; **Display** – структура, описывающая механизм представления данных пользователю; **Rules** – структура для опи-

сания бизнес-правил приложения; **Plugins** – структура для описания взаимодействия с внешними подключаемыми модулями, решающими задачи, не заложенными в базовый функционал. Рассмотрим некоторые структуры модели подробно.

Структура $Schema = \langle Tbls, Refs \rangle$ описывает множества таблиц $Tbls = \{t_1, \dots, t_n\}, n \in \mathbb{N}$ и связей $Refs = \{r_1, \dots, r_m\}, m \in \mathbb{N}$ между ними. $C = \{c_1, \dots, c_l\}$ – множество схем таблиц БД, $l \in \mathbb{N}, l \leq n$. $t_i(c_j)$ – таблица со схемой $c_j, i = 1, \dots, n, j = 1, \dots, m$, где $c_j = \{a_1, \dots, a_p\}, p \in \mathbb{N}$. a_h – атрибут таблицы, $h = 1, \dots, p$.

Каждый атрибут относится к определенному типу данных: $\forall a_h \exists Type(a_h)$ – тип атрибута. Определены восемь типов $Type = \{I, F, S, D, B, X, G, SD\}$, где I – множество целых чисел, F – множество действительных чисел, S – множество строковых значений, D – множество значений дат и времени, $B = \{true, false\}$ – булевы значения, X – множество бинарных данных, G – множество растровых данных, SD – множество пространственных данных. Тип атрибута таблицы влияет на способ представления и обработки данных из этого поля.

Каждая таблица состоит из конечного множества записей $t_i(c_j) = \{k_1, \dots, k_q\}, q \in \mathbb{N} : \forall k_g \exists c_j \rightarrow Type, k_g(a_h) \in Type_f, f = 1, \dots, \delta, g = 1, \dots, q$, где $k_g(c_j)$ – запись со схемой $c_j, k_g(a_h)$ – значение записи k_g на атрибуте a_h , т.е. значение поля.

Для каждой таблицы могут быть определены ключи. Первичный ключ $PK = \{a_1, \dots, a_{pp}\} \subseteq c_j, pp = 1, \dots, p$, если

1. $\forall k_g, k_{gg} \in t_i : k_g(PK) \neq k_{gg}(PK), g \neq gg, g = 1, \dots, q, gg = 1, \dots, q;$
2. $\exists PK' \subseteq PK : k_g(PK') \neq k_{gg}(PK), g \neq gg, g = 1, \dots, q, gg = 1, \dots, q$

Часто в качестве РК используют так называемые искусственные ключи – автоинкрементные поля, например, $k_{g+1}(PK) = k_g(PK) + 1$. При этом в таблице могут присутствовать естественные ключи (например, «Район», «Улица», «Дом»). Комбинации полей естественных ключей $NF = \{a_1, \dots, a_{ppp}\} \subseteq c_j, ppp = 1, \dots, p$ будем называть именуемыми полями. Такие именуемые поля необходимы для организации удобного пользовательского интерфейса, например, по ним может быть организована автофильтрация или сортировка. $FK = \{a_1, \dots, a_{pp}\} \subseteq c_j$ – внешний ключ таблицы t_i , если

1. $\exists t_{ii}(c_{jj}), PK = \{a_1, \dots, a_{pp}\} \subseteq c_{jj};$
2. $\forall k_g \in t_i(c_j) \exists k_{gg} \in t_{ii}(c_{jj}) : k_g(FK) = k_{gg}(PK), gg = 1, \dots, q$

Структура $Refs = \{r_1, \dots, r_m\}$ описывает множество связей между таблицами. Данная структура необходима для автоматического создания пользовательского интерфейса, поскольку метаинформации СУБД о ключах недостаточно для понимания, как должна быть организована работа с таблицами с точки зрения пользователя. Определены связи трех видов: $r_w \in \{LR, PR, DR\}, w = 1, \dots, m$, где LR – простая связь, PR – связь из первичного ключа, DR – связь типа детали.

$r_w(t_i, t_{ii}) \in LR$, если $k_g(FK) = k_{gg}(PK) \vee \emptyset, FK \in c_j, PK \in c_{jj}, k_g \in t_i(c_i), k_{gg} \in t_{ii}(c_{jj});$
 $r_w(t_i, t_{ii}) \in PR$, если $k_g(PK) = k_{gg}(PK), PK \in c_j, PK \in c_{jj}, k_g \in t_i(c_i), k_{gg} \in t_{ii}(c_{jj});$
 $r_w(t_i, t_{ii}) \in DR$, если $k_g(PK) = k_1(FK) \wedge \dots \wedge k_{gg}(FK), FK \in c_j, PK \in c_{jj}, k_g \in t_i(c_i), k_{gg} \in t_{ii}(c_{jj}), \{k_1, \dots, k_{gg}\} \in t_{ii};$

При наличии связей LR или PR можно построить представления. Представление $v(c') \in V$ – это множество записей из таблицы $t_i(c_j) \in Tbls$, где $c' \subseteq c_j$, либо $v(c') = (t_i(c_j), V', LR', PR')$, где $V' \subseteq V, \forall v \in V' \exists r_w(t_i, v) \in LR' \vee \exists r_w(t_i, v) \in PR' \wedge c' \subseteq c_j \cup c'_1 \cup \dots \cup c'_l$.

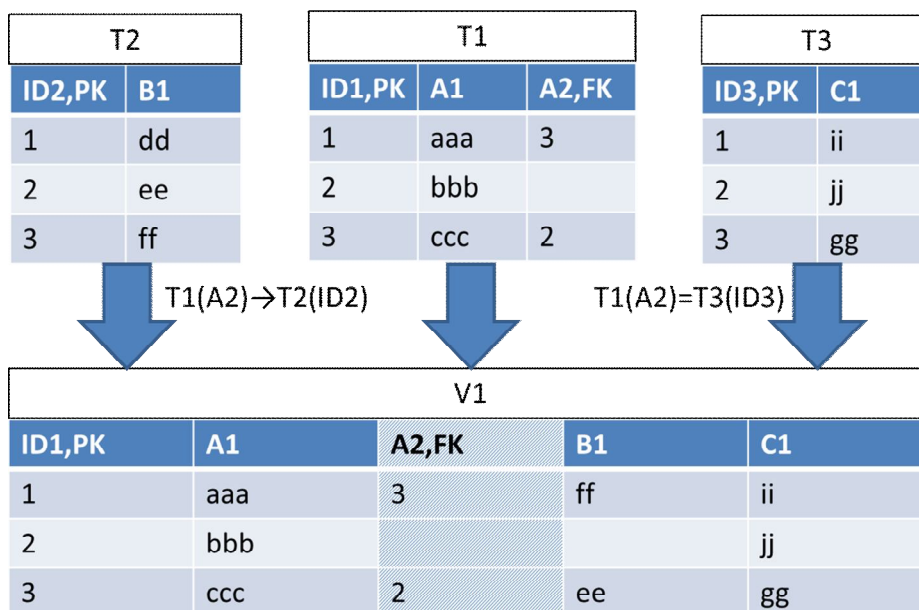


Рис. 1. Пример представления

В отличие от View популярных СУБД, которые являются сохраненными SQL-запросами, представления в предлагаемой модели являются наборами данных (рис. 1), которые построены на одной базовой таблице и содержат информацию о декодировании связей для получения данных из связанных таблиц. Такие представления позволяют автоматически организовать работу со связанными таблицами как с одним набором данных.

Структура $Display = \langle C, U, E, C \rightarrow E \rangle$ отвечает за то, какими элементами пользовательского интерфейса будут представлены данные из полей таблиц и представлений. $U = (grd, frm)$ – режимы работы (табличный, форма). $E = \{Btn, Edt, Cbx, Dtp, Chkt, Img, fn(Cbx), Grd, GB, TS\}$ – множество элементов управления пользовательского интерфейса. Отображение полей в элементы пользовательского интерфейса зависит от типа поля:

$$k(a_h) \rightarrow Edt, \text{ если } Type(a_h) = I \vee F \vee S, k \in t_i, h = 1, \dots, p, a_h \in C;$$

$$k(a_h) \rightarrow Dtp, \text{ если } Type(a_h) = D, k \in t_i, h = 1, \dots, p, a_h \in C;$$

$$k(a_h) \rightarrow Chkt, \text{ если } Type(a_h) = B, k \in t_i, h = 1, \dots, p, a_h \in C;$$

$$k(a_h) \rightarrow Cbx, \text{ если } Type(a_h) = I \vee F \vee S, k(a_h) \in v(c') = (t_i(c_j), V', LR', PR'),$$

$$h = 1, \dots, p, a_h \in C;$$

$$k(a_h) \rightarrow Img, \text{ если } Type(a_h) = G, k \in t_i, h = 1, \dots, p, a_h \in C;$$

$$k(a_h) \rightarrow fn(Cbx), \text{ если } Type(a_h) = I \vee F \vee S, k(a_h) \in v(c') = (t_i(c_j), V', LR', PR'),$$

$$a_h \in NF \subseteq c_j, h = 1, \dots, p, \text{ где } fn(Cbx) - Cbx \text{ с функциональностью автофильтрации.}$$

В случаях, когда поля получены по ссылкам (в представлениях) или являются FK, рядом с ними присутствует кнопка (Btn) для перехода к записям по ссылке. Несколько полей могут быть выделены в отдельные блоки. Поддерживается два вида блоков – группа (GB) и закладка (TS).

Структура $Rules = \{RO, CHK, MO\}$ отвечает за режимы и правила работы с данными. Например, если $RO(t_i)$, то работа с таблицей t_i будет в режиме «только для чтения», поля таблицы будут неактивны для редактирования. Режим $CHK(k(a_h), Rg), Rg = \{0,1\}$ отвечает за проверку вводимых данных, при этом существует два режима (Rg) проверки: строгое соответствие (1) и предупреждение (0). Меха-

низм $MO(t_i, t_{ii})$ позволяет автоматически создавать пространственные объекты для записей из t_i на цифровой топооснове по метрике из таблицы деталей t_{ii} .

Структура $Plugins = \langle Tbls, D, Dll \rangle$ отвечает за вызов внешних приложений для обработки данных. Данная структура обеспечивает для $t_i \in Tbls$ передачу данных во внешнее приложение $dll_z \in Dll$, вызов которого осуществляется из пользовательского интерфейса $d_x \in Display$.

Спецификации приложений баз данных

Для представления и хранения моделей приложений баз данных применяются декларативные спецификации, которые строятся на специально разработанном языке ЯПБД [6]. Конструкции языка позволяют в декларативном виде описывать все элементы модели приложения БД, а также некоторые общие настройки системы (например, способ подключения к БД, взаимодействие с другими БД и цифровыми картами). Язык представления баз данных принадлежит к классу LL(1) [8] грамматик. Предложения ЯПБД имеют следующую структуру:

<Стартовое слово> <Список обязательных выражений> [список необязательных выражений]

Начало предложений определяется по принадлежности первого слова к множеству стартовых слов. Каждое предложение состоит из обязательных и необязательных выражений. Все выражения разделены пробелами. Каждое выражение содержит зарезервированное (служебное) слово, описывающее имя какого-либо атрибута системы, и может включать значение этого атрибута через знак «=». Например,

READONLY или **SCHEMA**=<Значение атрибута>.

В некоторых случаях за служебным словом может идти список выражений в скобках «(...)» через запятую «,»:

FIELDS (<список выражений>, ..., <список выражений>)

Определены следующие предложения:

- способы подключения к БД,
- общие настройки работы приложения,
- описание таблицы (для каждой таблицы, с которой необходимо работать),
- описание представления (для каждого представления, с которым необходимо работать),
- описание дерева таблиц и представлений,
- описание механизма интеграции с картой,
- описание расширений (Plugins).

Разработанный язык обеспечивает возможность настройки большого перечня параметров приложения. Предложения, описывающие таблицы и представления (рис. 2), сочетают в себе информацию о всех уровнях модели приложения $M = \langle \text{Schema, View, Rules, Plugins} \rangle$. С одной стороны, конструкции языка позволяют достаточно детально описать структуру БД (уровень Schema), с другой, позволяют представить эту информацию с точки зрения работы приложения, то, как информация из таблиц БД будет представлена пользователю (уровень Display). Кроме того, конструкции ЯПБД позволяют поддерживать модульное программирование, т.е. при разработке спецификации приложения БД можно ссылаться на готовые спецификации других приложений. Такая возможность позволяет создавать

АИС интегрирующие данные из нескольких предметных АИС, а также ускорить процесс разработки в целом.

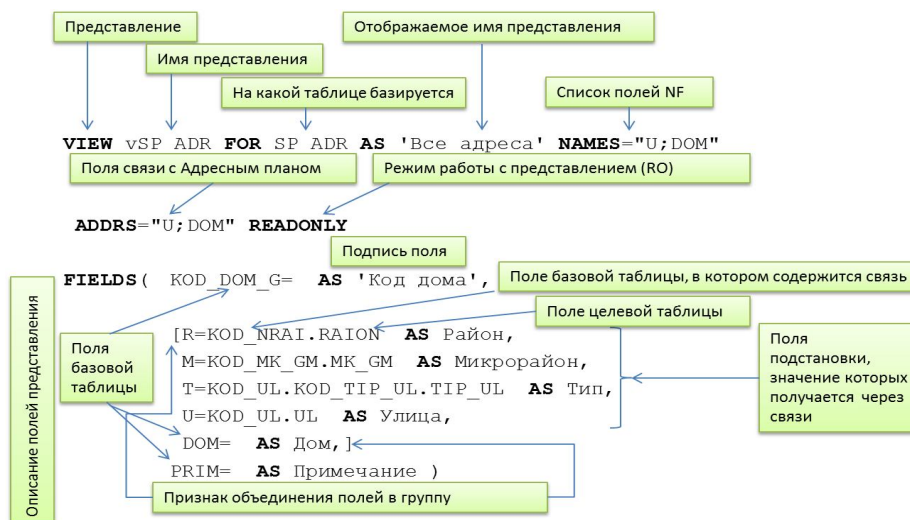


Рис. 2. Пример представления на ЯПБД

Инструментальная система создания приложений баз данных. Для создания и модернизации спецификаций приложений БД авторами была разработана инструментальная система ГеоАРМ [7]. Данная инструментальная система позволяет интерактивно управлять структурами моделей приложений баз данных, настраивать взаимодействие с БД, внешними подсистемами, цифровыми картами и сохранять спецификации в файлы на ЯПБД. Кроме того, при помощи спецификаций инструментальная система автоматически формирует пользовательский интерфейс, становясь предметной АИС.

Инструментальная система состоит из следующих подсистем: «Ядро», «Управление спецификациями», «Редактор БД», «Построитель пользовательских запросов», «Карта», модуль «Программный интерфейс» (рис. 3).

Ядро системы отвечает за взаимодействие всех подсистем АИС с СУБД, обеспечивая преобразование команд в терминах спецификации в команды интерфейсов для работы с СУБД.

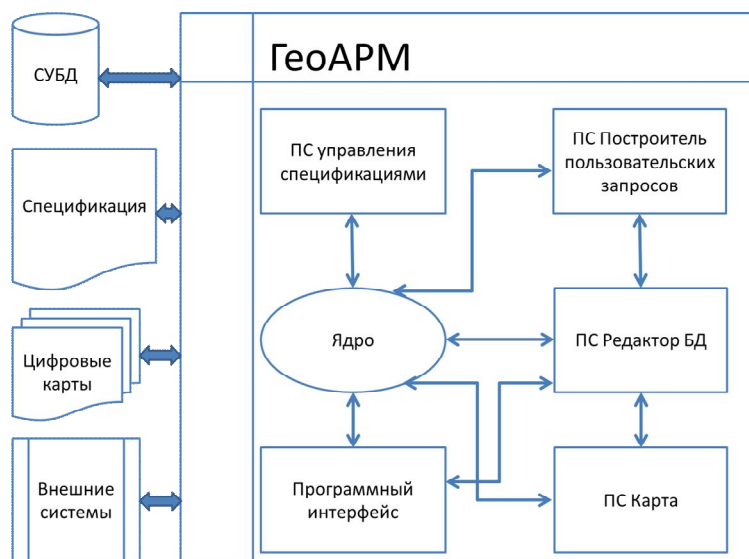


Рис. 3. Архитектура инструментальной системы создания приложений БД

Подсистема управления спецификациями приложений БД является визуальным пользовательским интерфейсом, позволяющим решать задачи создания и модернизации спецификаций приложений БД: настройка соединения с БД, выбор и загрузка метаданных из СУБД, управление описанием таблиц, управление описанием представлений, управление описанием надстроек, настройка параметров взаимодействия с цифровой картой.

Подсистема «Редактор БД» реализует структуру Display модели приложения БД и представляет собой динамический интерфейс для работы с таблицами БД, создаваемый автоматически при помощи спецификации. Редактор БД (рис. 4) содержит главное меню, дерево сущностей, область задач и область данных. Главное меню содержит все функции АИС для быстрого доступа. В дереве сущностей (слева на рис. 4) отображаются описанные в спецификации имена таблиц и представлений АИС, объединённые в смысловые группы, исходя из их назначения или функций.

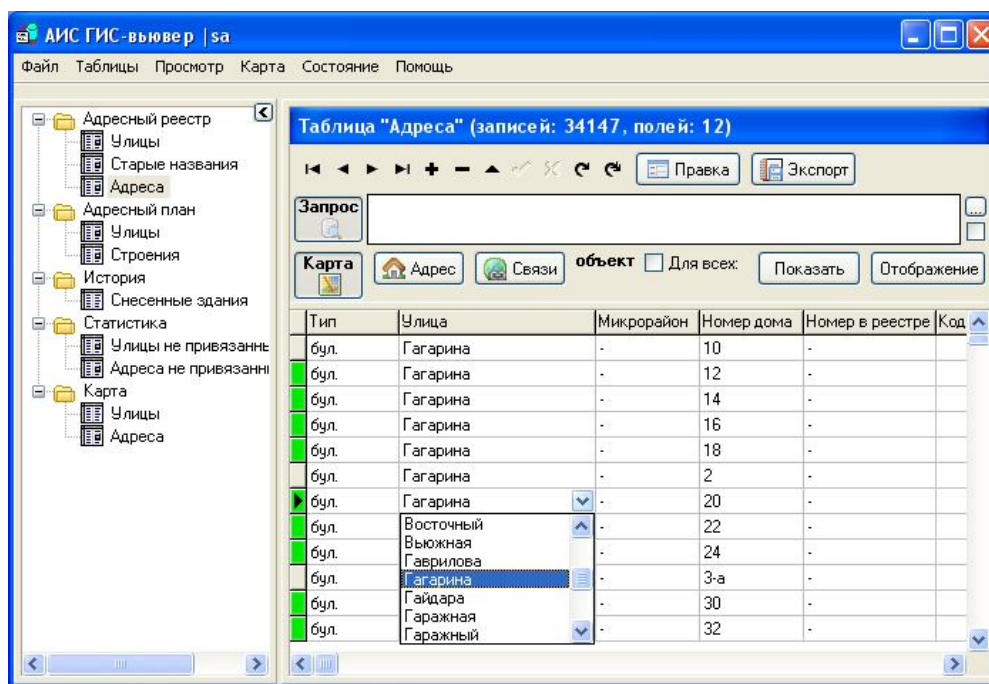


Рис. 4. ПС Редактор БД в режиме «таблица»

В области данных (справа на рис. 4) отображаются элементы управления для работы с таблицами и представлениями. Работа с данными осуществляется в двух режимах: в режиме таблицы (режим $U=grd$) для просмотра набора записей или в режиме формы (режим $U=frm$) для работы с конкретной записью. Состав элементов управления и стиль работы с полями таблиц или представлений определяется автоматически на основе информации из спецификации. Например, в режиме таблицы для полей представлений, полученных по ссылкам ($k(a_h) \in v(c') = (t_i(c_j), V', LR', PR')$), автоматически формируется список значений, причём, если осуществляется работа с несколькими полями таблицы-источника, то может быть определена последовательность их выбора. В режиме «форма» (рис. 5) поля, полученные по одной ссылке, автоматически объединяются в группу типа GB, если не определены иные группировки полей. Для полей $Type(a_h) = D$ поддерживается ввод данных непосредственно в Edt и через «Календарь». Рядом с полями $Type(a_h) = G$ автоматически создаются кнопки для вызова стандартных диалогов открытия и сохранения графических файлов. При наличии связей типа $r_w(t_i, t_{ii}) \in DR$ в нижней части формы отображаются записи из таблицы-деталей. За оптимальную, с точки зрения удобства работы пользователя, расстановку в области задач элементов управления отвечает менеджер компоновки [9].

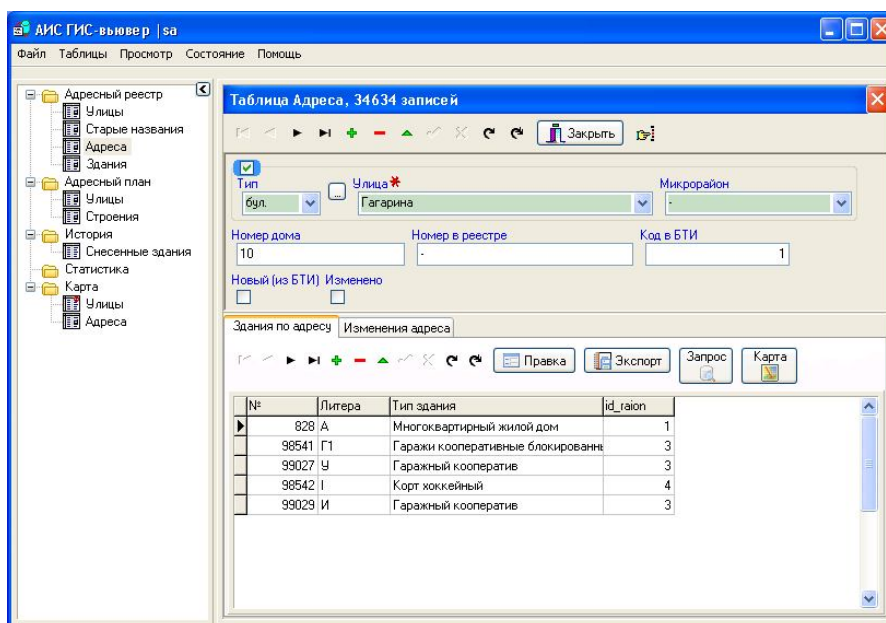


Рис. 5. PC Редактор БД в режиме «форма»

Построитель пользовательских запросов является универсальной формой, которая настраивается на работу с конкретной таблицей или представлением при помощи спецификаций. Построитель может работать как в упрощенном, так и в расширенном режиме. В упрощенном режиме пользователь видит таблицу с именами полей, в которой он может задать ограничения на значения некоторых из них, при этом конъюнкция условий образует условие запроса. При необходимости задать более сложное условие, можно перейти в расширенный режим, в котором можно редактировать выражения в виде горизонтально ориентированного дерева (рис. 6).

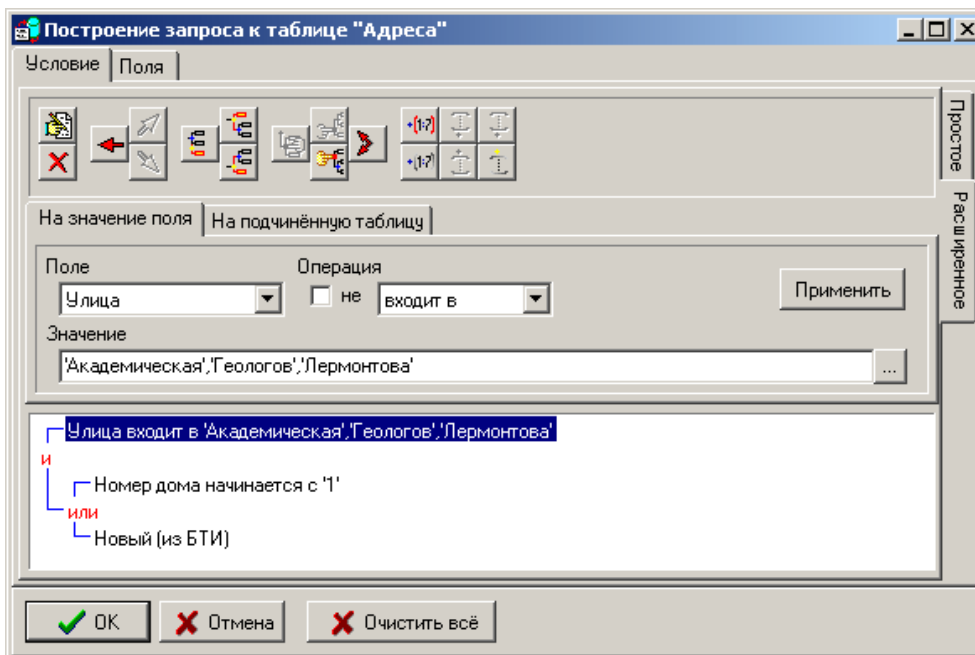


Рис. 6. Построитель пользовательских запросов

В расширенном режиме поддерживается формирование условий на записи подчиненных таблиц. При формировании таких условий построитель запросов вызывается рекурсивно. В любом режиме способ редактирования условий на значения поля определяется информацией о виде поля таблицы или представления из спецификации АИС.

Программный интерфейс предназначен для взаимодействия с надстройками, позволяющими решать специфические, не заложенные в архитектуру «ГеоАРМ» задачи. Надстройки реализуются в виде динамически подключаемых библиотек и позволяют расширять возможности системы.

Подсистема «Карта» обеспечивает возможность работы с цифровой топоосновой. Данный модуль реализован при помощи пакета GIS ToolKit из ГИС Панорама [10]. В системе реализован механизм «Адресный план» – поиск на карте объектов по их почтовым адресам.

Картографический модуль позволяет просматривать картографическую информацию, находить на карте объекты, информация о которых содержится в БД и, наоборот, находить в БД информацию, связанную с объектом карты. Также в модуле реализованы стандартные механизмы просмотра карты, такие, как загрузка карты, масштабирование, перетаскивание карты, создание, удаление объекта, получение информации об объекте.

Привязка БД к карте может быть осуществлена двумя способами: через геокодирование, т.е. по полям, содержащим адреса домов, или привязкой к произвольным объектам карты через таблицу связей.

Заключение

Предложенная модель приложения баз данных позволила разделить информацию о структуре БД и пользовательском интерфейсе на отдельные структуры, что обеспечило возможность создания универсальных алгоритмов обработки данных из таблиц БД и автоматического создания пользовательского интерфейса. Представление моделей приложений БД в виде декларативных спецификаций удобно своей компактностью по сравнению с программами на императивных языках, предметной ориентированностью и выразительностью, а также широкой возможностью интерпретации и обработки различными трансформационными процедурами. Разработанная инструментальная система ГеоАРМ позволяет визуально создавать (модернизировать) спецификации приложений БД. Кроме того, ГеоАРМ может интерпретировать спецификации, становясь предметной АИС, обеспечивающей работу с БД и пространственными данными. В целом разработанная технология позволяет быстро создавать информационные системы для работы с БД и ГИС.

Литература

1. Архангельский А.Я. Приемы программирования в Delphi на основе VCL. – М.: Бином-Пресс, 2009. – 944 с.
2. Griffiths T. et al., A Model-Based User Interface Development Environment for Object Databases / Interacting with Computers. – Vol. 14, No. 1, Dec. 2001. – P. 31–68.
3. Рамбо Дж. UML 2.0. Объектно-ориентированное моделирование и разработка. – СПб.: Питер, 2007. – 544 с.
4. Чарнецки К., Айзенкер У. Порождающее программирование. Методы, инструменты, применение. – СПб.: Питер, 2005. – 736 с.
5. Грибова В. В., Клещев А.С. Управление проектированием и реализацией пользовательского интерфейса на основе онтологии // Пробл. управл. – 2006. – № 2. – С. 58–62.
6. Фереферов Е.С., Хмельнов А.Е. Язык представления баз данных // Инфокоммуникационные и вычислительные технологии и системы: материалы III Междунар. конф. – Улан-Удэ: Изд-во БГУ, 2010. – С. 269-272.
7. Фереферов Е.С., Бычков И.В., Ружников Г.М., Хмельнов А.Е. Инструментальное средство автоматизации создания приложений баз данных на основе декларативных спецификаций // Вестн. Бурят. гос. ун-та. – 2011. – № 9. – С. 118-122.
8. Aho A.V., Lam M.S., Sethi R., Ulman J.D. Compilers. Principles, techniques and tools: Second Edition // Addison-Wesley Publishing Company, Inc. 2008.

9. Фереферов Е.С., Хмельнов А.Е. Реализация менеджера размещения визуальных компонентов в Delphi // Вычислительные и информационные технологии в науке, технике и образовании: материалы Междунар. конф. – Алматы; Новосибирск, 2008. – Т. 13. – С. 283-287.

10. ГИС Панорама. URL: <http://www.gisinfo.ru>

Фереферов Евгений Сергеевич, научный сотрудник ИДСТУ СО РАН, 664033, г. Иркутск, ул. Лермонтова, 134, тел. (395-2) 453071, e-mail: fereferov@icc.ru

Хмельнов Алексей Евгеньевич, кандидат технических наук, заведующий лабораторией ИДСТУ СО РАН, тел. (395-2) 453071, e-mail: hmelnov@icc.ru

Fereferov Evgeny Sergeevich, researcher, Institute for System Dynamics and Control Theory SB RAS, 664033, Irkutsk, Lermontov str., 134, e-mail: fereferov@icc.ru

Khmelnov Alexey Evgenievich, candidate of technical sciences, head of laboratory, Institute for System Dynamics and Control Theory SB RAS, e-mail: hmelnov@icc.ru

УДК 004.91

© А.О. Шигаров

ПРОЕКТ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ИЗВЛЕЧЕНИЯ ТАБЛИЧНОЙ ИНФОРМАЦИИ ИЗ НЕСТРУКТУРИРОВАННЫХ ТЕКСТОВ¹

Данная работа посвящена актуальной проблеме — разработке средств автоматизации для извлечения информации из произвольных таблиц со сложной компоновкой, представленных в неструктурированных текстах. Такие таблицы не включают формальную модель данных (например, реляционную модель данных), но имеют различные идентификаторы структуры (например, графическое форматирование, элементы компоновки и разметки).

Ключевые слова: анализ и обработка табличной информации, извлечение информации из таблиц, анализ и распознавание документов, системы анализа документов, информационный поиск, обработка неструктурированной информации.

© А.О. Shigarov

DESIGN OF INTELLEAGENT SYSTEM FOR TABLE INFORMATION EXTRACTION FROM UNSTRUCTURED TEXTS

The paper is addressed to actual problems of development of tools for automation information extraction from random tables with a complex arrangement in unstructured texts. These tables do not include a formal data model (e.g. relational data model), but they have different structure identifiers, (e.g. graphical formatting, layout, or markup)..

Keywords: analysis and processing of table information, information extraction from tables, document analysis and recognition, document analysis systems, information search, unstructured information processing.

Введение

В настоящее время многие исследователи, в т. ч. один из наиболее известных исследователей в области управления данными – Inmon В. [1] – отмечают важность проблем управления и интеграции неструктурированной информации. Под термином «неструктурированная информация» или «неструктурированные данные» (англ. unstructured information or unstructured data), как правило, понимается любая информация, которая не имеет predetermined формальной модели данных или не укладывается в таблицу реляционной базы данных [2]. Примерами такой информации являются изображения, научные статьи, отчеты, e-mail сообщения. Следует отметить, что иногда также используются более аккуратные термины — слабоструктурированные документы (англ. weakly structured documents) для обозначения текстовой информации, включающей в себя некоторые индикаторы структуры в виде элементов графики, компоновки или разметки, и полуструктурированные документы (англ. semi-structured documents) для обозначения текстовой информации, включающей в себя некоторые структурные и содержательные метаданные (например, HTML страницы). Подробное обсуждение этих терминов можно найти в работе Feldman R. и Sanger J. [3]. По аналогии со слабоструктурированными документами в данной работе под слабоструктурированными понимаются таблицы, не имеющие формальной модели данных, но при этом включающие некоторые индикаторы структуры, а также, возможно, некоторые структурные метаданные. Поскольку в таких таблицах отсутствует схема данных, предназначенная для высокоуровневой машинной обработки (например, выполнение запросов к данным по аналогии с SQL), то в определенном ранее смысле они также являются примером неструктурированной текстовой информации.

В литературе преобразование табличной информации от неструктурированной формы представления к структурированной принято называть извлечением информации из таблиц [4]. Сложность

¹ Работа выполнена при финансовой поддержке РФФИ грант № 12-07-31051, и Совета по грантам Президента РФ СП-3387.2013.5.

этой исследовательской проблемы обусловлена огромным разнообразием способов и форм представления, изображения таблиц. В зависимости от этого необходимо решать задачи анализа физической компоновки (обнаружения и сегментации) и логической компоновки таблицы (восстановление отношений и типов данных). Современное состояние данной области исследований не позволяет говорить о полном решении проблем извлечения информации из таблиц. Известные методы и системы анализа и обработки табличной информации преимущественно ориентированы на заранее определенные компоновки и особенности таблиц и документов. Эффективность таких методов и систем во многом зависит от вида таблиц, на которые они ориентированы. Являясь эффективными для таблиц определенного вида, они могут оказаться неэффективными для таблиц другого рода. Например, известные программные продукты OmniPage (Nuance Communications), Cuneiform (Cognitive Technologies), FineReader (ABBYY), PDF2XL (Cogniview) и Solid Converter PDF (Solid Documents) выполняют только обнаружение и сегментацию таблиц. Причем класс обрабатываемых ими таблиц ограничен простой «решеточной» компоновкой, когда табличные строки непосредственно соответствуют записям реляционной таблицы, а столбцы — ее полям. В работе [5] показано, что большинство исследований в этой области посвящено решению проблем низкоуровневой обработки табличной информации — обнаружению и сегментации таблиц из растровых изображений документов. При этом проблемы анализа логической компоновки и извлечения информации из слабоструктурированных таблиц остаются малоизученными.

Известные методы и системы основаны на различных подходах, например, на основе динамического программирования [8], машинного обучения [9] или статистических методов [10]. Однако для большинства из них при построении алгоритмов анализа характерно использование предположений (часто достаточно сильных) о компоновках обрабатываемых таблиц. По сути, такие знания (предположения) встроены в эти известные алгоритмы анализа и поэтому ограничивают классы эффективно (точно и полно) обрабатываемых ими таблиц. Использование логического вывода дает возможность разделить алгоритмы анализа слабоструктурированных таблиц и знания об обрабатываемых таблицах. В отличие от известных, такой подход позволяет реализовать различные программы (базы знаний) для различных форм (способов) представления и оформления слабоструктурированных таблиц. Данный подход положен в основу предлагаемой интеллектуальной системы извлечения табличной информации. Это обеспечивает обработку более широкого класса слабоструктурированных таблиц со сложной компоновкой по сравнению с известными исследованиями и программными продуктами.

1. Родственные работы

Наиболее близкими к данному исследованию являются работы S. Douglas и др. [6] и Y. Tijerino и др. [7]. В этих работах рассматривается преобразование (структурирование) табличной информации, называемое канонизацией таблицы (англ. table canonicalization). В работе S. Douglas и др. приводится определение канонической формы таблицы, под которой условно понимается отношение в терминах реляционных баз данных, а также предлагается метод интерпретации и канонизации таблиц, которые содержатся в спецификациях, используемых в строительной промышленности. Для этого S. Douglas S. и др. предлагают использовать обработку естественного языка. В частности, для разделения табличных заголовков и данных, а также для отнесения заголовков к определенным типам данных (доменам) предлагается использовать онтологию предметной области (подъязык спецификаций строительной промышленности). Основным недостатком этой системы является ориентированность на узкий класс таблиц. В работе Y. Tijerino и др. описывается платформа TANGO (Table ANalysis for Generating Ontologies, <http://tango.byu.edu>), предназначенная для инкрементальной генерации концептуальной онтологии из таблиц, содержащихся в web-страницах. В TANGO приведение таблицы к канонической форме является первым этапом в процессе генерации онтологии. Предлагаемый ими способ канонизации основан на использовании библиотеки фреймов, содержащей знания о лексическом содержании таблиц. Каждый фрейм данных описывает один тип данных и используется для отнесения выражений на естественном языке (табличных заголовков и значений) к этому типу. Для описания типов данных Y. Tijerino и др. предлагают использовать регулярные выражения, словари, лексическую базу данных английского языка WordNet (<http://wordnet.princeton.edu>) и другие ресурсы. Следует отметить, что приведение к канонической форме используется для слабоструктурированных таблиц со сложной компоновкой. Таким образом, в TANGO канонизация таблицы выполняется на

основе анализа ее естественно-языкового содержания. На практике этого не всегда достаточно. Для более точного и полного извлечения информации из таблицы часто также требуется анализ пространственной и графической информации.

2. Модель таблицы

На основе представлений табличной информации в основных широко распространенных форматах данных документов – табличного процессора Excel, текстового процессора Word, страниц HTML и системы LaTeX – предлагается оригинальная, достаточно общая, модель слабоструктурированной таблицы. В данной модели сделано несколько общих для этих представлений предположений: 1) ячейка может располагаться в одной или нескольких соседних строках и в одном или нескольких соседних столбцах (например, атрибуты “colspan” и “rowspan” в HTML) и имеет прямоугольную форму в пространстве строк и столбцов; 2) внутри ячейки не могут располагаться другие ячейки (это не допускается в Excel); 3) текстовое содержимое ячейки может являться либо меткой (заголовком), либо входением (данными); 4) метки могут адресовать вхождения либо в строках – метки строк, либо в столбцах — метки столбцов. Сделанные предположения описывают широкий класс обрабатываемых таблиц. Пример описания слабоструктурированной таблицы в терминах предложенной модели приводится на рис. 1.

Используемые в данной работе термины “метка” и “вхождение” соответствуют смыслу терминов “label” и “entry” из работы X. Wang [11]. Предлагаемая модель слабоструктурированной таблицы включает два уровня, которые в упрощенном виде можно описать следующим образом.

1. Уровень физической компоновки $Tp=(Sr, Sc, C)$ состоит из: пространства строк — Sr и столбцов — Sc ; набора ячеек — C , в котором каждая ячейка — $c=(p, c', F, G)$ включает информацию о своих координатах в пространстве строк Sr и столбцов Sc (cl — левой, rt — верхней, cr — правой и rb — нижней гранях соответственно) — $p=(cl, rt, cr, rb)$, содержании — c' ; графическом форматировании (цветовые схемы, шрифтовые метрики, выравнивание и др.) — F , и разграфке (оформлении границ) — G .

2. Уровень логической компоновки $Tl=(D, Lr, Lc, A, E)$ состоит из: набора измерений — D ; дерева меток строк — Lr , дерева меток столбцов — Lc , отражающих связи между метками — $I=(I', Di)$, где I' — содержание метки, а Di — сопоставленное ей измерение из набора D ; набора значений измерений $A=\{Ai\}$, где Ai — подмножество значений измерения Di , представленных в обрабатываемой таблице; набора вхождений — E , в котором каждое вхождение — $e=(e', A', L')$ включает информацию о своем содержании — e' , связанных с ней значениях измерений — A' (A' является подмножеством A) и метках — L' (L' является подмножеством объединения меток из деревьев Lr и Lc).

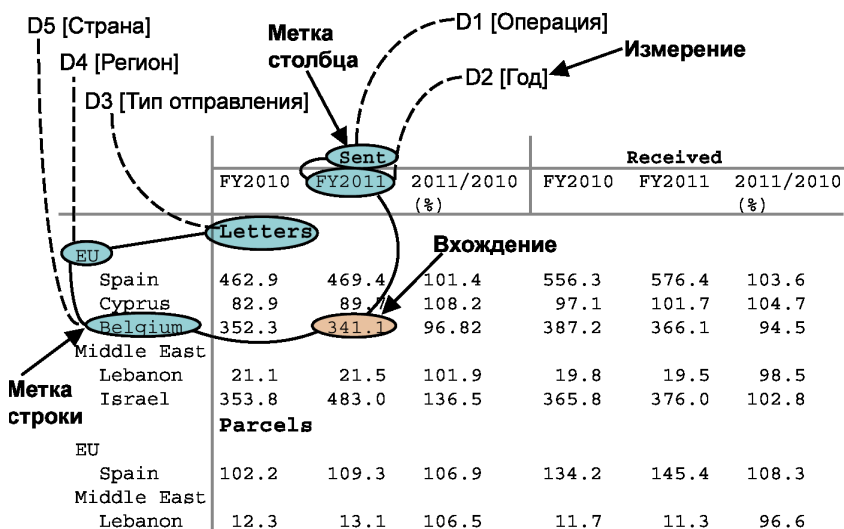


Рис. 1. Пример описания слабоструктурированной таблицы в терминах предложенной модели

3. Ядро анализа табличной компоновки

Основная идея, лежащая в основе предлагаемого подхода к анализу табличной компоновки, состоит в следующем. Обычно внутри тематической коллекции документов от одного поставщика таблицы компонуются и форматируются однообразно. Для такой коллекции документов можно определить набор формализованных правил анализа табличной компоновки, который удовлетворяет всем или (почти всем) ее таблицам. Эти правила можно представить в виде базы знаний, а процесс анализа табличной компоновки реализовать как логический вывод. При этом база фактов формируется из доступной в неструктурированном текстовом источнике табличной информации и внешней информации (фактах о типах данных и отношениях между понятиями). В результате логического вывода формируются новые факты о табличной компоновке, т.е. выполняется восстановление информации, отсутствующей в исходном представлении. На основе восстановленной информации выполняется структурирование (канонизация) таблицы.

Пример 1

```
rule "Recovering row labels in the stub"
  dialect "mvel"
  when
    $c : CCell( cl == 1 )
  then
    modify ( $c ) { setRole( Role.ROWLABEL ) };
end
```

Пример 2

```
rule "Binding column labels in the head"
  dialect "mvel"
  when
    $c1 : Ccell( role == Role.COLLABEL )
    $c2 : Ccell( role == Role.COLLABEL,
                rt == $c1.rb + 1,
                ( $c1.cl <= cl && cr < $c1.cr ) ||
                ( $c1.cl < cl && cr <= $c1.cr ) )
  then
    $c1.addConnectedCell( $c2 );
end
```

Пример 3

```
rule "Recovering ignored row labels without nulls"
  dialect "mvel"
  when
    $c : CCell( role == null, cl == 1, cl == cr,
                content matches "(?i).*(total)|.*(average)" )
  then
    modify ( $c ) { setRole( Role.IGNORED_ROWLABEL ) }
end
```

Пример 4

```
rule "Binding entries and column labels"
  dialect "mvel"
  when
    $l : CCell( role == Role.COLLABEL )
    $e : CCell( role == Role.ENTRY, cl == $l.cl, cr == $l.cr)
  then
    $e.addConnectedCell( $l );
end
```

Рис. 2. Примеры правил анализа логической компоновки на языке MVEL

Табличная информация, представленная на уровне физической компоновки предлагаемой модели, формируется в результате анализа физической компоновки (обнаружения внутри источника и сегментация таблицы) исходной информации (изображений, ASCII-текста, PDF, Excel и др.). Информация, представленная на уровне логической компоновки предлагаемой модели, формируется в результате анализа логической компоновки информации о физической компоновке. В данной работе обсуждаются именно задачи анализа логической компоновки, которые остаются малоизученными в мире по сравнению с задачами анализа физической компоновки.

Предлагаемый анализ компоновки основан на логическом выводе, который выполняется в свободной системе исполнения правил Drools Expert (JBoss Community, <http://www.jboss.org/drools>). Продукционные правила анализа табличной компоновки записываются на языке выражений MVEL (<http://mvel.codehaus.org>) и используют предлагаемые структуры данных. Для анализа логической компоновки такие правила отображают доступную информацию — позиции (координаты) элементов таблицы (ячеек, строк и столбцов), графическое форматирование (цветовые схемы, шрифтовые метрики, выравнивание и др.) и текстовое содержание ячеек, в отсутствующую изначально информацию — семантические отношения между ячейками, роли (метки или вхождения) и типы данных (домены) содержимого ячеек. С помощью этого решаются следующие задачи: 1) разделение ячеек на метки и вхождения; 2) восстановление связей между вложенными и охватывающими метками, между метками и вхождениями; 3) восстановление типов данных для меток.

На рис. 2 приводятся ряд простых примеров возможных правил анализа логической компоновки на языке MVEL. Пример 1: если ячейка “\$c” находится в первом столбце “cl==1”, то необходимо пометить, что ее содержание является меткой строки “modify (\$c) {setRole(Role.ROWLABEL)}”. Пример 2: если две ячейки “\$c1” и “\$c2” содержат метки столбцов “role==Role.COLLABEL” и вторая ячейка “\$c2” расположена непосредственно под первой “\$c1” таким образом, что первая ячейка “\$c1” полностью охватывает вторую “\$c2” по столбцам, то необходимо задать связь между этими ячейками “\$c1.addConnectedCell(\$c2)”. Пример 3: если ячейка “\$c” с неопределенной ролью “role==null” расположена полностью в первом столбце “cl == 1, cl==cr” и содержит текст, удовлетворяющий регулярному выражению вида “(?i).*(total).*(average)”, то необходимо ее отметить как игнорируемую в дальнейшем процессе обработки. Пример 4: если ячейка “\$e” содержит вхождение “role==Role.ENTRY” и находится в одном столбце с ячейкой “\$l”, содержащей метку столбца “role==Role.COLLABEL”, то необходимо задать связь между этими ячейками “\$e.addConnectedCell(\$l)”.

На практике для одного или нескольких схожих источников, из которых необходимо извлечь табличную информацию, можно разработать набор подобных правил. Например, для обработки таблиц из одного источника (Statistical Handbook of Japan 2007. Statistics Bureau of Japan) потребовалось составить 9 правил, которые занимали 94 строки кода на языке MVEL. По сравнению с известными предлагаемый подход требует развития моделей анализа табличной информации (наборов правил) для различных случаев исходных данных, но при этом он применим для существенно более широкого класса слабоструктурированных таблиц без потери эффективности (точности и полноты) извлечения информации.

Дополнительные алгоритмы обработки табличной информации

Для нормализации и структурирования информации о табличной компоновке, восстановленной в процессе логического вывода, предлагается ряд оригинальных алгоритмов. Далее приводится их краткое описание.

Метки внутри одной или нескольких таблиц могут различаться по написанию, но иметь одно лексическое значение, т.е. могут являться синонимами. Например, следующие метки: “2010”, “FY2010”, “Year 2010” и “Previous Year”, “2010 г.”, “Текущий год” могут быть синонимами, означающими 2010 г. В качестве эталонного значения приведенных выражений может использоваться “2010”. Нормализация меток начинается с замены таких выражений на их эталоны путем сопоставления со словарем эталонов, который содержит набор отношений вида (S, R), где S — естественно-языковое или регулярное выражение, а R — соответствующее ему эталонное выражение. Например, если в словаре задать отношения вида (“FY[2][0][0-1][0-3]”, “[2][0][0-1][0-3]”), то все заголовки, соответствующие регулярному выражению “FY[2][0][0-1][0-3]”, т.е. “FY2000”, ..., “FY2013”, будут заменены на соответ-

ствующие эталоны “2000”,..., “2013”. Кроме того, в таблице может содержаться несколько меток с одинаковым значением. Предполагается, что они являются экземплярами одной растиражированной метки. В процессе нормализации все экземпляры такой метки заменяются ею. При этом для каждого вхождения его связь с экземпляром метки заменяется связью с растиражированной меткой.

Данные	Операция	Год	Тип отправления	Регион	Страна
462.9	Sent	2010	Letters	EU	Spain
82.9	Sent	2010	Letters	EU	Cyprus
352.3	Sent	2010	Letters	EU	Belgium
21.1	Sent	2010	Letters	Middle East	Lebanon
353.8	Sent	2010	Letters	Middle East	Israel
102.2	Sent	2010	Parcels	EU	Spain
12.3	Sent	2010	Parcels	Middle East	Lebanon
469.4	Sent	2011	Letters	EU	Spain
89.7	Sent	2011	Letters	EU	Cyprus
341.1	Sent	2011	Letters	EU	Belgium
21.5	Sent	2011	Letters	Middle East	Lebanon
483.0	Sent	2011	Letters	Middle East	Israel
109.3	Sent	2011	Parcels	EU	Spain
13.1	Sent	2011	Parcels	Middle East	Lebanon
556.3	Received	2010	Letters	EU	Spain
97.1	Received	2010	Letters	EU	Cyprus
387.2	Received	2010	Letters	EU	Belgium
19.8	Received	2010	Letters	Middle East	Lebanon
365.8	Received	2010	Letters	Middle East	Israel
134.2	Received	2010	Parcels	EU	Spain
11.7	Received	2010	Parcels	Middle East	Lebanon
576.4	Received	2011	Letters	EU	Spain
101.7	Received	2011	Letters	EU	Cyprus
366.1	Received	2011	Letters	EU	Belgium
19.5	Received	2011	Letters	Middle East	Lebanon
376.0	Received	2011	Letters	Middle East	Israel
145.4	Received	2011	Parcels	EU	Spain
11.3	Received	2011	Parcels	Middle East	Lebanon

Рис. 3. Пример канонической формы для слабоструктурированной таблицы из рис. 1

Структурирование информации о логической компоновке таблицы включает дополнительную разметку, которая состоит в сопоставлении метки с измерением. Этот процесс основан на использовании словаря измерений, в котором естественно-языковые и регулярные выражения сопоставляются с заранее заданными измерениями. Словарь измерений содержит набор отношений вида (S, Di) , где S — выражение, а Di — соответствующее измерение. Дополнительно в процессе разметки могут определяться метки, указывающие на вычислимость данных (“проценты”, “итоги” и др.). Такие данные часто целесообразно исключать из дальнейшего преобразования табличной информации к целевой форме. Описанная разметка также доступна и в процессе логического вывода, а соответствующие словарям и измерениям структуры данных могут использоваться в определениях правил при разработке баз знаний.

По разметке выполняется восстановление используемых в таблице значений измерений. Из деревьев меток Lg и Lc исключаются метки, которые являются значениями измерений из D . Каждая исключаемая метка добавляется в список значений соответствующего измерения — Ai . В случае, когда каждая метка соотнесена с некоторым измерением, деревья меток вырождаются. Из невырожденных деревьев строятся пути меток от листьев до корней. По вхождениям (данным), построенным путям меток, и восстановленным значениям измерений формируется таблица в канонической форме, соответствующая по смыслу отношению в терминах реляционной модели данных. Предлагаемая канони-

ческая таблица включает следующие атрибуты (поля): DATA — поле вхождений (данных); ROW_LABEL — поле путей из невырожденного дерева меток строк, COL_LABEL — поле путей из невырожденного дерева меток столбцов; D1, D2, ..., Dn — поля значений соответствующих измерений. Пример канонической формы обработанной таблицы приводится на рис. 3. Сформированная каноническая таблица может экспортироваться в формат реляционной базы данных с помощью стандартных средств управления данными.

Реализация и практическое использование

На основе предложенных в данной работе подходов, модели и алгоритмов разработан прототип интеллектуальной системы извлечения информации из слабоструктурированных таблиц со сложной компоновкой, который был успешно применен в практической задаче.

В прототипе предлагаемой системы реализованы алгоритмы: 1) загрузки исходной табличной информации, представленной в формате табличного процессора Excel (тестовых данных со специализированной разметкой); 2) нормализации и структурирования табличной информации, восстановленной в процессе логического вывода; 3) экспорта результатов в формате Excel. Предлагаемая в работе модель таблицы реализована в виде ряда структур данных, основные из которых: CELL, ENTRY, LABEL, LABELNODE. Структура CELL предназначена для представления ячейки и прежде всего информации о ее физической компоновке, однако она также включает уровень логической компоновки ячейки (т.е. она позволяет накапливать информацию о ее связях с другими ячейками, ее роли и типе данных). На практике это позволяет разрабатывать правила анализа логической компоновки в более лаконичной манере по сравнению со случаем, при котором используются дополнительные структуры данных для представления информации уровня логической компоновки. Структуры ENTRY, LABEL, LABELNODE используются исключительно на уровне логической компоновки. ENTRY служит для представления вхождения, а LABEL — метки. Структура LABELNODE является оболочкой для структуры LABEL и обеспечивает представление деревьев меток. Все предложенные структуры данных и алгоритмы реализованы на платформе Java.

В рамках совместного российско-монгольского проекта РФФИ № 11-07-92204 с помощью данного прототипа наполнено хранилище данных по социально-экономическому положению территорий Монголии из неструктурированных текстовых источников, предоставленных Институтом национального развития Монголии (<http://mdi.gov.mn>). Информация извлекалась из более чем 40 слабоструктурированных таблиц в формате табличного процессора Excel. Примеры таких таблиц представлены на рис. 4. Для анализа этой информации потребовалось описать следующие измерения: “время” — (D1={"19\d\d.0", "200\d.0", "201[0-2].0"}), “пространство” — (D2={"Bayan-Olgii", "Govi-Altai", ..., "Ulaanbaatar"}), соответствующее административному делению Монголии, и “отрасли” — (D3={"Agriculture", "Industry", ..., "Services"}). Еще два дополнительных измерения “показатели” и “меры” были сформированы из названий таблиц в процессе обработки их контекста. Также потребовалось составить 9 правил, занимающих 81 строку кода на языке MVEL. В условиях (левых частях) данных правил используется информация о расположении ячеек и их текстовом содержимом. При этом некоторые правила определяли игнорируемые метки и вхождения. Например, в процессе логического вывода для меток, удовлетворяющих регулярному следующему выражению: "(?i).*(total)|.*(region)|.*(average)|(\s*)", задавалась роль игнорируемых меток — "IGNORED_ROW_LABEL" (пример 3 на рис. 2). Кроме того, в правой части некоторых правил использовались дополнительные вычисления по преобразованию текстовых выражений (строковых представлений чисел) к числам с помощью регулярных выражений и методов класса Math из состава платформы исполнения Java.

	2003			...	2011	
	Total, bln.tog	Share to total, %			Total, bln.tog	Services
		Agriculture	Services			
TOTAL	1479,70	20,60	54,00	10829689,60	52,00	
Western region	118,50	63,50	32,20	455942,30	41,40	
<i>Bayan-Olgii</i>	27,30	59,30	36,70	91968,90	51,90	
<i>Govi-Altai</i>	14,60	62,00	39,10	60036,60	49,40	
...						
<i>Khentii</i>	25,90	74,80	20,30	99575,30	29,10	
Ulaanbaatar	861,50	0,80	73,00	6991314,80	68,80	

<i>Aimags and the Capital</i>	1992	1993	...	2011
TOTAL	806,00	765,40		1037,70
Total	160,10	152,00		167,80
<i>Bayan-Olgii</i>	26,70	22,00		42,01
<i>Govi-Altai</i>	26,90	25,30		24,44
...				
Total	78,20	70,20		76,70
<i>Dornod</i>	30,80	27,60		25,77
...				
<i>Khentii</i>	29,00	25,50		26,20
Ulaanbaatar	206,30	195,80		361,40

Рис. 4. Примеры обрабатываемых таблиц

В разработанном прототипе системы извлечения табличной информации не реализованы механизмы анализа физической компоновки таблиц, и, в частности, средства обнаружения таблиц внутри текста. Чтобы избежать сложности, связанной с анализом физической компоновки таблиц, использовалась дополнительная разметка внутри листов табличного процессора Excel. Как показано на рис. 5, используются теги, указывающие позиции начала — \$START и конца — \$END таблицы, а также элементы ее контекста: название — \$NAME и единицу измерения — \$MEASURE. На данном этапе исследования использование формата Excel с дополнительной разметкой в качестве исходного представления позволяет свести анализ физической компоновки таблицы к ряду достаточно тривиальных процедур загрузки и предобработки данных. Всего в процессе обработки этих таблиц извлечено более 15 000 неагрегированных значений данных (вхождений). Время обработки этих данных составляет менее 40 с на компьютере со следующими параметрами: Intel Core 2 Quad 2,66ГГц, 4Гб.

	\$NAME	Table 1. ...
	\$MEASURE	bln.tog
\$START		
	...	

	...	
		\$END

Рис. 5. Дополнительная разметка таблицы внутри листа Excel

Заключение

В работе изложен оригинальный подход к анализу компоновки слабоструктурированных таблиц на основе логического вывода. Использование этого подхода позволяет разрабатывать различные программы (базы знаний) для различных форм (способов) представления и оформления таблиц. Данный подход положен в основу разрабатываемой интеллектуальной системы извлечения табличной информации из неструктурированных текстов. Реализованный прототип этой системы успешно применен для автоматизации ввода в хранилище данных информации из таблиц по социально-экономическому положению Монголии, которые представлены изначально в неструктурированных текстах.

Литература

1. Inmon B. Matching unstructured data and structured data // The data administration newsletter. 2006. URL: <http://www.tdan.com/view-articles/5009>
2. Blumberg R., Atre S. The problem with unstructured data // DM Review, 2003. URL: http://soquelgroup.com/Articles/dmreview_0203_problem.pdf
3. Feldman R., Sanger J. The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data // Cambridge University Press. 2006. 422 p.
4. Embley D.W., Hurst M., Lopresti D., Nagy G. Table-processing paradigms: a research survey // Int. J. on Document Analysis and Recognition. Springer-Verlag. 2006. Vol. 8, No. 2. P. 66-86.
5. e Silva A.C., Jorge A.M., Torgo L. Design of an end-to-end method to extract information from tables // Int. J. on Document Analysis and Recognition. Springer-Verlag. 2006. Vol. 8, No. 2. P. 144-171.
6. Douglas S., Hurst M., Quinn D. Using Natural Language Processing for Identifying and Interpreting Tables in Plain Text // In Proc. of the 4th Annual Symposium on Document Analysis and Information Retrieval. Las Vegas. 1995. P. 535-546.
7. Tijerino Y., Embley D., Lonsdale D., Nagy G.: Towards ontology generation from tables // World Wide Web: Internet and Web Information Systems. 2005. Vol. 8, No 3. P. 261-285.
8. Hu J., Kashi R., Lopresti D., Wilfong G. Medium-independent table detection // In Proc. of the Document Recognition and Retrieval VII. 2000. P. 291-302.
9. Tengli A., Yang Y., Ma N. L. Learning table extraction from examples // In Proc. of the 20-th Int. Conf. on Computational Linguistics. Stroudsburg, USA. 2004. Article 987.
10. Pinto D., McCallum A., Wei X., Croft B. Table extraction using conditional random fields // In Proc. of the 26-th Int. ACM SIGIR Conf. on Research and Development in Informaion Retrieval. New York, USA. 2003. P. 235-242.
11. Wang X. Tabular Abstraction, Editing, and Formatting. PhD Thesis. Waterloo, Ontario, Canada. 1996.

Шигаров Алексей Олегович, кандидат технических наук, старший научный сотрудник ИДСТУ СО РАН, тел. (3952) 453102, e-mail: shigarov@icc.ru

Shigarov Alexey Olegovich, candidate of technical sciences, senior researcher, Institute for System Dynamics and Control Theory SB RAS. Ph.: (3952) 453102, e-mail: shigarov@icc.ru

3. Функциональный анализ и дифференциальные уравнения

УДК 517.956.6

© А.А. Алсыкова

НАЧАЛЬНО-КРАЕВАЯ ЗАДАЧА С ИНТЕГРАЛЬНЫМИ ГРАНИЧНЫМИ УСЛОВИЯМИ ДЛЯ ОДНОГО КЛАССА УРАВНЕНИЙ СОСТАВНОГО ТИПА¹

Исследована разрешимость начально-краевой задачи для уравнения соболевского типа

$$Lv \equiv Av_{tt} + Bv = f$$

с эллиптическим оператором A и оператором B произвольного типа с интегральным краевым условием.

Ключевые слова: начально-краевая задача, уравнение соболевского типа, априорные оценки, теоремы существования единственного решения.

© А.А. Alsykova

INITIAL BOUNDARY VALUE PROBLEM WITH INTEGRAL BOUNDARY CONDITIONS FOR A CLASS OF COMPOSITE TYPE EQUATIONS

Solvability of the initial-boundary value problem for an equation of Sobolev type is researched

$$Lv \equiv Av_{tt} + Bv = f$$

with the elliptic operator A and the operator B of arbitrary type with integral boundary condition.

Keywords: initial-boundary value problem, the equation of Sobolev type, a priori estimates, theorems of existence a unique solution.

Пусть Ω – ограниченная область пространства R^n с гладкой (для простоты, бесконечно дифференцируемой) границей Γ , Q – цилиндр $\Omega \times (0, T)$ конечной высоты T , $S = \Gamma \times (0, T)$ его боковая граница, $a^{ij}(x)$, $b^{ij}(x, t)$, $b^i(x, t)$, ($i, j = 1, \dots, n$), $a(x)$, $b(x, t)$, $f(x, t)$, и $K(x, y, t)$ – заданные при $x \in \bar{\Omega}$, $y \in \bar{\Omega}$, $t \in [0, T]$ функции, A , B и L – операторы, действие которых на функцию $v(x, t)$ определяется равенствами

$$Av = \frac{\partial}{\partial x_i} (a^{ij}(x)v_{x_j}) + a(x)v,$$

$$Bv = b^{ij}(x, t)v_{x_i x_j} + b^i(x, t)v_{x_i} + b(x, t)v,$$

$$Lv = Av_{tt} + Bv$$

(здесь и далее по повторяющимся индексам ведется суммирование в пределах от i до n).

Краевая задача: найти функцию $u(x, t)$, являющуюся в цилиндре Q решением уравнения

$$Lu = f(x, t) \tag{1}$$

и такую, что для нее выполняются условия

$$u(x, 0) = u_t(x, 0) = 0, \quad x \in \Omega, \tag{2}$$

$$u(x, t)|_{(x, t) \in S} = \int_{\Omega} K(x, y, t)u(y, t)dy|_{(x, t) \in S}. \tag{3}$$

Краевые задачи с интегральным граничным условием вида (3) прежде всего для параболических и гиперболических уравнений, для $(2m+1)$ -параболических уравнений, ультрапараболических уравнений активно изучаются в последнее время; как наиболее близкие к настоящей работе по постановке и технике отметим работы [1–7]. С другой стороны, уравнение (1) близко по типу к уравнениям составного типа и, в частности, содержит известное уравнение Буссинеска [8], но краевые задачи с граничным условием (3) интегрального вида для таких уравнений ранее не изучались.

Перейдем к содержательной части работы.

¹ Работа выполнена при поддержке гранта БГУ-2012

Определим оператор M равенством

$$(Mu)(x, t) = u(x, t) - \int_{\Omega} K(x, y, t)u(y, t)dy,$$

для удобства значение оператора M на функции $u(x, t)$ обозначим как $w(x, t)$.

Пусть V есть следующее пространство функций

$$V = \{v(x, t) : v(x, t) \in L_2(0, T; W_2^2(\Omega)), v_i(x, t) \in L_2(0, T; W_2^2(\Omega)), v_{ii}(x, t) \in L_2(0, T; W_2^2(\Omega))\},$$

норму в этом пространстве определим естественным образом

$$\|v\|_V = \|v\|_{L_2(0, T; W_2^2(\Omega))} + \|v_i\|_{L_2(0, T; W_2^2(\Omega))} + \|v_{ii}\|_{L_2(0, T; W_2^2(\Omega))}.$$

Введем обозначение

$$\begin{aligned} \Phi(x, t, u) = & \int_{\Omega} K(x, y, t)a(y)u_{ii}(y, t)dy + \int_{\Omega} K(x, y, t)A_{0y}u_{ii}(y, t)dy + \\ & + \int_{\Omega} K(x, y, t)B_yu(y, t)dy - \int_{\Omega} A_xK(x, y, t)u_{ii}(y, t)dy - 2 \int_{\Omega} A_xK_i(x, y, t)u_i(y, t)dy - \\ & - \int_{\Omega} A_xK_{ii}(x, y, t)u(y, t)dy - \int_{\Omega} B_xK(x, y, t)u(y, t)dy, \end{aligned}$$

где

$$A_xu = \frac{\partial}{\partial x_i}(a^{ij}(x)u_{x_j}) + a(x)u, \quad A_yu = \frac{\partial}{\partial y_i}(a^{ij}(y)u_{x_j}) + a(y)u,$$

$$A_{0y}u = \frac{\partial}{\partial y_i}(a^{ij}(y)u_{y_j}), \quad B_xu = b^{ij}(x, t)u_{x_ix_j} + b^i(x, t)u_{x_i} + b(x, t)u,$$

$$B_yu = b^{ij}(y, t)u_{y_iy_j} + b^i(y, t)u_{y_i} + b(y, t)u.$$

Заметим, что для функции $v(x, t)$ из пространства V , такой, что

$$v(x, 0) = v_t(x, 0) = 0, \tag{2'}$$

$$v(x, t)|_S = 0, \tag{3'}$$

имеют место следующие неравенства

$$\int_{\Omega} v^2(x, t)dx \leq R_1 \sum_{i=1}^n \int_{\Omega} v_{x_i}^2(x, t)dx \leq R_2 \sum_{i,j=1}^n \int_{\Omega} v_{x_ix_j}^2(x, t)dx, \tag{4}$$

в которых постоянные R_1 и R_2 определяются областью Ω ,

$$v^2(x, t) \leq T \int_0^{\tau} v_{\xi}^2(x, \xi)d\xi \leq T^2 \int_0^{\xi} \int_0^{\xi} v_{zz}^2(x, z)dzd\xi. \tag{5}$$

Положим

$$N(x, y, t) = K(x, y, t)a(y) - A_xK(x, y, t),$$

$$m_1 = \max_{0 \leq t \leq T} \int_{\Omega} \int_{\Omega} N^2(x, y, t)dydx, \quad m_2 = \max_{i,j=1,\dots,n} \max_{0 \leq t \leq T} \int_{\Omega} \int_{\Omega} [K(x, y, t)a^{ij}(y)]^2dydx,$$

$$m_3 = \max_{i,j=1,\dots,n} \max_{0 \leq t \leq T} \int_{\Omega} \int_{\Omega} [K(x, y, t)a_{y_i}^{ij}(y)]^2dydx, \quad m_4 = \max_{i,j=1,\dots,n} \max_{0 \leq t \leq T} \int_{\Omega} \int_{\Omega} K_{x_i}^2(x, y, t)dydx,$$

$$m_5 = \max_{i,j=1,\dots,n} \max_{0 \leq t \leq T} \int_{\Omega} \int_{\Omega} K_{x_ix_j}^2(x, y, t)dydx.$$

Утверждение 1. Пусть $w(x, t)$ есть функция из пространства V , для которой выполняются условия (2') и (3'), $u(x, t)$ есть функция $M^{-1}w$, и пусть выполняются условия

$$\begin{aligned} a^{ij}(x) \in C^1(\bar{\Omega}), \quad b^{ij}(x, t) \in C(\bar{\Omega}), \quad b^i(x, t) \in C(\bar{\Omega}), \quad i, j = 1, \dots, n, \\ a(x) \in C(\bar{\Omega}), \quad b(x, t) \in C(\bar{\Omega}), \quad K(x, y, t) \in C^3(\bar{\Omega} \times \bar{\Omega} \times [0; T]), \end{aligned}$$

$$b_0 \int_{\Omega} w^2(x, t) dx \leq \int_{\Omega} u^2(x, t) dx \leq b_1 \int_{\Omega} w^2(x, t) dx, \\ b_0 \geq 0, t \in [0; T], u(x, t) \in L_2(0, T; W_2^2(\Omega)). \quad (6)$$

Тогда справедливо неравенство

$$\int_{\Omega} \Phi^2(x, t, u) dx \leq k_1 \sum_{i,j=1}^n \int_{\Omega} w_{y_i y_j}^2(y, t) dy + k_2 \sum_{i=1}^n \int_{\Omega} w_{y_i}^2(y, t) dy + \\ + k_3 \int_{\Omega} w_{tt}^2(y, t) dy + C_2 \left(\int_0^t \int_{\Omega} w_{\tau\tau}^2(y, \tau) dy d\tau + \sum_{i,j=1}^n \int_0^t \int_0^{\tau} \int_{\Omega} w_{y_i y_j \xi \xi}^2(y, \xi) dy d\xi d\tau + \right. \\ \left. + \sum_{i=1}^n \int_0^t \int_0^{\tau} \int_{\Omega} w_{y_i \xi \xi}^2(y, \xi) dy d\xi d\tau + \int_0^t \int_0^{\tau} \int_{\Omega} w_{\xi \xi}^2(y, \xi) dy d\xi d\tau \right),$$

где k_1 , k_2 и k_3 есть соответственно величины

$2m_2(n^2 + n)(1 + \delta_1^2)(1 + \delta_4^2)$, $2m_3(n^2 + n)(1 + \delta_1^2)(1 + \delta_3^2)$ и $2b_1 m_1(1 + \delta_1)(1 + \delta_1^2)(1 + \delta_2^2)[m_1 + (n^2 + n)(m_2 m_5(2 + \delta_4^2) + m_3 m_4(2 + \delta_3^2))]$, с произвольными сколь угодно малыми положительными числами $\delta_1, \delta_2, \delta_3$ и δ_4 , C_2 есть число, определяемое коэффициентами операторов A и B , числами $\delta_1, \delta_2, \delta_3, \delta_4$, а также функцией $K(x, y, t)$.

Доказательство. Учитывая введенные обозначения и используя неравенства Гельдера и Юнга, нетрудно получить следующее неравенство

$$\int_{\Omega} \Phi^2(x, t, u) dx \leq (1 + \delta^2) \left[m_1 \int_{\Omega} u_t^2(x, t) dx + \right. \\ \left. + 2m_2(n^2 + n) \sum_{i,j=1}^n \int_{\Omega} u_{x_i x_j}^2(x, t) dx + 2m_3(n^2 + n) \sum_{i=1}^n \int_{\Omega} u_{x_i}^2(x, t) dx \right] + \\ C_1 \left(\sum_{i,j=1}^n \int_{\Omega} u_{y_i y_j}^2(y, t) dy + \sum_{i=1}^n \int_{\Omega} u_{y_i}^2(y, t) dy + \int_{\Omega} u^2(y, t) dy + \int_{\Omega} u_t^2(y, t) dy \right), \quad (7)$$

где постоянная C_1 определяется лишь коэффициентами операторов A и B , а также числом δ и функцией $K(x, y, t)$.

Имеет место следующее равенство

$$u_t(x, t) - \int_{\Omega} K(x, y, t) u_t(y, t) dy = w_t(x, t) + \int_{\Omega} K_t(x, y, t) u(y, t) dy.$$

Обозначим через $w_1(x, t)$ правую часть данного равенства.

Учитывая условие (6), получаем

$$\int_{\Omega} u_t^2(x, t) dx \leq b_1 \int_{\Omega} w_1^2(x, t) dx.$$

Используя неравенство Гельдера, а также условие (6), оценим $w_1^2(x, t)$:

$$w_1^2(x, t) = \left(w_t(x, t) + \int_{\Omega} K_t(x, y, t) u(y, t) dy \right)^2 \leq \\ \leq 2w_t^2(x, t) + 2 \int_{\Omega} K_t^2(x, y, t) dy \int_{\Omega} u^2(y, t) dy \leq \\ \leq 2w_t^2(x, t) + 2b_1 \int_{\Omega} K_t^2(x, y, t) dy \int_{\Omega} w^2(y, t) dy.$$

Отсюда

$$\int_{\Omega} u_t^2(x, t) dx \leq 2b_1 \int_{\Omega} w_t^2(y, t) dy + 2b_1^2 \int_{\Omega} K_t^2(x, y, t) dy \int_{\Omega} w^2(y, t) dy. \quad (8)$$

Справедливы равенства

$$\begin{aligned} u_{tt}(x, t) - \int_{\Omega} K(x, y, t) u_{tt}(y, t) dy &= w_{tt}(x, t) + \\ &+ 2 \int_{\Omega} K_t(x, y, t) u_t(y, t) dy + \int_{\Omega} K_{tt}(x, y, t) u(y, t) dy, \\ u_{x_{it}}(x, t) &= w_{x_{it}}(x, t) + \sum_{i=1}^n \int_{\Omega} K_{x_i}(x, y, t) u_{tt}(y, t) dy + \\ &+ 2 \sum_{i=1}^n \int_{\Omega} K_{x_{it}}(x, y, t) u_t(y, t) dy + \sum_{i=1}^n \int_{\Omega} K_{x_{it}}(x, y, t) u(y, t) dy, \\ u_{x_i x_{jt}}(x, t) &= w_{x_i x_{jt}}(x, t) + \sum_{i,j=1}^n \int_{\Omega} K_{x_i x_j}(x, y, t) u_{tt}(y, t) dy + \\ &+ 2 \sum_{i,j=1}^n \int_{\Omega} K_{x_i x_{jt}}(x, y, t) u_t(y, t) dy + \sum_{i,j=1}^n \int_{\Omega} K_{x_i x_{jt}}(x, y, t) u(y, t) dy, \end{aligned}$$

из которых, используя (8), нетрудно получить следующие неравенства:

$$\int_{\Omega} u_{tt}^2(x, t) dx \leq b_1(1 + \delta_2^2) \int_{\Omega} w_{tt}^2(x, t) dx + d_1 \left(\int_{\Omega} w_t^2(x, t) dx + \int_{\Omega} w^2(x, t) dx \right), \quad (9)$$

$$\begin{aligned} \sum_{i=1}^n \int_{\Omega} u_{x_{it}}^2(x, t) dx &\leq (2 + \delta_3^2) \sum_{i=1}^n \int_{\Omega} w_{x_{it}}^2(x, t) dx + (2 + \delta_3^2)(1 + \delta_2^2) b_1 m_4 \int_{\Omega} w_{tt}^2(x, t) dx + \\ &+ d_2 \left(\int_{\Omega} w_t^2(x, t) dx + \int_{\Omega} w^2(x, t) dx \right), \end{aligned} \quad (10)$$

$$\begin{aligned} \sum_{i=1}^n \int_{\Omega} u_{x_i x_{jt}}^2(x, t) dx &\leq (2 + \delta_4^2) \sum_{i,j=1}^n \int_{\Omega} w_{x_i x_{jt}}^2(x, t) dx + (2 + \delta_4^2)(1 + \delta_2^2) b_1 m_5 \int_{\Omega} w_{tt}^2(x, t) dx + \\ &+ d_3 \left(\int_{\Omega} w_t^2(x, t) dx + \int_{\Omega} w^2(x, t) dx \right). \end{aligned} \quad (11)$$

Здесь d_1 , d_2 и d_3 – постоянные, зависящие от функции $K(x, y, t)$.

Используя полученные оценки (8) – (11) и неравенства (4), (5), продолжим (7)

$$\begin{aligned} \int_{\Omega} \Phi^2(x, t, u) dx &\leq 2m_2(n^2 + n)(1 + \delta_1^2)(2 + \delta_4^2) \sum_{i,j=1}^n \int_{\Omega} w_{y_i y_{jt}}^2(y, t) dy + \\ &+ 2m_3(n^2 + n)(1 + \delta_1^2)(2 + \delta_3^2) \sum_{i=1}^n \int_{\Omega} w_{y_{it}}^2(y, t) dy + \\ &+ 2b_1 m_1(1 + \delta_1^2)(2 + \delta_2^2) [m_1 + (n^2 + n)(m_2 m_5(2 + \delta_4^2) + \\ &+ m_3 m_4(2 + \delta_3^2))] \int_{\Omega} w_{tt}^2(y, t) dy + C_2 \left(\int_0^t \int_{\Omega} w_{\tau\tau}^2(y, \tau) dy d\tau + \right. \\ &\left. + \sum_{i,j=1}^n \int_0^t \int_0^{\tau} \int_{\Omega} w_{y_i y_j \xi \xi}^2(y, \xi) dy d\xi d\tau + \sum_{i=1}^n \int_0^t \int_0^{\tau} \int_{\Omega} w_{y_i \xi \xi}^2(y, \xi) dy d\xi d\tau + \int_0^t \int_0^{\tau} \int_{\Omega} w_{\xi \xi}^2(y, \xi) dy d\xi d\tau \right), \end{aligned}$$

где постоянная C_2 определяется функцией $K(x, y, t)$, а также постоянными $\delta_1, \delta_2, \delta_3$ и δ_4 .

Утверждение доказано.

Ниже нам понадобится второе основное неравенство для эллиптических операторов [11]

$$\sum_{i=1}^n \int_{\Omega} v_{x_i}^2(x, t) dx + \sum_{i,j=1}^n \int_{\Omega} v_{x_i x_j}^2(x, t) dx \leq c_0 \|Av\|_{L_2(\Omega)}^2 + c_1 \|v\|_{L_2(\Omega)}^2, \quad (12)$$

в котором числа c_0 и c_1 определяются лишь оператором A и областью Ω .

Теорема 1. Пусть выполняются условия

$$a^{ij}(x) \in C^1(\bar{\Omega}), a^{ij}(x) = a^{ji}(x), i, j = 1, \dots, n, x \in \bar{\Omega}, K(x, y, t) \in C^3(\bar{\Omega} \times \bar{\Omega} \times [0; T]),$$

$$a^{ij}(x) \xi_i \xi_j \geq k_0 |\xi|^2, k_0 > 0, a(x) \leq -a_0 < 0, \xi \in R^n, x \in \bar{\Omega}, \quad (13)$$

$$b_0 \int_{\Omega} w^2(x, t) dx \leq \int_{\Omega} u^2(x, t) dx \leq b_1 \int_{\Omega} w^2(x, t) dx, b_0 > 0,$$

$$t \in [0; T], u(x, t) \in L_2(0, T; W_2^2(\Omega)) (w = Mu), \quad (14)$$

$$\frac{1}{2a_0} + \frac{1}{2} < \min\left(\frac{1}{4k_1 c_0}, \frac{k_0}{k_2}, \frac{a_0}{4k_3}\right). \quad (15)$$

Тогда краевая задача (1)-(3) имеет решение $u(x, t)$, принадлежащее пространству V , и это решение единственно.

Доказательство. Пусть $g(x, t)$ есть произвольная функция из пространства $L_2(Q)$. Рассмотрим вспомогательную краевую задачу: найти функцию $u(x, t)$, являющуюся в цилиндре Q решением уравнения

$$Lw = g(x, t) + \Phi(x, t, u), w = Mu, \quad (16)$$

удовлетворяющую условиям

$$w(x, t)|_{(x,t) \in S} = 0, \quad (17)$$

$$w(x, 0) = w_t(x, 0) = 0, x \in \Omega. \quad (18)$$

Уравнение (16) представляет собой уравнение "составного типа" относительно функции $u(x, t)$ – его старшая часть представлена в виде суперпозиции двух операторов различных типов. С другой стороны, вследствие взаимной однозначности оператора M , вытекающей из условия (14), уравнение (16) можно рассматривать как уравнение относительно функции $w(x, t)$

$$Lw = g(x, t) + \Phi(x, t, M^{-1}w). \quad (19)$$

Найдя решение $w(x, t)$ данной задачи, т.е. задачи (16) – (18), вновь вследствие взаимной однозначности оператора M , мы сможем найти и собственно функцию $u(x, t)$.

Пусть λ есть число из отрезка $[0, 1]$. Рассмотрим семейство краевых задач: найти функцию $u(x, t)$, являющуюся в цилиндре Q решением краевой задачи

$$Lw = g(x, t) + \lambda \Phi(x, t, u), \quad (16_\lambda)$$

и удовлетворяющую условиям (17), (18).

Обозначим для краткости:

$$F(x, t) = g(x, t) + \lambda \Phi(x, t, u).$$

Обозначим через Λ множество тех чисел λ из отрезка $[0, 1]$, для которых краевая задача (16_λ), (17), (18) имеет решение $w(x, t)$, принадлежащее пространству V для любой функции $g(x, t)$ из $L_2(Q)$. Известно [9], что если множество Λ не пусто, открыто и замкнуто, то оно совпадает со всем отрезком $[0, 1]$.

Непустота множества Λ следует из того, что задача (16₀), (17), (18) разрешима в пространстве V [10].

Открытость и замкнутость Λ следуют из априорной оценки [9]

$$\|w\|_V \leq K.$$

Установим ее наличие.

Умножим уравнение (16_λ), записанное в переменных x и τ , на функцию $-w_\tau(x, \tau)$, результат проинтегрируем по области Ω и по переменной τ в пределах от 0 до t :

$$-\int_0^t \int_\Omega Aw_{\tau\tau}w_\tau dx d\tau = \int_0^t \int_\Omega Bw_\tau w_\tau dx d\tau - \int_0^t \int_\Omega Fw_\tau dx d\tau.$$

Интегрируя по частям и используя неравенство Юнга, а также условия (14) и (15), нетрудно получить следующее неравенство

$$k_0 \sum_{i=1}^n \int_\Omega w_{x_i}^2(x, t) dx + a_0 \int_\Omega w_t^2(x, t) dx \leq P_1 \left(\int_0^t \int_\Omega w_\tau^2(x, \tau) dx d\tau + \sum_{i,j=1}^n \int_0^t \int_\Omega w_{x_i x_j}^2(x, \tau) dx d\tau + \int_0^t \int_\Omega g^2(x, \tau) dx d\tau \right) + \frac{\delta_1^2}{2} \int_0^t \int_\Omega \Phi^2(x, \tau, u) dx d\tau, \quad (20)$$

где число δ_1 – произвольное положительное число, P_1 есть число, определяющееся коэффициентами оператора B , а также областью Ω и числом δ_1 .

Далее, умножим уравнение (16_λ), записанное в переменных x и τ , на функцию $-w_{\tau\tau}(x, \tau)$, результат проинтегрируем по области Ω и по переменной τ в пределах от 0 до t :

$$-\int_0^t \int_\Omega Aw_{\tau\tau}w_{\tau\tau} dx d\tau = \int_0^t \int_\Omega Bw_{\tau\tau}w_{\tau\tau} dx d\tau - \int_0^t \int_\Omega Fw_{\tau\tau} dx d\tau.$$

Интегрируя по частям и оценивая правую сверху, используя неравенства (4), (5), неравенство Юнга, условия (13), (14), получаем следующую оценку

$$k_0 \sum_{i=1}^n \int_0^t \int_\Omega w_{x_i \tau\tau}^2(x, \tau) dx d\tau + \frac{a_0}{2} \int_0^t \int_\Omega w_{\tau\tau}^2(x, \tau) dx d\tau \leq \delta_2^2 \int_0^t \int_\Omega w_{\tau\tau}^2(x, \tau) dx d\tau + P_2 \left(\sum_{i,j=1}^n \int_0^t \int_\Omega w_{x_i x_j}^2(x, \tau) dx d\tau + \int_0^t \int_\Omega g^2(x, \tau) dx d\tau \right) + \frac{1}{2a_0} \int_0^t \int_\Omega \Phi^2(x, \tau, u) dx d\tau. \quad (21)$$

Теперь для получения оценок умножим уравнение (16_λ), записанное в переменных x и τ , на Aw_τ и затем на $Aw_{\tau\tau}$, интегрируя и используя неравенство Юнга, имеем

$$\int_\Omega (Aw_t(x, t))^2 dx \leq \left(1 + \frac{1}{\delta_1^2}\right) \int_0^t \int_\Omega (Aw_\tau(x, \tau))^2 dx d\tau + P_3 \left(\int_0^t \int_\Omega w_{x_i x_j}^2(x, \tau) dx d\tau + \int_0^t \int_\Omega g^2(x, \tau) dx d\tau \right) + \delta_1^2 \int_0^t \int_\Omega \Phi^2(x, \tau, u) dx d\tau, \quad (22)$$

$$\frac{1}{2} \int_0^t \int_\Omega (Aw_{\tau\tau}(x, \tau))^2 dx d\tau \leq \delta_2^2 \int_0^t \int_\Omega (Aw_{\tau\tau}(x, \tau))^2 dx d\tau + P_2 \left(\sum_{i,j=1}^n \int_0^t \int_\Omega w_{x_i x_j}^2(x, \tau) dx d\tau + \int_0^t \int_\Omega g^2(x, \tau) dx d\tau \right) + \frac{1}{2} \int_0^t \int_\Omega \Phi^2(x, \tau, u) dx d\tau. \quad (23)$$

Здесь δ_2 – произвольное положительное число, числа P_2 и P_3 определяются аналогично числу P_1 . Зафиксируем δ_2

$$\delta_2^2 = \min\left(\frac{1}{2}; \frac{a_0}{2}\right).$$

Тогда из (20) – (23) вытекает неравенство

$$\begin{aligned}
 & k_0 \sum_{i=1}^n \int_{\Omega} w_{x_i t}^2(x, t) dx + a_0 \int_{\Omega} w_t^2(x, t) dx + k_0 \sum_{i=1}^n \int_0^t \int_{\Omega} w_{x_i \tau \tau}^2(x, \tau) dx d\tau + \\
 & + \frac{a_0}{4} \int_0^t \int_{\Omega} w_{\tau \tau}^2(x, \tau) dx d\tau + \int_{\Omega} (Aw_t(x, t))^2 dx + \frac{1}{4} \int_0^t \int_{\Omega} (Aw_{\tau \tau}(x, \tau))^2 dx d\tau \leq \\
 & \leq P_4 \left(\int_0^t \int_{\Omega} w_{\tau}^2(x, \tau) dx d\tau + \sum_{i,j=1}^n \int_0^t \int_{\Omega} w_{x_i x_j}^2(x, \tau) dx d\tau + \int_0^t \int_{\Omega} g^2(x, \tau) dx d\tau \right) + \\
 & + \left(1 + \frac{1}{\delta_1^2} \right) \int_0^t \int_{\Omega} (Aw_{\tau}(x, \tau))^2 dx d\tau + \left(\frac{3\delta_1^2}{2} + \frac{1}{2a_0} + \frac{1}{2} \right) \int_0^t \int_{\Omega} \Phi^2(x, \tau, u) dx d\tau.
 \end{aligned}$$

Полученное неравенство можно продолжить, используя утверждение 1

$$\begin{aligned}
 & k_0 \sum_{i=1}^n \int_{\Omega} w_{x_i t}^2(x, t) dx + a_0 \int_{\Omega} w_t^2(x, t) dx + k_0 \sum_{i=1}^n \int_0^t \int_{\Omega} w_{x_i \tau \tau}^2(x, \tau) dx d\tau + \\
 & + \frac{a_0}{4} \int_0^t \int_{\Omega} w_{\tau \tau}^2(x, \tau) dx d\tau + \int_{\Omega} (Aw_t(x, t))^2 dx + \frac{1}{4} \int_0^t \int_{\Omega} (Aw_{\tau \tau}(x, \tau))^2 dx d\tau \leq \\
 & \leq P_4 \left(\int_0^t \int_{\Omega} w_{\tau}^2(x, \tau) dx d\tau + \sum_{i,j=1}^n \int_0^t \int_{\Omega} w_{x_i x_j}^2(x, \tau) dx d\tau + \int_0^t \int_{\Omega} g^2(x, \tau) dx d\tau \right) + \\
 & + \left(1 + \frac{1}{\delta_1^2} \right) \int_0^t \int_{\Omega} (Aw_{\tau}(x, \tau))^2 dx d\tau + \left(\frac{3\delta_1^2}{2} + \frac{1}{2a_0} + \frac{1}{2} \right) \left[k_1 \sum_{i,j=1}^n \int_0^t \int_{\Omega} w_{y_i y_j \tau \tau}^2(y, \tau) dy d\tau + \right. \\
 & k_2 \sum_{i=1}^n \int_0^t \int_{\Omega} w_{y_i \tau \tau}^2(y, \tau) dy d\tau + k_3 \int_0^t \int_{\Omega} w_{\tau \tau}^2(y, \tau) dy d\tau + k_4 \int_0^t \int_{\Omega} w_{\tau}^2(y, \tau) dy d\tau + \\
 & + k_5 \int_0^t \int_{\Omega} w^2(y, \tau) dy d\tau + C_2 \left(\sum_{i,j=1}^n \int_0^t \int_0^{\tau} \int_{\Omega} w_{y_i y_j \xi \xi}^2(y, \xi) dy d\xi d\tau + \right. \\
 & \left. \left. + \sum_{i=1}^n \int_0^t \int_0^{\tau} \int_{\Omega} w_{y_i \xi \xi}^2(y, \xi) dy d\xi d\tau + \int_0^t \int_0^{\tau} \int_{\Omega} w_{\xi \xi}^2(y, \xi) dy d\xi d\tau \right) \right]. \quad (24)
 \end{aligned}$$

Число P_4 определяется числами P_2 и P_3 . Уменьшим число δ_1 так, чтобы выполнялись неравенства

$$\begin{aligned}
 & \frac{a_0}{4} - \left(\frac{3\delta_1^2}{2} + \frac{1}{2a_0} + \frac{1}{2} \right) k_3 > 0, \quad k_0 - k_2 \left(\frac{3\delta_1^2}{2} + \frac{1}{2a_0} + \frac{1}{2} \right) > 0, \\
 & \frac{1}{4} - \left(\frac{3\delta_1^2}{2} + \frac{1}{2a_0} + \frac{1}{2} \right) k_1 c_0 > 0 \quad (25)
 \end{aligned}$$

(это возможно в силу условия (15)).

Отсюда и из неравенства (12) следует, что (24) можно привести к виду

$$\begin{aligned}
 & k_0 \sum_{i=1}^n \int_{\Omega} w_{x_i t}^2(x, t) dx + a_0 \int_{\Omega} w_t^2(x, t) dx + \\
 & \left(\frac{a_0}{4} - \left(\frac{3\delta_1^2}{2} + \frac{1}{2a_0} + \frac{1}{2} \right) k_3 \right) \int_0^t \int_{\Omega} w_{\tau \tau}^2(x, \tau) dx d\tau + \int_{\Omega} (Aw_t(x, t))^2 dx + \frac{1}{4} \int_0^t \int_{\Omega} (Aw_{\tau \tau}(x, \tau))^2 dx d\tau \leq \\
 & \leq P_4 \left(\int_0^t \int_{\Omega} w_{\tau}^2(x, \tau) dx d\tau + \sum_{i,j=1}^n \int_0^t \int_{\Omega} w_{x_i x_j}^2(x, \tau) dx d\tau + \int_0^t \int_{\Omega} g^2(x, \tau) dx d\tau \right) +
 \end{aligned}$$

$$+ \left(1 + \frac{1}{\delta_1^2}\right) \int_0^t \int_{\Omega} (Aw_t(x, \tau))^2 dx d\tau + \left(\frac{3\delta_1^2}{2} + \frac{1}{2a_0} + \frac{1}{2}\right) k_1 c_0 \int_0^t \int_{\Omega} (Aw_{\tau\tau}(x, \tau))^2 dx d\tau + \\ + \left(\frac{3\delta_1^2}{2} + \frac{1}{2a_0} + \frac{1}{2}\right) |k_2 - k_0| c_0 \int_0^t \int_{\Omega} (Aw_{\tau\tau}(x, \tau))^2 dx d\tau.$$

В результате, учитывая неравенство (25), получим

$$k_0 \sum_{i=1}^n \int_{\Omega} w_{x_i t}^2(x, t) dx + a_0 \int_{\Omega} w_t^2(x, t) dx + a_1 \int_0^t \int_{\Omega} w_{\tau\tau}^2(x, \tau) dx d\tau + \\ + \int_{\Omega} (Aw_t(x, t))^2 dx + a_2 \int_0^t \int_{\Omega} (Aw_{\tau\tau}(x, \tau))^2 dx d\tau \leq P_4 \left(\int_0^t \int_{\Omega} w_{\tau}^2(x, \tau) dx d\tau + \right. \\ \left. + \sum_{i,j=1}^n \int_0^t \int_{\Omega} w_{x_i x_j}^2(x, \tau) dx d\tau + \int_0^t \int_{\Omega} g^2(x, \tau) dx d\tau \right) + \left(1 + \frac{1}{\delta_1^2}\right) \int_0^t \int_{\Omega} (Aw_t(x, \tau))^2 dx d\tau.$$

Имеет место представление

$$w_{x_i x_j}(x, \tau) = \int_0^{\tau} w_{x_i x_j \xi}(x, \xi) d\xi = \int_0^{\tau} \int_0^{\xi} w_{x_i x_j \eta \eta}(x, \eta) d\eta d\xi.$$

Из этого равенства, вновь из второго основного неравенства для эллиптических операторов и леммы Гронуолла, следует оценка

$$\int_{\Omega} (Aw_t(x, t))^2 dx + \int_0^t \int_{\Omega} (Aw_{\tau\tau}(x, \tau))^2 dx d\tau \leq K_0.$$

Эта оценка и дает требуемую оценку

$$\|w\|_V \leq K$$

Как уже говорилось выше, этого достаточно для разрешимости краевой задачи (16) – (18). Возьмем функцию $g(x, t)$ специальным образом: $g(x, t) = Mf(x, t)$. Имеем

$$LMu = Mf + \Phi$$

или

$$M(Lu - f) = 0.$$

В силу взаимной однозначности оператора M отсюда следует

$$Lu = f.$$

Выполнение условий (2) и (3) для функции $u(x, t)$ очевидно. Единственность решений также очевидна.

Теорема доказана.

Приведем еще один вариант теоремы о разрешимости задачи (1) – (3).

Теорема 2. Пусть выполняются условия (13) и (14) теоремы 1, и пусть также выполняется условие интегральное уравнение

$$v(x, t) = \lambda \int_{\Omega} K(x, y, t)v(y, t) dy + \varphi(x, t) \tag{14'}$$

однозначно разрешимо в пространстве $L_2(Q)$ для всех t из отрезка $[0, T]$ и для всех λ из отрезка $[0; 1]$, и при этом для всех λ из отрезка $[0; 1]$ выполняется неравенство

$$\|v\|_{L_2(Q)} \leq R_1 \|\varphi\|_{L_2(Q)}$$

с некоторым фиксированным числом R_1 . Тогда краевая задача (1) – (3) имеет решение $u(x, t)$, принадлежащее пространству V , и это решение единственно.

Доказательство. Воспользуемся методом продолжения по параметру.

Пусть λ есть число из отрезка $[0; 1]$. Рассмотрим семейство краевых задач: найти решение урав-

нения (1), для которого выполняются условия (2), а также условие

$$u(x,t)|_{(x,t) \in S} = \lambda \int_{\Omega} K(x,y,t)u(y,t)dy|_{(x,t) \in S}. \quad (3_{\lambda})$$

Обозначим через Λ множество тех чисел λ , для которых краевая задача (1), (2), (3_{λ}) разрешима в пространстве V для произвольной функции $f(x,t)$ из пространства $L_2(Q)$. Если будет доказано, что множество Λ не пусто, открыто и замкнуто (в топологии отрезка $[0,1]$), то оно будет совпадать со всем отрезком $[0,1]$ [9].

Непустота Λ очевидна, т.к. число 0 принадлежит ему. Покажем, что множество Λ будет открытым. Пусть λ_0 есть число из множества Λ . Покажем, что при малых $|\tilde{\lambda}|$ число $\lambda_0 + \tilde{\lambda}$ также будет принадлежать Λ .

Определим множество V_0 :

$$V_0 = \{v(x,t) : v(x,t) \in V, v(x,0) = 0, v_t(x,0) = 0 \text{ при } x \in \Omega\}.$$

Очевидно, что множество V_0 есть подпространство пространства V .

Пусть $v(x,t)$ есть произвольная функция из пространства V_0 . Рассмотрим краевую задачу: найти функцию $u(x,t)$ являющуюся в цилиндре Q решением уравнения (1), такую, что для нее выполняются условия (2), а также условие

$$u(x,t)|_{(x,t) \in S} = \lambda_0 \int_{\Omega} K(x,y,t)u(y,t)dy|_{(x,t) \in S} + \tilde{\lambda} \int_{\Omega} K(x,y,t)v(y,t)dy|_{(x,t) \in S}. \quad (3_{\lambda_0, \tilde{\lambda}})$$

Покажем, что эта задача разрешима в пространстве V .

Пусть $v_0(x,t)$ есть функция, являющаяся решением интегрального уравнения

$$v_0(x,t) - \lambda_0 \int_{\Omega} K(x,y,t)v_0(y,t)dy = \tilde{\lambda} \int_{\Omega} K(x,y,t)v(y,t)dy. \quad (26)$$

Заметим, что функция $v_0(x,t)$ определена корректно, она будет принадлежать пространству V , и при этом будут выполняться равенства $v_0(x,0) = 0, v_{0t}(x,0) = 0$ (вследствие условий (13) и (14)'), а также равенств $v(x,0) = 0, v_t(x,0) = 0$. Положим

$$w = (x,t) = u(x,t) - u_0(x,t), \tilde{f}(x,t) = f(x,t) - Lv_0(x,t).$$

Имеют место равенства

$$Lw = \tilde{f}(x,t), (x,t) \in Q, w(x,0) = w_t(x,0) = 0, x \in \Omega, \\ w(x,t)|_{(x,t) \in S} = \lambda_0 \int_{\Omega} K(x,y,t)w(y,t)dy|_{(x,t) \in S}.$$

Эти равенства представляют собой краевую задачу (1), (2), $(3_{\lambda_0, \tilde{\lambda}})$; поскольку функция $\tilde{f}(x,t)$ принадлежит пространству $L_2(Q)$ и поскольку число λ_0 принадлежит множеству Λ , то данная краевая задача разрешима в пространстве V_0 . Возвращаясь к функции $u(x,t)$, получим, что краевая задача (1), (2), $(3_{\lambda_0, \tilde{\lambda}})$ имеет решение $u(x,t)$, принадлежащее пространству V_0 . Другими словами, данная краевая задача порождает оператор G , переводящий пространство V_0 в себя: $G(v) = u$. Покажем, что при малой величине $|\tilde{\lambda}|$ у оператора G имеются неподвижные точки.

Пусть $v_1(x,t)$ и $v_2(x,t)$ есть произвольные функции из пространства V_0 .

Положим $u_1 = G(v_1), u_2 = G(v_2), u = u_1 - u_2, v = v_1 - v_2$. Имеют место равенства

$$Lu = \tilde{f}(x,t), u(x,0) = u_t(x,0) = 0, x \in \Omega, \\ u(x,t)|_{(x,t) \in S} - \lambda_0 \int_{\Omega} K(x,y,t)u(y,t)dy = \tilde{\lambda} \int_{\Omega} K(x,y,t)v(y,t)dy.$$

Перейдем к функции $w(x,t)$. Имеем

$$Lw(x,t) = -Lv_0(x,t), w(x,0) = w_t(x,0) = 0, x \in \Omega.$$

Справедливо неравенство

$$\|w\|_V \leq R_0 \|Lv_0\|_{L_2(Q)}, \tag{27}$$

в котором число R_0 определяется коэффициентами операторов A и B , а также областью Ω и числом T (доказательство теоремы 1).

Оценим $\|Lv_0\|_{L_2(Q)}$. Имеем

$$v_0(x,t) - \lambda_0 \int_{\Omega} K(x,y,t)v_0(y,t)dy = \tilde{\lambda} \int_{\Omega} K(x,y,t)v(y,t)dy. \tag{28}$$

Из условия (14) и уравнения (26) следует справедливость оценки

$$\|v_0\|_{L_2(Q)} \leq |\tilde{\lambda}| R_2 \|v\|_{L_2(Q)}.$$

Переходя от уравнения (28) к уравнению для производных $v_{0t}(x,t), v_{0x_i}(x,t), v_{0x_i x_j}(x,t)$ и т.д., получим, что аналогичные оценки справедливы и для производных. Отсюда следует, что имеет место неравенство

$$\|v_0\|_V \leq |\tilde{\lambda}| R_3 \|v\|_V. \tag{29}$$

Постоянная R_3 в этом неравенстве определяется числом R_1 , а также коэффициентами операторов A и B , областью Ω и числом T .

Неравенства (27) и (29) дают оценку

$$\|w\|_V \leq |\tilde{\lambda}| R_0 \|v\|_V. \tag{30}$$

Если теперь число $\tilde{\lambda}$ настолько мало, что выполняется неравенство $|\tilde{\lambda}| R_0 < 1$, то оператор G будет сжимающим.

Как известно, сжимающий оператор имеет единственную неподвижную точку. Другими словами, существует функция $u(x,t)$, принадлежащая пространству V_0 и такая, что $G(u) = u$. Очевидно, что для функции $u(x,t)$ выполняются равенства

$$Lu = f(x,t), \quad u(x,0) = u_t(x,0) = 0, \quad x \in \Omega, \\ u(x,t)|_{(x,t) \in S} = (\lambda_0 + \tilde{\lambda}) \int_{\Omega} K(x,y,t)u(y,t)dy|_{(x,t) \in S}.$$

Другими словами, функция $u(x,t)$ является решением краевой задачи (1), (2), $(3_{\lambda_0 + \tilde{\lambda}})$, принадлежащим пространству V . А это и означает, что число $\lambda_0 + \tilde{\lambda}$ при выполнении указанного выше неравенства $|\lambda| R_0 < 1$, будет принадлежать множеству Λ , и далее – что множество Λ открыто.

Докажем, что множество Λ замкнуто.

Пусть $\{\lambda_m\}$ есть последовательность точек из множества Λ такая, что $\lambda_m \rightarrow \lambda_0$ при $m \rightarrow \infty$. Покажем, что число λ_0 также принадлежит Λ .

Поскольку каждое число λ_m принадлежит Λ , то существует функция $u_m(x,t)$, принадлежащая пространству V и являющаяся решением задачи (1), (2), (3_{λ_m}) . Обозначим $w_{mk}(x,t) = u_m(x,t) - u_k(x,t)$. Для функций $w_{mk}(x,t)$ имеют место равенства

$$Lw_{mk} = 0, \quad w_{mk}(x,0) = w_{mkt}(x,0) = 0, \quad x \in \Omega, \\ w_{mk}(x,t)|_{(x,t) \in S} = \lambda_m \int_{\Omega} K(x,y,t)w_{mk}(y,t)dy|_{(x,t) \in S} + (\lambda_m - \lambda_k) \int_{\Omega} K(x,y,t)u_k(y,t)dy|_{(x,t) \in S}.$$

Повторяя доказательство неравенства (30), получим оценку

$$\|w_{mk}\|_V \leq R_0 |\lambda_m - \lambda_k|.$$

Отсюда следует, что последовательность $\{u_m(x,t)\}$ будет фундаментальной в пространстве V . Следовательно, существует функция $u(x,t)$, принадлежащая пространству V и являющаяся пределом последовательности $\{u_m(x,t)\}$. Для предельной функции $u(x,t)$ будут выполняться уравнение (1), условие (2), а также условие (3_{λ_0}) . А это означает, что число λ_0 будет принадлежать множеству Λ .

Принадлежность множеству Λ любой его предельной точки и означает, что Λ – замкнуто. Итак, множество Λ не пусто, открыто и замкнуто. Как уже говорилось выше, множество Λ будет совпадать со всем отрезком $[0; 1]$. А это означает, что краевая задача (1) – (3) разрешима в пространстве V .

Теорема доказана.

Замечание. Отметим, что в теореме 1 требуется однозначная разрешимость интегрального уравнения

$$v(x,t) = \lambda \int_{\Omega} K(x,y,t)v(y,t)dy + \varphi(x,t)$$

лишь при $\lambda = 1$, а в теореме 2 – при всех λ из отрезка $[0; 1]$, но при этом в первой теореме требуются условия малости (15), во второй таких условий нет.

Литература

1. Пулькина Л.С. Нелокальная задача для уравнения теплопроводности // Неклассические уравнения математической физики. – Новосибирск: ИМ СО РАН, 2005. – С. 231 – 239.
2. Пулькина Л.С. Нелокальная задача с двумя интегральными условиями для гиперболического уравнения на плоскости // Неклассические уравнения математической физики. – Новосибирск: ИМ СО РАН, 2007. – С. 232 – 236.
3. Кожанов А.И., Пулькина Л.С. Краевые задачи с интегральным граничным условием для многомерных гиперболических уравнений // ДАН. 2005. – Т.404. – №5. – С.589 – 592.
4. Кожанов А.И., Пулькина Л.С. О разрешимости краевой задачи с нелокальным граничным условием интегрального вида для многомерных гиперболических уравнений // Дифференц. уравнения. – 2006. – Т.42. – №9. – С. 1166 – 1179.
5. Абдрахманов А.М., Кожанов А.И. Задача с нелокальным граничным условием для одного класса уравнений нечетного порядка // Изв. вузов. Математика. – 2007. – №5. – С. 3 – 12.
6. Абдрахманов А.М. О разрешимости краевой задачи с интегро-дифференциальным граничным условием для некоторых классов уравнений составного типа // Мат. заметки ЯГУ. – 2011. – Т. 18. – Вып. 2. – С. 3 – 10.
7. Лукина Г.А. Краевые задачи с интегральными граничными условиями для ультрапараболических уравнений // Мат. заметки ЯГУ. – 2011. – Т. 18. – Вып. 2. – С. 113 – 127.
8. Демиденко Г.В., Успенский С.В. Уравнения и системы, не разрешенные относительно старшей производной. – Новосибирск: Научная книга, 1998. – 456 с.
9. Треногин В.А. Функциональный анализ. – М.: Наука, 1980.
10. Якубов С.Я. Линейные дифференциально-операторные уравнения и их приложения. – Баку: Элм, 1985.
11. Ладыженская О.А., Уральцева Н.Н. Линейные и квазилинейные уравнения эллиптического типа. – М.: Наука, 1973.

Алсыкова Аюна Андреевна, аспирант кафедры математического анализа и методики преподавания математики Бурятского государственного университета, e-mail: 888552@mail.ru

Alsykova Ayuna Andreevna, postgraduate student, department of mathematical analysis and teaching methodology of mathematics, Buryat State University, e-mail: 888552@mail.ru

УДК 517.98

© А.Д. Мижидон, С.Г. Баргуев

КРАЕВАЯ ЗАДАЧА ДЛЯ ОДНОЙ ГИБРИДНОЙ СИСТЕМЫ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ¹

В статье рассматривается краевая задача для гибридной системы дифференциальных уравнений, содержащих сингулярности типа δ -функций. Такого класса гибридная система дифференциальных уравнений имеет место при описании динамики механических систем с сосредоточенными и распределенными параметрами, представляющих собой упругий стержень с закрепленными краями и прикрепленными на нем с помощью пружин твердыми телами. В качестве теоретических основ исследования предлагается единый подход к построению частотных уравнений таких систем.

Ключевые слова: гибридная система дифференциальных уравнений, краевая задача, обобщенное решение, частотное уравнение.

© A.D. Mizhidon, S.G. Barguev

BOUNDARY VALUE PROBLEM FOR ONE HYBRID SYSTEM OF DIFFERENTIAL EQUATIONS

The article is devoted to the boundary value problem for hybrid system of differential equations containing δ -functions. This class of hybrid systems of differential equations takes place in the description of dynamics of mechanical systems with lumped and distributed parameters. These mechanical systems are an elastic rod with fixed edges and solids attached at it by springs. As a theoretical basis of research a unified approach to the construction of frequency equations of such systems has been proposed.

Keywords: hybrid system of differential equations, boundary value problem, generalized solution, frequency equation.

Введение

При исследовании механических колебаний элементов различных конструкций, деталей и механизмов во многих случаях расчетными схемами исследования является твердое тело (или система твердых тел), соединенное упругими связями со стержнем [1-6]. Для вывода уравнений движения систем используется вариационный принцип Гамильтона, который справедлив как для систем с сосредоточенными, так и для систем с распределенными параметрами. Полученные на основании принципа Гамильтона уравнения движений для приведенных выше механических систем являются гибридными системами дифференциальных уравнений [1-6]. Под гибридными системами дифференциальных уравнений понимается система дифференциальных уравнений, состоящая из обыкновенных дифференциальных уравнений и уравнений в частных производных.

Для примера рассмотрим механическую систему, состоящую из массы m , установленной с помощью двух пружин жесткости c_1 и c_2 на упругом стержне, длины l (рис. 1). Масса m может перемещаться поступательно в направлении оси O_1z и совершать угловые отклонения φ . Перемещение точек стержня описывается функцией $u(x,t)$.

¹ Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 12-08-00309 а

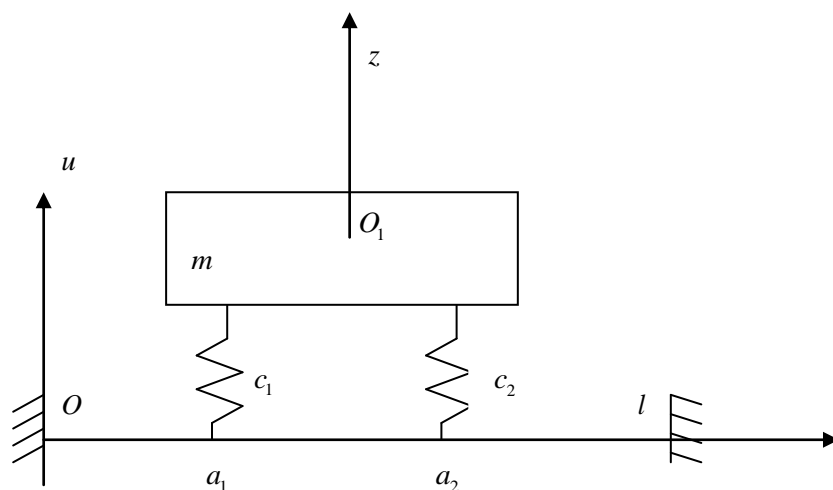


Рис. 1. Расчетная схема механической системы, состоящей из массы, установленной на упругом стержне

Полученные на основании принципа Гамильтона для этой механической системы уравнения движений имеют вид

$$\begin{cases} m\ddot{z} + c_1(z - d_1\varphi - u(a_1, t)) + c_2(z + d_2\varphi - u(a_2, t)) = 0, \\ J_\varphi\ddot{\varphi} - c_1d_1(z - d_1\varphi - u(a_1, t)) + c_2d_2(z + d_2\varphi - u(a_2, t)) = 0, \\ \rho F \frac{\partial^2 u}{\partial t^2} + EJ \frac{\partial^4 u}{\partial x^4} = c_1(z - d_1\varphi - u(x, t))\delta(x - a_1) + c_2(z + d_2\varphi - u(x, t))\delta(x - a_2), \end{cases}$$

где J_φ - момент инерции твердого тела относительно центра масс, при повороте на угол φ ; d_1 - расстояние от центра масс до оси пружины, закрепленной в точке a_1 ; d_2 - расстояние от центра масс до оси пружины, закрепленной в точке a_2 ; ρ - плотность материала стержня; F - площадь поперечного сечения; E - модуль упругости стержня; J - момент инерции поперечного сечения стержня относительно нейтральной оси сечения, перпендикулярной к плоскости колебаний.

На функцию $u(x, t)$, необходимо наложить граничные условия, соответствующие условиям, накладываемым на правый и левый конец стержня. В случае жесткой заделки граничные условия имеют вид

$$u(0, t) = u(l, t) = 0, \quad \frac{\partial u}{\partial x}(0, t) = \frac{\partial u}{\partial x}(l, t) = 0. \quad (4)$$

В данной работе рассматривается краевая задача для одного класса гибридных систем дифференциальных уравнений, описывающих в общем случае динамику механических систем, представляющих собой упругий стержень с закрепленными краями и прикрепленными на нем с помощью пружин твердыми телами. Под краевой задачей понимается задача нахождения обобщенного решения гибридной системы дифференциальных уравнений, удовлетворяющей краевым условиям (4).

1. Постановка задачи

Рассмотрим гибридную систему дифференциальных уравнений

$$\begin{cases} A\ddot{z} + Bz + C(Dz - \bar{u}) = 0, \\ k \frac{\partial^2 u}{\partial t^2}(x, t) + b \frac{\partial^4 u}{\partial x^4}(x, t) = \sum_{i=1}^m q_i (d^i z(t) - u(x, t))\delta(x - a_i), \end{cases} \quad (5)$$

где $z(t)$ - n -мерная вектор-функция; $u(x, t)$ - скалярная функция; $\bar{u}(t)$ - m -мерная вектор-функция с компонентами $u(a_1, t), \dots, u(a_m, t)$; A, B - заданные, постоянные $n \times n$ - матрицы; C - за-

данная, постоянная $n \times m$ – матрица; D – заданная, постоянная $m \times n$ – матрица; d^i – n – мерный вектор, составленный из строк матрицы D ; $k, b, a_i, q_i, (i = \overline{1, m})$ – заданные постоянные, причем $0 \leq a_i \leq l$; штрих $()'$ – здесь и ниже операция транспонирования.

Отметим, система (4)-(5) является общей математической моделью механических систем, представляющих собой упругий стержень с закрепленными краями и прикрепленными на нем с помощью упругих связей системой твердых тел, соединенных между собой упругими связями.

Введем понятие обобщенного решения гибридной системы дифференциальных уравнений (5), удовлетворяющего краевым условиям (4).

Для этого рассмотрим множество вектор-функций

$$K = \left\{ (y(\cdot), v(\cdot, \cdot)) : y(\cdot) \in C_{\infty, [0, T]}^n, v(\cdot, \cdot) \in C_{\infty, D}^2 \right\}, \quad (6)$$

где $D = \{(x, t) \in R^2 : 0 \leq x \leq l, 0 \leq t \leq T\}$ – прямоугольник.

Отметим, в случае рассмотрения краевой задачи для механических систем, представляющих собой упругий стержень с закрепленными краями и прикрепленными на нем с помощью пружин твердыми телами, класс основных функций можно трактовать как допустимые вариации обобщенных координат в принципе Гамильтона.

Потребуем, чтобы любая вектор-функция из множества функций K удовлетворяла условиям

$$\begin{aligned} y(0) = y(T) = 0, \\ v(x, 0) = v(x, T) = 0, \\ v(0, t) = v(l, t) = 0, \quad \frac{\partial v}{\partial x}(0, t) = \frac{\partial v}{\partial x}(l, t) = 0. \end{aligned} \quad (7)$$

Введенные вектор-функции назовем основными.

Определение 1. Вектор-функцию $z(\cdot) \in C_{2, [0, T]}^n$, скалярную функцию $u(\cdot, \cdot) \in C_{4, 2, D}^2$ назовем обобщенным решением краевой задачи (4)-(5), если для любой основной вектор-функции $(y(\cdot), v(\cdot, \cdot)) \in K$ имеет место тождество

$$\begin{aligned} & \int_0^T (A\ddot{z} + Bz + C(Dz - \bar{u}), y(t)) dt + \\ & + \iint_D \left(k \frac{\partial^2 u}{\partial t^2} + b \frac{\partial^4 u}{\partial x^4} - \sum_{i=1}^m q_i (d^i z(t) - u(x, t)) \delta(x - a_i) \right) \cdot v(x, t) dx dt = 0 \end{aligned}$$

2. Вспомогательная краевая задача

Подставив в систему (5) $z(t), u(x, t)$ в виде

$$z(t) = Z \sin \omega t, \quad u(x, t) = V(x) \sin \omega t,$$

где ω – собственная частота, Z – n – мерный вектор амплитуд колебаний масс, $V(x)$ – амплитуда колебаний точек упругого стержня, после преобразований получим

$$(-\omega^2 A + B + CD)Z - C\bar{V} = 0, \quad (8)$$

$$-\omega^2 kV(x) + b \frac{d^4 V(x)}{dx^4} = \sum_{i=1}^m q_i (d^i Z - V(x)) \delta(x - a_i), \quad (9)$$

где \bar{V} – m – мерный вектор с компонентами $V(a_1), \dots, V(a_m)$.

В силу граничных условий (4) функция $V(x)$ должна удовлетворять условиям

$$V(0) = V(l) = 0, \quad \frac{dV}{dx}(0) = \frac{dV}{dx}(l) = 0. \quad (10)$$

Рассмотрим вспомогательную краевую задачу (9)-(10).

Определение 2. Скалярную функцию $V(\cdot) \in C_{4,[0,T]}$ назовем обобщенным решением краевой задачи (9)-(10), если для любой компоненты $v(\cdot, \cdot)$ основной вектор-функции $(y(\cdot), v(\cdot, \cdot)) \in K$, имеет место тождество при любом $t \in [0, T]$

$$\int_0^l \left(-\omega^2 k V(x) + b \frac{d^4 V(x)}{dx^4} - \sum_{i=1}^m q_i (d^{i'} Z - V(x)) \delta(x - a_i) \right) \cdot v(x, t) dx = 0.$$

Теорема 1. При любых значениях ω и Z функция

$$V(x) = \sum_{i=1}^m G_i(x - a_i) q_i (d^{i'} Z - V(a_i)) \quad (11)$$

является обобщенным решением краевой задачи (9)-(10), где функции $G_i(x)$, $(i=1, \dots, m)$ – обобщенные решения уравнения

$$-\omega^2 k G_i(x) + b \frac{d^4 G_i(x)}{dx^4} = \delta(x), \quad (i=1, \dots, m), \quad (12)$$

удовлетворяющие краевым условиям

$$\begin{cases} G_i(-a_i) = G_i(l - a_i) = 0, \\ \frac{dG_i}{dx}(-a_i) = \frac{dG_i}{dx}(l - a_i) = 0, \end{cases} \quad (i=1, \dots, m) \quad (13)$$

Доказательство: Для функции (11) справедливость выполнения краевых условий (10) непосредственно следует из краевых условий (13) для функций $G_i(x)$, $(i=1, \dots, m)$.

В том, что (11) является решением уравнения (9) убедимся непосредственной подстановкой (11) в исходное уравнение (9).

Для этого представим (11) в виде

$$V(x) = \sum_{i=1}^m \int_0^l \left(G_i(x - \xi) q_i (d^{i'} Z - V(\xi)) \delta(\xi - a_i) \right) d\xi. \quad (14)$$

Подставим (14) в левую часть уравнения (9), умножив на $v(x, t)$ из класса основных функций, проинтегрируем по x в пределах от 0 до l . Далее, меняя порядок интегрирования и учитывая (12), получим

$$\begin{aligned} & \int_0^l \left[\int_0^l \sum_{i=1}^m \left[\left(-k\omega^2 G_i(x - \xi) + b \frac{d^4 G_i(x - \xi)}{dx^4} \right) q_i (d^{i'} Z - V(\xi)) \delta(\xi - a_i) \right] d\xi \right] \cdot v(x, t) dx = \\ & = \sum_{i=1}^m \int_0^l \left[q_i (d^{i'} Z - V(\xi)) \delta(\xi - a_i) \cdot \int_0^l \left(-k\omega^2 G_i(x - \xi) + b \frac{d^4 G_i(x - \xi)}{dx^4} \right) v(x, t) dx \right] d\xi = \\ & = \sum_{i=1}^m \int_0^l \left[q_i (d^{i'} Z - V(\xi)) \delta(\xi - a_i) \cdot \int_0^l v(x, t) \delta(x - \xi) dx \right] d\xi = \\ & = \sum_{i=1}^m \int_0^l \left[q_i (d^{i'} Z - V(\xi)) v(\xi, t) \delta(\xi - a_i) \right] d\xi = \sum_{i=1}^m \left[q_i (d^{i'} Z - V(a_i)) v(a_i, t) \right]. \end{aligned}$$

Аналогично, подставив (14) в правую часть уравнения (9), после преобразований получим

$$\begin{aligned} & \int_0^l \left[\sum_{i=1}^m q_i (d^{i'} Z - V(x)) \delta(x - a_i) \right] v(x, t) dx = \\ & = \sum_{i=1}^m \int_0^l q_i (d^{i'} Z - V(x)) \delta(x - a_i) v(x, t) dx = \sum_{i=1}^m \left[q_i (d^{i'} Z - V(a_i)) v(a_i, t) \right], \end{aligned}$$

т.е. в результате проделанных преобразований левая и правая части исходного уравнения тождественно совпадают. Таким образом, выражение (11) является решением уравнения (9) в обобщенном смысле.

Теорема доказана.

Замечание 1. Для нахождения функций $G_1(x), G_2(x), \dots, G_m(x)$, входящих в (11), имеем m краевых задач для уравнения

$$-\omega^2 kG(x) + b \frac{d^4 G(x)}{dx^4} = \delta(x), \tag{15}$$

с условиями (13).

Общее решение $G(x)$ уравнения (15) можно найти в виде суммы общего обобщенного решения $G_0(x)$ однородного уравнения

$$-\omega^2 kG(x) + b \frac{d^4 G(x)}{dx^4} = 0, \tag{16}$$

и некоторого обобщенного решения $\tilde{G}(x)$ неоднородного уравнения (15), т. е.

$$G(x) = G_0(x) + \tilde{G}(x). \tag{17}$$

Общее решение $G_0(x)$ однородного уравнения (16) можно записать в виде

$$G_0(x) = c_1 S_1(\beta x) + c_2 S_2(\beta x) + c_3 S_3(\beta x) + c_4 S_4(\beta x),$$

где c_1, c_2, c_3, c_4 – произвольные постоянные; $S_1(\beta x), S_2(\beta x), S_3(\beta x), S_4(\beta x)$ – функции Крылова, которые определяются следующим образом:

$$S_1(\beta x) = \frac{\text{ch}(\beta x) + \cos(\beta x)}{2}, \quad S_2(\beta x) = \frac{\text{sh}(\beta x) + \sin(\beta x)}{2},$$

$$S_3(\beta x) = \frac{\text{ch}(\beta x) - \cos(\beta x)}{2}, \quad S_4(\beta x) = \frac{\text{sh}(\beta x) - \sin(\beta x)}{2}.$$

Здесь $\beta = \sqrt[4]{\frac{\omega^2 k}{b}}$. Отметим, что выражение для обобщенного решения $G_0(x)$ совпадает с классическим решением.

Частное обобщенное решение $\tilde{G}(x)$ неоднородного уравнения (15) можно определить в виде [6]

$$\tilde{G}(x) = \theta(x) \frac{kS_4(\beta x)}{b\beta^3}, \tag{18}$$

где $\theta(x)$ – функция, определяемая следующим образом:

$$\theta(x) = \begin{cases} 1, & x > 0, \\ 0, & x < 0. \end{cases}$$

2. Частотные уравнения

Для нахождения функций $G_1(x), G_2(x), \dots, G_m(x)$, согласно представлению (17), определим произвольные константы c_1, c_2, c_3, c_4 из условий выполнения соответствующих граничных условий.

Далее, принимая в (11) последовательно значения $x = a_1, x = a_2, \dots, x = a_m$, получим систему линейных алгебраических уравнений относительно $V(a_1), V(a_2), \dots, V(a_m)$

$$(1 + G_j(0)q_j V(a_j) + \sum_{\substack{i=1, \\ i \neq j}}^m G_i(a_j - a_i)q_i V(a_i)) = \sum_{i=1}^m G_i(a_j - a_i)q_i d^{i'} Z, \quad (j = 1, \dots, m). \tag{19}$$

Используя матричные обозначения, систему (19) можно записать в виде

$$M\bar{V} = NZ,$$

где M – матрица системы размерности $m \times m$:

$$M = \begin{pmatrix} 1 + G_1(0)q_1 & G_2(a_1 - a_2)q_2 & \dots & G_m(a_1 - a_m)q_m \\ G_1(a_2 - a_1)q_1 & 1 + G_2(0)q_2 & \dots & G_m(a_2 - a_m)q_m \\ \dots & \dots & \dots & \dots \\ G_1(a_m - a_1)q_1 & G_2(a_m - a_2)q_2 & \dots & 1 + G_m(0)q_m \end{pmatrix},$$

N – матрица размерности $m \times n$

$$N = \begin{pmatrix} \sum_{i=1}^m G_i(a_1 - a_i)q_i d_1^i & \sum_{i=1}^m G_i(a_1 - a_i)q_i d_2^i & \dots & \sum_{i=1}^m G_i(a_1 - a_i)q_i d_n^i \\ \sum_{i=1}^m G_i(a_2 - a_i)q_i d_1^i & \sum_{i=1}^m G_i(a_2 - a_i)q_i d_2^i & \dots & \sum_{i=1}^m G_i(a_2 - a_i)q_i d_n^i \\ \dots & \dots & \dots & \dots \\ \sum_{i=1}^m G_i(a_m - a_i)q_i d_1^i & \sum_{i=1}^m G_i(a_m - a_i)q_i d_2^i & \dots & \sum_{i=1}^m G_i(a_m - a_i)q_i d_n^i \end{pmatrix},$$

\bar{V} – m -мерный вектор с компонентами $V(a_1), V(a_2), \dots, V(a_m)$.

Таким образом, решение линейной системы алгебраических уравнений (17) представим в виде

$$\bar{V} = M^{-1}NZ \tag{20}$$

Подставив (20) в систему (8), получим систему линейных, однородных алгебраических уравнений относительно вектора амплитуд Z

$$(-\omega^2 A + B + CD - CM^{-1}N) Z = 0 \tag{21}$$

Система (21) имеет ненулевые решения, если ее определитель равен нулю. Приравняв определитель системы (21) к нулю, получим уравнение собственных частот

$$\det(-\omega^2 A + B + CD - CM^{-1}N) = 0. \tag{22}$$

Отметим, уравнение собственных частот (22) является трансцендентным, содержащим периодические тригонометрические функции, а также монотонные гиперболические функции, в силу чего является бесконечный дискретный набор собственных частот.

3. Сравнительный анализ предложенного подхода

Применение предложенного подхода для построения частотного уравнения механической системы, состоящей из массы m , установленной с помощью двух пружин жесткости c_1 и c_2 на упругом стержне (рис. 1), привело к уравнению вида

$$\begin{aligned} & (-m\omega^2 + c_1 + c_2 - \alpha c_1 - \mu c_2)(-J_\phi \omega^2 + c_1 d_1^2 + c_2 d_2^2 + c_1 d_1 \gamma - c_2 d_2 \nu) = \\ & = (-c_1 d_1 + c_2 d_2 + c_1 d_1 \alpha - c_2 d_2 \mu)(-J_\phi \omega^2 + c_1 d_1^2 + c_2 d_2^2 + c_1 d_1 \gamma - c_2 d_2 \nu), \end{aligned} \tag{23}$$

где

$$\alpha = \frac{\mu_1 \beta_2 - \mu_2 \beta_1}{\Delta}, \quad \gamma = \frac{\nu_1 \beta_2 - \nu_2 \beta_1}{\Delta}, \quad \mu = \frac{\alpha_1 \mu_2 - \alpha_2 \mu_1}{\Delta}, \quad \nu = \frac{\alpha_1 \nu_2 - \alpha_2 \nu_1}{\Delta},$$

$$\Delta = \alpha_1 \beta_2 - \alpha_2 \beta_1, \quad \alpha_1 = 1 + e_1 G_1(0), \quad \alpha_2 = e_1 G_1(a_2 - a_1),$$

$$\beta_1 = e_2 G_2(a_1 - a_2), \quad \beta_2 = 1 + e_2 G_2(0),$$

$$\mu_1 = e_1 G_1(0) + e_2 G_2(a_1 - a_2), \quad \mu_2 = e_1 G_1(a_2 - a_1) + e_2 G_2(0),$$

$$\nu_1 = -e_1 d_1 G_1(0) + e_2 d_2 G_2(a_1 - a_2), \quad \nu_2 = -e_1 d_1 G_1(a_2 - a_1) + e_2 d_2 G_2(0),$$

$$e_1 = \frac{c_1}{\rho F}, \quad e_2 = \frac{c_2}{\rho F}.$$

Замечание 2. В данной задаче для нахождения $G_1(x), G_2(x)$, согласно замечанию 1, принимается $k = \rho F$ и $b = EJ$.

Для проведения сравнительного анализа, с целью проверки адекватности предлагаемой методики получения собственных частот, был произведен расчет собственных частот согласно уравнению (23). Для этого были взяты данные из работы [7], в которой для получения собственных частот автором был использован метод допускаемых мод или форм, разработанный в [8]. Полученные им результаты сравнивались с результатами, полученными в работе [9], где авторы использовали аналитико-численно-комбинированный метод (АЧКМ) и метод конечных элементов (МКЭ).

Данные для стержня: модуль упругости $E = 2,069 \times 10^{11} \text{ N} \cdot \text{m}^{-2}$, плотность материала стержня $\rho = 15,3875 \text{ kg} \cdot \text{m}^{-3}$, момент инерции поперечного сечения $J = 3,0679 \times 10^{-7} \text{ m}^4$, длина $l = 1,0 \text{ m}$. Для упруго присоединенной упругой системы: масса твердого тела $m = 1,53875 \text{ kg}$, момент инерции твердого тела относительно центра масс $J_{\varphi} = 1,53875 \text{ kg} \cdot \text{m}^2$, жесткость пружин $c_1 = c_2 = 6,34761 \times 10^6 \text{ N} \cdot \text{m}^{-1}$, координаты точек крепления пружин a_1, a_2 соответственно расстояния от точек крепления пружин до центра масс $d_1 = 0,06667 \text{ m}$, $d_2 = 0,13333 \text{ m}$ соответственно.

В таблице 1 приведены результаты проведенного расчета согласно уравнению частот (23), а также результаты расчета тремя методами, приведенными в [7]. Из таблицы видно хорошее совпадение полученных результатов.

Таблица 1

$\omega_i, \text{ rad} \cdot \text{s}^{-1}$	АЧКМ	МКЭ	Подход Philip D.Cha [7]	Предлагаемый подход
ω_1	273.8904	273.8565	273.8892	273.8564
ω_2	1388.6244	1388.5937	1388.6073	1388.5914
ω_3	2880.5511	2879.7694	2880.0323	2879.7628
ω_4	4222.2172	4221.9181	4221.9610	4221.8472
ω_5	7837.1068	7837.4548	7836.9696	7836.9522

Заключение

Предложен единый подход к построению частотных уравнений для одного класса динамических систем, описываемых гибридной системой дифференциальных уравнений. Рассматриваемая система дифференциальных уравнений является общей математической моделью механических систем, представляющих собой упругий стержень с закрепленными краями и прикрепленными на нем с помощью упругих связей системой твердых тел, соединенных между собой упругими связями. Сравнительный анализ численных расчетов, проведенных предложенным методом с расчетами, проведенными другими способами, известными из литературы, показал достоверность и универсальность предлагаемого подхода.

Литература

1. Миждон А.Д., Баргуев С.Г. О вынужденных колебаниях механической системы, установленной на упругом стержне // Современные технологии. Системный анализ. Моделирование. – 2004. – №1. – С. 32-34.
2. Миждон А.Д., Баргуев С.Г., Лебедева Н.В. К исследованию виброзащитной системы с упругим основанием // Современные технологии. Системный анализ. Моделирование. – 2009. – №2(22). – С. 13-203.
3. Баргуев С.Г., Миждон А.Д. Определение собственных частот простейшей механической системы на упругом основании // Вестн. Бурят. гос. ун-та. – Вып. 9. Математика и информатика. – 2009. – С. 58-66.
4. Исследование возможности гашения колебаний n масс, установленных на упругом стержне / С.Г. Баргуев, Е.В. Елтошкина, А.Д. Миждон, М.Ж. Цыцыренова // Современные технологии. Системный анализ. Моделирование. – 2010. – №4(28). – С. 78-84.
5. Баргуев С.Г., Миждон А.Д., Цыцыренова М.Ж. О пределах применимости классической схемы расчета собственных частот в виброзащитной системе с двумя защищаемыми объектами // Вестн. Бу-

рят. гос. ун-та. – Вып. 9. – Математика и информатика. – 2010. – С. 135-144.

6. Мижидон А.Д., Баргуев С.Г. О собственных колебаниях механической системы каскадного типа, установленной на упругом стержне // Вестн. Вост.-Сиб. гос. технологического ун-та. –2010. – № 1. – С. 26-33.

7. D. Cha Philip, Free vibrations of a uniform beam with multiple elastically mounted two-degree-of-freedom systems, Journal of Sound and Vibration 307 (2007), P. 386-392.

8. Meirovitch L., Fundamental of Vibrations, McGraw-Hill Companies, New York, 2001.

9. Wu J.-J., Whittaker A.R., The natural frequencies and mode shapes of a uniform cantilever beam with multiple two-DOF spring-mass systems, Journal of Sound and Vibration 227(1999), P. 386-392. 361-381.

Мижидон Арсалан Дугарович, доктор технических наук, заведующий кафедрой Восточно-Сибирского государственного университета технологий и управления, тел. (902-5) 633204, e-mail: miarsdu@esstu.ru

Баргуев Сергей Ганжурович (Гаврилович), кандидат физико-математических наук, заведующий кафедрой Бурятского филиала Сибирского государственного университета телекоммуникаций и информатики, тел. (950-3) 808172, e-mail: barguev@yandex.ru

Mizhidon Arsalan Dugarovich, doctor of technical sciences, professor, head of applied mathematics department, East-Siberian State University of Technology and Management., ph. (902-5) 633204, e-mail: miarsdu@esstu.ru

Barguev Sergey Ganzhurovich (Gavriilovich), candidate of physical and mathematical sciences, head of the department, Buryat Branch of the Siberian State University of Telecommunications and Informatics, ph. (950-3) 808172, e-mail: barguev@yandex.ru

УДК 519.622

© А.Е. Новиков, Е.А. Новиков, Л.В. Кнауб

КОНТРОЛЬ УСТОЙЧИВОСТИ МЕТОДА ЧЕСКИНО ВТОРОГО ПОРЯДКА ТОЧНОСТИ¹

Построено неравенство для контроля устойчивости метода Ческино второго порядка точности. Приведены результаты расчетов жестких задач, подтверждающие повышение эффективности алгоритма за счет контроля устойчивости.

Ключевые слова: явные методы, контроль точности и устойчивости, жесткие задачи.

© А.Е. Novikov, E.A. Novikov, L.V. Knaub

STABILITY CONTROL FOR A SECOND ORDER CESCHINO'S METHOD

The inequation for stability control for a Ceschino's method of second order is constructed. Calculation results of stiff problems modeling are given, they confirm a growth of algorithm efficiency at the expense of stability control.

Keywords: explicit methods, control of accuracy and stability, stiff problems.

Введение

При численном решении задачи Коши для жестких систем обыкновенных дифференциальных уравнений большой размерности возникает необходимость использования алгоритмов на основе явных методов [1–3]. Методы интегрирования на основе неявных или полужявных численных схем, как правило, используют обращение матрицы Якоби [2]. В данном случае это отдельная трудоемкая задача. В такой ситуации предпочтительнее применять алгоритмы на основе явных формул, если жесткость задачи позволяет за разумное время получить приближение к решению [3].

Обычно алгоритм управления шагом интегрирования строится на контроле точности численной схемы. Это естественно, потому что основным критерием является точность нахождения решения. Однако при применении алгоритмов интегрирования на основе явных формул для решения жестких задач данный подход приводит к потере эффективности и надежности [4]. Это связано с тем, что на участке установления противоречие между точностью и устойчивостью приводит к большому количеству повторных вычислений решения, а шаг выбирается значительно меньше максимально допустимого. Этого можно избежать дополнительным контролем устойчивости численной схемы. В настоящее время можно выделить два подхода к контролю устойчивости [3, 5]. Первый связан с оценкой максимального собственного числа матрицы Якоби через ее норму с последующим контролем (наряду с точностью) неравенства $h\|f_y\| \leq D$ [5], где положительная постоянная D связана с размером области устойчивости. Ясно, что для явных методов, где матрица Якоби не участвует в вычислительном процессе, это приводит дополнительно к ее нахождению и, следовательно, к увеличению вычислительных затрат. Второй подход основан на оценке максимального собственного числа λ_{\max} матрицы Якоби степенным методом через приращения правой части системы дифференциальных уравнений с последующим контролем неравенства $h\|\lambda_{\max}\| \leq D$ [3]. Во всех рассмотренных ситуациях такая оценка не приводит к увеличению вычислительных затрат [4].

Здесь с применением предложенного в [6] способа оценки максимального собственного числа матрицы Якоби построено неравенство для контроля устойчивости метода Ческино второго порядка точности [7]. Приведены результаты расчетов, подтверждающие повышение эффективности алгоритма интегрирования за счет контроля устойчивости.

1. Метод Ческино

Рассмотрим задачу Коши для системы обыкновенных дифференциальных уравнений

$$y' = f(t, y), \quad y(t_0) = y_0, \quad t_0 \leq t \leq t_k, \quad (1)$$

¹Работа поддержана РФФИ (проекты 11-01-00106 и 11-01-00224)

где y и f – N -мерные вещественные вектор-функции, t – независимая переменная, которая изменяется на заданном интервале $[t_0, t_k]$. Для решения (1) будем использовать явные формулы типа Рунге-Кутты следующего вида

$$y_{n+1} = y_n + \sum_{i=1}^4 p_{mi} k_i$$

$$k_1 = hf(t_n, y_n), \quad k_2 = hf\left(t_n + \frac{1}{4}h, y_n + \frac{1}{4}k_1\right), \quad (2)$$

$$k_3 = hf\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right), \quad k_4 = hf\left(t_n + h, y_n + k_1 - 2k_2 + 2k_3\right),$$

где h – шаг интегрирования, k_i , $1 \leq i \leq 4$, – стадии метода, p_{mi} , $1 \leq i \leq 4$, – числовые коэффициенты, m – порядок точности метода. При значениях коэффициентов

$$p_{21} = 1, \quad p_{22} = -2, \quad p_{23} = 2, \quad p_{24} = 0 \quad (3)$$

схема (2), (3) имеет второй порядок точности [7].

2. Контроль точности вычислений

Согласно [7], численная формула (2) с коэффициентами

$$p_{41} = \frac{1}{6}, \quad p_{42} = 0, \quad p_{43} = \frac{2}{3}, \quad p_{44} = \frac{1}{6} \quad (4)$$

имеет четвертый порядок. Тогда для контроля точности схемы второго порядка можно использовать оценку ошибки δ_n вида

$$\delta_n = \sum_{i=1}^4 (p_{4i} - p_{2i}) k_i. \quad (5)$$

В результате для контроля точности вычислений применяется неравенство

$$\|\delta_n\| \leq \varepsilon, \quad (6)$$

где $\|\cdot\|$ – некоторая норма в R^N , ε – требуемая точность расчетов. Имеет место соотношение $\delta_n = O(h^3)$, шаг h^{ac} по точности выбирается по формуле $h^{ac} = qh$, где значение q находится из уравнения $q^3 \|\delta_n\| = \varepsilon$. Если $q < 1$, то происходит повторное вычисление решения (возврат) с шагом h , равным qh . В противном случае вычисляется приближенное решение, а прогнозируемый шаг вычисляется по формуле $h_{n+1} = qh$. Неравенство (6) хорошо зарекомендовало себя при решении многих практических задач, и ниже будет использоваться здесь. В дальнейшем алгоритм переменного шага на основе схемы (2) с коэффициентами (3), (4) и неравенством для контроля точности (6) назовем CESCH42.

3. Контроль устойчивости численной схемы

Построим неравенство для контроля устойчивости схемы (2). Для этого применим (2) для решения линейной задачи $y' = Ay$ с постоянной матрицей A . Первые три стадии k_1 , k_2 и k_3 схемы (2), применительно к данной задаче, имеют вид

$$k_1 = Xy_n, \quad k_2 = \left(X + \frac{1}{4}X^2\right)y_n, \quad k_3 = \left(X + \frac{1}{2}X^2 + \frac{1}{8}X^3\right)y_n,$$

где $X = hA$. Нетрудно видеть, что имеют место соотношения

$$k_1 - 2k_2 + k_3 = \frac{1}{8}X^3y_n, \quad 0.5(k_2 - k_1) = \frac{1}{8}X^2y_n.$$

Теперь оценку максимального собственного числа матрицы Якоби системы (1) можно вычислить степенным методом. Введем обозначение

$$v_n = 2 \cdot \max_{1 \leq i \leq N} \left\{ \frac{|(k_1 - 2k_2 + k_3)_i|}{|(k_2 - k_1)_i|} \right\}. \quad (7)$$

Тогда, согласно [6], для контроля устойчивости метода Ческино можно применять неравенство

$$v_n \leq D, \quad (8)$$

где постоянное число D ограничивает интервал устойчивости.

Устойчивость методов типа Рунге-Кутты обычно исследуется на скалярном тестовом уравнении

$$y' = \lambda y, \tag{9}$$

где λ есть произвольное комплексное число, $\text{Re}(\lambda) < 0$. Смысл λ – некоторое собственное число матрицы Якоби задачи (1). Так как λ есть произвольное комплексное число с отрицательной вещественной частью, то для устойчивых задач все собственные числа матрицы Якоби системы (1) попадают в (9). Применяя (2) с коэффициентами (3) для решения (9), получим, что функция устойчивости $Q_2(x)$ метода второго порядка точности имеет вид

$$Q_2(x) = 1 + x + \frac{1}{2}x^2 + \frac{1}{4}x^3, \quad x = h\lambda.$$

Область устойчивости метода второго порядка приведена на рис. 1 (приведены линии уровня $|Q_2(x)|=q$ при q равном 0.3, 0.7 и 1.0).

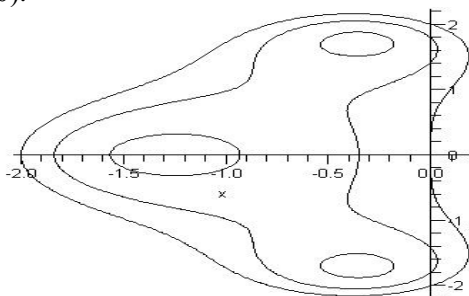


Рис. 1. Область устойчивости метода второго порядка точности

Из рис. 1 следует, что длина интервала устойчивости метода второго порядка точности (2) приблизительно равна 2. Функция устойчивости $Q_4(x)$ метода четвертого порядка точности имеет вид

$$Q_4(x) = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4, \quad x = h\lambda.$$

Область устойчивости метода четвертого порядка приведена на рис. 2. Интервал устойчивости метода четвертого порядка приблизительно равен 2.8. Поэтому в неравенстве (8) положим $D=2$. Учитывая, что $v_n = O(h)$, шаг h^{st} по устойчивости можно выбирать по формуле $h^{st} = rh$, где r вычисляется из равенства $rv_n = D$.

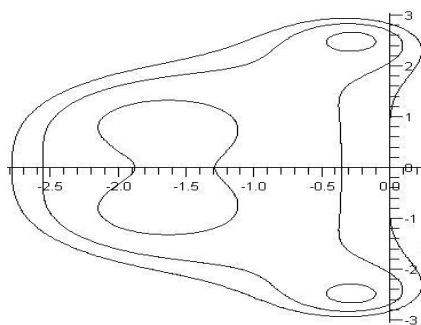


Рис. 2. Область устойчивости метода четвертого порядка точности

Оценка (7) является грубой, потому что вовсе не обязательно максимальное собственное число сильно отделено от остальных, в степенном методе применяется мало итераций, и дополнительные искажения вносит нелинейность задачи (1). Поэтому здесь контроль устойчивости используется как ограничитель на размер шага интегрирования. В результате прогнозируемый шаг вычисляется по формуле

$$h_{n+1} = \max[h_n, \min(h^{ac}, h^{st})], \tag{10}$$

где h_n есть последний успешный шаг интегрирования. Отметим, что формула (10) применяется для прогноза величины шага интегрирования h_{n+1} после успешного вычисления решения с предыдущим шагом h_n , и поэтому фактически не приводит к увеличению вычислительных затрат. Если шаг по устойчивости меньше последнего успешного, то он уменьшен не будет, потому что причиной этого

может быть грубость оценки максимального собственного числа. Однако шаг не будет и увеличен, потому что не исключена возможность неустойчивости численной схемы. Если шаг по устойчивости должен быть уменьшен, то в качестве следующего шага будет применяться последний успешный шаг h_n . В результате для выбора шага и предлагается формула (10). Данная формула позволяет стабилизировать поведение шага на участке установления решения, где определяющую роль играет устойчивость. Именно наличие данного участка существенно ограничивает возможности применения явных методов для решения жестких задач.

В дальнейшем алгоритм переменного шага с дополнительным контролем устойчивости численной схемы будем называть CESCH42st. Данный алгоритм основан на численной формуле невысокого (второго) порядка точности, и поэтому он ориентирован на решение нежестких задач с небольшой точностью расчетов (порядка 1% и ниже), а также задач умеренной жесткости. Отметим, что в зависимости от интервала интегрирования и размерности системы дифференциальных уравнений одна и та же задача может быть отнесена к нежесткой, средней жесткости или жесткой.

4. Дифференциальные уравнения химической кинетики

Кинетическая схема химической реакции состоит из элементарных стадий, записываемых в виде

$$\sum_{i=1}^{N_r} \alpha_{ij} x_i \rightarrow \sum_{i=1}^{N_s} \beta_{ij} x_i, \quad (11)$$

где x_i , $1 \leq i \leq N_r$ – реагенты; N_r и N_s – соответственно число реагентов и число стадий в реакции; α_{ij} и β_{ij} , $1 \leq i \leq N_r$, $1 \leq j \leq N_s$ – стехиометрические коэффициенты. Для каждой элементарной реакции заданы константы скоростей стадии k_j , $1 \leq j \leq N_s$. Процессу (11) в рамках сосредоточенной модели изотермического реактора постоянного объема соответствует система обыкновенных дифференциальных уравнений

$$C' = A^T V \quad (12)$$

с заданным начальным условием $C(0) = C_0$. Здесь A^T – стехиометрическая матрица, C и V – соответственно вектор концентраций реагентов и скоростей стадий. В случае протекания реакции в неизотермических условиях к данной системе добавляется уравнение теплового баланса

$$T' = \frac{Q^T V - \alpha(T - T_{01})}{C_V^T C}.$$

где T – температура смеси в реакторе, T_{01} – температура стенок реактора, Q^T – вектор удельных теплот стадий, C_V^T – вектор теплоемкостей реагентов, $\alpha = \dot{a}s/r$, \dot{a} – коэффициент теплоотдачи, s и r – площадь поверхности и объем реактора соответственно. Верхний индекс T у векторов Q^T и C_V^T означает транспонирование. Теплоемкости реагентов и коэффициент теплоотдачи могут быть функциями концентраций реагентов c_i , $1 \leq i \leq N_r$, а α также может еще зависеть от температуры.

Если реакция протекает в изотермическом реакторе постоянного объема с обменом вещества (открытая система, реактор идеального смешения), система дифференциальных уравнений записывается в виде

$$C' = A^T V + \frac{1}{\Theta} (C_p - C),$$

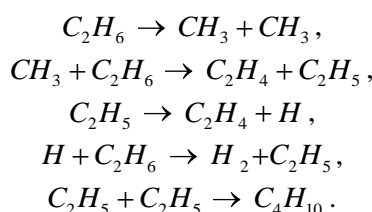
где C_p – вектор концентраций реагентов на входе в реактор, Θ – время пребывания смеси в реакторе, $\Theta = r/u$, u – объемная скорость течения смеси через реактор. При протекании реакции в неизотермических условиях эта система дополняется уравнением теплового баланса

$$T' = \frac{Q^T V - \alpha(T - T_{01})}{C_V^T C} - \frac{1}{\Theta} (T - T_{02}),$$

где T_{02} – температура смеси на входе в реактор. Температура реагирующей смеси может задаваться в виде функции времени t и концентраций c_i , $1 \leq i \leq N_r$, т.е. $T = T(t, C)$.

5. Численное моделирование пиролиза этана

Пиролиз этана в отсутствие кислорода описывается небольшой последовательностью стадий. Механизм пиролиза этана неоднократно обсуждался в литературе. Здесь принята схема реакции, предложенная и исследованная в работе [8]



Здесь константы скоростей стадий имеют вид

$$\begin{aligned} k_1 &= 1.34 \cdot 10^{-5}, \quad k_2 = 3.73 \cdot 10^2, \quad k_3 = 3.69 \cdot 10^3, \\ k_4 &= 3.66 \cdot 10^5, \quad k_5 = 1.62 \cdot 10^7. \end{aligned}$$

Обозначим концентрации реагентов следующим образом:

$$\begin{aligned} c_1 &= [C_2H_6], \quad c_2 = [CH_3], \quad c_3 = [CH_4], \quad c_4 = [C_2H_5], \\ c_5 &= [C_2H_4], \quad c_6 = [H], \quad c_7 = [H_2], \quad c_8 = [C_4H_{10}]. \end{aligned}$$

Тогда соответствующая система состоит из 8 обыкновенных дифференциальных уравнений вида

$$\begin{aligned} c_1' &= -k_1c_1 - k_2c_1c_2 - k_4c_1c_6, \\ c_2' &= 2k_1c_1 - k_2c_1c_2, \quad c_3' = k_2c_1c_2, \\ c_4' &= k_2c_1c_2 - k_3c_4 + k_4c_1c_6 - 2k_5c_4^2, \\ c_5' &= k_3c_4, \quad c_6' = k_3c_4 - k_4c_1c_6, \\ c_7' &= k_4c_1c_6, \quad c_8' = k_5c_4^2. \end{aligned} \tag{13}$$

Начальная концентрация этана $c_1=[C_2H_6]$ равна 0.14, для остальных реагентов концентрации равны нулю.

Расчеты осуществлялись на PC Intel(R) Core(TM) i7-3770S CPU@3.10GHz с двойной точностью с задаваемой точностью $\epsilon=10^{-2}$. Эффективность алгоритмов интегрирования оценивалась по числу вычислений правой части if задачи (13) на интервале интегрирования. Численное решение осуществлялось на промежутке $[0,0.26]$ с начальным шагом $h=10^{-5}$. Данная задача удовлетворяет ”классическому” определению жесткости. В начале интервала интегрирования наблюдается переходный участок (сотые доли секунды), а затем происходит медленное установление.

Сравнение эффективности построенного алгоритма (CESCH42st) проводилось с известным методом Мерсона (MERSION) [9]. Для обоих методов фактическая точность не хуже задаваемой точности. Алгоритму CESCH42st для нахождения решения потребовалось 20 403 вычислений правой части задачи (13), а для метода Мерсона $if=25\ 796$.

Заключение

Из результатов расчетов можно сделать следующие выводы. Во-первых, построенный алгоритм интегрирования второго порядка с контролем точности вычислений и устойчивости численной схемы можно применять для решения достаточно жестких задач. Во-вторых, по вычислительным затратам алгоритм CESCH42st эффективнее метода Мерсона. Это является следствием контроля устойчивости численной схемы. Представляется, что при достаточно большой размерности задачи (1) метод CESCH42st может конкурировать с неявными методами на задачах умеренной жесткости, потому что в нем не обращается матрица Якоби. При решении двенадцати тестовых задач [2] и десяти примеров [10] преимущество алгоритма CESCH42st значительно выше. В частности, при моделировании модифицированного орегонатора [11] преимущество CESCH42st по сравнению с методом Мерсона более чем в полтора раза.

Литература

1. Хайрер Э., Нерсетт С., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи. – М.: Мир, 1990. – 512 с.
2. Хайрер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи. – М.: Мир, 1999. – 685 с.
3. Новиков Е.А. Явные методы для жестких систем. – Новосибирск: Наука, 1997. – 197 с.
4. Новиков Е.А., Шорников Ю.В. Компьютерное моделирование жестких гибридных систем. –

Новосибирск: Изд-во НГТУ. 2012. – 450 с.

5. Shampine L.M. Implementation of Rosenbrock methods // ACM Transaction on Mathematical Software. – 1982. №5. P. 93–113.

6. Новиков Е.А., Новиков В.А. Контроль устойчивости явных одношаговых методов интегрирования обыкновенных дифференциальных уравнений // ДАН СССР. – Т. 27. – №5. – 1984. – С. 1058–1062.

7. Ceschino F., Kuntzman J. Numerical solution of initial value problems. – New Jersey: Prentice-Hall, Englewood Clis, 1966. – 318 p.

8. Kulich D.M., Taylor J.E. Mathematical simulation of the oxygen ethane reaction // J. Chem. Kinetic. 1975. V. 8. P. 89–97.

9. Merson R.H. An operational methods for integration processes // Australia: Proc. of Symp. on Data Processing. 1957. P. 329–331.

10. Enright W.H., Hull T.E. Comparing numerical methods for the solutions of systems of ODE's // ВПТ. – 1975. – №15. – P. 10–48.

11. Новиков Е.А. Численное моделирование модифицированного орегонатора (2,1)-методом решения жестких задач // Вычислительные методы и программирование. 2010. –Т.11. – №2. – С. 123–130.

Новиков Антон Евгеньевич, преподаватель кафедры Сибирского федерального университета, тел. +7(391)2495979. E-mail: aenovikov@bk.ru.

Новиков Евгений Александрович, главный научный сотрудник Института вычислительного моделирования СО РАН, тел. +7(391)2494724. E-mail: novikov@icm.krasn.ru.

Кнауб Людмила Владимировна, доцент кафедры Сибирского федерального университета, тел. +7(391)2912213. E-mail: aenovikov@bk.ru

Novikov Anton Evgenevich, lecturer, Siberian Federal University, ph. +7(391)2495979. E-mail: aenovikov@bk.ru.

Novikov Evgeny Alexandrovich, chief researcher, Institute of Calculation Modeling SB RAS, ph. +7(391)2494724. E-mail: novikov@icm.krasn.ru.

Knaub Lyudmila Vladimirovna, associate professor, Siberian Federal University, ph. +7(391)2912213. E-mail: aenovikov@bk.ru

УДК 517.948

© Г.А. Шишкин

ФУНКЦИЯ ГИБКОЙ СТРУКТУРЫ И ЕЕ МОДИФИКАЦИЯ ПРИ РЕШЕНИИ КРАЕВЫХ ЗАДАЧ ДЛЯ УРАВНЕНИЙ С ФУНКЦИОНАЛЬНЫМ ЗАПАЗДЫВАНИЕМ

В статье, используя стандартную форму функции гибкой структуры, применяемую при решении начальных задач функциональных уравнений, исследуется возможность построения ее аналога для решения краевых задач.

Ключевые слова: функция гибкой структуры, функциональные запаздывания, краевая задача, разрешающие уравнения.

© G.A. Shishkin

FUNCTION OF FLEXIBLE STRUCTURE AND ITS UPDATING AT THE SOLUTION OF BOUNDARY VALUE PROBLEMS FOR EQUATIONS WITH FUNCTIONAL DELAY

Using the standard form of function of flexible structure applied at the solution of initial problems of functional equations, the article researches a possibility of construction of its analogue for solution of boundary value problems.

Keywords: function of flexible structure, functional delays, boundary value problem, resolving equations.

Введение

В работе [4] рассматривались начальные задачи для линейных интегро-дифференциальных уравнений Вольтерра с запаздывающим аргументом запаздывающего, нейтрального и опережающего типов. Определение типов в этой работе, как и в следующих, осуществлялось в соответствии с классификацией, приведенной в работе [1].

Проведено исследование возможностей преобразования начальных задач для линейных интегро-дифференциальных уравнений Вольтерра с запаздывающим аргументом с помощью функции гибкой структуры к интегральным уравнениям с обыкновенным аргументом. Доказано, что задача Коши для всех интегро-дифференциальных уравнений Вольтерра с отклоняющимся аргументом запаздывающего типа с помощью функции гибкой структуры преобразуется к разрешающему интегральному уравнению типа Вольтерра с обыкновенным аргументом, решение которого существует и притом единственное – при выполнении условий ограниченности функций, входящих в уравнение. Рассмотрены возможности решения в замкнутом виде и вариант приближенного решения, если точное решение найти затруднительно.

В данной работе исследуем вопрос о возможности построения аналога функции гибкой структуры для решения краевых задач функциональных уравнений.

Постановка краевой задачи

Выпишем общий вид линейных интегро-дифференциальных уравнений Вольтерра с запаздывающим аргументом

$$\sum_{j=0}^l \sum_{i=0}^n [f_{ij}(x)y^{(i)}(u_j(x)) + \lambda \int_a^x K_{ij}(x,\eta)y^{(i)}(u_j(\eta))d\eta] = f(x), \tag{1}$$

где $u_0(x) \equiv x$, $u_j(x) \leq x$, $u_j(x) \neq x$, $j = \overline{1, l}$, $f_{ij}(x)$, $f(x)$ и $u_j(x)$ - непрерывны, ядра $K_{ij}(x,\eta)$ - регуляры в квадрате $a \leq x, \eta \leq b$.

с начальными функциями

$$y^{(i)}(u_j(x)) = y^{(i)}(x_0)\varphi^{(i)}(u_j(x)), \quad i = \overline{0, n-1}, \quad x \in E_{x_0}, \tag{2}$$

где $E_{x_0} = \bigcup_{j=0}^l E_{x_0}^j$, $E_{x_0}^j$ - множество точек, для которых соответствующие $u_j(x) \leq x$ при $x \geq x_0 \quad \forall j = \overline{1, l}$, а

$E_{x_0}^0 = [a, x_0]$, функции $\varphi_i(x)$ - заданы и $\varphi_i(x_0) = 1, \quad \forall i = \overline{0, n-1}$.

Рассмотрим уравнение (1) с линейными билокальными краевыми условиями

$$\sum_{i=0}^{n-1} [\alpha_{i\tau} y^{(i)}(x_0) + \beta_{i\tau} y^{(i)}(x_1)] = \gamma_\tau, \quad (3)$$

где $\tau = \overline{0, n-1}$, $a \leq x_0 < x_1 \leq b$, α_i и β_i - постоянные числа.

Функция гибкой структуры для решения начальных задач

В работе [4] для уравнения (1) исследуются возможности решения начальных задач, применив для преобразований одну из модификаций функции гибкой структуры [3]

$$y^{(i)}(u_j(x)) = D^{-1} \left[\sum_{s=1}^n y^{(s-1)}(x_0) \frac{d^i \Delta_s(u_j(x) - x_0)}{dx^i} + \int_{x_0}^{u_j(x)} \frac{\partial^i \Delta_n(u_j(x) - t)}{\partial x^i} \mu(t) dt \right] + \gamma_i u_j^n(x) \mu(u_j(x)), \quad (4)$$

где $i = \overline{0, n}$, $D = D(r_1, r_2, \dots, r_n)$ - определитель Вандермонда, составленный из неопределенных параметров r_1, r_2, \dots, r_n , которые определяются в ходе решения задачи исходя из оптимальности ее решения. Определители $\Delta_s(x-t), s = \overline{1, n}$ получаются из определителя D заменой s -ой строки строкой $\exp r_1(x-t), \exp r_2(x-t), \dots, \exp r_n(x-t)$, $\mu(x)$ - новая неизвестная функция и $\gamma_n = 1, \gamma_i = 0 \quad \forall i = \overline{0, n-1}$.

Т.к. любую непрерывную n раз дифференцируемую функцию можно представить в виде функции с гибкой структурой [2]-[3], то преобразования, выполненные с ее помощью, приведут к разрешающему интегральному уравнению, эквивалентному первоначально поставленной задаче. За счет оптимального выбора параметров функции гибкой структуры можно получать решения в замкнутой форме, а если это не удастся, то ускорять процесс приближенного решения и влиять на объем вычислений.

Параметры в определителе D могут быть и равными, в этом случае пределы выражений $D^{-1} \frac{\partial^i \Delta_s(u_j(x) - t)}{\partial x^i}$ имеют вполне определенные значения, вычисляемые по правилу Лопиталья.

Наименьший из корней уравнений $u_j(x) = x_0$ на отрезке $x \in [x_0, b]$ обозначим через c_j , если таких нет, то принимаем $c_j = b$.

Построение модификации функции гибкой структуры для решения краевых задач

Используя начальные функции (2), краевые условия (3) и функцию гибкой структуры и ее производные (4) определим $y^{(i)}(x_0)$ через новую неизвестную функцию $\mu(x)$, при этом могут возникнуть три возможные ситуации:

1. $x_0 < x_1 \leq c_j \quad \forall j = \overline{0, l}$;
2. $x_0 < c_j \leq x_1 \quad \forall j = \overline{0, l}$;
3. x_1 таково, что $\exists j = \overline{0, l}$, что для некоторых выполняется $x_0 < x_1 \leq c_j$, а для других $x_0 < c_j \leq x_1$.

1. Первый случай наиболее простой, т.к., подставив $x = x_1$ в начальные функции (2) при $j = 0$ и затем значения $y^{(i)}(x_1) = y^{(i)}(x_0) \varphi_i(x_1)$ в краевые условия (3), получим алгебраическую систему

$$\begin{cases} \sum_{i=0}^{n-1} y^{(i)}(x_0)[\alpha_{i\tau} + \beta_{i\tau}\varphi_i(x_1)] = \gamma_\tau, \\ \tau = \overline{0, n-1} \end{cases} \quad (5)$$

для определения значений $y^{(i)}(x_0)$.

Обозначив через ω главный определитель этой системы

$\omega = \det[\alpha_{i\tau} + \beta_{i\tau}\varphi_i(x_1)]$, $i, \tau = \overline{0, n-1}$ и через $\omega_{i\tau}$ – алгебраические дополнения к элементам главного определителя по формулам Крамера найдём

$$y^{(i)}(x_0) = \omega^{-1} \sum_{\tau=0}^{n-1} \gamma_\tau \omega_{i\tau}, \quad i = \overline{0, n-1}. \quad (6)$$

Краевая задача в этом случае свелась к решению начальной задачи, рассмотренной в [4], только при ее решении в формулах функции гибкой структуры значения $y^{(i)}(x_0)$ следует заменить в соответствии с формулой (6). Тогда начальные функции для краевых задач в этом случае примут вид

$$\varphi_i^*(u_j(x)) = \varphi_i(u_j(x)) \omega^{-1} \sum_{\tau=0}^{n-1} \gamma_\tau \omega_{i\tau}. \quad (2^*)$$

Также получим новые формулы и для функции гибкой структуры, пронумеровав их теми же номерами, что и первоначально выписанные формулы, но пометим эти номера звёздочкой

$$\begin{aligned} y^{(i)}(u_j(x)) = D^{-1} \left[\sum_{s=1}^n \omega^{-1} \sum_{\tau=0}^{n-1} \gamma_\tau \omega_{(s-1)\tau} \Delta_s^{(i)}(u_j(x) - x_0) + \right. \\ \left. + \int_{x_0}^{u_j(x)} \frac{\partial^i \Delta_n(u_j(x) - t)}{\partial x^i} \mu(t) dt \right] + \gamma_i u_j^{(n)}(x) \mu(u_j(x)), \end{aligned} \quad (4^*)$$

где $i = \overline{0, n}$, $\gamma_n = 1$, $\gamma_i = 0 \quad \forall i = \overline{0, n-1}$, $j = \overline{0, l}$.

2. Во втором и третьем случаях, применив функцию гибкой структуры (2) и её производные (3)-(4), выразим значения $y^{(i)}(x_1)$, $i = \overline{0, n-1}$ через новую неизвестную функцию $\mu(x)$ и начальные значения искомой функции $y^{(i)}(x_0)$

$$\begin{aligned} y^{(i)}(x_1) = D^{-1} \left[\sum_{s=1}^n y^{(s-1)}(x_0) \Delta_s^{(i)}(x_1 - x_0) + \right. \\ \left. + \int_{x_0}^{x_1} \frac{\partial^i \Delta_n(x_1 - t)}{\partial x^i} \mu(t) dt \right], \quad i = \overline{0, n-1}. \end{aligned}$$

Подставив полученные выражения $y^{(i)}(x_1)$ в краевые условия (3)

$$\sum_{i=0}^{n-1} \{ \alpha_{i\tau} y^{(i)}(x_0) + \beta_{i\tau} D^{-1} [\sum_{s=1}^n y^{(s-1)}(x_0) \Delta_s^{(i)}(x_1 - x_0) + \int_{x_0}^{x_1} \frac{\partial^i \Delta_n(x_1 - t)}{\partial x^i} \mu(t) dt] \} = \gamma_\tau, \quad \tau = \overline{0, n-1}$$

и, перегруппировав слагаемые, придём к системе алгебраических уравнений относительно $y^{(i)}(x_0)$

$$\begin{aligned} \sum_{i=0}^{n-1} [\alpha_{i\tau} + \beta_{i\tau} D^{-1} \Delta_{i+1}^{(i)}(x_1 - x_0)] y^{(i)}(x_0) = \\ = \gamma_\tau - D^{-1} \sum_{i=0}^{n-1} \beta_{i\tau} \int_{x_0}^{x_1} \frac{\partial^i \Delta_n(x_1 - t)}{\partial x^i} \mu(t) dt, \quad \tau = \overline{0, n-1}. \end{aligned} \quad (7)$$

Обозначив главный определитель системы (7) через ω

$$\omega = \det [\alpha_{i\tau} + \beta_{i\tau} D^{-1} \Delta_{i+1}^{(i)}(x_1 - x_0)], \quad i, \tau = \overline{0, n-1},$$

а алгебраические дополнения к элементам главного определителя через $\omega_{i\tau}$, по формулам Крамера найдём

$$y^{(i)}(x_0) = \sum_{\tau=0}^{n-1} \frac{\omega_{i\tau}}{\omega} [\gamma_{\tau} D^{-1} \sum_{k=0}^{n-1} \beta_{k\tau} \int_{x_0}^{x_1} \frac{\partial^k \Delta_n(x_1-t)}{\partial x^k} \mu(t) dt], \quad i = \overline{0, n-1}. \quad (8)$$

В соответствии с (8) начальные функции (2) и функции гибкой структуры (4) для краевой задачи в случаях **2** и **3** примут вид

$$y^{(i)}(u_j(x)) = \varphi_i(u_j(x)) \sum_{\tau=0}^{n-1} \frac{\omega_{i\tau}}{\omega} [\gamma_{\tau} D^{-1} \sum_{k=0}^{n-1} \beta_{k\tau} \int_{x_0}^{x_1} \frac{\partial^k \Delta_n(x_1-t)}{\partial x^k} \mu(t) dt], \quad (2^{**})$$

$$i = \overline{0, n-1}, \quad j = \overline{0, l}, \quad x \in E_{x_0},$$

$$y^{(i)}(u_j(x)) = D^{-1} \left\{ \sum_{s=1}^n \frac{\partial^i \Delta_s(u_j(x) - x_0)}{\partial x^i} \sum_{\tau=0}^{n-1} \frac{\omega_{s\tau}}{\omega} [\gamma_{\tau} - D^{-1} \cdot \sum_{k=0}^{n-1} \beta_{k\tau} \int_{x_0}^{x_1} \frac{\partial^k \Delta_n(x_1-t)}{\partial x_1^k} \mu(t) dt] + \int_{x_0}^{u_j(x)} \frac{\partial^i \Delta_n(u_j(x) - t)}{\partial x^i} \mu(t) dt \right\} + \gamma_i u_j'^n(x) \mu(u_j(x)). \quad (4^{**})$$

где $i = \overline{0, n}$, $\gamma_n = 1$, $\gamma_i = 0 \quad \forall i = \overline{0, n-1}$, $j = \overline{0, l}$.

Полученные выше формулы начальных функций и функции гибкой структуры с ее производными для краевой задачи в случаях **2** и **3** помечены теми же номерами, что и первоначально выписанные формулы, но с двумя звездочками.

Заключение

Функции гибкой структуры, построенные в данной статье, могут быть использованы при исследовании и решении краевых задач для различных видов функциональных уравнений, как линейных, так и нелинейных.

В следующих работах предполагается исследовать возможности построения для краевых задач дифференциальных и интегро-дифференциальных уравнений с функциональным запаздыванием разрешающих уравнений с обыкновенным аргументом, а в дальнейшем рассмотреть возможности оптимизации нахождения точных или приближенных решений за счет оптимального выбора параметров функции гибкой структуры и разработать программы решения поставленных задач на ЭВМ. Этому будут посвящены дальнейшие исследования и разработки программ.

Литература

1. Громова П.С. Некоторые вопросы качественной теории интегро-дифференциальных уравнений с отклоняющимся аргументом / П.С. Громова // Труды семинара по теории дифференциальных уравнений с отклоняющимся аргументом. Т.5. – М., 1967. – С. 61-76.
2. Куликов Н.К. Инженерный метод решения и исследования обыкновенных дифференциальных уравнений / Н.К. Куликов. – М.: Высшая школа, 1964. – 207 с.
3. Куликов Н.К. Решение и исследование обыкновенных дифференциальных уравнений на основе функций с гибкой структурой / Н.К. Куликов // Тематический сб. МТИПП. – М., 1974. – С. 47-57.
4. Шишкин Г.А. Линейные интегро-дифференциальные уравнения Вольтерра с функциональным запаздыванием. – Улан-Удэ: Издательство БГУ. – 2009. – 64 с.

Шишкин Геннадий Александрович, кандидат физико-математических наук, профессор кафедры прикладной математики Бурятского государственного университета. Тел. (3012)217733.

Shishkin Gennady Alexandrovich, candidate of physical and mathematical sciences, professor, department of applied mathematics, Buryat State University.

4. Математическое моделирование

УДК 51-73

© *В.И. Баранов, Н.Н. Голенко, Л.А. Компаниец, В.Т. Пака, Т.В. Якубайлик*

ПРОСТРАНСТВЕННО-ВРЕМЕННАЯ ИЗМЕНЧИВОСТЬ ОСНОВНЫХ ХАРАКТЕРИСТИК ОЗЕРА ШИРА В СЕЗОНЕ НАБЛЮДЕНИЙ 2011-2012 ГГ.¹

В работе обсуждаются новые данные, полученные в результате натурных наблюдений летом 2012 г. на озере Шира, которые дополняют исследования в течение трех летних сезонов 2009-2011 гг. Описываются приборы, которыми проводились измерения, приводятся результаты натурных измерений.

Ключевые слова: стратифицированное озеро, натурные наблюдения, гидрологические характеристики.

© *V.I. Baranov, N.N. Golenko, L.A. Kompaniets, V.T. Paika, T.V. Yakubaylik*

SPATIAL AND TEMPORAL VARIABILITY OF THE MAIN CHARACTERISTICS OF LAKE SHIRA IN THE SEASON OF OBSERVATIONS 2011-2012

This paper discusses the new data obtained from field observations in the summer of 2012 at Lake Shira, which supplement research over three summer seasons of 2009-2011. The instruments used for measurements are described, as well as the results of field measurements.

Keywords: stratified lake, field observations, hydrological characteristics.

Введение

В 2012 г. продолжались натурные измерения основных гидрологических характеристик озера Шира (максимальная глубина 25 м) [1]. При этом основное внимание уделялось измерениям, позволяющим воссоздать картину непрерывного распределения определяющих величин в вертикальных разрезах, например вдоль малой оси озера. В частности, были проведены измерения по нескольким разрезам таких величин, как температура, соленость, плотность, скорость течения, а также измерения колебаний теплозапаса в фиксированной точке разреза, предположительно обусловленных колебаниями термоклина. При анализе временных рядов и измерений на разрезах были использованы метеоданные с ближайшей к оз. Шира метеостанции.

1. Методы измерений основных гидрологических величин

Измерения температуры и солености озера проводились CTD зондом YSI (Д.Ю. Рогозин, сотрудник ИБФ СО РАН, канд. физ.-мат. наук) и зондом IDRONAUT (В.И. Баранов, сотрудник АОИО РАН им. П.П. Ширшова). Положение разреза в 2011 г. показано на рис. 1.

¹Работа выполнена при финансовой поддержке междисциплинарного интеграционного проекта СО РАН №56.



Рис. 1. Точки гидрологических измерений в 2011 г.

Измерения вектора скорости течения проводились профилографами ADCP Workhorse Sentinel 600 кГц и 1200 кГц, коммутированными с полевым компьютером Panasonic Touchbook с установленным программным обеспечением для сбора, контроля, накопления и обработки информации. Измерения производились путем буксировки ADCP с борта маломерного судна с подвесным мотором, обеспечивающим необходимый для измерений малый ход. За день 4 августа 2012 г. было пройдено 8 промерных галсов с расстоянием примерно 1 км между ними. На рис. 2 показаны галсы буксировки ADCP, а также положение гидрологических разрезов 2012 г., которое практически совпадало с галсами буксировки ADCP. Галс, соответствующий номеру 5 в 2012 г., почти совпадает с разрезом, на котором проводились измерения в 2011 г.

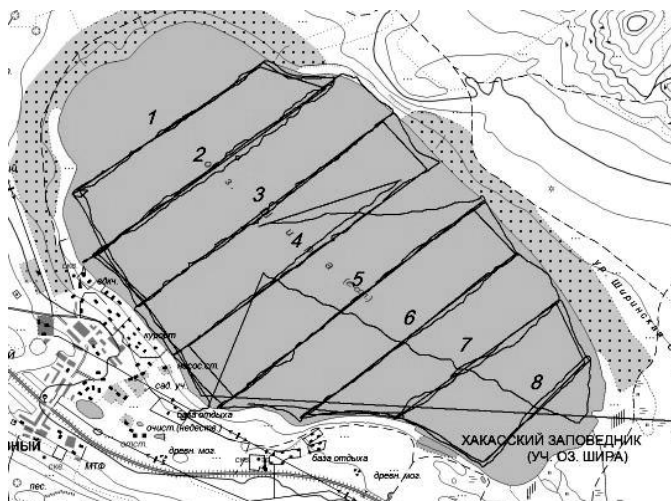


Рис. 2. Точки гидрологических измерений в 2012 г.

Также проводились длительные измерения путем постановки ADCP на дно в центральной точке озера. Прибор ADCP 600 был установлен в мае и июне 2012 г.

Измерение колебаний теплозапаса в фиксированной точке проводилось с использованием автономных распределенных датчиков температуры (РДТ), отличающихся от обычных точечных датчиков тем, что его чувствительный элемент вытянут в линию, располагаемую вертикально и пересекающую слой температурного скачка. РДТ регистрирует среднюю температуру в выбранном интер-

вале глубин, пропорциональную теплозапасу, который реагировал главным образом на вертикальные смещения слоя температурного скачка. Автономные приборы на заякоренных буях обеспечивают продолжительные измерения в выбранном районе. В нашем распоряжении имелось три таких датчика (один в виде гирлянды последовательно соединенных термисторов и два в виде длинного медного провода), которые были установлены в наиболее глубокой части озера.

2. Обсуждение полученных данных

2.1. Формула для определения плотности

СТД зонды наряду с другими величинами измеряют электропроводность воды, на основании которой по встроенным в зонд формулам вычисляются соленость и плотность.

Зонд YSI недостаточно удобен для экспресс-съемки, для этого был использован Idronaut, который хотя и дает систематическую погрешность при расчете солености и плотности по океанским формулам, но тем не менее, позволяет оперативно строить разрезы, которые отражают динамику водоема.

Сотрудником ИБФ СО РАН канд. физ.-мат. наук Д.Ю. Рогозиным были проведены опытные исследования для определения солености в оз. Шира. Им рекомендована формула, которую он вывел в 2011 г.

$$S = 0,8286 \cdot K25 + 0,1937. \tag{1}$$

Кондуктивность *Cond*, измеренная в озере при температуре *in situ*, пересчитывалась в кондуктивность при постоянной температуре 25°C (*K25*) по формуле $K25 = Cond / (1 + 0.0202(t - 25))$. Тогда значения солености, полученные по формуле (1), для натуральных наблюдений 2011 г. в полтора раза превосходят значения, полученные по стандартным формулам.

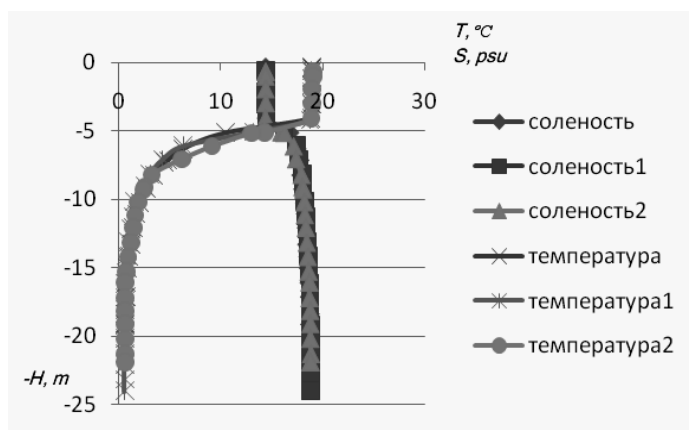


Рис.3. Графики температуры и солености в трех центральных точках озера на 27.06.11 г.

Для подсчета плотности воды соленого озера по известной температуре и солености надо проверить допустимость использования формул для плотности океанской воды. Озеро неглубокое, и в уравнении состояния можно не учитывать зависимость от давления. Мы использовали формулу Бусинеска

$$\rho(T, S) = \rho_1 \cdot \left(\varepsilon_1 + \varepsilon_2 \cdot \frac{T}{T_0} + \varepsilon_3 \cdot \frac{S}{S_0} \right),$$

где $\rho_1 = 1,02541 \text{ г/см}^3$;

$T_0 = 17,5^\circ\text{C}$, $S_0 = 35 \text{ psu}$;

$\varepsilon_1 = 0,97529$; $\varepsilon_2 = -0,00317$; $\varepsilon_3 = 0,02737$.

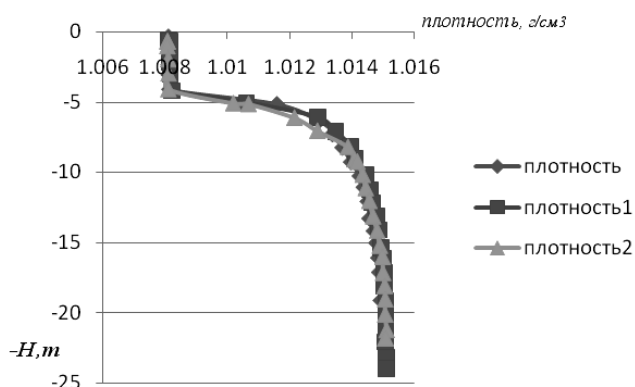


Рис. 4. Плотность по формуле Буссинеска в трех точках озера в соответствии с рис.3.

В придонном слое вклад изменения солёности в изменение плотности сильнее вклада изменений температуры, поэтому за счет солёности в озере возникает устойчивая плотностная стратификация.

В первом приближении можно считать, что структура распределения плотности по глубине "трех-слойная".

Верхний квазиоднородный слой простирается в 2011 г. до 4,5 м. Слой скачка плотности располагается на глубине от 4,5 до 10 м. После этого идет слой почти постоянной температуры.

В 2012 г. квазиоднородный слой опускается до 6 м, слой скачка температуры заканчивается на 13 м.

2.2. Распределение гидрологических величин по разрезам

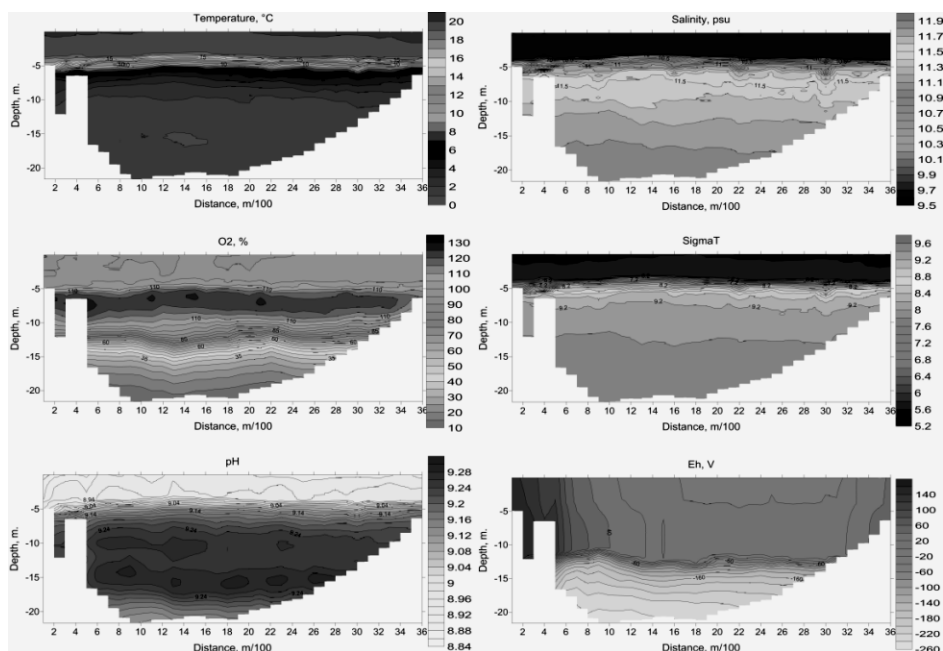


Рис. 5. Гидрологический разрез в оз. Шира 01.07.2011 г.

Главной гидрологической особенностью озера является солёность, распределение которой определяет распределение плотности, а также температуры и остальных гидрологических характеристик. Верхний слой распреснен осадками и тальми водами. Ветро-волновое перемешивание происходит только в верхнем слегка распресненном слое, он быстрее и равномернее нагревается и обогащается кислородом. Гипоксия в придонном слое с резким окисклином и скачком редокс-потенциала свидетельствуют о слабом перемешивании глубинных вод и наличии поглотителей кислорода. Видно, что для всех полей, кроме кислорода, прослеживается однотипная структура, повторяющая структуру

распределения температуры с выделением верхнего квазиоднородного слоя и слоя скачка. У кислорода присутствуют две зоны скачка – 5 и 12 м. В последнем измерении (появление сероводорода в придонном слое) зона скачка приходится на глубину 12 м.

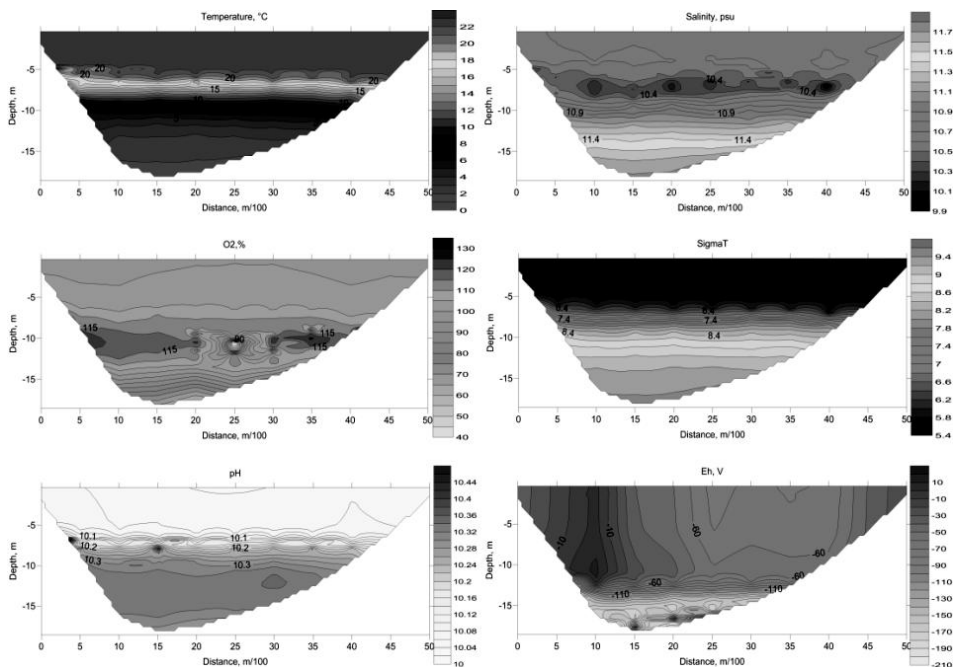


Рис. 6. Гидрологический разрез в оз. Шира 05.08.2012 г.

Зоны скачка кислорода в 2012 г. спустились до 6 и 14 м.

2.3. Горизонтальные циркуляции в озере

В 2011-2012 гг. проведены измерения величины и направления скорости течения в центральной точке озера (рис. 7-12).

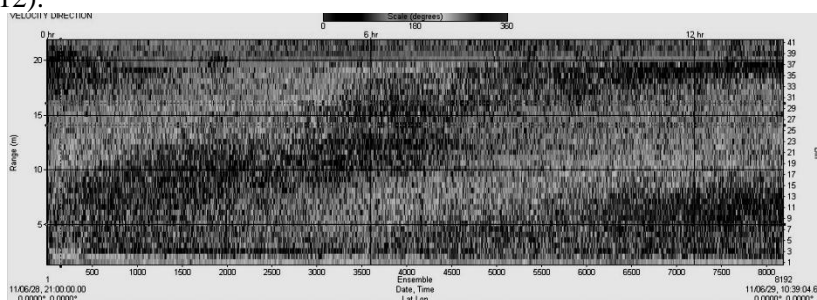


Рис. 7. Направление горизонтальной скорости (июль 2011 г.)

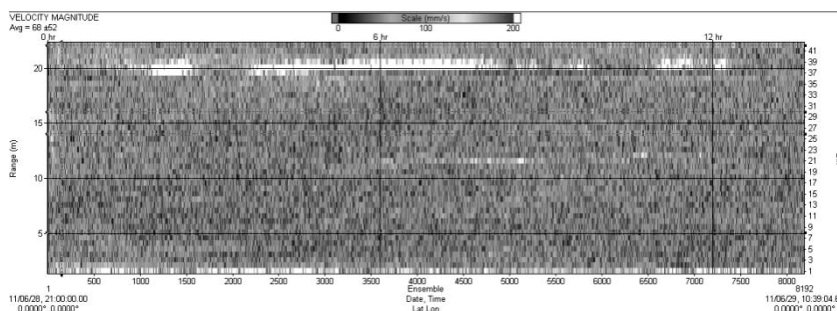


Рис. 8. Величина вектора скорости (июль 2011 г.)

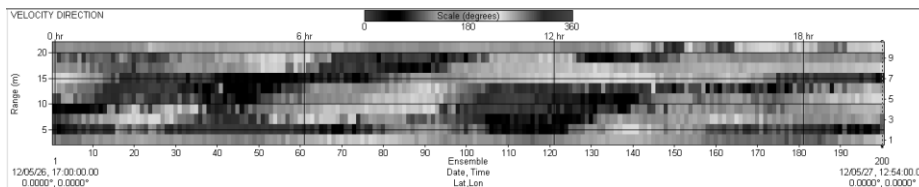


Рис. 9. Направление горизонтальной скорости (май 2012 г.)

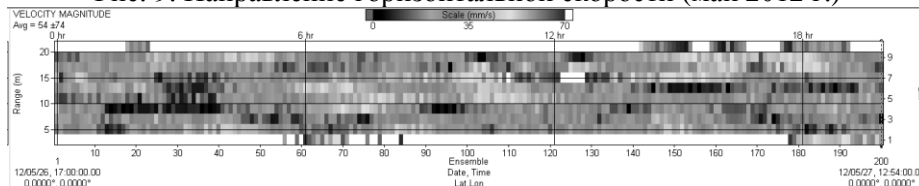


Рис.10. Величина вектора скорости (май 2012 г.)

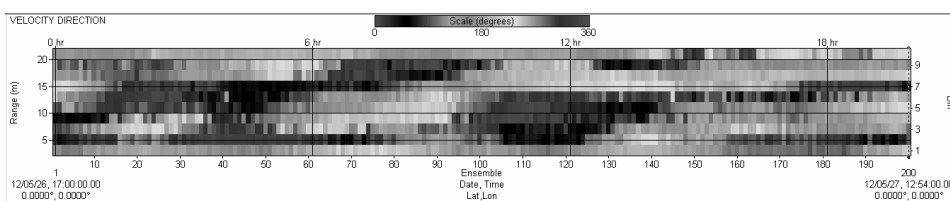


Рис.11. Направление горизонтальной скорости (июль 2012 г.)

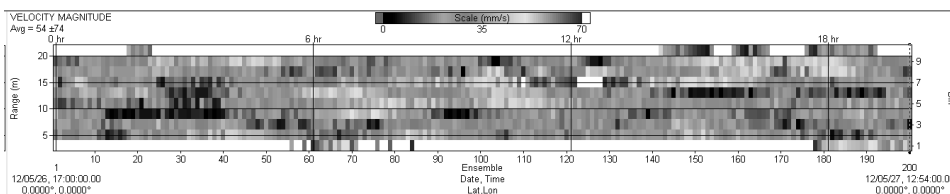


Рис. 12. Величина вектора скорости (июль 2012 г.)

При всех измерениях наблюдаются горизонтальные циркуляции.

При измерениях в июле 2011 г. (рис. 7), мае 2012 г. (рис. 9) и июле 2012 г. (рис. 11) полный поворот направления скорости в точке измерений происходит во всех точках вертикали, лежащих ниже 5 м, с периодом около 12 часов.

При буксировании прибора ADCP 600 вдоль малой оси озера в августе 2012 г. также наблюдались горизонтальные циркуляции.

Например, при прохождении 4-го галса (рис. 2) получено следующее распределение характеристик скорости течения по глубине.

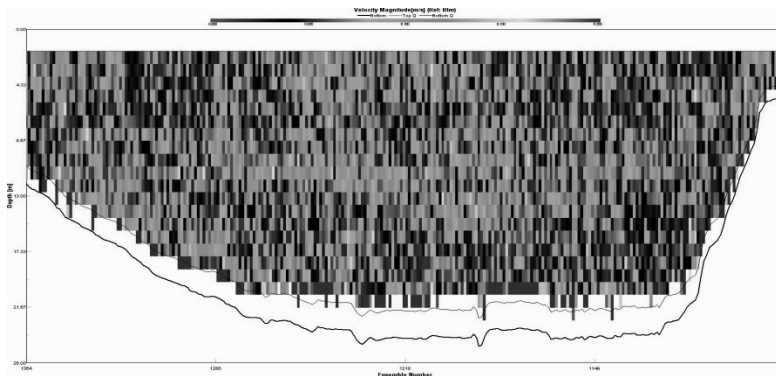


Рис. 13. Величина вектора скорости вдоль 4-го галса

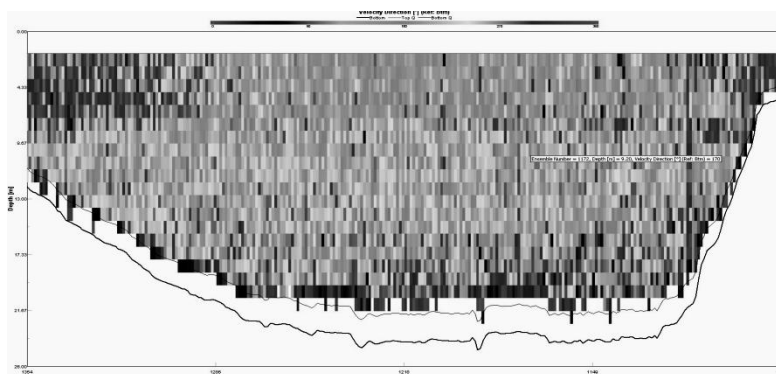


Рис. 14. Направление горизонтальной скорости течения

При данном измерении выделяются глубоководная и прибрежная зоны. В прибрежной зоне есть поверхностный слой до 7 м, где скорости по направлению перпендикулярны направлению течения воды в глубоководной зоне.

3. Исследование колебаний интегральной температуры водяного столба

Исследование колебаний интегральной температуры (ИТ) водяного столба проводилось с использованием распределенных датчиков температуры (РДТ).

Поскольку РДТ измеряет интегральную температуру, то по его показаниям можно судить о колебаниях, обусловленных смещением термоклина вследствие внутренних волн различного периода.

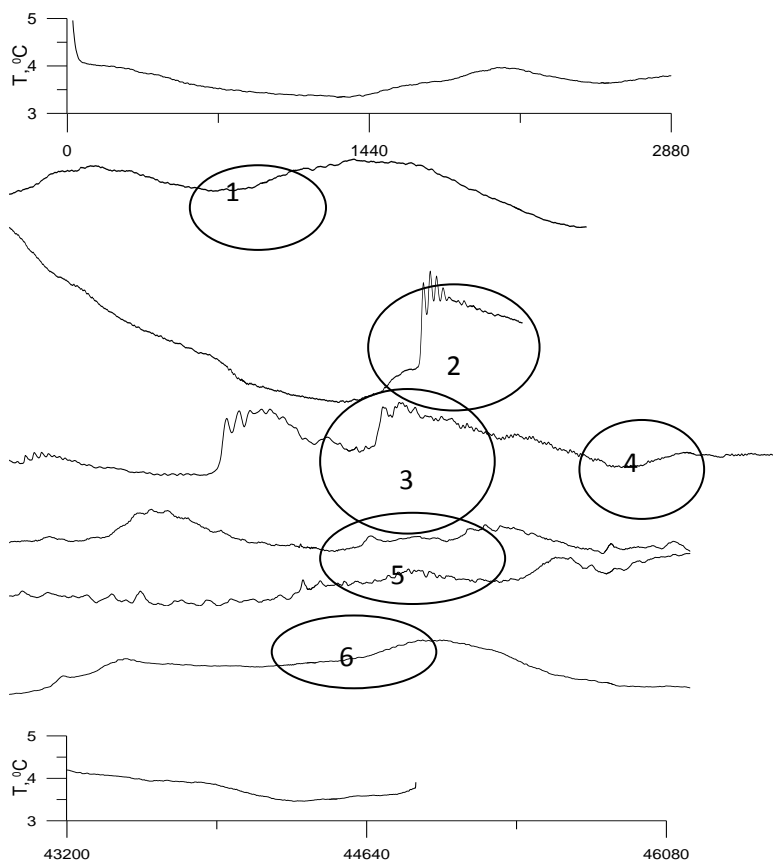


Рис. 15. Растяжки записи интегральной температуры, июль 2011 г.

На рис. 15 показана растяжка записи ИТ одного из РДТ. Каждый участок занимает 4 часа. Отмечены следующие особенности (см. пронумерованные кружки).

1 – пример низкой активности внутренних волн. Предполагаем штилевые условия.

2 – резкий скачок ИТ, и вслед за ним – цуг правильных короткопериодных внутренних волн. Можно предположить прохождение ударной волны (гидроудар). Похожие явления, но слабее, отмечены номерами 3 и 4.

5 – пример беспорядочной толчеи внутренних волн, возникшей предположительно после предшествующих когерентных движений (одиночных волн) после начала сильного ветра.

6 – пример возвращения волнового поля к спокойному состоянию.

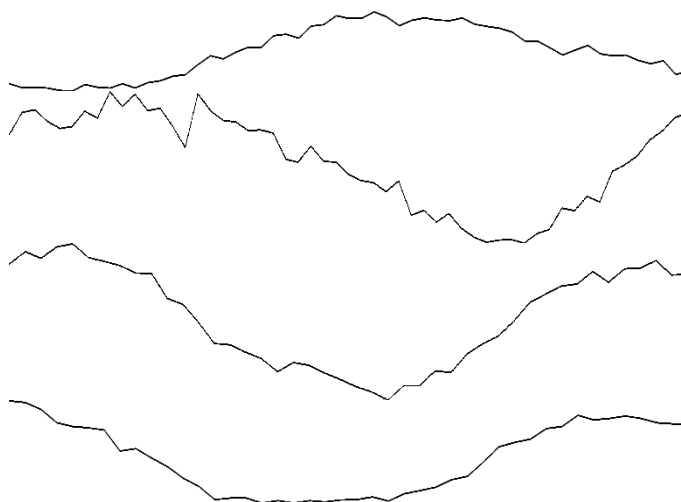


Рис. 16. Растяжки записи интегральной температуры 13-15 августа 2012 г., каждый участок занимает 3 часа

Последний рисунок соответствует периоду, когда в районе оз. Шира наблюдался сильный (14 м/с) ветер.

Заключение

В рамках натурного изучения термохалинной структуры и течений озера в 2011-2012 г. получены следующие результаты.

1. Картина течения в оз. Шира сложная и нестационарная. Устойчивая стратификация и переменная ветровая картина предрасполагают к появлению сейш, их распространению и исчезновению.

2. Устойчивая стратификация имеет место даже для сильных (14 м/с) ветров (толщина перемешанного слоя возрастает, но вблизи дна сохраняется более высокая плотность).

3. Структура распределения плотности неглубоких озер может иметь специфику, связанную с тем, что здесь область постоянной плотности ниже слоя главного термоклина мала или отсутствует вовсе. Это означает, что при исследовании внутренних волн необходимо учитывать тонкую структуру или даже микроструктуру распределения плотности по глубине.

Различие в поведении внутренних волн для сезонов 2011-2012 гг. может быть объяснено следующим образом. В обоих случаях в первом приближении структура распределения плотности по глубине трехслойная. В рамках трехслойной структуры распределения плотности упрощенные математические модели [2] дают максимальные колебания изотерм в середине области скачка (для озера Шира это приблизительно 7-8 м в 2011-2012 гг.). Это соответствует наблюдениям 2012 г., но не соответствует наблюдениям 2011 г., когда максимальные колебания изотерм наблюдались на глубине 12 м. Детальный анализ тонкой структуры распределения плотности в 2011 г. показывает, что, в отличие от 2012 г., распределение плотности в области пикноклина в этом случае можно трактовать как многоволноводную систему. В случае многоволноводной системы возможно появление частот, где групповая скорость достигает экстремальных значений, и в окрестности этих частот могут возникать мощные цуги волн (волны Эйри), что и иллюстрируют наблюдения 2011 г. Кроме того, в многоволноводной системе возможен переход энергии из одного волновода в другой, что могло привести в 2011 г. к варианту течения, когда максимальные колебания изотерм наблюдались на глубине 12 м.

Литература

1. Компаниец Л.А., Якубайлик Т.В., Володько О.С. Анализ характеристик озера Шира на основе натуральных наблюдений // Вестн. Бурят. гос. ун-та. – 2012. – Вып. 9. – С. 167-176.

2. Waves and oscillations in the ocean and in lakes. In Continuum mechanics in environmental science and geophysics / edited by K.Hutter – Springer-Verlag. Berlin, New York, 1993. – 434 p.

Баранов Владимир Иванович, инженер АО ИО РАН, г. Калининград, тел. (4012) 956911, e-mail: baranovvlad@mail.ru.

Голенко Николай Николаевич, кандидат физико-математических наук, ведущий научный сотрудник АО ИО РАН, г. Калининград, тел. (4012) 956911.

Компаниец Лидия Алексеевна, кандидат физико-математических наук, доцент, старший научный сотрудник ИВМ СО РАН, г. Красноярск, тел. (3912) 498811, e-mail: kla@icm.krasn.ru.

Пака Вадим Тимофеевич, доктор физико-математических наук, заведующий лабораторией АО ИО РАН, г. Калининград, тел. (4012) 956911, e-mail: vpaка@mail.ru.

Якубайлик Татьяна Валерьевна, младший научный сотрудник ИВМ СО РАН, г. Красноярск.

Baranov Vladimir Ivanovich, engineer, Atlantic Department of P.P.Shirshov Institute of Oceanology RAS (Kaliningrad), e-mail: baranovvlad@mail.ru

Golenko Nikolay Nikolaevich, candidate of physical and mathematical sciences, chief researcher, Atlantic Department of P.P.Shirshov Institute of Oceanology RAS (Kaliningrad), e-mail: kla@icm.krasn.ru.

Kompaniets Lidiya Alekseevna, candidate of physical and mathematical sciences, senior researcher, Institute of Computational Modeling SB RAS (Krasnoyarsk), (3912) 498811, kla@icm.krasn.ru.

Paка Vadim Timofeevich, doctor of physical and mathematical sciences, professor, head of the laboratory, Atlantic Department of P.P.Shirshov Institute of Oceanology RAS (Kaliningrad), e-mail: vpaка@mail.ru

Yakubaylik Tatiana Valerevna, junior researcher, Institute of Computational Modeling SB RAS (Krasnoyarsk).

УДК 004.922

© А.М. Бубенчиков, К.Л. Базаров, З.Г. Буянтуева

КОМПЬЮТЕРНАЯ ГРАФИКА, ПОРОЖДЕННАЯ ЛИНЕЙНЫМИ ПРЕОБРАЗОВАНИЯМИ КООРДИНАТ

Найдены простейшие координатные зависимости, позволяющие проводить перемещения, повороты и масштабирование объектов. На примере совершенного объекта сферической формы показана возможность параметризации коэффициентов линейного преобразования координат, иллюстрирующая всплытие сферического газового пузыря.

Ключевые слова: компьютерная графика, перемещение, повороты, масштабирование, линейные преобразования координат.

© А.М. Bubenchikov, K.L. Bazarov, Z.G. Buyantueva

COMPUTER GRAPHICS GENERATED BY TRANSFORMATIONS OF LINEAR COORDINATES

The simplest coordinate dependences to make displacements, rotations or scaling objects have been found. On the example of the perfect object of spherical shape the possibility of coefficients parameterization of linear transformation of coordinates has been shown, it illustrates the emersion of a spherical gas bubble.

Keywords: computer graphics, displacement, rotation, scaling, linear transformation of coordinates.

Задача компьютерной графики заключается в моделировании движения объектов в пространстве, то есть в моделировании их поступательных и вращательных перемещений, а также возможного изменения масштаба. В [1] для решения отмеченных задач, а также задач отображения объектов предполагается применить матричную алгебру. Авторы настоящей работы считают, что более естественной, а главное, более простой технологией решений поставленных задач является технология, опирающаяся на линейные преобразования координат.

Рассмотрим прежде всего преобразование, которое отвечает за вращение объекта вокруг некоторой оси. Как показал опыт решения задач классической механики, для этой цели удобнее всего пользоваться подходом Эйлера [2] (задачи динамики твердого тела). С подвижным объектом свяжем ортогональную декартову систему координат $Oxyz$, а систему $O\xi\eta\zeta$, имеющую то же начало, свяжем с пространством, в котором осуществляется движение. Эту последнюю систему называют обычно неподвижной или абсолютной.

Координаты точек объекта в обеих этих системах связаны между собой соотношениями [3]:

$$\left. \begin{aligned} \xi &= \alpha_1 x + \alpha_2 y + \alpha_3 z, \\ \eta &= \beta_1 x + \beta_2 y + \beta_3 z, \\ \zeta &= \gamma_1 x + \gamma_2 y + \gamma_3 z. \end{aligned} \right\} \quad (1)$$

В представленной записи $\alpha_1, \alpha_2, \alpha_3$ – направляющие косинусы осей $Oxyz$ по отношению к оси $O\xi$; $\beta_1, \beta_2, \beta_3$ – те же величины по отношению к оси $O\eta$; $\gamma_1, \gamma_2, \gamma_3$ – направляющие косинусы подвижных осей по отношению к оси $O\zeta$. Эти девять величин Эйлер предложил определять через три независимые угловые координаты θ, ψ, φ , получившие впоследствии название углов Эйлера:

$$\left. \begin{aligned}
 \alpha_1 &= \cos \varphi \cdot \cos \theta \cdot \cos \psi - \sin \varphi \cdot \sin \psi, \\
 \alpha_2 &= -\cos \varphi \cdot \cos \theta \cdot \sin \psi - \sin \varphi \cdot \cos \psi, \\
 \alpha_3 &= \cos \varphi \cdot \sin \theta, \\
 \beta_1 &= \sin \varphi \cdot \cos \theta \cdot \cos \psi + \cos \varphi \cdot \sin \psi, \\
 \beta_2 &= -\sin \varphi \cdot \cos \theta \cdot \sin \psi + \cos \varphi \cdot \cos \psi, \\
 \beta_3 &= \sin \varphi \cdot \sin \theta, \\
 \gamma_1 &= -\sin \theta \cdot \cos \psi, \quad \gamma_2 = \sin \theta \cdot \sin \psi, \quad \gamma_3 = \cos \theta.
 \end{aligned} \right\} \quad (2)$$

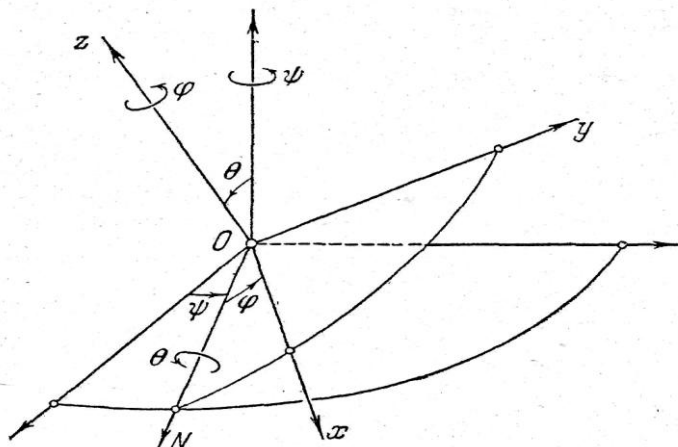


Рис. 1. Взаимное расположение подвижной и неподвижной систем отсчета, определяемое углами Эйлера

Углы, показанные на рисунке 1, называются: φ – угол собственного вращения, ψ – угол прецессии; θ – угол нутации. Задавая θ из интервала $[0, \pi]$, мы отклоняем тело и связанную с ним ось Oz от оси $O\xi$ на соответствующий угол θ . Задавая, кроме этого, ψ из интервала $[0, 2\pi]$, мы поворачиваем на ψ систему координат $Oxuz$ вокруг оси $O\xi$. Наконец, фиксируя еще значение φ в интервале $[0, 2\pi]$, мы поворачиваем тело вокруг оси собственного вращения (оси Oz). Таким образом моделируются все возможные повороты перемещающегося и видоизменяющегося объекта.

Если правые части (1) умножить на числа a_1, a_2, a_3 , то мы изменим масштабы объекта по осям $O\xi, O\eta, O\zeta$ соответственно. Наряду с этим добавим к правым частям (1) постоянные величины b_1, b_2, b_3 . Это обеспечит сдвиг объекта по соответствующим осям.

Таким образом, результирующие формулы поворота, масштабирования и сдвига будут выглядеть как формулы линейного преобразования:

$$\left. \begin{aligned}
 \xi &= \alpha'_1 x + \alpha'_2 y + \alpha'_3 z + b_1, \\
 \eta &= \beta'_1 x + \beta'_2 y + \beta'_3 z + b_2, \\
 \zeta &= \gamma'_1 x + \gamma'_2 y + \gamma'_3 z + b_3.
 \end{aligned} \right\} \quad (3)$$

$$\text{Здесь } \alpha'_i = a_i \alpha_i, \quad \beta'_i = a_i \beta_i, \quad \gamma'_i = a_i \gamma_i, \quad i = 1, 2, 3. \quad (4)$$

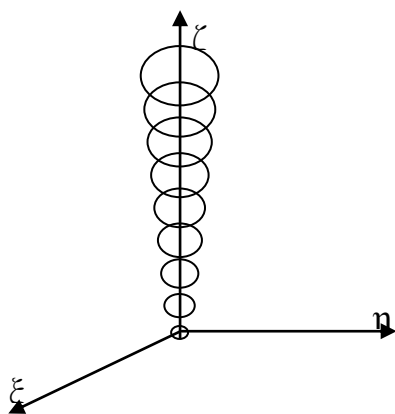


Рис. 2. Подъем газового пузыря по вертикальной траектории

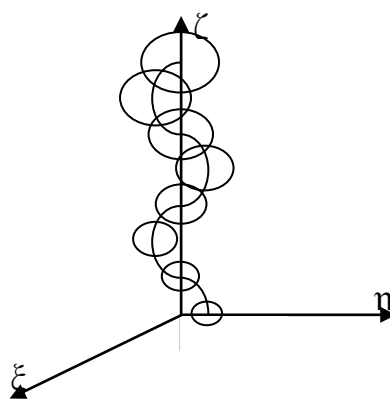


Рис. 3. Движение пузыря по спирали

Движение объекта, показанное на рис. 2, определяется его перемещением вдоль оси Oz и одновременным изменением масштаба по всем трем координатным осям Ox , Oy , Oz . Нетрудно понять, что пропорционального со временем роста размера пузыря можно добиться, задав коэффициенты a_i , входящие в (4) следующим образом:

$$a_i = 1 + v_0 t, \quad i = 1, 2, 3 \quad (5)$$

Здесь v_0 – относительная скорость роста линейного размера пузырька.

В рассматриваемом примере в начальный момент времени центр сферы находится в центре O , являющемся началом систем отсчета (рис. 1), а в последующие моменты времени он всегда находится на оси Oz . Поэтому зависимости углов Эйлера от времени могут быть следующими:

$$\Theta = 0, \quad \varphi = \varphi t, \quad \psi = \psi t \quad (6)$$

Здесь ввиду полярной симметрии объекта функции $\varphi(t)$ и $\psi(t)$ могут быть произвольными, в частности они могут быть нулями: $\varphi t \equiv 0, \psi t \equiv 0$.

Перемещение сферы по оси Oz задается следующими коэффициентами:

$$b_1 = b_2 = 0, \quad b_3 = wt \quad (7)$$

где w – скорость движения центра сферы по оси Oz .

Движение объекта на рис. 3 является более сложным. Здесь центр сферы перемещается по винтовой линии. При этом с высотой пузырек также увеличивается в размерах. Все метаморфозы, происходящие со сферой, в этом случае описываются параметрическими зависимостями для коэффициентов (5), (6). Причем в этом случае целесообразно принять $\varphi t = \omega t, \psi t \equiv 0$. Зависимости (7) обобщаются и принимают следующий вид:

$$b_1 = x_0, \quad b_2 = y_0, \quad b_3 = wt \quad (7')$$

Здесь x_0, y_0 – координаты (не равные нулю), определяющие начальное положение сферы.

Сам объект (сфера) может быть параметризован следующим образом:

$$x - x_0 = R \sin \lambda \cos \delta,$$

$$y - y_0 = R \sin \lambda \sin \delta,$$

$$z - wt = R \cos \lambda.$$

Здесь угловые параметры λ и δ изменяются в следующих диапазонах: $\lambda \in 0, \pi, \delta \in 0, 2\pi$.

Литература

1. Соснин Н.В. Компьютерная графика. Математические основы. Версия 1.0 [Электронный ресурс] : электрон. учеб. пособие. – Красноярск : ИПК СФУ, 2008.
2. Лойцянский Л.Г., Лурье А.И. Курс теоретической механики. – М. : Изд-во ТТЛ, 1955. – 520 с.

3. Арнольд Р.Н., Мондер Л. Гиродинамика и ее технические применения. – М.: Машиностроение, 1964. – 468 с.

Бубенчиков Алексей Михайлович, доктор физико-математических наук, профессор, зав. кафедрой теоретической механики ММФ ТГУ. Тел: 89138500937.

Базаров Кирилл Леонидович, магистрант БГУ. Тел: 89503952036. E-mail: kl.bazarov@jmail.com

Буянтуева Зоригма Гомбожаповна, магистрант БГУ. Тел: 89085962114. E-mail: zori90@mail.ru

Bubenchikov Aleksey Mikhailovich, doctor of physical and mathematical sciences, professor, head of theoretical mechanics department, faculty of mechanics and mathematics, Tomsk State University. Ph.: 89138500937

Bazarov Kirill Leonidovich, graduate student, Buryat State University. ph: 89503952036. E-mail: kl.bazarov@jmail.com

Buyantueva Zorigma Gombozhapovna, graduate student, Buryat State University. ph.: 89085962114. E-mail: zori90@mail.ru

УДК 556.013:556.536.2:556.555

© *В.Н. Данчев, П.Ю. Пушистов*

КОЛИЧЕСТВЕННАЯ ОЦЕНКА ПРОГНОСТИЧЕСКОГО ПОТЕНЦИАЛА ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНОГО КОМПЛЕКСА «ТЕЛЕЦКОЕ ОЗЕРО»

Представлены конкретные результаты численного эксперимента с информационно-вычислительным комплексом гидродинамики, термического и ледового режимов Телецкого озера, полученные при учете внутрисуточной и синоптической изменчивости переменных метеорологического и гидрологического форсингов по данным наблюдений за апрель-июль 1968 г.

Ключевые слова: Телецкое озеро, информационно-вычислительный комплекс, термический режим озера.

© *V.N. Danchev, P.Yu. Pushistov*

QUANTITATIVE ASSESSMENT OF PROGNOSTIC POTENTIAL OF DATA-COMPUTING COMPLEX «TELETSKOYE LAKE»

The results of numerical experiment with data-computing complex of hydrodynamics, thermal and ice regimes of Lake Teletskoye are presented, they were received taking into account the intra daily and synoptic variability variables of meteorological and hydrological forcings according to observational data for April-July 1968.

Keywords: Teletskoye Lake, data-computing complex, thermal regime of lake.

Введение

Информационно-вычислительный комплекс, разработанный авторами этой статьи, для численного воспроизведения переменных гидродинамики, термического и ледового режимов единой лимнологическо-гидрологической системы: устьевой участок р. Чулышман – Телецкое озеро (включая Кыгинский и Камгинский заливы) с основными наиболее крупными боковыми притоками – исток р. Бии, имеет в своем составе четыре проблемно-ориентированных модуля: 1) модуль генерации модели сетки – WMS [1]; 2) базовый прогностический модуль ИВК – модель CE-QUAL-W2 [2]; 3) модуль пре- и постпроцессорной обработки входной и выходной информации – W2i и AGPM-2D [3]; 4) БД для обеспечения работы модулей 1 – 3. Структура, функциональные возможности и ограничения этого комплекса (далее ИВК «Телецкое озеро») подробно описаны в работах [4, 5].

Телецкое озеро, часто называемое «Малым Байкалом» [6, 7], – крупнейший водоем бассейна реки Обь. Длина озера 78,6 км, максимальные ширина – 5,2 км, глубина – 323 м. Основная часть притока в озеро (более 70 %) поступает через р. Чулышман в южную часть озерной котловины, а весь сток осуществляется через р. Бия в противоположной северо-западной части. В 1998 г. Телецкое озеро внесено ЮНЕСКО в реестр объектов Всемирного природного наследия. Половина акваторий Телецкого озера и его восточное побережье входит в состав Алтайского государственного биосферного природного заповедника и является особо охраняемой природной территорией [6, 7]. Однако за последние годы наблюдается резкий рост антропогенной нагрузки как на экосистему озера, так и на его бассейн [7]. Прежде всего это связано с массовым отдыхом и туризмом, сопровождающимся ростом потока отдыхающих (от 150 до 250 тыс. человек в год [5]), строительством объектов туристической и резким увеличением транспортных средств, в том числе маломерных судов. Научно обоснованная система оперативного контроля и управления антропогенной нагрузкой на акваторию и бассейн Телецкого озера отсутствует [7]. Таким образом, проблема разработки методики расчета допустимой антропогенной нагрузки на экосистему и бассейн Телецкого озера является весьма актуальной, и решать ее, по мнению авторов статьи, необходимо в рамках проекта создания системы поддержки принятия решения при интегрированном управлении экосистемой и бассейном озера (далее СППР «Телецкое озеро») [5, 8]. При этом имеется в виду создание полноструктурной СППР, состоящей из набора следующих подсистем: измерительно-коммуникационная / мониторинговая, информационная, мо-

делирующая, экспертно-аналитическая и интерфейсов пользователей. Подчеркнем, что разработка упомянутого выше ИВК исходно осуществлялась на основе идеи его перспективного использования при проектировании и реализации СППР «Телецкое озеро». При таком подходе особую значимость приобретает количественная оценка прогностического потенциала ИВК «Телецкое озеро» как фундаментальной компоненты вычислительной основы будущей СППР.

Цель данной статьи – дать количественную оценку прогностического потенциала ИВК «Телецкое озеро» на основе сравнения результатов моделирования пространственно-временной динамики термического режима озера и соответствующих данных наблюдений за период с 1 апреля по 1 августа 1968 г.

1. Параметры базового численного эксперимента по оценке прогностических возможностей ИВК «Телецкое озеро»

При планировании базового численного эксперимента (ЧЭ), результаты которого анализируются в следующем разделе данной статьи, ставилась конкретная задача – провести количественную оценку прогностических возможностей ИВК «Телецкое озеро» по воспроизведению динамики термического режима Телецкого озера при задании внешних метеорологических и гидрологических воздействий на основе использования фактических суточных данных гидрометеорологических наблюдений за период с 1 апреля по 1 августа 1968 г. По рекомендации В.В. Селегея 1968 г. был выбран в качестве базового года для проведения верификации ИВК «Телецкое озеро», поскольку именно этот год характеризуется наиболее полным набором доступных данных гидрометеорологических наблюдений Озерной станции Западно-Сибирского УГМС [6]. В основе указанной верификации лежит сравнение данных измерений температуры воды в озере (T_L) на всех 11 пунктах наблюдений (рейдовые вертикали и вертикали гидрологических разрезов) Озерной станции [9] с результатами расчетов T_L в базовом численном эксперименте.

Начальные поля T_L и толщины ледового покрова в этом ЧЭ задавались с привлечением максимально полного набора данных наблюдений Озерной станции в конце марта – начале апреля 1968 г. Конкретно рассматривается случай исходно (1 апреля) стратифицированного водоема с T_L в меридиональной части равной $0,8\text{ }^\circ\text{C}$ на поверхности озера и с вертикальным градиентом температуры $\gamma_L = 0,85\text{ }^\circ\text{C} / 100\text{ м}$ в слое от поверхности до глубины 190 м, ниже этой глубины – изотермия и в широтной части – от $0,1$ до $0,6\text{ }^\circ\text{C}$ в подледном слое воды толщиной 10 м и с $\gamma_L = 0,6\text{ }^\circ\text{C} / 100\text{ м}$ ниже глубины 10 м до глубины 190 м, ниже этой глубины – изотермия. Начальная средняя толщина ледового покрова в широтной части Телецкого озера задавалась (согласно работе [6]) равной $0,5\text{ м}$. Начальная температура воды участка р. Чулышман была задана равной $0,7\text{ }^\circ\text{C}$. Значение начального уровня воды (η) в Телецком озере составляло $432,85\text{ мБС}$. На участке р. Чулышман начальные условия для η рассчитывались с помощью линейной интерполяции значений уровня между гидрологическим постом Балыкча и озером. Переменные метеорологического форсинга в этом ЧЭ сформированы на основе 8-срочных данных наблюдений с метеостанций Бея и Яйлю для меридионального и широтного участков соответственно. Переменные гидрологического форсинга в ЧЭ задавались на основе проконтролированных данных суточных наблюдений на всех гидропостах Озерной станции за период с 1 апреля по 1 августа 1968 г. Все данные наблюдений, использованные в базовом ЧЭ, были подвергнуты экспертной оценке на достоверность с участием В.В. Селегея.

Задание других параметров ИВК «Телецкое озеро» в базовом ЧЭ подробно описано в работе [5].

2. Анализ результатов прогнозов температуры воды в поверхностном, среднем по глубине и придонном слоях Телецкого озера

Для количественной оценки прогностического потенциала ИВК «Телецкое озеро» воспользуемся величинами абсолютных ошибок прогноза в поверхностном (AE_S), среднем по глубине (AE_M) и придонном (AE_B) слоях озера, а также значениями:

$ME = \frac{1}{N} \sum_{N=1}^N T_{LH} - T_{LM}$ – среднеарифметической,

$MAE = \frac{1}{N} \sum_{N=1}^N |T_{LH} - T_{LM}|$ – средней абсолютной, $RMSE = \sqrt{\frac{1}{N} \sum_{N=1}^N (T_{LH} - T_{LM})^2}$ – среднеквадратической

ошибками прогнозов температуры воды, где T_{LM} – прогностическое значение T_L , T_{LH} – данные наблюдений.

Ниже приведена таблица 1, которая содержит значения AE_S , AE_M и AE_B , рассчитанные для каждого из 11 пунктов наблюдений (рис. 1) за период с 18 апреля по 20 июля 1968 г.

Таблица

Абсолютные ошибки прогноза температуры воды ($^{\circ}C$) в поверхностном (AE_S), среднем по глубине (AE_M) и придонном (AE_B) слоях Телецкого озера за апрель – июль 1968 г. для каждого из 11 пунктов наблюдений

Дата	18.4	27.4	11.5	21.5	5.6	11.6	20.6	29.6	9.7	20.7
Пункт набл.	Кыгинский залив, вертикаль 28, сегмент 203									
AE_S	0.29	0.32	0.42	0.77	0.10	2.80	0.30	1.20	0.00	0.73
AE_M	0.10	0.20	0.40	0.50	0.20	0.00	0.10	0.00	0.00	0.00
AE_B	0.00	0.00	0.16	0.19	0.26	0.08	0.08	0.08	0.01	0.08
Пункт набл.	мыс Кырсай, вертикаль 19, сегмент 103									
AE_S	0.05	0.39	0.32	0.27	0.62	3.96	3.30	3.30	0.87	1.70
AE_M	0.04	0.20	0.20	0.10	0.88	0.55	0.08	0.02	0.06	0.02
AE_B	0.09	0.11	0.26	0.53	0.07	0.09	0.08	0.14	0.18	0.14
Пункт набл.	р. Челюш, вертикаль 27, сегмент 120									
AE_S	0.05	0.49	0.47	0.32	0.40	0.54	2.43	2.81	1.57	1.60
AE_M	0.11	0.08	0.00	0.00	0.08	0.03	0.09	0.17	0.12	0.01
AE_B	0.05	0.03	0.18	0.19	0.31	0.17	0.10	0.20	0.46	0.22
Пункт набл.	р. Кокши, вертикаль 13, сегмент 130									
AE_S	0.06	0.14	0.08	0.11	0.16	0.22	0.05	0.06	0.12	0.02
AE_M	0.10	0.37	0.01	0.31	0.38	0.22	0.24	0.30	0.27	0.34
AE_B	0.19	0.37	0.39	0.34	0.35	0.33	0.32	0.27	0.09	3.80
Пункт набл.	р. Б. Корбу, вертикаль 26, сегмент 145									
AE_S	0.00	0.07	0.28	0.32	0.08	0.08	0.15	0.00	0.09	0.40
AE_M	0.10	0.04	0.05	0.03	0.08	0.15	0.22	0.23	0.27	0.14
AE_B	0.06	0.11	0.09	0.00	0.33	0.31	0.23	0.38	0.32	0.17
Пункт набл.	Камгинский залив, вертикаль 25, сегмент 335									
AE_S	–	–	0.02	0.05	–	3.19	0.20	0.90	0.01	0.00
AE_M	–	–	0.20	0.28	–	0.08	0.02	0.03	0.02	0.20
AE_B	–	–	0.57	0.37	–	0.08	0.08	0.08	0.08	0.46
Пункт набл.	п. Яйлю, вертикаль 6, сегмент 224									
AE_S	0.08	0.09	0.25	0.42	0.13	0.34	0.33	0.34	0.11	0.13
AE_M	0.05	0.17	0.18	0.10	0.05	0.09	0.03	0.12	0.19	0.19
AE_B	0.00	0.14	0.07	0.14	0.08	0.10	0.08	0.30	0.34	0.12
Пункт набл.	мыс Ажи, вертикаль 24, сегмент 237									
AE_S	–	–	0.22	0.30	0.15	0.30	0.09	0.16	0.23	1.20
AE_M	–	–	0.05	0.12	0.05	0.00	0.06	0.13	0.05	0.09
AE_B	–	–	0.37	0.09	0.01	0.03	0.15	0.26	0.23	0.19
Пункт набл.	мыс Караташ, вертикаль 23, сегмент 272									
AE_S	–	–	0.08	0.11	0.30	0.11	0.39	0.47	1.64	0.13
AE_M	–	–	0.14	0.09	0.38	0.01	0.11	0.14	0.45	0.80
AE_B	–	–	0.08	0.01	0.40	0.01	0.00	0.16	0.17	0.78

Пункт набл.	с. Артыбаш, вертикаль 22, сегмент 318										
АЕ _С	–	–	0.90	0.10	0.47	0.14	0.08	1.10	2.47	0.40	
АЕ _М	–	–	0.38	0.10	0.36	0.09	0.32	0.60	0.1	0.08	
АЕ _В	–	–	0.6	0.1	0.47	0.08	0.15	0.20	1.42	0.01	
Пункт набл.	г/п Артыбаш, вертикаль 2, сегмент 324										
Дата	10.4	20.4	30.4	10.5	20.5	30.5	10.6	20.6	30.6	10.7	20.7
АЕ _С	0.30	0.20	0.20	0.10	0.30	0.10	0.10	0.20	0.30	0.20	0.30

Судя по таблице 1, прогнозы температуры воды в поверхностном слое озера высокого качества ($MAE_S < 0,5 \text{ }^\circ\text{C}$) были получены за весь период моделирования для следующих пунктов: г/п Артыбаш ($MAE_S = 0,23 \text{ }^\circ\text{C}$), мыс Караташ ($0,40 \text{ }^\circ\text{C}$), мыс Ажи ($0,33 \text{ }^\circ\text{C}$), п. Яйлю ($0,22 \text{ }^\circ\text{C}$), р. Б. Корбу ($0,15 \text{ }^\circ\text{C}$). Показатель успешности прогнозов $T_{LS} - SF$ ($SF_S = 100 \%$, если $AE_S < 3 \text{ }^\circ\text{C}$, и $SF_S = 0 \%$, если $AE_S \geq 3 \text{ }^\circ\text{C}$) для всех этих пунктов равен 100% . В трех пунктах, а именно: с. Артыбаш, р. Челюш и рейдовая вертикаль 28 Кыгинского залива, при значениях показателя $SF_S = 100 \%$ фиксируются отдельные даты с несколько большими ошибками прогноза T_{LS} .

Определенную трудность в феноменологической интерпретации причины вызывает случай большой ошибки $AE_S = 3,19 \text{ }^\circ\text{C}$ на 11 июня 1968 г. для рейдовой вертикали Камгинского залива при высоком качестве прогноза T_{LS} в предшествующий (с 11.05 по 21.05.68) и последующий (с 19.06 по 20.07.68) периоды.

Значительная ошибка в прогнозе T_{LS} (прогноз – $12,8 \text{ }^\circ\text{C}$, измеренная величина – $9,0 \text{ }^\circ\text{C}$) в период летнего нагревания (20.07.68) при прямой (устойчивой) стратификации эпилимниона получена для пункта р. Кокши. Одна из вероятных причин возникновения такой ошибки, следуя работам [6, 10], – неучет сугубо локальной, характерной для этого пункта мезомасштабной циркуляции в приводном слое атмосферы, приводящей к местному усилению скорости ветра, которое не фиксируется в данных наблюдений на м/с Беля (расстояние между пунктом р. Кокши и м/с Беля составляет 22 км).

Значительные ошибки прогноза T_{LS} за вторую и третью декады июня 1968 г., как видно из таблицы 1, зафиксированы 11, 20 и 29 июня в пункте «мыс Кырсай». Указанные ошибки прогнозов в данном пункте обусловлены, по нашему мнению, поздним прохождением прогностического фронта речного термического бара (РТБ). Причин таких ошибок может быть, по меньшей мере, две. Первая из них, возможно, связана с тем, что в районе м. Кырсай ширина Телецкого озера оказывается максимальной (около 5,2 км), и исходное требование, при котором были получены уравнения модели CE-QUAL-W2 (см. уравнения (1) – (6) в [4]), – существенное превышение характерного горизонтального масштаба по длине водоема над соответствующим поперечным масштабом – может оказаться нарушенным именно для южной оконечности меридиональной части озера. Другими словами, участок озера от устья р. Чулышмана до гидрологического разреза «мыс Кырсай», включая Кыгинский залив, является скорее морфометрически трехмерным, нежели двумерным, как это предполагается в поперечно–осредненной модели CE-QUAL-W2 [2].

Вторая возможная причина заключается в пренебрежении силой Кориолиса в модели CE-QUAL-W2, которая, как это показано в работе [11], приводит к эффекту ускоренного продвижения фронта РТБ за счет формирования под ее действием струеобразного «прижимного» течения вдоль правого берега (в нашем случае – вдоль восточного берега) меридионально ориентированного водоема. Радикальным вариантом устранения указанных недостатков модели CE-QUAL-W2 может быть переход к какому–либо варианту трехмерной модели, например GEMSS [12]. Альтернативный вариант – модификация расчетной сетки с целью устранения ошибок прогноза скорости перемещения фронта РТБ на участке южной оконечности озера, примыкающей к устью реки Чулышман.

Исходя из анализа таблицы 1, прогнозы T_L на средних горизонтах измерений для всех 11 пунктов наблюдений за апрель – июль 1968 г. можно оценить как вполне успешные, поскольку максимальные значения AE_M не превышают $1 \text{ }^\circ\text{C}$.

Прогнозы придонной температуры озера, как видно из таблицы 1, для всех 11 пунктов наблюдений за апрель – июль 1968 г. можно оценить также как вполне успешные. Определенную трудность в феноменологической интерпретации причины вызывает единственный случай значительной ошибки

$AE_B = 1,42 \text{ } ^\circ\text{C}$ в период летнего нагревания (9.07.68) для пункта с. Артыбаш при высоком качестве прогноза T_{LB} в этом пункте в предшествующие (с 11.05 по 30.06.68) и последующий (20.07) сроки наблюдений.

Информация о средних за период с 18 апреля по 20 июля 1968 г. значениях MAE_S и MAE_B в каждом пункте измерения, а также о средних величинах показателей успешности прогноза $T_L - SF_S$ и SF_B ($SF_B = 100 \%$, если $AE_B < 1 \text{ } ^\circ\text{C}$; SF_B , соответственно, равен 0% , если $AE_B \geq 1 \text{ } ^\circ\text{C}$) по всем 11 пунктам наблюдений представлена на рис. 1.

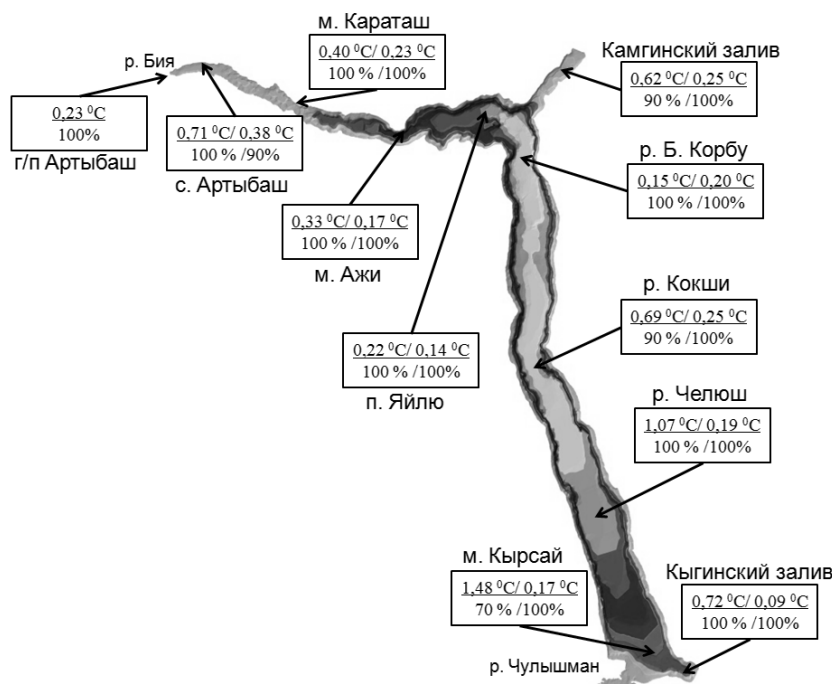


Рис. 1. Информация о средних значениях абсолютных ошибок прогноза T_L (верхняя строка – MAE_S / MAE_B , $^\circ\text{C}$) и средних значениях показателя успешности прогноза (нижняя строка – SF_S / SF_B , %) по 11 пунктам наблюдений Озерной станции ЗС УГМС с апреля по июль 1968 г.

Итоговые значения средних ошибок прогнозов T_{LS} для всех пунктов наблюдений за апрель – июль 1968 г. (всего оценено 103 прогноза) составили: $ME_S = 0,10 \text{ } ^\circ\text{C}$, $MAE_S = 0,59 \text{ } ^\circ\text{C}$ и $RMSE_S = 0,89 \text{ } ^\circ\text{C}$ при средней величине $SF_S = 95 \%$.

Итоговые средние ошибки прогнозов T_L на средних горизонтах измерений для всех пунктов наблюдений за апрель – июль 1968 г. (оценен 91 прогноз) составили: $ME = 0,03 \text{ } ^\circ\text{C}$, $MAE = 0,15 \text{ } ^\circ\text{C}$, $RMSE = 0,20 \text{ } ^\circ\text{C}$ при средней величине SF_M (которая рассчитывалась так же, как SF_B) равной 100% .

Итоговые значения средних ошибок прогнозов T_{LB} для всех пунктов наблюдений за апрель – июль 1968 г. (всего оценен 91 прогноз) составили: $ME_B = -0,05 \text{ } ^\circ\text{C}$, $MAE_B = 0,20 \text{ } ^\circ\text{C}$ и $RMSE_B = 0,27 \text{ } ^\circ\text{C}$ при средней величине $SF_B = 99 \%$.

Заключение

1. Проведенный детальный интерпретационно-сопоставительный анализ результатов прогностических расчетов базового ЧЭ с данными наблюдений на всех 11 пунктах Озерной станции ЗС УГМС за период с 1 апреля по 1 августа 1968 г. показал, что ИВК «Телецкое озеро» обладает значительным прогностическим потенциалом и качественно и количественно верно воспроизводит особенности реальной динамики термического режима поверхностного, среднего по глубине и придонного слоев водоема при учете фактической суточной изменчивости параметров метеорологического и гидрологического форсингов. В статье представлены результаты детального анализа возможных причин отдельных случаев значительных ошибок прогноза температуры воды в поверхностном слое и предложены конкретные способы и методы их устранения.

2. Поскольку ИВК «Телецкое озеро» реально обладает высокими прогностическими возможностями, то он может быть использован как один из базовых компонентов вычислительной основы при разработке проекта системы поддержки принятия решения при интегрированном управлении экосистемой и бассейном Телецкого озера.

Благодарности

Авторы выражают признательность чл.-корр. РАН В.Н. Лыкосову, д-ру геогр. наук В.А. Земцову, д-ру физ.-мат. наук В.Н. Крупчатникову и канд. биол. наук В.В. Кириллову за поддержку данной работы, ценные советы и рекомендации. Особой благодарности заслуживает известный исследователь Телецкого озера В.В. Селегей, который оказал нам большую помощь в получении данных наблюдений и интерпретации результатов расчетов.

Литература

1. Aquaveo is a water resources engineering consulting firm: [сайт]. URL: <http://www.aquaveo.com>.
2. Cole T.M., Wells S.A. "CE-QUAL-W2" A two-dimensional, laterally averaged, Hydrodynamic and Water Quality Model, Version 3.5. Instruction Report EL-06-1, US Army Engineering and Research Development Center, Vicksburg, MS, 2006. – 681 p.
3. Loginetics, Inc: [сайт]. URL: <http://www.loginetics.com>.
4. Данчев В.Н., Пушистов П.Ю. Опыт разработки и результаты применения информационно-вычислительного комплекса для моделирования гидродинамики и качества воды рек и водоемов бассейна Оби. Часть 2 - Телецкое озеро и участок реки Чулышман // Вестн. Бурят. гос. ун-та. – 2012. – Вып. 9. – С. 154-161.
5. Пушистов П.Ю., Данчев В.Н. Информационно-вычислительные комплексы водных объектов бассейна Оби. Ч. 1: ИВК «Северная Сосьва». Ч. 2: ИВК «Телецкое озеро» / науч. ред. В.Н. Лыкосов., В.А. Земцов. – Saarbrücken : LAP Lambert Academic Publishing, 2013. – 160 с.
6. Селегей В.В., Селегей Т.С. Телецкое озеро. Гидрометеорологический режим озер и водохранилищ СССР. – Л. : Гидрометеиздат, 1978. – 143 с.
7. Селегей В.В. Телецкое озеро: очерки истории: в 3 кн. – Барнаул: Пять плюс, 2011. – Кн. 3. – 244 с.
8. Loucks D.P., Beek E. Water resources systems planning and management: an introduction to methods, models and applications. – Paris: UNESCO Publishing, 2005. – 680 p.
9. Материалы наблюдений на озерах и водохранилищах. Дополнение к Гидрологическому ежегоднику 1968 г. / Главное управление гидрометеорологической службы при СМ СССР. – Новосибирск, 1970. – Т. 6. Вып. 0-9. Т. 7. Вып. 0,15-8 – 450 с.
10. Carmack E.C. et al. Importance of lake-river interaction on seasonal patterns in the general circulation of Kamloops Lake, British Columbia // Limnol. Oceanogr. 1979. 24(4). – P. 634–644.
11. Carmack E.C. et al. Mechanisms influencing the circulation and distribution of water mass in a medium residence time lake // Limnol. Oceanogr. – 1986. – 31(2). – P. 249–265.
12. ERM's Surfacewater Modeling Group: [сайт]. URL : <http://www.erm-smg.com>.

Данчев Василий Николаевич, младший научный сотрудник, Институт водных и экологических проблем СО РАН, e-mail: danchev_vasilii@mail.ru

Пушистов Петр Юрьевич, доктор физико-математических наук, профессор Бурятского государственного университета, главный научный сотрудник Сибирского регионального научно-исследовательского гидрометеорологического института, e-mail: pushtcmpr@mail.ru

Danchev Vasily Nikolaevich, junior researcher, Institute for Water and Environmental Problems SB RAS, e-mail: danchev_vasilii@mail.ru

Pushistov Pyotr Yurevich, doctor of physical and mathematical sciences, professor, Buryat State University, chief researcher, Siberian Regional Scientific-Research Hydro meteorological Institute, e-mail: pushtcmpr@mail.ru

ОБ ОДНОЙ МОДИФИКАЦИИ МОДЕЛИ МЕЖОТРАСЛЕВОГО БАЛАНСА¹

Описывается новая модель межотраслевого баланса, основанная на матрице суммарных затрат, включающей помимо коэффициентов прямых материальных затрат расходы домашних хозяйств по отраслям. На основе межотраслевого баланса совокупных расходов и отраслевого мультипликатора дохода проведены расчеты мультипликационных эффектов в экономике.

Ключевые слова: межотраслевой баланс, матрица суммарных расходов.

© Z.B.-D. Dondokov, A.S. Buldaev

ON ONE MODIFICATION OF INTERBRANCH BALANCE

A new model of interbranch balance is described. It has been founded on the matrix of total expenses which includes besides direct material expenses the household outlay by branches. Calculations of multiplying effects in economy have been performed on the basis of interbranch balance of total expenditures and branch income multiplier.

Keywords: interbranch balance, matrix of total expenses.

Введение

Разработанная В.В. Леонтьевым модель межотраслевого баланса (модель «затраты-выпуск», МОБ) является важнейшим инструментом оптимального управления макроэкономическими системами [1, 2]. На основе межотраслевых моделей практически во всех странах производится анализ экономического развития, осуществляются плановые и прогнозныe расчеты, определяются балансы производства и распределения продукции. Основные положения МОБ легли в основу макроэкономической статистики и системы национальных счетов. В качестве эндогенных переменных в модели используются показатели промежуточного потребления, искомой величиной является валовой выпуск.

Вместе с тем основным показателем, определяющим оптимальность управления макроэкономическими системами, является валовой внутренний продукт (ВВП), расчет которого в рамках классической модели МОБ не осуществляется. Решение данной проблемы возможно при использовании модели баланса суммарных расходов (МОБСР), являющейся синтезом классической модели межотраслевого баланса и кейнсианской модели мультипликатора.

1. Описание классической модели межотраслевого баланса

Межотраслевой баланс описывается экономико-математической моделью, характеризующей взаимосвязи между производством и распределением продукции по отраслям экономики страны. Каждая отрасль является, с одной стороны, производителем продукции. Например, продукцию электроэнергетики (электроэнергию) потребляют остальные отрасли экономики. В свою очередь отрасль является потребителем продукции других отраслей. Так, электроэнергетика использует в производстве продукцию угольной промышленности и т.п.

Сводная таблица межотраслевого баланса строится из типов уравнений:

а) производство продукции:

$$x_i = \sum_{j \in I} x_{ij} + y_i, \quad i \in I, \quad (1)$$

б) распределение продукции:

$$x_j = \sum_{i \in J} x_{ij} + z_j, \quad j \in J. \quad (2)$$

¹ Работа выполнена при финансовой поддержке РФФИ (проекты №№ 12-01-00914-а, 12-01-98011-р_сибирь_а, 13-01-92200-Монг_а).

В системах равенств (1) и (2) используются следующие обозначения: $i \in I$ – индекс отрасли-производителя продукции; $j \in J$ – индекс отрасли-потребителя продукции; x_i – валовой продукт (объем распределенной продукции) i -й отрасли; x_j – валовой продукт (объем производства продукции) j -й отрасли; x_{ij} – затраты продукции i -й отрасли на производство продукции j -й отрасли; y_i – конечное использование продукции i -й отрасли (непроизводственное потребление, возмещение выбытия и расширение основных фондов, экспорт, возмещение потерь, запасы и резервы на конец периода); z_j – объем конечной продукции j -й отрасли.

Рассматриваемая модель отражает баланс (равенство) между производством и распределением продукции по отраслям, что означает равенство x_i и x_j , если $i = j$.

Сводная таблица межотраслевого баланса состоит из четырех основных разделов-квадрантов (табл. 1).

Таблица 1

Схема межотраслевого баланса

Распределение продукции (j) Затраты на производство (i)	Текущее производственное потребление в отраслях					Конечная продукция (по элементам)	Валовой продукт
	1	2	...	n	итого		
Материальные затраты отраслей							
1	x_{11}	x_{12}	...	x_{1n}	$\sum_{j=1}^n x_{1j}$	y_1	x_1
2	x_{21}	x_{22}	...	x_{2n}	$\sum_{j=1}^n x_{2j}$	y_2	x_2
· · ·			Квадрант I			Квадрант II	
n	x_{n1}	x_{n2}	...	x_{nn}	$\sum_{j=1}^n x_{nj}$	y_n	x_n
ИТОГО	$\sum_{i=1}^n x_{i1}$	$\sum_{i=1}^n x_{i2}$...	$\sum_{i=1}^n x_{in}$	$\sum_{i=1}^n \sum_{j=1}^n x_{ij}$	$\sum_{i=1}^n y_i$	$\sum_{i=1}^n x_i$
Валовая добавленная стоимость	z_1	z_2	Квадрант III	z_n	$\sum_{j=1}^n z_j$	Квадрант IV	
Валовой продукт	x_1	x_2		x_n	$\sum_{j=1}^n x_j$		

I квадрант — таблица промежуточного использования продукции по отраслям (возмещение потребленных в производстве предметов труда и услуг);

II квадрант — материально-вещественный состав конечной продукции (использование продуктов на конечное потребление, валовое накопление капитала и экспорт);

III квадрант — компоненты добавленной стоимости (оплата труда, другие налоги за минусом субсидий на производство, потребление основного капитала и чистая прибыль) по отраслям;

IV квадрант — перераспределение конечной продукции.

Главную связующую роль в системе показателей межотраслевого баланса общественного продукта играет I квадрант. Он содержит основной массив информации, используемой для анализа межотраслевых связей.

Соотношение I и II квадрантов межотраслевого баланса общественного продукта (1) характеризует зависимости между величинами x_i, y_i, x_{ij} . Для построения математической модели решающее значение имеет предположение о том, что x_{ij} – есть функция от объема производства этой продукции: $x_{ij} = \varphi_{ij}(x_j)$, $i, j \in I$. Подставив значения x_{ij} в (1), получим систему из n уравнений:

$$x_i = \sum_{j \in I} \varphi_{ij}(x_j) + y_i, \quad i \in I. \quad (3)$$

В простейшей модели Леонтьева используется предположение о пропорциональной зависимости между затратами и объемами производства, т.е. вводятся линейные однородные функции производственных затрат:

$$x_{ij} = a_{ij}x_j. \quad (4)$$

Коэффициент пропорциональности $a_{ij} \geq 0$ называют *коэффициентом прямых затрат продукции i на производство единицы продукции j* . Эти коэффициенты в совокупности образуют квадратную матрицу $A = (a_{ij})$, $i, j \in I$.

В предположении (4) из (3) получаем систему из n линейных алгебраических уравнений:

$$x_i = \sum_{j \in I} a_{ij}x_j + y_i. \quad (5)$$

В векторно-матричной форме имеем

$$X = AX + Y, \quad (6)$$

где $X = (x_i)$ – вектор-столбец объемов производства; $Y = (y_i)$ – вектор-столбец конечной продукции. Матрицу $A = (a_{ij})$ с неотрицательными коэффициентами будем обозначать $A \geq 0$. Экономическая трактовка величин X, Y, A накладывает специфические ограничения в модели (6): $X \geq 0, Y \geq 0, A \geq 0$.

Матрица $A \geq 0$ (и модель (6)) называется продуктивной, если существует единственное решение $X \geq 0$ системы (6) для любого заданного $Y \geq 0$.

Для продуктивной модели получаем

$$X = (E - A)^{-1}Y \geq 0, \quad (7)$$

где E – единичная матрица порядка n , $(E - A)^{-1} \geq 0$ – матрица, обратная к $(E - A)$, которая называется матрицей полных затрат, а также мультипликатором Леонтьева.

На основе классической модели межотраслевого баланса (6) рассчитываются косвенные и полные эффекты в экономике, возникающие в связи с изменением объемов конечного потребления (потребления домашних хозяйств, государственных расходов, инвестиций, чистого экспорта).

Вместе с тем модели МОБ (6) не в полном объеме отражают воздействие изменения автономного спроса на экономические процессы. Оценка мультипликационных эффектов возможна лишь в отношении валового продукта. Матрица полных затрат является мультипликатором валового продукта, а не мультипликатором дохода. На наш взгляд, этого явно недостаточно, так как наибольший интерес представляет исследование изменений показателей дохода – валового национального продукта, национального дохода и т.д.

2. Модель межотраслевого баланса суммарных расходов

В основе предлагаемого подхода лежит гипотеза об однородности производственного и непроизводственного потребления, в силу чего при оценке косвенных эффектов необходимо суммировать производственные и потребительские расходы [3].

Доходы и расходы домашних хозяйств предлагается рассматривать в отраслевом разрезе. Проводится группировка домашних хозяйств по отраслям в соответствии с источниками доходов отдельной семьи.

В новой модели мультипликатора вектор-столбец потребления домашних хозяйств $C = (c_i)$ определяется с помощью матрицы потребления $F = (f_{ij})$ следующим образом. Общий объем потребления домашними хозяйствами продукции i -й отрасли представляется в виде суммы потребления этой продукции домашними хозяйствами, получающими доход в различных отраслях:

$$c_i = \sum_j f_{ij}, \tag{7}$$

где f_{ij} – потребление продукции i -ой отрасли домашними хозяйствами, получающими доход в j -ой отрасли. Матрица F по структуре аналогична матрице текущего производственного потребления (I квадрант межотраслевого баланса).

Введем матрицу P , элементами которой являются коэффициенты потребления p_{ij} :

$$p_{ij} = \frac{f_{ij}}{x_j}, \tag{8}$$

где x_j – валовой выпуск j -й отрасли. Тогда $PX = C$ и систему уравнений (6) можно представить в виде

$$X = AX + PX + Y^*, \tag{9}$$

где $Y^* = Y - C$ – вектор-столбец конечного потребления за вычетом потребления домашних хозяйств.

Однородность матриц позволяет преобразовать систему уравнений (9) в виде

$$X = (A + P)X + Y^* = BX + Y^*, \tag{10}$$

где $B = (b_{ij}) \geq 0$ – матрица суммарных расходов.

Система (10), рассматриваемая совместно с условиями

$$PX = Y - Y^*, Y \geq 0, Y^* \geq 0, Y - Y^* \geq 0, \tag{11}$$

является эквивалентной системе (6) относительно множества решений X . Следовательно, из продуктивности матрицы A следует продуктивность матрицы $B = A + P$ в модели (10), (11), и наоборот.

Принципиальное отличие предлагаемого подхода от других моделей, основанных на методе межотраслевого баланса, заключается в том, что в качестве «утечек» в новой модели рассматривается матрица B , элементы которой b_{ij} представляют собой коэффициенты суммарных «затрат» продукции i -й отрасли на производство единицы продукции j -й отрасли. Это позволяет при оценке мультипликационных эффектов учитывать воздействие на валовую продукцию отрасли как материальных затрат, так и потребительских расходов.

В продуктивной модели (10), (11) имеем:

$$X = (E - B)^{-1} Y^* \geq 0. \tag{12}$$

Назовем $(E - B)^{-1}$ матрицей полных суммарных расходов, аналогичной матрице полных затрат В.В. Леонтьева.

Вместе с тем определение прироста валового продукта, связанного с изменением автономного спроса, не является конечной и важнейшей целью. Наибольший интерес в исследовании процесса мультипликации представляет количественная оценка прироста дохода, в том числе по отдельным отраслям. Важнейшей задачей является определение мультипликаторов дохода. Классическая модель Леонтьева не дает такой возможности.

В силу этого модель мультипликатора, основанная на матрице $(E - B)^{-1}$, дополнена специальным множителем – вектором $R = (r_j)$, коэффициенты которого показывают долю дохода в валовом выпуске j -й отрасли:

$$r_j = 1 - \sum_i a_{ij},$$

где a_{ij} – коэффициенты прямых материальных затрат продукции i -й отрасли на производство единицы продукции j -й отрасли.

Использование вектора R позволяет вычленив из валового выпуска вновь созданную стоимость и рассчитать прирост валового внутреннего продукта по отраслям.

Произведение вектора R и $(E - B)^{-1}$ – матрицы полных суммарных расходов – представляет собой мультипликатор дохода K :

$$K = R(E - B)^{-1}. \quad (13)$$

Элементы вектора K показывают прирост валового внутреннего продукта по отраслям экономики. Это позволяет провести ранжирование отраслей по максимальному приросту валового внутреннего продукта.

На основе матрицы полных суммарных расходов можно осуществлять также расчет и других специальных мультипликаторов, в т.ч. мультипликаторов занятости, налогового мультипликатора и т.п.

Эффективное использование моделей включает решение оптимизационных задач по критериям, типичным для экономической науки. При этом в качестве критерия оптимальности часто рассматривается максимум значения накопленного дохода. Предлагаемый новый подход к анализу доходов дает возможность разрабатывать актуальные задачи оптимального управления региональной экономической динамикой в рамках развиваемых динамических балансовых моделей, современный обзор которых достаточно полно приведен в работе [4].

3. Результаты расчетов

З.Б.-Д. Дондоковым было рассмотрено действие механизма мультипликации на примере 22-отраслевого межотраслевого баланса России за 1997, 2000 и 2003 гг. [5-7].

Для исследования межотраслевых эффектов в экономике необходимо выделить потребление импортной и отечественной продукции. Лишь использование последней обеспечивает реальный прирост валовой продукции, валового внутреннего продукта и других макроэкономических показателей. В силу этого введем следующие показатели:

A_d – матрица коэффициентов прямых затрат отечественной продукции;

F_d – матрица потребления домашними хозяйствами отечественной продукции;

P_d – матрица коэффициентов потребления домашними хозяйствами отечественной продукции;

$Q = (E - A_d)^{-1}$ – матрица полных материальных затрат отечественной продукции;

$S = (E - (A_d + P_d))^{-1}$ – матрица полных суммарных расходов отечественной продукции;

$M_c = (m_{cj})$ – мультипликатор полных внутренних материальных затрат (мультипликатор Леонтьева), где $m_{cj} = \sum_i Q_{ij}$;

$M_t = (m_{tj})$ – мультипликатор суммарных расходов отечественной продукции, где $m_{tj} = \sum_i S_{ij}$;

$K_m = RS$ – отраслевой мультипликатор дохода.

Результаты расчетов, приведенные в таблице 2, позволяют проранжировать отрасли по приросту валового продукта и увеличению дохода (валового внутреннего продукта).

Таблица 2

Мультипликаторы валового выпуска и отраслевого дохода в России

Отрасли экономики	1997 г.			2000 г.			2003 г.		
	M_c	M_t	K_m	M_c	M_t	K_m	M_c	M_t	K_m
1. Э/энергетика	1.42	1.85	1.22	1.99	2.94	1.35	2.07	3.16	1.52
2. Нефтегазовая пром.	1.59	2.04	1.21	1.96	2.84	1.32	1.86	2.95	1.53
3. Угольная пром.	1.63	2.17	1.24	1.78	2.69	1.29	2.20	3.21	1.41
4. Прочая топл. пром.	1.46	1.94	1.23	1.53	2.41	1.33	1.55	2.60	1.44
5. Черная металлургия	1.74	2.16	1.13	2.01	2.88	1.21	2.16	3.15	1.38
6. Цветная металл.	1.76	2.24	1.16	2.05	2.91	1.22	2.10	3.13	1.44
7. Хим/нефт. пром.	1.79	2.23	1.16	2.03	2.88	1.18	2.19	3.11	1.29
8. Маш/стр. и мет/обр.	1.71	2.18	1.14	2.06	2.94	1.21	2.14	3.09	1.32
9. Лесная, дер/обр.	1.72	2.21	1.21	2.04	2.97	1.29	2.14	3.15	1.40
10. Пром. строит/мат.	1.59	2.06	1.21	2.12	3.05	1.29	2.17	3.21	1.45
11. Легкая пром.	1.60	2.01	1.03	1.78	2.41	0.86	1.82	2.57	1.04
12. Пищевая пром.	1.96	2.46	1.18	2.26	3.14	1.20	2.22	3.19	1.34
13. Прочая пром.	1.78	2.29	1.19	2.16	3.05	1.23	2.14	3.13	1.38
14. Строительство	1.47	1.95	1.20	1.86	2.79	1.28	1.83	2.87	1.42
15. С/х, лес. хозяйство	1.77	2.34	1.27	1.82	2.85	1.36	1.75	2.83	1.50
16. Транспорт, связь	1.38	1.93	1.28	1.65	2.60	1.32	1.77	2.83	1.47
17. Сфера обращения	1.33	1.96	1.33	1.43	2.48	1.41	1.47	2.57	1.54
18. Проч. сфера мат/пр.	1.42	2.07	1.29	1.59	1.86	1.01	1.53	2.56	1.43
19. Просвещение	1.50	2.07	1.26	1.79	2.76	1.30	1.59	2.64	1.47
20. ЖКХ	1.49	2.04	1.28	1.87	2.85	1.35	1.80	2.88	1.51
21. Управление, финан.	1.59	2.13	1.23	1.86	2.79	1.26	1.68	2.74	1.46
22. Наука	1.65	2.15	1.16	2.06	3.01	1.32	1.95	3.01	1.48
Максимум	1.96	2.46	1.33	2.26	3.14	1.41	2.22	3.21	1.54
Минимум	1.33	1.85	1.03	1.43	1.86	0.86	1.47	2.56	1.04

Высокие значения мультипликационных эффектов получены в отраслях с низкой долей «утечек» (высокой долей добавленной стоимости (оплаты труда и прибыли) и низким удельным весом импортной продукции в составе валового продукта): промышленность строительных материалов, электроэнергетика, сфера обращения. Наихудшие результаты зафиксированы в легкой промышленности, в продукции которой преобладает импорт. В классической модели «затраты-выпуск» в данной отрасли существенные отклонения мультипликационных эффектов не фиксируются.

Заключение

Разработанная методика позволяет более точно определить влияние развития отдельных отраслей на темпы экономического роста в стране. Проведенное исследование позволило сделать следующие выводы:

1. Результаты расчетов, проведенных по модели мультипликатора совокупных затрат значительно отличаются от значений мультипликационных эффектов, рассчитанных на основе других мультипликаторов.
2. Расчеты позволяют ранжировать отрасли по величине эффектов мультипликации.
3. Отрасли с низкой долей «утечек» в виде импорта и высокой долей дохода в валовом продукте обеспечивают наибольший мультипликационный прирост валового внутреннего продукта.
4. Государству целесообразно поощрять развитие отраслей, в стоимостной продукции которых доля отечественных товаров как производственного, так и потребительского назначения максимальна.

Литература

1. Гранберг А.Г. Основы региональной экономики. – 3-е изд. – М.:ГУ ВШЭ, 2003. – 495 с.
2. Леонтьев В. Межотраслевая экономика / пер. с англ. – М.: Экономика, 1997.
3. Дондоков З.Б.-Д. Методологические вопросы оценки мультипликационных эффектов в экономике. – Улан-Удэ: Изд-во ВСГТУ, 2000. – 145 с.
4. Проблемы и перспективы социо-эколого-экономических моделей / В.И. Гурман и др. // Вестн. Бурят. гос. ун-та. – 2012. – Спецвып. Д. – С. 73-79.
5. Система таблиц «Затраты - Выпуск» России за 1996-1997 гг.: стат. сб./ Госкомстат России. – М., 2001.
6. Система таблиц “Затраты-Выпуск” России за 2000 г.: стат. сб./ Госкомстат России. – М., 2003.
7. Система таблиц “Затраты-Выпуск” России за 2003 г.: стат. сб./ Госкомстат России. – М., 2006.

Дондоков Зорикто Бато-Дугарович, доктор экономических наук, профессор, заведующий отделом, Бурятский научный центр СО РАН. E-mail: dzorikto@mail.ru

Булдаев Александр Сергеевич, доктор физико-математических наук, профессор кафедры прикладной математики Бурятского государственного университета. E-mail: buldaev@mail.ru

Dondokov Zorikto Bato-Dugarovich, doctor of economic sciences, professor, head of department, Buryat Scientific Center SB RAS. E-mail: dzorikto@mail.ru

Buldaev Alexander Sergeevich, doctor of physical and mathematical sciences, professor of applied mathematics department of Buryat State University. E-mail: buldaev@mail.ru

УДК 519.1; 519. 248

© Г.В. Попков, В.К. Попков

СИСТЕМА РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ ДАННЫХ

В статье рассматривается гиперсетевой подход, связанный с обработкой информации, передаваемой по разнотипным сетям передачи данных. Данный подход обеспечивает универсальную технологию для любых видов информации.

Ключевые слова: гиперсети, обработка информации, системы распределения информации.

© G.V. Popkov, V.K. Popkov

A SYSTEM OF DISTRIBUTED DATA PROCESSING

In the article a hyper network approach concerned with processing of information transmitted over polytypic networks of data transmission is considered. This approach provides universal technology for any sort of information.

Keywords: hyper networks, information processing, systems of information distribution.

Введение

В прикладных задачах транспорта, связи, систем управления и др. систем часто возникает задача обработки разнообразной и разнотипной информации для принятия решений или других действий. Для этой цели используются сети передачи информации разного вида по разнотипным каналам. В данной работе предлагается вариант такой системы, основанной на применении теории S – гиперсетей [1]. Эта теория позволяет описывать взаимодействие сетей различного типа с учетом их пространственного расположения. Т.е. метрическое пространство всего круга взаимодействующих сетей задается первичной сетью S – гиперсети и тем самым определяется как минимум пространственное взаимодействие сетей. Узел первичной сети при этом рассматривается не как точка, а как некоторая замкнутая область, в которую отображаются все вторичные сети. Структура вторичных сетей задается не только графами и гиперграфами, но и гиперсетями. Таким образом, отобразив вершины в первичную сеть, можно написать процедуру отображения ребер этих сетей в ветви первичной сети. Аналогичным образом поступаем с сетями 2-го уровня, отображая их вершины в вершины вторичных сетей 1-го уровня, затем каждому ребру сети второго уровня сопоставляем маршрут в некотором графе вторичной сети 1-го уровня. И так далее согласно структурному дереву T(SH) S- гиперсети (рис.1).

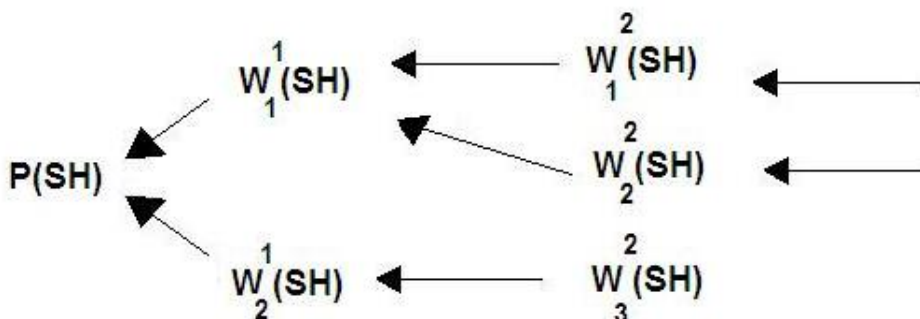


Рис.1. Структурное дерево гиперсети SH = (P, W₁¹, (W₁² W₂²), W₂¹ (W₃²))

1. Основные определения

В своей основе математическая модель системы распределенной обработки данных (СРОД) представляет собой многоуровневую нестационарную S-гиперсеть, в которой в качестве узлов первичной сети выступают программные модули различного назначения и которые объединяются в группы (узлы-сети) с целью организации «спец-вычислителей», предназначенных для реализации различных процессов (как в пространстве, так и во времени). Очевидно, что каждому сообщению в данной S-гиперсети будет соответствовать некоторое «виртуальное» ребро, которое сопоставляется цепочке из разных процессов. Вообще говоря, процессу также сопоставляется либо закрепленная, либо «виртуальная» последовательность, состоящая из последовательности модулей.

Для понимания постановки задачи дадим некоторые определения из теории гиперсетей.

Шестерка, состоящая из трех множеств и трех отображений $S=(X,V,R; P,F,W)$, называется гиперсетью, если:

$$\forall v \in V \mid P(v) \mid = 2,$$

$$\forall r \in R \mid W(r) \mid = 2,$$

$$\forall r \in R \text{ множество } F(r) \subseteq V \text{ составляет маршрут в графе } PS=(X,V).$$

Таким образом, первичная PS и вторичная сети WS гиперсети S являются графами, а F отображает ребра WS=(X,R) в маршруты графа PS=(X,V).

Так как множество F(r) является маршрутом, то отображение F единственным образом определяет отображение W. Действительно, концевые вершины маршрута F(r) являются одновременно концами ребра r, то есть гиперсеть S можно задать пятеркой (X,V,R; P,F).

В гиперсети вида $S = Y, V, R$ узел $y \in Y$ заменяется на граф вида $y = x_j^i, E_j$ – граф узла у структурированной гиперсети, где x_j^i – j-я вершина вторичной сети WS_i , отображенная в узел у структурированной гиперсети $SA = Y, V, G(X_i, R_i)$. Таким образом, в отличие от гиперсетей, вершины вторичных сетей помещаются в узлы первичной сети независимо друг от друга, без ограничения на характер отображения, т.е. возможны варианты отображения нескольких вершин одной вторичной сети в один узел первичной сети.

Дадим формальное определение S – гиперсети. Пусть задано множество графов (гиперграфов) $G_0 = (X^0, V), G_1 = (X^1, U^1), \dots, G_k = (X^k, U^k)$ и корневое дерево $T_0 = (Z, R)$, где $Z = z_0, z_1, \dots, z_k, R = r_1, \dots, r_k$, определяющее вложение графов G_j в G_i ($i < j$) аналогично вложениям, определяемым в гиперсетях, за тем лишь исключением, что вершины x_k^i и x_l^j графов G_i и G_j не тождественны, а инцидентны. Очевидно, что одной и той же вершине x_k^i могут быть инцидентны несколько вершин $X_k^j = \{x_{k_1}^{j_1}, x_{k_2}^{j_2}, \dots, x_{k_l}^{j_l}\}$ из l графов $\{G_{j_s}\}, s = 1, \dots, l$. На множестве вершин X_k^j определяется $L^j = (X_k^j, E)$. Вершины $x_{k_j}^{j_i}$ и $x_{k_s}^{j_s}$ смежны в L^j , если соответствующие графы G_{j_i} и G_{j_s} в вершине x_k^i имеют некоторую системообразующую связь $l(x^{j_i}, x^{j_s})$. В противном случае эти вершины не связаны. Так же, как в гиперсетях, ребру $u_l^j \in G_j$ в графе G_i сопоставляется цепь или некоторая связанная часть между соответствующими вершинами из G_i . Здесь необходимо отметить, что системообразующие связи типа $\{l(x,y)\}$, вообще говоря, могут иметь разную природу и, как правило, существенно зависят от времени. В некоторых случаях, например, в сетях разного типа (кабельные, ВОЛС, лазерные линии связи и др.). В этом случае имеет смысл рассматривать объединение всех вторичных сетей. Однако для некоторых задач имеет смысл рассматривать сумму всех графов гиперсети H, включая и первичную сеть PS, т.е. $\underline{G} = G_0 + G_1 + \dots + G_n + \{L^j\}$. Более подробно об этом можно прочитать в [2].

S-гиперсеть состоит из разного типа сетей:

- иерархический граф в виде дерева с четырьмя уровнями (граф первичной сети);
- полносвязный граф между вершинами (узлы-сети), который вкладывается в граф первичной сети;

- цепочки различной природы, соответствующие технологическим последовательностям обработки различных сообщений;

- связанные компоненты для отображения очередей процессов обработки с обратными связями.

Кроме того, определяется оптимальное расписание работы элементов S-гиперсети с целью оптимальной упаковки цепей различного вида, предназначенных для оптимизации обработки потока сообщений.

В результате на построенной S-гиперсети и составленного расписания работы ее элементов можно находить многопродуктовый поток заданной величины и минимальной стоимости. При этом модель позволяет управлять многими параметрами и тем самым изучать ее различные свойства и показатели в зависимости от входного многопродуктового потока.

В качестве прототипа разрабатываемой модели на основе S-гиперсетей можно взять нестационарные абстрактные гиперсети [1]. Инструментальные возможности теории нестационарных S-гиперсетей позволяют строить адекватные структурные модели очень сложных инфокоммуникационных сетей и, в частности, построить математическую модель СРОД для решения широкого спектра структурных задач рассматриваемой системы.

Для описания динамических систем сетевой структуры вводится понятие нестационарной S-гиперсети, которое позволяет параметры, структуру и атрибуты S – гиперсети сделать зависимыми от времени. В этом случае над множеством S-гиперсетей определяется система управления, которая согласно заданному расписанию видоизменяет значения параметров S-гиперсети и существование ее элементов в определенный момент времени. Причем расписание работы S- гиперсети может быть как детерминированным, так и случайным. Вершины S-гиперсети являются не абстрактной точкой, а некоторой системой, способной перерабатывать входящий поток и хранить содержимое этого потока в определенном месте. Таким образом, определенный математический объект (нестационарная S-гиперсеть) можно подвергнуть классификации, приведенной ниже и подробно описанной в [2].

2. Отображения и классификация нестационарных S-гиперсетей

Многообразие отображений вторичных сетей в первичную сеть делает возможным отображение различных коллизий, возникающих между реальными сетями.

2.1. Отображения

При отображении графа вторичной сети WS в первичную сеть PS возникают четыре класса вложений ребер WS в ветви PS .

Для гиперсети $H = (PS, WS)$:

1. Ребра вторичной сети WS не отображаются в ребра первичной, т.е. отображаются только вершины WS в вершины PS . Таким образом, матрицы $N_j^i(b_p^d)$ в представлении гиперсети отсутствуют (рис.2в). Здесь имеют место экс-отображения $WS \rightarrow PS$ и соответственно **экс-гиперсеть**.

2. Если ребра вторичной сети $WS = (X^1, R)$ идут рядом (параллельно) с ветвями первичной сети $PS = (X^0, V)$, то имеет место пара-отображение $WS \xrightarrow{p} PS$ (рис.2с), которое порождает **пара-гиперсеть** $H = (WS, PS)$.

3. В том случае, когда ребра вторичной сети WS располагаются на «плоских» ветвях первичной сети PS , то имеет место экто-отображение $WS \xrightarrow{эк} PS$ (см. рис.2d) и соответственно **экто-гиперсети**.

4. В последнем случае ребра вторичных сетей располагаются внутри ветвей первичной сети. На рис.2е. приведен пример эндо-отображения $WS \xrightarrow{эн} PS$, которое порождает **эндо-гиперсеть**.

Вершины также могут по-разному отображаться друг в друга. Здесь так же, как с ребрами, имеем, те же четыре способа отображения:

1. Вершина вторичной сети x абстрактно отображается в вершину y первичной сети, если их взаимное расположение безразлично, т.е. имеет место **экс-отображение**.

2. Если эта вершина x отображается рядом с узлом y , то имеет место *пара-отображение*.
3. При *экто-отображении* вершин одна вершина располагается на другой.
4. И наконец, при отображении одной вершины вовнутрь другой имеет место *эндо-отображение* вершин.

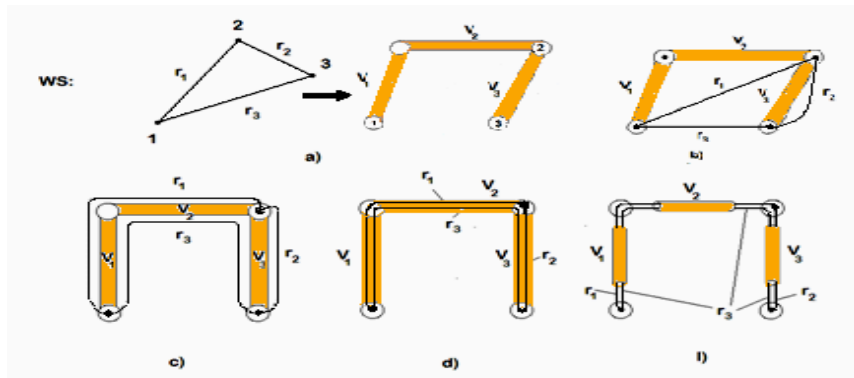


Рис.2. Примеры вложений вторичной сети в первичную

Соответственно, S-гиперсети можно называть согласно отображениям элементов. Очевидно, что в одной и той же S-гиперсети разные элементы одновременно могут отображаться в другие разными способами. Таким образом, формально определенное взаимоотношение вторичных сетей в первичной позволит более подробно и адекватно описывать взаимоотношение вторичных сетей в первичной.

2.2. Классификация S-гиперсетей

Широта применения математической теории определяется не только свойствами математического объекта, но его пластичностью. В частности, подробная классификация S-гиперсетей даёт представление о её возможности адекватно описывать различные системы для их структурного анализа и оптимизации.

Системная типология

СТ-1. Системная типология S- гиперсетей:

Классификационные признаки: характеристики S-гиперсетей

Типы вложений: экс- / пара- / экто- / эндо- гиперсеть

Размерность: конечная / бесконечная

Потоковая функция: дискретная / непрерывная

Атрибуты: детерминированная / случайная

Стабильность: стационарная / нестационарная

Структура гиперсети: каскадная / смешанная / параллельная

Разнообразие вторичных сетей: однотипная / K-типная

СТ-2. Системная типология первичных и вторичных сетей:

Классификационные признаки: характеристики сетей

Тип математического объекта: граф / гиперграф / гиперсеть

Ориентация: ориентированный / смешанный / неориентированный

Сложность: пустой / дерево / планарный / полный / произвольный

Однородность: однородный / неоднородный

Весовые характеристики: взвешенный / невзвешенный

Атрибуты: детерминированная / случайная

Стабильность: стационарная / нестационарная

СТ-3. Системная типология элементов:

Классификационные признаки: характеристики элементов

Типы элементов: узлы (вершины) / ветви (ребра) / полюса (полусвязь)

Весовые характеристики:	взвешенные / невзвешенные
Ориентация:	ориентированный / неориентированный
Время существования:	постоянно / по расписанию / случайное
Представление:	абстрактное / геометрическое
Число полюсов:	всех полюсов / свободных полюсов

СТ-4. Системная типология связей:

Классификационные признаки:	характеристики связей
Способ соединения:	слияние / инцидентность / слабая инцидентность / смежность
Ориентация:	ориентированный / неориентированный
Время существования связи:	постоянно / по расписанию / случайное
Надежность соединения:	с резервированием / без резервирования
Связь точечных элементов:	абстрактная / физическая
Связь элементов:	экс- / пара- / экто- / эндо- вложение

Топологическая типология

ТТ-5. Топологическая типология S-гиперсетей:

Классификационные признаки:	топологические характеристики S-гиперсетей
Размерность пространства:	одномерное / двумерное / трехмерное
Род поверхности первичной сети:	плоская / K-го рода
Число уровней в иерархии:	обычная гиперсеть / K уровней (слоев)
Число сетей на K-ом уровне:	N(K) число сетей на уровне K
Размерность:	конечная / бесконечная
Тип вложений вторичных сетей в первичной:	типы вложений в СТ-1

ТТ-6. Топологическая типология вторичных сетей

Классификационные признаки:	структурные характеристики вторичных сетей
Тип математического объекта:	граф / гиперграф / ультраграф / гиперсеть
Наличие циклов:	циклический / ациклический
Ориентация:	ориентированный / неориентированный
Род графа:	плоский / K-го рода
Связанность графов вторичных сетей:	связанные / независимые

ТТ-7. Топологическая типология элементов.

Классификационные признаки:	топологические характеристики элементов
Типы элементов:	узел (вершина) / ветвь (ребро) / полюс (полу-связь)
Размерность:	без размера / линейный / плоский / объемный
Число полюсов:	без полюсов / K-полюсный
Представление:	абстрактное / геометрическое

ТТ-8. Топологическая типология связей

Классификационные признаки:	топологические характеристики связей
Ориентация полюсов:	неориентированные / ориентированные
Число полюсов в элементах:	неоднородные / однородные
Число типов полюсов:	однотипные / разнотипные
Занятость полюсов:	занятые / свободные
Тип связи полюсов:	слияние / инцидентность / полуинцидентность

Приведенная классификация позволяет практически автоматизировать технологию синтеза S-гиперсетей для моделирования сложных динамических систем сетевой структуры. В частности, ис-

следование структурных и временных характеристик СРОД, связанных с прохождением данных по иерархической сети этой системы, становится возможным на специально синтезированных нестационарных S-гиперсетях. Такой подход позволяет создать унифицированное алгоритмическое обеспечение для решения различных задач на гиперсетях разного типа. Из данной классификации следует, что распределенная система обработки данных может быть промоделирована нестационарными S-гиперсетями с заданными характеристиками и расписанием. Из вышесказанного следует, что модель нестационарной S-гиперсети для СРОД имеет следующий вид:

$N(SH) = (SH, A, П, P, T)$, где:

SH – обобщенная структура S- гиперсети SH;

A – атрибуты и параметры SH;

П – память в элементах SH;

P – расписание (таблица значений атрибутов и параметров SH, расписанных на временной оси);

T – период времени работы модели.

Из этого определения следует, что реализация этой модели возможна только на компьютере, поэтому общая схема моделирования $N(SH)$ имеет вид:



Рис.3. Блок схема моделирования СРОД

Для данной формулировки обобщенной модели наиболее значимыми становятся задачи поиска кратчайшего пути или максимального потока за период T в нестационарной гиперсети $N(SH)$.

3. Модель СРОД на базе теории нестационарных S-гиперсетей

Теперь рассмотрим концептуальную модель СРОД с привлечением аппарата теории нестационарных S-гиперсетей. Вполне очевидно, что в общем случае речь идет о большом потоке информации величины l , протекающей по сети СРОД. При этом часть информации обрабатывается в сети СРОД, хранится в ней и пересылается (возможно с потерей объема) вверх. Другая, возможно более значительная, часть исчезает из СРОД (как ненужная).

Каждому ребру S-гиперсети SH сопоставляется тройка чисел:

c_{ij}^j - стоимость единицы потока в u_{ij} ребре в WH^1 вторичной сети;

t_{ij} – время прохождения потока в u_{ij} ребре в WH^1 вторичной сети;

$b(V_i)$ – ограничение на суммарный поток в v_i ребре;

x_{ij} – поток в u_{ij} ребре.

Таким образом, при данном условии можно сформулировать две задачи прохождения информации через СРОД.

3.1. Задачи СРОД

1. Найти заданный s, h – поток величины l за минимальную стоимость. Соответствует задаче минимизации используемых ресурсов (памяти, вычислительных, канальных) при обнаружении m - сообщений за заданное время Δt (ресурсоэффективность).

$$\text{Минимизировать } \varphi = \sum c_{ij} x_{ij} \quad (1.1)$$

при условиях

$$\sum_i x_{ij} - \sum_k x_{jk} = \begin{cases} -l, j = s \\ 0, j \neq s \\ l, j = h \end{cases} \tag{1.2}$$

$$0 \leq \sum_{x_{ij} \in F_z, (U_{ij})=V_z} x_{ij} \leq b(V_z) \tag{1.3}$$

$$\rho_t^{W^*}(s, h) \leq \Delta t. \tag{1.4}$$

Ограничения (1.2) стандартны для задач поиска потоков. В нашем случае это ограничение поддерживается тем, что в S-гиперсети SH вводится вторичная сеть, на которую уходит поток информации, не нужный для дальнейшей обработки.

Ограничение (1.3) гарантирует, что суммарная величина потока через вершину или ветвь V_z не превосходит заданного значения в (V_z) . Здесь под суммарной величиной потока понимается либо одновременный объем информации, обрабатываемый в узле V_z , либо в ветви V_z , (где $\{V_z\}$ – множество элементов первичной сети).

Ограничение (1.4) показывает, что в полученном графе потоков W^* расстояние по времени между вершинами s,t не превосходит заданную величину Δt .

2. Найти заданный s,h – поток, проходящий сеть SH за минимальное время. Данная постановка соответствует минимизации времени обнаружения сигналов на заданной совокупности M – сообщений. Чем меньше заданный поток V, тем точнее для заданных M – сообщений решение.

Формально задача ставится следующим образом:

$$\text{минимизировать } \psi = \rho_t^{W^*}(s, h) \tag{2.1}$$

при условиях: в найденном графе потока W^* максимальный

$$m(s, h) \geq V, \tag{2.2}$$

где V – заданная величина потока,

$m(s, h)$ – максимальный поток,

$\rho_t^{W^*}(s, h)$ – длина кратчайшей по времени s,h –цепи в найденном потоковом графе W^* .

Условия (1.2) и (1.3) остаются верными и для этой задачи. Относительно стоимости имеет место ограничение

$$\sum c_{ij} x_{ij} \leq C. \tag{2.3}$$

Т.е. суммарная стоимость всего потока из s в h в графе W^* не превосходит величину C.

Так как в задачах присутствуют время и объем пропуска обрабатываемой информации, то мы имеем дело с нестационарной S-гиперсетью.

3.2. Структура сети СРОД

Все подсистемы СРОД с сетевой составляющей имеют свой аналог в нестационарной S-гиперсети. В частности:

1. Каждому функциональному блоку программ Π_i сопоставляется вторичная сеть $W\Pi_i = (x^i, u^i)$.
2. Поток данных, состоящий из различных по типу подпотоков PD_j , направляется на соответствующие данному типу сети обработки данных WOD_j (блок ПТО).
3. Сети $\{W\Pi_i\}$ и $\{WOD_j\}$ объединяются в одну структуру WO для обработки (передачи) входной информации, которая является аналогом ПУВР.
4. Так как в нестационарной гиперсети очередность срабатываний, показатели, атрибуты и т.д. зависят от времени, т.е. определяются расписанием, то процедура его составления и модификация определяется системой управления CV(SH). Здесь можно заметить, что аналогом такой системы управления является нейроэндокринная система человека.
5. Так как во всех алгоритмах существенно используется тот факт, что в любой вершине объем входящей информации равен объему исходящей, то для поддержки этого баланса вводится ориентированная сеть утилизации WU, которая определена на том же множестве вершин, что и обрабатывающий граф. Пропускные возможности ребер этой сети практически неограничены. Однако именно этим параметром (пропускной способностью) можно регулировать процент обрабатываемой информации.

6. Управляющая сеть WY в данной ситуации создает дополнительную нагрузку на обрабатывающую сеть WO, а управляющее воздействие осуществляет CY(SH) через таблицы расписаний для SH.

Отсюда следует, что нестационарная S-гиперсеть HS имеет структуру $HS = (PS, \{W\Pi_i\}, \{WOD_j\}, WO, WU, WY)$. Структура отображений определяется отношениями в СПОД и будет определена в процессе реального компьютерного моделирования HS.

Заключение

В работе рассматривается задача обработки в реальном времени информации, проходящей в различных информационных системах на больших территориях, например в системах Интернет, радио, телевидения и т.п. Такие системы представляют интерес для рекламных и информационных агентств и многих других служб, для которых необходима регулярная обработка массовой информации.

Литература

1. Попков В.К. Математические модели связности. – Новосибирск: Изд-во ИВМиМГ СО РАН, 2006. – 490 с.
2. Попков В.К. Применение теории S-гиперсетей для моделирования систем сетевой структуры // Проблемы информатики. – 2010. – №4. – С. 17-40.

Попков Глеб Владимирович, кандидат технических наук, доцент, старший научный сотрудник Института вычислительной математики и математической геофизики СО РАН, моб. тел. 89134789130, e-mail: glebpopkov@rambler.ru

Попков Владимир Константинович, доктор физико-математических наук, профессор, главный научный сотрудник Института вычислительной математики и математической геофизики СО РАН, моб. тел. 89139426496, e-mail: popkov@sscc.ru

Popkov Gleb Vladimirovich, candidate of technical sciences, associate professor, senior researcher, Institute of Computing Mathematics and Mathematical Geophysics SB RAS, mob.ph.: 89134789130, e-mail: glebpopkov@rambler.ru

Popkov Vladimir Konstantinovich, doctor of physical and mathematical sciences, professor, chief researcher, Institute of Computing Mathematics and Mathematical Geophysics SB RAS, mob. ph.: 89139426496, e-mail: popkov@sscc.ru

УДК 621.391

© *М.В. Харинов*

МОДЕЛЬ ЛОКАЛИЗАЦИИ ОБЪЕКТОВ НА ЦИФРОВОМ ИЗОБРАЖЕНИИ

Статья посвящена проблеме автоматического выделения объектов на цифровом изображении или «проблеме сегментации», эффективная формализация которой по постановке задачи в настоящее время ограничивается случаями простейших изображений типа «объект-фон».

Ключевые слова: пиксели изображения, кластеризация, слияние, дробление, модель Мамфорда–Шаха.

© *M.V. Kharinov*

A MODEL FOR LOCALIZATION OF OBJECTS IN DIGITAL IMAGES

The article is devoted to the problem of automatic detection of objects in digital images or so-called “segmentation problem”. At present the effective formalization of problem statement is available only in the case of simplest images of “object-background” type.

Keywords: image pixels, clustering, merging, splitting, Mumford–Shah model.

Введение

Проблему локализации можно охарактеризовать как проблему разбиения обрабатываемого изображения на изображения объектов, подлежащие дальнейшему признаковому анализу и идентификации автоматической системой распознавания. Из-за недостаточной формализации понятия «изображение» в настоящее время данная проблема относится к числу недостаточно формализованных и обычно рассматривается для изображений из заданной предметной области или решается в контексте распознавания заданных объектов. Тем не менее в случае простейших изображений объектов на однородном фоне она, как правило, не возникает. Поэтому имеет непосредственный смысл разработка алгоритмов, которые позволили бы установить взаимосвязь решения общей проблемы локализации объектов с решением частной задачи их выделения на однородном фоне.

Целью статьи является обсуждение вычислительной модели локализации объектов на цифровом изображении, разработанной в предположении иерархической природы объектов посредством обобщения и развития вычислений в методах Оцу [1-3], модели Мамфорда–Шаха [4-10] и методе K -средних [11-19], реализуемых на современных компьютерах.

1. Постановка задачи

Перечисленные подходы, в отличие от большинства других подходов к выделению объектов на цифровых изображениях, наиболее формализованы по постановке задачи, так как при построении разбиений опираются на определенный критерий качества. Критерием качества в методах Оцу и методе K -средних, а также в версии [7] модели Мамфорда–Шаха служит суммарная квадратичная ошибка E , или равносильная величина среднеквадратичного отклонения σ значений пикселей изображения от значений пикселей оцениваемого кусочно-постоянного приближения.

Метод Оцу [1-3] является одним из немногих, если не единственным методом точного вычисления оптимальных приближений, минимально отличающихся от изображения по суммарной квадратичной ошибке. Это классический метод кластеризации пикселей изображения посредством нахождения по яркостной гистограмме одного или нескольких пороговых значений яркости, которые разбивают шкалу яркости на диапазоны из условия минимального значения среднеквадратичного отклонения средних по диапазонам значений пикселей от значений пикселей исходного изображения. В методе Оцу одинаковые пиксели изображения всегда относят к одному кластеру, состоящему из некоторого числа сегментов (связных областей) изображения. То, что метод Оцу не позволяет относить одинаковые пиксели к различным кластерам, препятствует его использованию для локализации объектов посредством классификации пикселей в зависимости от особенностей их локального распределения. Принципиальным ограничением метода Оцу является то, что он не имеет очевидного обобщения на

случай классификации многомерных пикселей цветовых или многоспектральных изображений.

Метод K -средних является старейшим методом многомерного кластерного анализа [17-18] и интенсивно применяется не только в обработке изображений, но также и в ряде других областей, связанных с классификацией и распознаванием данных. В приложении к изображениям он сводится к итеративной коррекции разбиений пикселей на кластеры или сегменты посредством реклассификации пикселей из условия минимизации расстояния до центроидов в виде векторов (столбцов), покомпонентно усредненных по кластерам яркостных значений. В учебной и научной литературе принято обсуждать метод K -средних без предварительного аналитического обоснования. Между тем элементарные выкладки, приводимые в следующем разделе, показывают, что минимизация расстояний до центроидов кластеров не равносильна минимизации суммарной квадратичной ошибки. Поэтому для оптимизации приближений вместо метода K -средних оказывается эффективным применять метод коррекции разбиений в уточненном варианте [19].

Традиционная модель Мамфорда–Шаха [4-9] относится к теоретически и экспериментально обоснованным моделям сегментации изображения посредством итеративного слияния сегментов. Отождествление классов со связными сегментами, однако, ограничивает возможность отнесения пикселей экземпляров объекта и частей объекта к одному классу. Условие связности пикселей поддерживается в модели автоматически, благодаря использованию единственной операции слияния смежных сегментов, которая не выводит из их множества. С другой стороны, использование единственной операции слияния сегментов при последовательной генерации разбиений изображения опирается на предположение, что последовательность оптимальных разбиений изображения является иерархической. Поскольку последовательность разбиений изображения, минимизирующих E или σ , вообще говоря, отличается от иерархической [10], единственность операции слияния сегментов является принципиальным недостатком вычислений в традиционной модели Мамфорда–Шаха [4-9], который полезно исправить.

Характерной особенностью обсуждаемых классических методов сегментации изображений посредством оптимизации качества разбиений по суммарной квадратичной ошибке E или среднеквадратичному отклонению σ является прозрачность алгоритмов и необходимых вычислительных формул, которая часто утрачивается в многочисленных приложениях при эвристическом учете свойств «объектов интереса» и оптимизации итерационных вычислений по скорости. При этом постановочное условие оптимизации качества разбиений по E (или σ) обычно не контролируется в качестве обязательного критерия достоверности вычислений. В отличие от большинства практических решений для конкретных предметных областей, мы строим унифицированный алгоритм локализации объектов только на основании элементарных аналитических соотношений, непосредственно связанных с минимизацией целевого функционала.

2. Рабочие формулы

Рассмотрим приращение ΔE суммарной квадратичной ошибки E при *реклассификации* многомерных пикселей, т. е. исключении k пикселей, характеризуемых вектором средних значений I , из числа n_1 пикселей кластера 1, характеризуемого вектором средних значений яркости I_1 , и отнесении их к некоторому другому кластеру.

Случай $k = n_1$ сопоставляется включению всех n_1 пикселей кластера 1 в число пикселей некоторого ранее выделенного на изображении кластера 2, т. е. слиянию кластеров 1 и 2, которое описывается неотрицательным приращением суммарной квадратичной ошибки ΔE_{merge} :

$$\Delta E_{merge} = \frac{\|I_1 - I_2\|^2}{\frac{1}{n_1} + \frac{1}{n_2}} \geq 0, \quad (1)$$

где знак $\|\cdot\|^2$ обозначает суммирование квадратов отклонений $I_1 - I_2$ (евклидову норму векторов по цветовым или спектральным компонентам).

В случае $k < n_1$ предусматриваются две возможные операции реклассификации подмножества пикселей кластера 1 – дробление и коррекция.

При операции *дробления* кластера надвое k пикселей из кластера 1 выделяются в новый кластер. При этом кластер 1 преобразуется в кластер из $n_1 - k$ пикселей, которые дополняли обсуждаемые k пикселей в составе прежнего кластера. Операция дробления кластера сопровождается отрицательным, или нулевым, приращением суммарной квадратичной ошибки ΔE_{split} :

$$\Delta E_{split} = -\frac{\|I - I_1\|^2}{\frac{1}{k} - \frac{1}{n_1}} \leq 0. \tag{2}$$

При операции *коррекции* часть пикселей реклассифицируется из кластера–*донора* 1 в *акценторный* кластер 2, что сопровождается приращением суммарной квадратичной ошибки $\Delta E_{correct}$:

$$\Delta E_{correct} = \frac{\|I - I_2\|^2}{\frac{1}{k} + \frac{1}{n_2}} - \frac{\|I - I_1\|^2}{\frac{1}{k} - \frac{1}{n_1}}. \tag{3}$$

При этом второе, отрицательное, слагаемое в (3) описывает приращение суммарной квадратичной ошибки при выделении k пикселей из кластера 1 в самостоятельный кластер, а первое слагаемое описывает ее приращение при слиянии выделенного кластера с кластером 1 согласно (1) и (2). Таким образом, операция коррекции трактуется как составная.

Из (3) получаем для коррекции критерий убывания суммарной квадратичной ошибки E :

$$\|I - I_1\| > \alpha \cdot \|I - I_2\|, \tag{4}$$

где коэффициент $\alpha < 1$ описывает соотношение количества пикселей в рассматриваемых кластерах 1, 2 и выражается в виде:

$$\alpha \equiv \sqrt{\frac{n_2(n_1 - k)}{n_1(n_2 + k)}}. \tag{5}$$

Если коэффициент α в формуле (4) положить равным 1, то из нее выводится метод K –средних. Однако если не делать этого упрощения, которое практически не влияет на скорость вычислений на современных компьютерах, то, опираясь на формулу (4) с учетом коэффициента α , удается получить более сильный метод реклассификации пикселей [19]. Он позволяет избегать преждевременного останова минимизации суммарной квадратичной ошибки, точнее выбирать варианты реклассификации пикселей и эффективнее снижать величину суммарной квадратичной ошибки за счет реклассификации не отдельных пикселей, а их множеств.

3. Модель устойчивой локализации объектов

В нашем понимании изображение состоит из вложенных изображений. Поэтому наглядные понятия «изображение», «объект» и «кластер пикселей» принципиально не разделяются друг с другом.

Локализация объектов сводится к получению последовательности разбиений изображения на вложенные изображения из условия минимизации суммарной квадратичной ошибки или среднеквадратичного отклонения. Искомые разбиения содержат кластеры пикселей. Кластеры пикселей из различных разбиений, вообще говоря, перекрываются между собой. Допускается, что кластер пикселей может содержать более одного *сегмента* (связной области, компоненты связности). Поэтому операции дробления и коррекции, которые, вообще говоря, нарушают связность сегментов, не выводят из рассматриваемого множества кластеров.

Если два соседствующих на изображении пикселя принадлежат различным кластерам, то эти кластеры считаются *смежными* – так же, как и в случае сегментов.

Кластер содержит один или несколько несмежных сегментов. Каждый кластер из нескольких сегментов разделяется на два вложенных кластера без нарушения целостности сегментов. В свою очередь каждый сегмент, содержащий более одного пикселя, разделяется на два вложенных сегмента. Разделение каждого сегмента надвое задается бинарной иерархической последовательностью разбиений, которая порождается итеративным слиянием смежных сегментов и выполняется по формуле (1) до слияния всех исходных пикселей в один сегмент. Затем формирование бинарной иерархической последовательности разбиений кластера на составные части продолжается посредством итеративного

попарного объединения сегментов до их слияния в один кластер. При этом для каждого сегмента, как для отдельного изображения, выполняется укрупнение вложенных смежных сегментов в модели Мамфорда–Шаха [7, 10], очередная пара которых выбирается по критерию слияния, которым служит минимальная величина ΔE_{merge} :

$$\Delta E_{merge} = \min , \quad (6)$$

а затем тот же самый критерий применяется для объединения сегментов посредством вычисления значений ΔE_{merge} для каждой пары сегментов, составляющих рассматриваемый кластер. В целом для каждого кластера из p пикселей обсуждаемая бинарная иерархическая последовательность разбиений задает $2p - 1$ составных частей (вложенных кластеров, сегментов и их связанных компонент).

Для детализации разбиения изображения посредством дробления кластера на составную часть и ее дополнение используется формула (2). Критерием служит условие максимального падения суммарной квадратичной ошибки, т. е. ее минимального приращения ΔE_{split} :

$$\Delta E_{split} = \min . \quad (7)$$

В результате дробления кластера общее число кластеров увеличивается на единицу, а число сегментов увеличивается или остается прежним.

Формула (3) применяется для итеративного снижения суммарной квадратичной ошибки посредством циклически повторяемой коррекции пар кластеров. Выполняется до тех пор, пока на изображении обнаруживаются кластер-акцептор и кластер-донор, имеющий смежную с кластером-акцептором составную часть, реклассификация которой в кластер-акцептор приводит к падению суммарной квадратичной ошибки. Из множества вариантов требуемой реклассификации выбирается вариант, сопровождаемый минимальным приращением суммарной квадратичной ошибки $\Delta E_{correct}$:

$$\Delta E_{correct} = \min < 0 . \quad (8)$$

В процессе коррекции разбиения изображения на кластеры общее число кластеров не меняется. Если при коррекции реклассифицируемая составная часть кластера-донора содержит единственный сегмент, то число сегментов в кластере-акцепторе также остается прежним.

В результате коррекции получается *устойчивое* разбиение изображения на кластеры, которое нельзя улучшить по суммарной квадратичной ошибке посредством реклассификации составной части любого кластера в другой, смежный с этой частью.

Если в изложенной схеме коррекции разбиения изображения на кластеры допускается реклассификация любой части кластера-донора, необязательно смежной с кластером-акцептором, то такая коррекция называется *безусловной*. При безусловной коррекции разбиения изображения число кластеров сохраняется, но возрастает число сегментов в кластерах-акцепторах за счет слияния кластеров-акцепторов с несмежными составными частями кластеров-доноров. В результате безусловной коррекции получается разбиение изображения на кластеры, которое нельзя улучшить по суммарной квадратичной ошибке посредством реклассификации любой предусмотренной составной части одного кластера в другой.

Вычисления проводятся в алгоритме «наращивания кластеров», в котором коррекция разбиения изображения выполняется по очереди с увеличением числа кластеров на единицу за счет дробления кластера, отвечающего максимальному падению суммарной квадратичной ошибки. При этом генерируется последовательность устойчивых разбиений изображений на кластеры для любого начального разбиения (например, содержащего единственный сегмент, покрывающий все изображение) или для начального разбиения изображения на несколько сегментов. Во втором случае применение вместо дробления кластеров безусловной коррекции обеспечивает детализацию начального разбиения изображения за счет увеличения числа сегментов при неизменном числе кластеров.

Таким образом, в нашей интерпретации, локализация объектов реализуется благодаря вычислению последовательности кусочно-постоянных приближений изображения, порождаемых его устойчивыми разбиениями на кластеры. Кластеры пикселей изображения разделяются на составные части, образуемые сегментами (связными областями) изображения. При построении устойчивых разбиений кластеры обмениваются сегментами из условия снижения суммарной квадратичной ошибки E или среднеквадратичного отклонения σ . В процессе построения устойчивого разбиения при модифика-

ции кластеров обновляется и сопоставляемая каждому кластеру иерархия его составных частей. Поэтому устойчивые кластеры пикселей изображения характеризуются не только устойчивостью относительно обмена составными частями друг с другом, но и соответствием составных частей целостным кластерам.

4. Экспериментальные результаты

Экспериментальные результаты по локализации объектов в графическом виде демонстрируются на рис. 1.

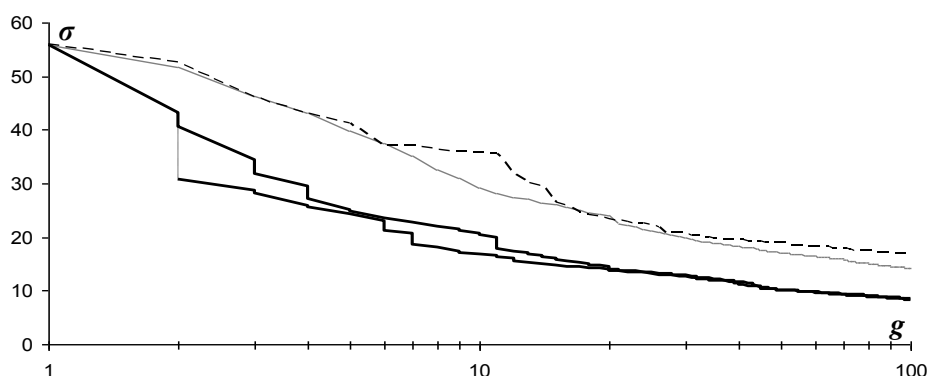


Рис. 1. Оценка качества разбиения в зависимости от числа кластеров

Рис. 1 на примере стандартного изображения «Лена»¹ иллюстрирует среднеквадратичное отклонение σ изображения от его кусочно–постоянных приближений в зависимости от числа кластеров g , которое совпадает с числом градаций средней яркости. Графики изображены в логарифмическом масштабе по горизонтальной оси в диапазоне числа кластеров от 1 до 100.

Две верхних переплетающихся кривых описывают иерархические разбиения на сегменты, полученные по версии FLSA модели Мамфорда–Шаха [8, 9] (верхняя штрихованная кривая) и по версии [7] (сплошная серая кривая). Они отражают известные результаты и приведены для наглядной демонстрации эффекта снижения среднеквадратичного отклонения σ при генерации устойчивых разбиений.

Пара нисходящих сливающихся друг с другом черных кривых описывает процесс автоматической генерации последовательности устойчивых разбиений изображения посредством чередования увеличения числа кластеров на 1 за счет дробления одного из них и циклической коррекции разбиения до преобразования его в устойчивое разбиение изображения. Верхняя кривая соответствует случаю тривиального начального разбиения, содержащего единственный кластер. Нижняя кривая соответствует начальному разбиению изображения на два кластера из нескольких сегментов. При этом каждому значению числа кластеров g на той и другой кривой соответствуют по две точки, которые для отдельных значений g на графиках разделяются, а для большинства остальных сливаются между собой, т. к. незначительно различаются по σ . Устойчивые разбиения изображения описываются для каждого значения g минимальным значением σ .

Отрезок вертикальной пунктирной прямой на рис. 1 начинается от минимального значения σ для $g = 2$ на верхней нисходящей кривой. Он описывает циклическую детализацию соответствующего инвариантного разбиения изображения по числу сегментов при неизменном числе кластеров в алгоритме безусловной коррекции. В результате безусловной коррекции получается устойчивое разбиение изображения на два кластера, которое характеризуется значением среднеквадратичного отклонения, равным $\sigma = 30.64564$. Характерно, что полученное разбиение совпадает с оптимальным разбиением изображения на два кластера пикселей [3, 7], которое вычисляется классическим методом Оцу [1]. Фактически обсуждаемый отрезок представляет 182 точки, сгущающиеся по мере приближения к

¹ URL: <http://sipi.usc.edu/database/>

оптимальному значению. При этом плавное снижение σ наглядно выражается в проявлении шумов на соответствующих кусочно–постоянных приближениях изображения. Поэтому для испытания алгоритма поочередного дробления и коррекции при нетривиальном начальном разбиении изображения (нижняя нисходящая кривая на рис. 1) в качестве начального устойчивого разбиения выбрано не оптимальное разбиение, а разбиение, полученное с подавлением шумов, которое получается на 11-й итерации безусловной коррекции и отвечает значению $\sigma = 30.79373$.

Для уточнения данных первые десять пар численных значений σ (для верхней нисходящей кривой) и σ' (для нижней нисходящей кривой), отвечающие устойчивым разбиениям изображения, перечислены для последовательных значений числа кластеров g в таблице 1 (по пять штук в каждой колонке).

Таблица

Значения σ и σ' для последовательных разбиений изображения

№	g	σ	σ'	№	g	σ	σ'
1	1	55.88322		11	6	23.62842	22.93591
2	1	55.88322		12	6	23.51916	21.17198
3	2	43.16748	30.79373	13	7	22.82637	20.56156
4	2	40.58435	30.79373	14	7	22.73704	18.64853
5	3	34.39798	28.72376	15	8	22.05335	18.04458
6	3	31.85193	28.08163	16	8	21.99891	18.01739
7	4	29.55488	25.85536	17	9	21.39022	17.43846
8	4	27.04307	25.73018	18	9	21.25647	17.19645
9	5	25.09586	24.31218	19	10	20.59120	16.82263
10	5	24.86835	24.21468	20	10	20.55978	16.77930

Значения среднеквадратичного отклонения, указанные в таблице 1 в строках с четными номерами, отвечают устойчивым разбиениям изображения. Сами устойчивые разбиения, которые наглядно описываются посредством кусочно-постоянных приближений изображения, представлены на рис. 2.



Рис. 2. Устойчивые разбиения изображения

В правом верхнем углу рис. 2 показано стандартное изображение «Лена». В первой колонке демонстрируются устойчивые разбиения изображения с числом кластеров от одного до пяти, полученные чередованием дробления очередного сегмента и циклической коррекции при тривиальном начальном разбиении изображения, отвечающего значению $\sigma = 55,88322$. Во второй колонке показаны устойчивые разбиения, полученные по тому же алгоритму, но при нетривиальном начальном разбиении.

нии изображения, которое отвечает значению $\sigma = 30.79373$ и размещено под стандартным изображением.

Графически последовательность разбиений рис. 2 описывается огибающими каждой из черных сливающихся кривых на рис. 1, которые строятся по точкам с меньшими значениями σ . По поводу рис. 2, следует заметить, что вычисляемые для кластеров значения средней яркости являются вещественными числами, которые при визуализации округляются до целых и могут не различаться при печати.

5. Интерпретация

Исходя из результатов эксперимента, можно утверждать, что по сравнению с традиционной моделью Мамфорда–Шаха, предложенная модель локализации объектов на цифровом изображении обеспечивает заметное улучшение кусочно-постоянных приближений по среднеквадратичному отклонению σ , (рис. 1), причем, без резкого увеличения числа сегментов, как в методе Оцу [3, 7]. При этом улучшение качества приближений наглядно выражается в более полном и точном «проявлении» объектов в приближениях изображения с ограниченным числом градаций яркости (рис. 2).

Характерно, что при генерации разбиений в алгоритме «наращивания кластеров» от различных начальных разбиений изображения, различие по σ результирующих устойчивых разбиений нивелируется, и соответствующие кривые на рис. 1 сближаются друг с другом, по-прежнему, заметно выигрывая по σ у традиционной модели. На изображении при этом выделяются сходные совокупные наборы объектов.

Отличительной особенностью пары сливающихся кривых на рис. 1, описывающих процесс получения последовательности устойчивых разбиений изображения, является наличие участков плавного снижения среднеквадратичного отклонения σ , разделенных его резкими перепадами. Резкое снижение σ описывает существенное изменение очередного разбиения изображения при модификации нескольких кластеров и заметном смещении их границ, как в верхних четырех приближениях слева на рис. 2. Плавное снижение σ описывает последовательное проявление на изображении очередного объекта при незначительной модификации границ, как в колонке приближений справа на рис. 2, размещенных под исходным изображением. Последнее дает основание предположить, что изображение имеет «кусочно–иерархическую» структуру, т.е. аппроксимируется последовательностями иерархических приближений по диапазонам кластеров, которые можно вычислить, детектируя участки плавного снижения среднеквадратичного отклонения σ на графиках рис. 1. По нашему мнению, сформулированная гипотеза представляет самостоятельный интерес для дальнейшего теоретического и экспериментального исследования.

Заключение

Таким образом, в настоящей статье мы развиваем оптимизацию приближений изображения для локализации и выделения объектов на основе модели Мамфорда–Шаха [7], в которой из соображений минимизации суммарной квадратичной ошибки E (среднеквадратичного отклонения σ) утроили число операций с множествами пикселей и перешли от вычислений со связными сегментами к вычислениям с кластерами из несмежных сегментов, порождаемыми дополнительными операциями дробления и коррекции кластеров пикселей изображения. Для оптимизации разбиений изображения по E (или σ) разработан и испытан алгоритм «наращивания» кластеров, аналитически выведенный из условий минимизации суммарной квадратичной ошибки (1)-(3) и (6)-(8), в котором именно соображениями минимизации целевого функционала определяется как порождаются кластеры, как они разделяются на составные части, а также как обмениваются составными частями друг с другом. При этом для минимизации функционала с сохранением числа кластеров применен метод многомерной кластеризации, который по эффективности превосходит метод K -средних, а в одномерном случае обеспечивает воспроизведение результатов, полученных классическим методом Оцу [1], что позволяет применять этот метод в качестве многомерного аналога метода Оцу.

В настоящее время продолжают работу по совершенствованию предложенной модели и ее применению для унификации автоматической локализации и выделения объектов на цифровых изображениях [20].

Литература

1. Otsu N. A Threshold Selection Method from Gray-Level Histograms // IEEE Transactions on systems, MAN, and CYBERNETICS. 1979. Vol. SMC-9, January. – №. 1. – P. 62-66.
2. Ping-Sung Liao, Tse-Sheng Chen, Pau-Choo Chung A Fast Algorithm for Multilevel Thresholding // J. Inf. Sci. Eng. 2001. Vol. 17 (5), P. 713–727.
3. Харинов М.В. Устойчивая сегментация изображения // Вестн. Бурят. гос. ун-та. – 2012. – Вып. 9. – С. 64-69.
4. Mumford D., Shah J. Boundary detection by minimizing functionals, I // Proc. IEEE Comput. Vision Patt. Recog. Conf., San Francisco. – 1985. – P. 22-26.
5. Mumford D., Shah J. Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems // Communications on Pure and Applied Mathematics. 1989. Vol. XLII, № 4. P. 577-685.
6. Koepfler G., Lopez C., Morel J. A Multiscale Algorithm for Image Segmentation by Variational Method // SIAM Journal on Numerical Analysis. – 1994. – Vol. 31, № 1. – P. 282–299.
7. Бугаев А.С., Хельвас А.В. Поисковые исследования и разработка методов и средств анализа и автоматического распознавания потоковой информации в глобальных информационных системах. Шифр «Лацкан» // Отчет по НИР. – М.: Изд-во МФТИ, 2001. – Т. 1. – 140 с.
8. Redding N.J. et al. An efficient algorithm for Mumford–Shah segmentation and its application to SAR imagery // Proc. Conf. Digital Image Computing Techniques and Applications (DICTA '99). 1999. P. 35–41.
9. Robinson B.J., Redding N.J., Crisp D.J. Implementation of a fast algorithm for segmenting SAR imagery // Scientific and Technical Report, Australia: Defense Science and Technology Organization, Australia. 2002. 1 January.
10. Харинов М.В. Устойчивая сегментация цифрового изображения // Компьютерная графика и зрение (Графикон'2012): материалы XXII Междунар. конф. по компьютерной графике и зрению. М.: Изд-во МГУ им. М. В. Ломоносова, 2012. С. 208–213.
11. Steinhaus H. Sur la division des corps materiels en parties // Bull. Acad. Polon. Sci., 1956. С1. III Vol. IV. P. 801–804.
12. Lloyd S.P. Least squares quantization in PCM // IEEE Transactions on Information Theory. 1982. № 28 (2). P. 129–137.
13. MacQueen J.B. Some Methods for classification and Analysis of Multivariate Observations // Proc. Fifth Berkeley Symp. Math. Stat. and Probab., Berkeley: University of California Press, 1967. Vol. 1. P. 281–297.
14. Likas A., Vlassis N. and Verbeek J. The Global K –Means Clustering Algorithm // Pattern Recognition. 2003. Vol. 36. P. 451–461.
15. Jain A.K., Murthy M.N., Flynn P.J. Data Clustering A Review // ACM Computing Surveys. 1999. Vol. 31, № 3. P. 264–323.
16. Jain A.K. Data Clustering: 50 Years Beyond K –Means // Pattern Recognition Letters. 2010. Vol. 31, № 8. P. 651–666.
17. Мандель И.Д. Кластерный анализ. – М.: Финансы и статистика. 1988. – 176 с.
18. Айвазян С.А. и др. Прикладная статистика: классификация и снижение размерности. – М.: Финансы и статистика, 1989. – 607 с.
19. Kharinov M.V. Reclassification formula that provides to surpass K –means method // arXiv preprint , arXiv:1209.6204, 28 Sep 2012. 10 p. URL: <http://arxiv.org/ftp/arxiv/papers/1209/1209.6204.pdf>
20. Харинов М.В. Обобщение трех подходов к оптимальной сегментации цифрового изображения // Труды СПИИРАН. – 2013. – Вып. 2(25). С. 294–316.

Харинов Михаил Вячеславович, кандидат технических наук, старший научный сотрудник лаборатории прикладной информатики Санкт-Петербургского института информатики и автоматизации, тел: (812) 3281919, e-mail: khar@iias.spb.su

Kharinov Mikhail Vyacheslavovich, candidate of technical sciences, senior researcher, laboratory of applied computer science, St. Petersburg Institute for Informatics and Automation RAS, e-mail: khar@iias.spb.su

Содержание

1. Управляемые системы и методы оптимизации

Колмакова А.И. Многометодная оптимизация управления в экономической модели выбора налоговой ставки для предприятий энергетической отрасли.....	3
Медвежонков Д.С. Экспериментальные исследования алгоритмов внутренних точек на нелинейных задачах потокораспределения.....	12
Мижидон А.Д., Мижидон К.А. Задача оптимального управления линейной системой при фазовых и смешанных ограничениях.....	17
Орлов А.В. Гибридный генетический алгоритм глобального поиска оптимистических решений в задачах двухуровневой оптимизации.....	25
Расина И.В., Усенко О.В. Достаточные условия оптимальности в дискретной иерархической модели.....	33
Хишектеуева И.-Х.Д. Алгоритм оптимизации параметров нелинейных динамических систем.....	39

2. Информационные системы и технологии

Абасов Н.В., Чернышов М.Ю., Каверзина А.В., Осипчук Е.Н., Чернышова И.А. Программный комплекс для синтеза прикладных программных моделей и метамоделей на основе принципов математического программирования.....	45
Бименова Ж.Б., Джумамуратов Р.А., Сидорова Е.А. Подход к построению русско-казахского тезауруса по информатике.....	53
Верхотурова Ю.С. Модель предметной области на языке описания онтологий.....	63
Курганская Г.С. Облачные технологии интернет-образования на основе KFS модели представления знаний.....	69
Ружников Г.М., Хмельнов А.Е., Гаченко А.С. Создание и развитие ГИС «Инвестора» г. Иркутска.....	76
Фаткиева Р.Р. Разработка метрик для обнаружения атак на основе анализа сетевого трафика.....	81
Федорченко Л.Н. Синтаксически управляемая обработка данных для практических задач.....	87
Фереферов Е.С., Хмельнов А.Е. Автоматизация создания пользовательского интерфейса на основе модели приложения баз данных.....	100
Шигаров А.О. Проект интеллектуальной системы извлечения табличной информации из неструктурированных текстов.....	110

3. Функциональный анализ и дифференциальные уравнения

Алсыкова А.А. Начально-краевая задача с интегральными граничными условиями для одного класса уравнений составного типа.....	119
Мижидон А.Д., Баргуев С.Г. Краевая задача для одной гибридной системы дифференциальных уравнений.....	130
Новиков А.Е., Новиков Е.А., Кнауб Л.В. Контроль устойчивости метода Ческино второго порядка точности.....	138
Шишкин Г.А. Функция гибкой структуры и ее модификация при решении краевых задач для уравнений с функциональным запаздыванием.....	144

4. Математическое моделирование

Баранов В.И., Голенко Н.Н., Компаниец Л.А., Пака В.Т., Якубайлик Т.В. Пространственно-временная изменчивость основных характеристик озера Шира в сезоне наблюдений 2011-2012 гг. ..	148
Бубенчиков А.М., Базаров К.Л., Буянтуева З.Г. Компьютерная графика, порожденная линейными преобразованиями координат.....	157
Данчев В.Н., Пушистов П.Ю. Количественная оценка прогностического потенциала информационно-вычислительного комплекса «Телецкое озеро».....	161
Дондоков З.Б.-Д., Булдаев А.С. Об одной модификации модели межотраслевого баланса.....	167
Попков Г.В., Попков В.К. Система распределённой обработки данных.....	174
Харинов М.В. Модель локализации объектов на цифровом изображении.....	182

Contents

1. Controlled systems and methods of optimization

Kolmakova A.I. Multimethod optimization of control in economic model of tax rate choice for companies of energy sector	3
Medvezhonkov D.S. Experimental researches of interior-point algorithms for solution of flow distribution nonlinear problems	12
Mizhidon A.D., Mizhidon K.A. The problem of optimal control for linear system with state and mixed constraints	17
Orlov A.V. Hybrid genetic algorithm of global search for optimistic solutions in bilevel optimization problems	25
Rasina I.V., Usenko O.V. Sufficient conditions of optimality in a discrete hierarchical model	33
Khishektueva I.-Kh.D. Algorithm for optimization of parameters of nonlinear dynamic systems.....	39

2. Information systems and technologies

Abasov N.V., Chernyshov M.Yu., Kaverzina A.V., Osipchuk E.N., Chernyshova I.A. A software complex for synthesis of applied software models and meta-models on the basis of mathematical programming principles	45
Bimenova Zh.B., Dzhumamuratov R.A., Sidorova E.A. The approach to development of the Russian-Kazakh thesaurus on computer science	53
Verkhoturova Yu.S. Model of the object domain in the language of ontology description	63
Kurganskaya G.S. Cloud technologies of the Internet Education based on KFS knowledge representation model	69
Ruzhnikov G.M., Khmel'nov A.E., Gachenko A.S. Working out and development of GIS "Investor" of Irkutsk	76
Fatkieva R.R. Development of metrics for attack detection on the basis of network traffic analysis	81
Fedorchenko L.N. Syntax-driven data processing for practice.....	87
Fereferov E.S., Khmel'nov A.E. Automatization of user interface creation based on models of database application	100
Shigarov A.O. Design of intelligent system for table information extraction from unstructured texts	110

3. Functional analysis and differential equations

Alsykova A.A. Initial boundary value problem with integral boundary conditions for a class of composite type equations	119
Mizhidon A.D., Barguev S.G. Boundary value problem for one hybrid system of differential equations	130
Novikov A.E., Novikov E.A., Knaub L.V. Stability control for a second order Ceschino's method...	138
Shishkin G.A. Function of flexible structure and its updating at the solution of boundary value problems for equations with functional delay	144

4. Mathematical modeling

Baranov V.I., Golenko N.N., Kompaniets L.A., Paka V.T., Yakubaylik T.V. Spatial and temporal variability of the main characteristics of Lake Shira in the season of observations 2011-2012.....	148
Bubenchikov A.M., Bazarov K.L., Buyantueva Z.G. Computer graphics generated by transformations of linear coordinates	157
Danchev V.N., Pushistov P.Yu. Quantitative assessment of prognostic potential of data-computing complex "Teletskoye Lake".....	161
Dondokov Z.B.-D., Buldaev A.S. On one modification of interbranch balance.....	167
Popkov G.V., Popkov V.K. A system of distributed data processing	174
Kharinov M.V. A model for localization of objects in digital images	182

ВЕСТНИК БУРЯТСКОГО ГОСУНИВЕРСИТЕТА

Вестник БГУ включен в подписной каталог Роспечати за № 18534 и Перечень изданий Российской Федерации, где должны быть опубликованы основные научные результаты диссертаций на соискание ученых степеней доктора и кандидата наук.

На основании постановления заседания Ученого совета БГУ за № 10 от 28 мая 2009 г. в «Вестнике БГУ» в 2013 г. публикуются статьи по следующим направлениям:

1. Педагогика (январь)

гл. ред. Дагбаева Нина Жамсуевна – тел. 21-04-11; 44-23-95

эл. адрес: vestnik_pedagog@bsu.ru

2. Экономика. Право (февраль)

гл. ред. Бадмаева Мария Валентиновна – тел. 21-37-44

эл. адрес: feu-bsu@mail.ru

3. Химия, физика (март)

гл. ред. Хахинов Вячеслав Викторович – тел. 43-42-58

эл. адрес: khakhinov@mail.ru

4. Биология, география (март)

гл. ред. Доржиев Цыдып Заятуевич – тел. 21-03-48

эл. адрес: vestnik_biology@bsu.ru

5. Психология, социальная работа (апрель)

гл. ред. Базарова Татьяна Содномовна – тел. 21-26-49

эл. адрес: decspf@mail.ru

6. Философия, социология, политология, культурология (апрель)

гл. ред. Осинский Иван Иосифович – тел. 21-05-62

эл. адрес: intellige2007@rambler.ru

7. История (май)

гл. ред. Митупов Константин Батомункич – тел. 21-64-47

эл. адрес: vestnik_history@bsu.ru

8. Востоковедение (май)

гл. ред. Бураев Дмитрий Игнатьевич – тел. 44-25-22

эл. адрес: railia@mail.ru

9. Математика, информатика (июнь)

гл. ред. Булдаев Александр Сергеевич – тел. 21-97-57

эл. адрес: vestnik_bsu_math@rambler.ru

10. Филология (сентябрь)

гл. ред. Имихелова Светлана Степановна – тел. 21-05-91

эл. адрес: 223015@mail.ru; map1955@mail.ru

11. Романо-германская филология (сентябрь)

гл. ред. Ковалева Лариса Петровна – тел. 21-17-98

эл. адрес: k1p@bsu.ru, khida@mail.ru

12. Медицина, фармация (октябрь)

гл. ред. Хитрихеев Владимир Евгеньевич – тел. 44-82-55

эл. адрес: vestnik_medicine@bsu.ru

13. Физкультура и спорт (октябрь)

гл. ред. Гасков Алексей Владимирович – тел. 21-69-89

эл. адрес: gaskov@bsu.ru

14. Философия, социология, политология, культурология (ноябрь)

гл. ред. Осинский Иван Иосифович – тел. 21-05-62

эл. адрес: intellige2007@rambler.ru

15. Теория и методика обучения (декабрь)

гл. ред. Очиров Михаил Надмитович – тел. 21-97-57

эл. адрес: vestnik_method@bsu.ru

Требования к оформлению статей, представляемых в «Вестник БГУ»

Отбор и редактирование публикуемых статей производится редакционной коллегией из ведущих ученых и приглашенных специалистов.

В «Вестник БГУ» следует направлять статьи, отличающиеся высокой степенью научной новизны и значимостью. Каждая статья имеет УДК, а также письменный развернутый отзыв (рецензию) научного руководителя или научного консультанта, заверенный печатью.

Автор статьи обязан заключить лицензионный договор о предоставлении неисключительных прав на использование созданного им произведения (статьи) ФГБОУ ВПО «Бурятский государственный университет». Образец лицензионного договора представлен на сайте БГУ.

Общие требования	Тексты представляются в электронном и печатном виде. Файл со статьей может быть на дискете или отправлен электронным письмом. На последней странице – подпись автора(ов) статьи. Название статьи и аннотация даются и на английском языке. После аннотации дать ключевые слова на русском и английском языках.
Электронная копия	Текстовый редактор Microsoft Word (версии 6.0, 7.0, 97). В имени файла указывается фамилия автора.
Параметры страницы	Формат А4. Поля: правое – 15 мм, левое – 25 мм, верхнее, нижнее – 20 мм.
Форматирование основного текста	С нумерацией страниц. Абзацный отступ – 5 мм. Интервал – полуторный.
Гарнитура шрифта	Times New Roman. Обычный размер кегля – 14 пт. Список литературы и аннотация – 12 пт.
Объем статьи (ориентировочно)	Кратких сообщений – до 3 с., статей на соискание ученой степени кандидата наук – 7–12 с., на соискание ученой степени доктора наук – 8–16 с.
Сведения об авторах	Указываются фамилия, имя, отчество (полностью), ученая степень, звание, должность и место работы, адрес с почтовым индексом, телефоны/факсы, e-mail (на русском и английском языках)

- Список литературы – все работы необходимо пронумеровать, в тексте ссылки на литературу оформлять в квадратных скобках.

- Материалы, не соответствующие предъявленным требованиям, к рассмотрению не принимаются.

- Решение о публикации статьи принимается редакцией «Вестника БГУ». Корректурa авторам не высылается, присланные материалы не возвращаются.

- Статьи принимаются в течение учебного года.

- Допустима публикация статей на английском языке, сведения об авторах, название и аннотацию которых необходимо перевести на русский язык.

- Формат журнала 60x84 1/8.

- Статья должна содержать минимум таблиц, формул, рисунков и графиков. Их присутствие допускается только в тех случаях, если описать процесс в текстовой форме невозможно или нецелесообразно. Желательно использование только вертикальных таблиц и рисунков. Все объекты должны быть черно-белыми без оттенков. Все формулы должны быть созданы с использованием компонента Microsoft Equation или в виде четких картинок. Символы можно вставлять с помощью операции в Word (Вставка – Символ). Диаграммы располагаются в тексте с использованием программы Microsoft Excel (Вставка – Объект – Создание – Диаграмма Microsoft Excel). Рисунки и графики должны иметь четкое изображение и быть выдержаны в черно-белой гамме, лучше применять штриховку (Формат автофигуры – Цвета и линии – Цвет – Способы заливки – Узор). Схемы создаются с помощью панели инструментов Рисование. Фотографии и рисунки в формате *.tif или *.jpg должны иметь разрешение не менее 300 dpi. Диаграммы, формулы, рисунки, графики должны прилагаться отдельными файлами, чтобы издательство имело возможность ввести в них правки.

Стоимость обработки 1 с. (формата А4) для преподавателей БГУ составляет 200 р., для остальных – 400 р. Для аспирантов – бесплатно.

Адрес: 670000, г. Улан-Удэ, ул. Смолина, 24 а, Издательство БГУ.

Тел. (301-2)-21-95-57. Факс (301-2)-21-05-88

Оплата производится при получении счета от бухгалтерии БГУ.