

**Научно-технический
журнал**
Выходит 4 раза в год

УЧРЕДИТЕЛЬ

Институт проблем управления
им. В.А. Трапезникова РАН

Главный редактор

И.В. Прангишвили

Заместители главного
редактора

А.Н. Шубин, Ф.Ф. Пащенко

Научный редактор

Л.П. Боровских

Региональные редсоветы
(руководители)

Владивосток — О.В. Абрамов
(4232) 31-02-02

Липецк — Л.А. Кузнецов
(0742) 32-80-44

Издатель ООО «СенСиДат»

Ген. директор Н.Н. Кузнецова

Адрес редакции
117997, ГСП-7, Москва,
ул. Профсоюзная, д. 65, к. 104.
Тел./факс (095) 330-42-66,
тел.: (095) 334-92-00,
334-90-20

E-mail: datchik@ipu.ru
www.ipu.ru/period/ru

Оригинал-макет
подготовлен ООО «АКИМ»

Отпечатано в типографии
ООО «TexKomИнтел»

Подписано в печать
29.12.2003 г.

Заказ № 01/02

Журнал зарегистрирован
в Министерстве
Российской Федерации
по делам печати,
телерадиовещания
и средств массовых
коммуникаций

Подписные индексы:
81708 в каталоге Роспечати
38006 в объединенном каталоге
«Пресса России»

Свидетельство о регистрации
ПИ №77-11963
от 06 марта 2002 г.

© СенСиДат, 2003 г.

ПРОБЛЕМЫ УПРАВЛЕНИЯ

4.2003

СОДЕРЖАНИЕ

Информационные технологии в управлении

Волгин Л.И., Климовский А.Б. Импликативная алгебра выбора как основа информационных технологий и систем управления в континуальной области	2
Затуливетер Ю.С., Фищенко Е.А. Организация распределенных вычислений в системе программирования ПАРСЕК на примере сжатия цифрового видео	6
Абрамов О.В., Катуева Я.В. Технология параллельных вычислений в задачах анализа и оптимизации	11
Закревский А.Д. Эффективные методы нахождения кратчайших решений систем линейных логических уравнений	16

Системный анализ и обработка данных

Гребенюк Е.А. Анализ и оперативная диагностика систем, описываемых нестационарными случайными процессами	23
Бахтадзе Н.Н., Назин А.В. Виртуальные анализаторы маркетинговой информации в системе управления предприятием	30
Клепарский В.Г., Ефремов В.А. Мультифрактальность, диссипация и устойчивость среднесрочных трендов на фондовом рынке	36

Надежность и техническая диагностика

Ведешенков В.А. Самодиагностирование цифровых систем с реконфигурацией	39
Микрин Е.А., Пелихов В.П. Анализ нештатных ситуаций и критичности программного обеспечения в проекте международной космической станции	52

Проблемно ориентированные системы управления

Жожикашвили В.А., Билик Р.В., Вертигб В.А. и др. Открытые системы массового обслуживания с распознаванием речи	55
---	----

Управление подвижными объектами

Калашников Л.М., Малышев Г.В., Свотин А.П. Управление свертыванием двухмодульной космической тросовой системы	63
--	----

Хроника

Михаил Александрович Гаврилов (к 100-летию со дня рождения)	67
Конференция «Теория и практика логического управления», посвященная памяти М.А. Гаврилова	73

* * *

Указатель статей, опубликованных в журнале «Проблемы управления» за 2003 г.	74
Contents and abstracts	76

УДК 510.6+512.57+681.51.011

ИМПЛИКАТИВНАЯ АЛГЕБРА ВЫБОРА КАК ОСНОВА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ УПРАВЛЕНИЯ В КОНТИНУАЛЬНОЙ ОБЛАСТИ

Л.И. Волгин, А.Б. Климовский

Ульяновский государственный технический университет, г. Ульяновск

Рассмотрены новые направления в теоретической информатике и математической кибернетике (континуальные алгебраические логики – специальные алгебры и общие логики), в аналоговой вычислительной технике (реляторная схемотехника), ориентированные на решение вычислительных задач управления на базе универсального логико-алгебраического аппарата импликативной алгебры выбора.

Классическая двузначная алгебра логики Дж. Буля (1815–1864) охватывает весь двоично-дискретный (виртуальный) мир, в котором переменные формул принимают значения из двухэлементного множества {1, 0}. «Ноль и единица от Бога, всё остальное дело рук человеческих» (Л. Кронекер (1823–1891)). По объему и многообразию применений булевой алгебры в информационных технологиях и во многих других областях науки и техники XX век можно назвать булевым.

Но физический макромир, технологии производства и управления, технические объекты, измеряемые и контролируемые параметры в подавляющем большинстве случаев сопровождаются и описываются континуальными (непрерывными, аналоговыми) процессами, что приводит к известным противоречиям применения двузначной булевой алгебры логики в континуальной области [1]. Указанное обуславливает необходимость смены парадигмы логико-алгебраического обеспечения континуальных информационных технологий.

Многие проблемы в информационных технологиях и системах искусственного интеллекта относятся к классу задач (естественнонаучных, научно-технических, экономических, социальных и др.) оптимального многокритериального (векторного) управления в континуальной области [2].

Исторически первой задача оптимизации при нескольких противоречивых критериях была поставлена и решена итальянским экономистом и социологом В. Парето (1848–1923) и опубликована в его учебнике по экономике в 1896 г. [3]. Парето предложил использовать в качестве критерия оптимизации равенство нулю якобиана оптимизируемой вектор-функции, компонентами которой являются частные критерии оптимальности исследуемого объекта, когда в пространстве переменных управления существуют точки, в каждой из которых хотя бы один критерий оптимизации пре-восходит другие. В математической экономике парето-оптимизация была развита Л.Н. Волгиным, сформулировавшим «принцип согласованного оптимума» [4].

Большой вклад в развитие теории оптимального управления внес Л.С. Понtryгин (1908–1988) [5].

Вопросы оптимального управления тесно связаны с задачами разработки алгоритмов обработки, синтеза логико-алгебраических моделей и сопутствующих им программной, аппаратурной или аппаратурно-программной реализаций.

Самостоятельное направление в теоретической информатике, математической кибернетике, в математической (символической) логике и, соответственно, в информационных технологиях, системах искусственного интеллекта и управления представ-



ляют собой рассматриваемые в настоящей работе континуальные алгебраические логики (специальные алгебры и общие логики), входящие в разработанную логико-алгебраическую метасистему взаимоотношений алгебраических логик и сопутствующих им логических исчислений [6]. Предметными переменными (аргументами) упомянутых логик изначально являются различного типа математические объекты.

Наиболее широкими функциональными, управляемыми, вычислительными, алгоритмическими, адаптационными и комбинационными возможностями обладает импликативная алгебра выбора (ИАВ), логико-алгебраические модели которой синтезируются через операции предметной и предикатной суперпозиций в базисе бинарных операций ИАВ-конъюнкции (Λ) и ИАВ-дизъюнкции (V):

$$Z_1 = \Lambda_I(y_1, y_2) = y_1 I(x_2 - x_1) + y_2 I(x_1 - x_2), \quad (1)$$

$$Z_2 = V_I(y_1, y_2) = y_1 I(x_1 - x_2) + y_2 I(x_2 - x_1),$$

где $I(x)$ – единичная функция, равная нулю при $x < 0$ и единице при $x > 0$, $I(0) \in \{0, 1\}$; $I = (I_{12}, I_{21})$ – кортеж (вектор) весовых коэффициентов $I_{12} = I(x_1 - x_2)$ и $I_{21} = \bar{I}_{12} = 1 - I_{12} = I(x_2 - x_1)$; $I_{12} + I_{21} = 1$ (условие комплементарности); y_1 и y_2 – предметные переменные (в общем случае любые физические величины и математические объекты, удовлетворяющие условиям мультиплективного выделения $y_i \cdot 1 = y_i$ и поглощения $y_i \cdot 0 = 0$), x_1 и x_2 – предикатные переменные (действительные числа). При этом предметные переменные y_1 и y_2 могут быть объектами любой физической природы и различной размерности.

При отождествлении в выражениях (1) предметных и предикатных переменных ($y_1 = x_1$ и $y_2 = x_2$) операции ИАВ-конъюнкции и ИАВ-дизъюнкции вырождаются в базовые бинарные операции непрерывной логики [7, 8]:

$$\begin{aligned} Z_1 &= \Lambda(x_1, x_2) = \min(x_1, x_2), \\ Z_2 &= V(x_1, x_2) = \max(x_1, x_2). \end{aligned} \quad (2)$$

Если континуальные переменные в выражениях (2) принимают k разрешенных уровней, то приходим к бинарным операциям многозначных (k -значных) логик, которые при $k = 2$ вырождаются в конъюнкцию $Z_1 = x_1 \wedge x_2$ и дизъюнкцию $Z_2 = x_1 \vee x_2$ двузначной булевой алгебры логики.

В свою очередь, ИАВ-операции (1) являются частными реализациями бинарных операций предикатной алгебры выбора (в публикациях до 2002 г. ИАВ, как подалгебра предикатной алгебры выбо-

ра, также называлась предикатной алгеброй выбора) [9, 10]:

$$\begin{aligned} Z_1 &= \Lambda_A(y_1, y_2) = y_1 \alpha_2 + y_2 \alpha_1, \\ Z_2 &= V_A(y_1, y_2) = y_1 \alpha_1 + y_2 \alpha_2, \end{aligned} \quad (3)$$

где $\alpha_i \in \{0, 1\}$, $\alpha_1 + \alpha_2 = 1$ (условие комплементарности весовых коэффициентов α_1 и α_2), $A = (\alpha_1, \alpha_2)$ – кортеж весовых коэффициентов α_1 и α_2 .

Выражения (1)–(3) воспроизводят различного типа операции альтернативного выбора. Выражения (3) при выполнении условия комплементарности $\alpha_1 + \alpha_2 = 1$ описывают бинарные операции континуальной комплементарной алгебры (условие двузначности весовых коэффициентов $\alpha_1, \alpha_2 \in \{0, 1\}$ не накладывается) [11].

Путем введения вместо единичных функций операторов отношения предпочтения (доминирования) $p(x_1, x_2)$ и $\bar{p}(x_1, x_2)$, где $p + \bar{p} = 1$, разработано теоретико-множественное обобщение бинарных операций ИАВ [2].

Из всех алгебраических логик метасистемы [6] наиболее содержательна ИАВ. Это обусловлено тем, что n -арные ИАВ-функции

$$\begin{aligned} Z &= W_A(y_1, y_2, \dots, y_n) = \\ &= y_1 a_1 + y_2 a_2 + \dots + y_n a_n \end{aligned} \quad (4)$$

определенны на двух задающих множествах – предметных $Y = \{y_1, y_2, \dots, y_n\}$ и предикатных $X = \{x_1, x_2, \dots, x_n\}$ переменных.

В выражении (4) операция W либо конъюнкция Λ , либо дизъюнкция V ; $a_i \in \{0, 1\}$ – составные (суперпозиционные) весовые коэффициенты, являющиеся функциями элементарных весовых коэффициентов I_{ij} ; $I_{ij} + I_{ji} = 1$; $i, j \in \{1, 2, \dots, n\}$, $i \neq j$. Для весовых коэффициентов a_i выполняется условие $a_1 + a_2 + \dots + a_n = 1$ (свойство суперпозиционной инвариантности условия комплементарности).

Универсальность, многофункциональность и многообразие применений ИАВ [2, 9] также определяется наличием двух различных по целевому назначению задающих множеств X и Y переменных, обслуживающих, соответственно, области управления (предикатные переменные) и исполнения (предметные переменные).

В области информационных технологий построение соответствующей поставленной задаче алгебраической логики (логико-алгебраической модели) начинается назначением (или поиском) подходящих базовых алгебраических операций, образующих полную систему исходных функций, через суперпозиции которых в заданной предметной области строятся необходимые логико-алгеб-

раические модели. Таким образом, от алгебры формул классической логики, логики высказываний и событий приходим к алгебрам функций.

Известна также обобщенная трактовка понятия функциональной системы (P, Ω) , где P – множество отображений, реализуемых управляющими системами из некоторого класса, а множество Ω состоит из операций, используемых при построении новых управляющих систем из заданных [12] и ориентированных на построение оптимальных логико-алгебраических моделей.

Следует выделять, как считает О.П. Кузнецов, два компонента науки об управлении – вычислительный и концептуальный. Соответственно различаются и два типа исполнителей: руководители конкретных разработок (проекты, стройки и пр.) и руководители с гораздо более общим ареалом деятельности (министры, мэры и пр.). Если первые из них должны профессионально владеть вычислительным компонентом науки, то для вторых достаточно компетентности в области ее концептуальных аспектов [13]. Формальная логико-алгебраическая теория управления начинается с элементарных (бинарных) операций альтернативного выбора (1)–(3) в континуальном (трехмерном) пространстве предикатных (управляющих) и предметных (исполняющих) переменных (вычислительный подход) или с использованием логико-концептуального подхода [14].

Аппаратурно ИАВ-модели реализуются в элементном базисе реляторов – универсальных логических схемных элементов, воспроизводящих бинарные операции ИАВ, непрерывной логики, многозначных и двузначной логик [10, 15–18].

Принципиальное отличие аналоговых реляторных электронно-вычислительных сетей и реляторных процессоров от цифровых заключается в том, что элементами задающих множеств являются аналоговые сигналы. Здесь в роли одного двоичного разряда выступает аналоговое число, т. е. необходимость двоично-поразрядного сканирования слов в последовательности кодовых комбинаций при вводе и считывании информации устраняется. Тем самым изначально обеспечивается глобальный параллелизм обработки и преобразования формы представления информации без промежуточного преобразования в цифровой код.

В элементном базисе реляторов возможно построение широкой номенклатуры аналоговых управляющих, коммутационных, логических, вычислительных, функциональных, нейронных, измерительных преобразователей и реляторных управляющих спецпроцессоров.

Континуальные алгебраические логики [2, 6–22] и реляторная схемотехника [15–21] являются са-

мостоятельным и эффективным направлением в теоретической информатике, математической кибернетике и, соответственно, в информационных технологиях и континуальных системах управления.

Многочисленные практические применения ИАВ в различных областях науки и техники приведены в работах [18–21] и др. Применение логико-математического аппарата ИАВ для решения прикладных задач (в том числе интеллектуальных) в различных предметных областях непрерывно расширяется. Это подтверждается географией возникновения новых научных направлений и школ в России, пользующихся логико-математическим аппаратом ИАВ, ИАВ-моделями и реляторной схемотехникой для решения задач в аналоговой области [20].

В частности, на кафедре «Конструирование и производство радиоаппаратуры» Пензенского государственного университета логико-алгебраический аппарат ИАВ применяется для синтеза формы теплонаагруженных элементов конструкций радиоэлектронных средств и построения их ИАВ-моделей на основе эволюционной оптимизации. Тем самым заложены основы теории вычисления элементов конструкций радиоэлектронной аппаратуры.

На кафедре вычислительной техники Курского государственного технического университета логико-алгебраический аппарат ИАВ применяется для построения алгоритмов, ИАВ-моделей структурных схем реляторных ячеек и устройств управления клеточной самоорганизацией мультимикроконтроллеров с программированным резервом, что позволяет устраниить необходимость преобразования континуальной формы представления информации в цифровую [20]. Разработаны синхронные и асинхронные варианты клеточно-нейронной среды с различными механизмами образования ре-продуцируемой логической структуры. Результаты моделирования подтвердили корректность предложенного подхода и перспективность его развития и применения. При этом элементы управляющей системы подобны эмбриональной клетке, выполняющей функцию в зависимости от места ее расположения. Для синтеза клеточной и клеточно-нейронной сред применен аппарат ИАВ и реляторный элементный базис [23].

На кафедре «Сервис бытовой радиоэлектронной аппаратуры» Тольяттинского государственного института сервиса развивается новое направление в области силовой электроники на основе информационных [17] и силовых [2, 20, 24] реляторов, переключательный канал которых построен на полевых транзисторах с изолированным затвором или на биполярных транзисторах с изолированной базой; коммутируемые токи составляют более сотни



ампер. Разработаны ИАВ-модели и соответствующие им реляторные структуры интеллектуальных устройств электропитания с высокой концентрацией типов энергетических преобразований и процедур управления.

На кафедре «Электронные измерительные системы» Московского инженерно-физического института для сокращения сроков проектирования и снижения финансовых затрат разработан аналоговый базовый матричный кристалл. На его основе реализованы реляторные микросхемы, в частности, в одном корпусе микросхемы можно разместить три релятора (ограничивающим фактором является допустимое число контактных площадок, равное 24) [20].

Направление, связанное с реляторной схемотехникой и ее логико-алгебраическими основами, традиционно развивается на кафедре измерительно-вычислительных комплексов Ульяновского государственного технического университета [2, 6–11, 15–22, 25, 26].

Развиваемое направление полностью соответствует концепциям Г. Башлара (1884–1962) «Мир культуры требует изменения логических ценностей, ... необходимо разработать столько логик, сколько существует типов объектов любой природы» [27] и В.А. Смирнова (1931–1996) «Цель науки – создание типовых методов, позволяющих стандартным образом решать целые классы задач» [28].

Авторы надеются, что реферативность изложенного компенсируется ссылками на соответствующие публикации.

ЛИТЕРАТУРА

1. Волгин Л.И., Мишин В.А. Будущее за цифровыми или аналоговыми технологиями? // Информационные технологии в электротехнике и электроэнергетике // Матер. II Все-росс. НТК.– Чебоксары, 1998. – С. 86–89.
2. Алгебраические логики, импликативная и предикатная алгебры выбора в задачах науки и техники: Тр. междунар. конф. «Континуальные алгебраические логики, исчисления и нейроинформатика в науке, технике и экономике» / Под ред. Л.И. Волгина. – Ульяновск: УлГТУ, 2003. – Т. 2. – 138 с.
3. Теория автоматического управления / Под ред. А.В. Нестушилова. – М.: Высшая школа, 1983. – 432 с.
4. Волгин Л.Н. Принцип согласованного оптимума. – М.: Сов. радио, 1977. – 144 с.
5. Понtryagin L.S. Принцип максимума в оптимальном управлении. – М.: Наука, 1989. – 64 с.
6. Волгин Л.И. Метасистема взаимоотношений алгебраических логик и сопутствующих исчислений, порождаемых функцией-аксиомой взвешенных степенных средних // Инфор. технологии. – 2002. – № 7. – С. 20–26.
7. Волгин Л.И. Алгебраические логики: взаимоотношения, законы и свойства. – М., 2003. – 24 с. – (Б-ка журн. «Информ. технологии», прилож. к № 6/2003).
8. Волгин Л.И., Левин В.И. Непрерывная логика и её применения. – Таллинн: АН Эстонии, 1990. – 210 с.
9. Волгин Л.И. Комплементарная алгебра и предикатная алгебра выбора. – Ульяновск: УлГТУ, 1996. – 68 с.
10. Волгин Л.И. Предикатная алгебра выбора и её модификации // Опыт, результаты, проблемы: Повышение конкурентоспособности радиоэлектронной аппаратуры. – Таллинн, 1986. – Вып. 4. – С. 64–104.
11. Волгин Л.И. Свойства и законы комплементарной алгебры // Изв. АН ЭССР. Физика. Математика. – 1988. № 4. – С. 417–427.
12. Карпенко А.С. Многозначные логики. – М.: Наука, 2000. – 223 с.
13. Мандель А.С. Вторая международная конференция по проблемам управления // Проблемы управления. – 2003. – № 2. – С. 60–64.
14. Шоломов Л.А. Логические методы исследования дискретных моделей выбора. – М.: Наука, 1989. – 288 с.
15. Волгин Л.И., Зарукин А.И. Развитие элементного базиса реляторной схемотехники // Датчики и системы. – 2002. – № 3. – С. 2–8.
16. Волгин Л.И., Зарукин А.И., Климовский А.Б. Классификация реляторов по доминантным признакам // Проектирование и технология электронных средств. – 2002. – № 3. – С. 32–38.
17. Волгин Л.И. Элементный базис реляторной схемотехники. – Тольятти: Поволж. технол. ин-т сервиса, 1999. – 70 с.
18. Волгин Л.И. Синтез устройств для обработки и преобразования информации в элементном базисе реляторов. – Таллинн: Валгус, 1989. – 210 с.
19. Волгин Л.И. Реляторные процессоры на основе графа Паскаля для адресно-ранговой идентификации, селекции и ранжирования аналоговых сигналов. – Тольятти: ТГИС, 2000. – 81 с.
20. Волгин Л.И. Релятор и реляторная схемотехника: логико-алгебраические основы и применения: Темат. библиогр. указ., комментарий и приложения. – Тольятти: ТГИС, 2003. – 213 с.
21. Волгин Л.И. Аналоговые реляторные сети для преобразования структуры данных с адресно-ситуационной идентификацией // Телекоммуникации. – 2002. – № 12. – С. 5–11.
22. Волгин Л.И. Алгебраические логики как основа интеллектуальных информационных технологий в континуальной области // Там же. – С. 3–7.
23. Колосков В.А., Медведева М.В. Синергетический подход к обеспечению отказоустойчивости процессорных систем // Импликативная алгебра выбора и непрерывная логика в прикладных задачах науки и техники: Тр. междунар. конф. «Континуальные алгебраические логики, исчисления и нейроматематика в науке, технике и экономике» / Под ред. Л.И. Волгина. – Ульяновск, 2002. – Т. 2. – С. 59–60.
24. Абрамов Г.Н., Кувшинов А.А. Однофазный инвертор AC-DC с коррекцией коэффициента мощности на базе силового релятора // Там же. – С. 61–65.
25. Волгин Л.И. Теоретико-множественная интерпретация бинарных операций импликативной алгебры выбора // Там же. – С. 8, 9.
26. Персональный библиографический указатель публикаций Л.И. Волгина: Юбил. изд. к 70-летию. – Ульяновск: УлГТУ, 2002. – 142 с.
27. Bachlard G. Le nouvel esprit scientifique.– Paris: Press Universitaires, 1934. Рус. пер.: Башляр Г. Новый рационализм: Пер. с фр. – М.: Прогресс, 1987. – 276 с.
28. Смирнов В.А. Творчество, открытие и логические методы поиска доказательства// Природа научного открытия. – М.; 1986. – С. 101–115.

(8422)-431312

E-mail: volgin@ulstu.ru



ОРГАНИЗАЦИЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ ПАРСЕК НА ПРИМЕРЕ СЖАТИЯ ЦИФРОВОГО ВИДЕО¹

Ю.С. Затуливетер, Е.А. Фищенко

Институт проблем управления им. В.А. Трапезникова, г. Москва

Рассмотрен новый подход к организации распределенных вычислений в сети Интернет, основанный на математически полной и замкнутой модели исчисления древовидных структур, реализованной посредством системы программирования ПАРСЕК. В язык ПАРСЕК введены операторы управления ресурсами и групповой передачи данных. На примере кодирования сжатия в реальном времени для видеоконференций в стандарте H.261 и цифрового видео в стандарте MPEG показана эффективность предложенного способа управления сетевыми ресурсами.

ВВЕДЕНИЕ

Следующий шаг в развитии Интернета (далее Сети) – предоставление пользователю суперкомпьютерных возможностей, что позволит ему решать задачи, требующие $10^{10} \dots 10^{18}$ операций в секунду и более, за счет совместного использования средств обработки информации одновременно на многих компьютерах Сети. В связи с этим стала весьма актуальной задача разработки средств регулярного программирования глобально распределенных вычислений.

Движение к глобальному программированию в Сети возможно по двум встречным направлениям. Одно – «снизу – вверх» (от сложившейся практики), другое – «сверху – вниз» (от новой теории).

Пример первого направления – Grid-технологии [1], подход к виртуальной организации вычислительных ресурсов Сети для распределенных вычислений путем создания глобально распределенной операционной системы. Это путь постепенного наращивания и расширения возможностей существующих средств индустриального программирования в рамках устоявшихся компьютерных платформ. Многочисленные и разнородные версии Grid-технологий толерантны к классической компьютерной парадигме, потому органично вписываются в арсенал индустриального программирования. Интеграция глобально распределенных вы-

числительных ресурсов осуществляется за счет внешних программных надстроек посредством силового преодоления комбинаторного сопротивления информационного шума, порождаемого разнородной компьютерной информацией.

Встречное направление начинается с новой концепции математически однородного поля компьютерной информации, в которой на уровне аксиоматики устранены причины воспроизведения информационного шума [2]. В этой концепции проблемы программирования, интеграции и масштабирования компьютерных решений и распределенных процессов перестают зависеть от технических особенностей компьютеров и сетей. Регулярный инструмент глобального программирования строится сначала на теоретическом уровне в виде математически полной и замкнутой модели исчисления древовидных структур, а затем алгоритмы функционирования модели программируются в ресурсах Сети посредством языка ПАРСЕК [3], воплощающего исчисление древовидных структур.

ОСОБЕННОСТИ СИСТЕМЫ ПРОГРАММИРОВАНИЯ ПАРСЕК

Различные формы древовидных структур широко применяются в языках программирования как формализованные типы данных, например, списки (Лисп). Их можно представить через общие структуры (Си, Паскаль) или через объекты (C++, Java). В прикладных системах с помощью древовидных структур реализуются удобные пользовательские интерфейсы, гипертекстовые спра-

¹ Работа выполнена при поддержке РФФИ, проект № 01-01-01047



вочники и др. Практика показывает: деревья упрощают пользование программными продуктами.

Язык и система программирования ПАРСЕК реализует исчисление древовидных структур с открытой интерпретацией, представленных для программиста в форме вертикальной ленты строк («одна строка – одна вершина»), которую еще Питер Нортон ввел в массовый компьютерный обиход, дав геометрическое представление файловой системы. В языке ПАРСЕК деревья в подобной геометрической форме дают универсальное и математически замкнутое представление компьютерной информации – как данных, так и программ. Решая задачи, программист создает необходимые деревья и функции их преобразования, по своему усмотрению назначая смысловую нагрузку и деревьям, и функциям их преобразования. Полнота и замкнутость исчисления древовидных структур в сочетании с их открытой содержательной интерпретируемостью дают практическую универсальность в применении к различным задачам. В значительной мере упрощается параметризация программ. На идеологическом уровне исключена опасность разрушения информации в памяти из-за переполнения индексных полей. При полностью динамическом управлении памятью ее расход ограничен заранее известными и приемлемыми для практики оценками.

СПОСОБ УПРАВЛЕНИЯ СЕТЕВЫМИ РЕСУРСАМИ

Для управления сетевыми ресурсами при организации распределенных вычислений создается специализированная база данных доступных ресурсов, с помощью которой производится динамическое выделение и освобождение ресурсов. В ней содержится информация о компьютерах с указанием их производительности и объемов памяти, пропускной способности маршрутов в сети и т. д.

От задачи идут запросы на выделение ресурсов: число компьютеров, размеры памяти, объемы и темп передачи данных и др. Формируется расписание использования выделенных компьютеров (рис. 1), в которых выполняются вычисления. По мере освобождения выделенных ресурсов они возвращаются базе данных для повторного использования.

Компьютер, посылающий базе данных запросы на выделение ресурсов, называется ведущим (*ведущий 1* – см. рис. 1). Он также осуществляет пуск задачи, динамическое распределение заданий, ввод/вывод данных, сбор результатов и освобождение ресурсов. Распределенные задания реализуют компьютеры сети, называемые ведомыми (*ведомые 1.1, 1.2, 1.3, 1.4*, которые могут быть ведущими (*ведущие 2.1, 2.2*) в отношении подзадач (*ведомые 2.1.1, 2.1.2, 2.2.1, 2.2.2*).

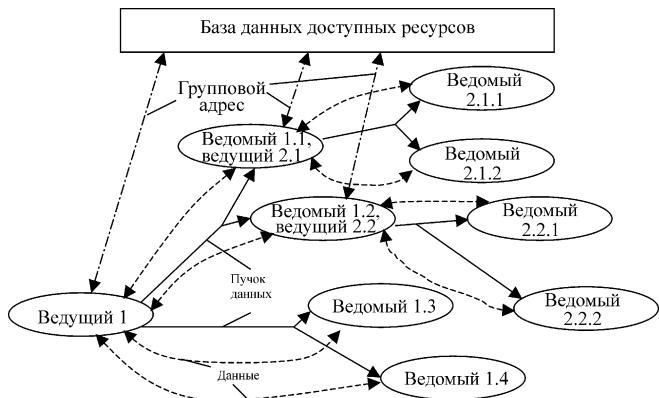


Рис. 1. Дерево заданий

В данной работе для эффективного управления распределенными вычислениями в сети предлагается ввести возможность работы группы ведомых компьютеров по одной программе и использовать функции передачи данных по групповому адресу, что позволит уменьшить время передачи данных и ускорить общее время решения задачи.

Цель данной работы – исследовать возможности предложенного способа управления сетевыми ресурсами для хорошо распараллелиемых задач, требующих производительности вычислительных ресурсов $10^{10} \dots 10^{14}$ оп/с. В качестве примеров рассмотрим задачи сжатия цифрового видео в реальном времени для видеоконференций в стандарте H.261 и кодирования в стандарте MPEG.

На примере задач высокой вычислительной сложности реального времени предлагается схема реализации распределенного счета применительно к системе ПАРСЕК, библиотека которой в поддержку распределенных вычислений расширяется следующими функциями работы с сетью:

gip = G_alloc(n) – выделение группы из *n* свободных компьютеров для исполнения *n* заданий. Компьютер посылает в базу данных запрос на выделение группы IP-адресов свободных компьютеров. Ответ на запрос *gip* – либо список адресов, либо групповой адрес, представляющий собой сетевой адрес с маскированными разрядами;

G_free(gip) – освобождение занятых компьютеров по групповому адресу *gip*;

Send(name₁, name₂, ..., name_M, gip) – пересылка данных с именами *name₁*, *name₂*, ..., *name_M* по групповому адресу *gip*;

Wait_D(name₁, name₂, ..., name_M) – оператор ожидания доставки данных с именами *name₁*, *name₂*, ..., *name_M*, отсылаемых функциями *Send(·)*; этот оператор необходим для получения полного набора входных данных для запуска заданий на исполнение.

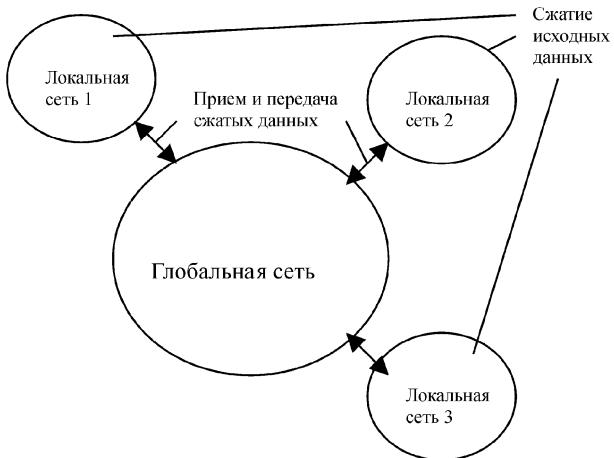


Рис. 2. Организация распределенных вычислений для видеоконференций

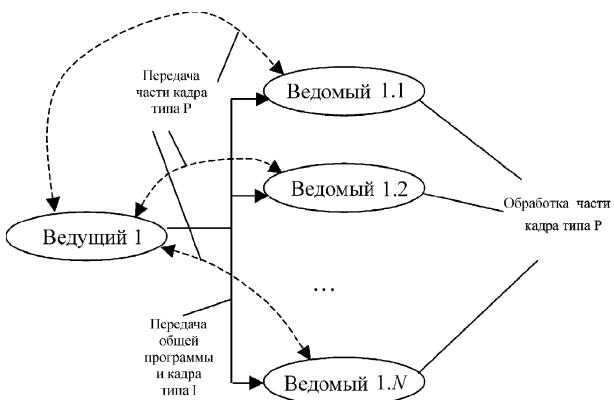


Рис. 3. Организация распределенных вычислений в локальной сети

Процедурный язык ПАРСЕК создавался под задачи с динамически сложными структурами, поэтому, расширяя его возможности, мы распространяем достоинства этого языка на решение задач в глобально распределенных ресурсах.

РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ ДЛЯ ВИДЕОКОНФЕРЕНЦИЙ В СТАНДАРТЕ H.261

Для видеоконференций в хорошо известном стандарте H.261 [4] построим распределенный алгоритм сжатия цифрового видео. Для получения качественного видеоряда видеоконференций для сжатия изображений в реальном времени требуется скорость обработки более 10^{10} коротких операций в секунду.

Требуемая скорость обработки рассчитана, исходя из выполнения наиболее трудоемкой операции поиска векторов движения для кадров формата CIF. Кадры состоят из макроблоков размером $m \times m$, для каждого из них необходимо осуществить

поиск векторов движения в окне. Число операций O_s , которые необходимо выполнить за одну секунду, определяется следующим образом: $O_s = 4FACW^2o_{mb}$, где F – число кадров, поступающих в секунду, A и C – число пикселей по горизонтали и вертикали в кадре, W – размер окна поиска в макроблоках, o_{mb} – число восьмиразрядных операций для сравнения пары макроблоков. При $F = 30$, $A = 352$, $C = 288$, $W = 1$, $o_{mb} = 768$ число операций в секунду равно 10^{10} . Если на выполнение одной восьмиразрядной операции тратится $0,25 \cdot 10^{-9}$ с, то на обработку данных, поступающих за одну секунду, потребуется 2,5 с, что не удовлетворяет условиям реального времени.

Для выполнения этих условий рассматриваются возможности реализации сжимающего кодирования видеоконференций параллельной обработкой на нескольких компьютерах сети. Для этого на стороне каждого из участников видеоконференции на компьютерах своей локальной сети производится распределенная обработка аудио- и видеоинформации. Сжатый в реальном времени (путем параллельной обработки) поток информации посыпается через глобальную сеть другим участникам видеоконференции, которые обрабатывают потоки данных в компьютерах своих локальных сетей (рис. 2).

Кодирование в локальной сети производится следующим образом (рис. 3). Каждый из участников видеоконференции должен иметь сеть с числом компьютеров $N + 1$ (значение N будет рассчитано далее), обменивающихся между собой со скоростью передачи данных не менее 10 Мб/с. Оцифрованные кадры поступают в ведущий компьютер, где производится их внутреннее кодирование в соответствии со стандартом MJPEG [4].

Время внутреннего кодирования T_i для кадров, поступающих за одну секунду, определяется как $T_i = F \frac{A}{8} \frac{C}{8} o_b t_i$, где o_b – число операций для одного блока, t_i – длительность одной операции. При $o_b = 1000$, $t_i = 10^{-9}$ с время внутреннего кодирования займет 0,05 с. Внутреннее кодирование позволяет в 8 раз сократить объемы данных, передаваемых ведомым компьютерам, и снизить требования к пропускной способности локальной сети.

Затем первый кадр группы, который должен быть преобразован в кадр типа I, групповым образом (один ко многим) передается N ведомым компьютерам. Последующие кадры типа P, для которых должно быть выполнено межкадровое кодирование, разбиваются на N частей. Каждая часть, содержащая $Y = \frac{AC}{m^2 N}$ макроблоков, передается соответствующему ведомому компьютеру.

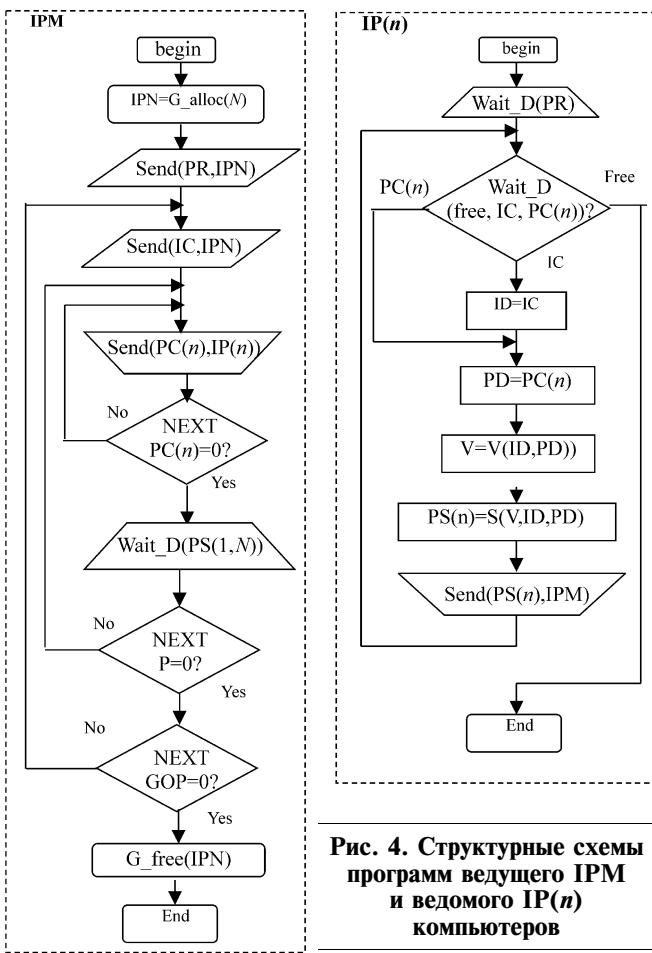


Рис. 4. Структурные схемы программ ведущего IPM и ведомого IP(*n*) компьютеров

Время S_i передачи кадров с внутренним кодированием, поступающих за одну секунду, определяется следующим образом: $S_i = \frac{FACp}{vk_i}$, где p – число разрядов в пикселе, k_i – коэффициент сжатия при внутреннем кодировании, v – скорость передачи данных в локальной сети. Для $p = 8$, $k_i = 8$, $v = 10$ Мб/с время передачи кадров с внутренним кодированием составляет 0,3 с.

Сжатые с помощью межкадрового кодирования кадры типа Р затем возвращаются ведущему компьютеру. Время S_{ni} передачи сжатых с помощью межкадрового кодирования кадров определяется следующим образом: $S_{ni} = \frac{FACp}{vk_i k_{ni}}$, где k_{ni} – коэффициент сжатия при межкадровом кодировании.

Для $k_{ni} = 3$ время передачи кадров с межкадровым кодированием S_{ni} , поступающих за одну секунду, составляет 0,1 с.

В каждом ведомом компьютере производится декодирование кадра типа I, декодирование и межкадровое кодирование Y макроблоков кадра

типа Р, время счета $T_{ni} = 4FYW^2 o_{mb} t_{mb}$. Для $t_{mb} = 0,25 \cdot 10^{-9}$ время счета $T_{ni} = 0,005 \cdot Y$.

Общее время обработки T_s для F кадров, поступающих за одну секунду, определяется следующим образом: $T_s = T_i + T_{ni} + S_i + S_{ni}$. Для выполнения условий реального времени $T_s \leq 1$; $T_s = 0,005Y + 0,1 + 0,3 + 0,05 \leq 1$, $Y \leq 110$, $N \geq 4$.

В данном случае число ведомых компьютеров должно быть не менее четырех.

Структурные схемы программ для ведущего компьютера с именем IPM и ведомого компьютера с именем IP(*n*) приведены на рис. 4. Принятые обозначения: PR – программа ведомого компьютера; IC – внутреннее кодирование кадра типа I; ID – декодирование кадра типа I; PC(*n*) – внутреннее кодирование Y макроблоков кадра типа Р для ведомого компьютера с номером *n*; PD – декодирование Y макроблоков кадра типа P; V – поиск векторов движения; PS(*n*) – компенсация движения для Y макроблоков кадров типа P; GOP – группа кадров, содержащая один кадр типа I и несколько кадров типа P, IPN – групповой адрес ведомых компьютеров, IP(*n*) – IP-адрес ведомого компьютера с номером *n*.

Для кодирования видеоконференций в стандарте H.261 при скорости передачи данных в сети 10 Мб/с 50% времени расходуется на передачу данных и 50% – на параллельные вычисления в локальной сети. Отметим, что возможность работы группы ведомых компьютеров по одной программе и использование функции пересылки данных по групповому адресу позволило уменьшить время передачи данных в два раза и обеспечить реальное время для кодирования видеоконференций.

РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ ДЛЯ СЖАТИЯ ЦИФРОВОГО ВИДЕО В СТАНДАРТЕ MPEG

Для кодирования сжатия цифрового видео в стандарте MPEG [4] необходимо решить проблему скорости обработки, которая составляет $10^{10} \dots 10^{14}$ операций/с. Стандарт MPEG отличается разнообразием видеоформатов и окон поиска, а также тем, что помимо кадров типов I и P в сжатой последовательности содержатся кадры типа B, в которых операция поиска векторов движения выполняется относительно двух опорных кадров типов I и P.

В работе [5] исследованы типы параллелизма, которые можно применять для кодирования сжатия цифрового видео в стандарте MPEG. В таблице приведено количество ведомых компьютеров, которое необходимо при сжатии цифрового видео в реальном времени, выраженное через операцию типа сложение двух операндов для различных видеоформатов и размеров окон поиска. Значения

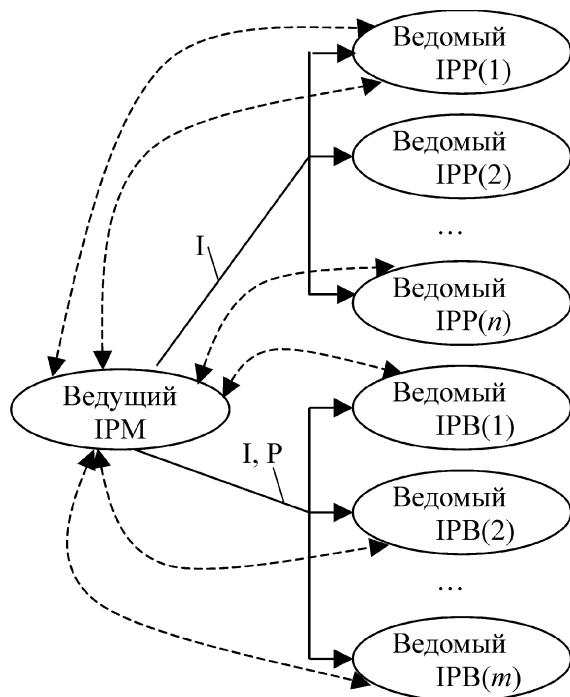


Рис. 5. Организация распределенных вычислений для стандарта MPEG

получены при следующих значениях параметров: время выполнения одной операции над восьмиразрядными операндами $0,25 \cdot 10^{-9}$ с, частота кадров 30 с^{-1} .

При организации распределенных вычислений целесообразна параллельная обработка многих макроблоков текущих кадров типа В, расположенных между опорными кадрами, и многих макроблоков одного кадра типа Р. При этом один макроблок одного текущего кадра может обрабатываться многими ведомыми компьютерами, и в каждом ведомом компьютере производится сравнение одного макроблока текущего кадра с частью окна поиска. Параллелизма, присущего задаче, при нали-

Необходимое количество ведомых компьютеров для кодирования сжатия цифрового видео в стандарте MPEG

Видеоформаты, размер кадра $A \times C$	Параметр окна поиска (в количестве макроблоков)				
	1	2	4	8	16
SIF 352×240	2	4	24	100	370
CCIR 601 720×486	7	26	102	412	1651
EDTV 960×486	9	35	137	550	2201
HDTV 1920×1080	38	152	608	2446	9784

чи неограниченного числа компьютеров в сети достаточно для организации сжатия цифрового видео в реальном времени.

Таким образом, в данном алгоритме распараллеливания будут задействованы один ведущий компьютер IPM и два типа ведомых, обрабатывающих кадры типа В – IPB(m) и кадры типа Р – IPP(n) (рис. 5). Необходимое число ведомых компьютеров зависит от используемого видеоформата (размера кадра) и окна поиска.

Кодирование в реальном времени предполагает обмен между компьютерами со скоростью 100 Мб/с для несжатых кадров или со скоростью 10 Мб/с для сжатых с помощью внутреннего кодирования кадров. Если скорость передачи данных в сети ниже, то сжатие в реальном времени невозможно.

ЗАКЛЮЧЕНИЕ

В настоящее время кодирование сжатия цифрового видео на персональном компьютере в реальном времени не производится. Для сжатия применяются или супер-ЭВМ, или специализированные параллельные процессоры высокой производительности. Распределенные вычисления в сетях из недорогих компьютеров позволят решать задачи сжатия цифрового видео в реальном времени.

В рассмотренном случае для эффективного управления распределенными вычислениями в Сети при сжатии цифрового видео существенно использование функции передачи данных по групповому адресу, позволяющее уменьшить время передачи данных и ускорить общее время решения задачи. Передачу данных по групповому адресу в сети Интернет можно обеспечить в соответствии с шестой версией протокола TCP/IP.

Анализ предложенного способа управления сетевыми ресурсами для задач сжатия цифрового видео показывает возможности эффективного применения языка ПАРСЕК к решению распределенных задач реального времени.

ЛИТЕРАТУРА

1. Foster I., Kesselman C., Tuecke S. The Anatomy of the Grid. Enabling Scalable Virtual Organizations. <http://www.globus.org/research/papers/anatomy.pdf>.
2. Затуливетер Ю.С. К единому полю компьютерной информации в концепции исчисления древовидных структур // Тр. Ин-та пробл. управл. РАН. – 2002. – Т. XVIII. – С.113–126.
3. Затуливетер Ю.С., Халатян Т.Г. ПАРСЕК – язык компьютерного исчисления древовидных структур с открытой интерпретацией. Стендовый вариант системы программирования. – М.: Ин-т пробл. управл. РАН, 1997.
4. Фищенко Е.А. Сжатие цифрового видео: методы и стандарты // Датчики и системы. – 2002. – № 8. – С.48–63.
5. Затуливетер Ю.С., Фищенко Е.А., Сухов Е.Г. Метод оптимизации параллельного процессора для сжатия цифрового видео // Тр. Ин-та пробл. управл. РАН. – 2002. – Т. XVIII. – С.94–112.

☎ (095) 334-92-09

E-mail: zvt@ipu.rssi.ru, fish@ipu.rssi.ru



УДК 681.5.015.63-192

ТЕХНОЛОГИЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В ЗАДАЧАХ АНАЛИЗА И ОПТИМИЗАЦИИ¹

О.В. Абрамов, Я.В. Катуева

Институт автоматики и процессов управления, г. Владивосток

Рассмотрена задача оптимального выбора параметров аналоговых технических устройств и систем по стохастическим критериям. Предложены параллельные алгоритмы много-вариантного анализа и оптимизации, ориентированные на компьютерные системы с массовым параллелизмом. Уделено внимание влиянию особенностей архитектуры параллельной вычислительной системы, механизма передачи данных и организации вычислительного процесса на эффективность решения задач.

ВВЕДЕНИЕ

Создание технических систем связано с решением задач синтеза либо анализа. Любой технический объект можно представить в виде двойки $W = \langle X, S \rangle$, где $X = (x_1, x_2, \dots, x_n)$ – параметры объекта, а S – его структура. Критерий качества функционирования объекта можно задать в виде функционала $\Phi = \Phi(W, Q)$, где Q – условия эксплуатации объекта.

Задача синтеза сводится к решению оптимизационной задачи

$$\Phi(W, Q) \rightarrow \underset{W \in \Omega_W}{\text{extr}} \quad W_Q^0,$$

где Ω_W – допустимое множество, в рамках которого может варьироваться объект, W_Q^0 – оптимальный в условиях Q объект. Задача синтеза допускает естественную декомпозицию на параметрическую и структурную:

$$\Phi(X, S, Q) \rightarrow \underset{X \in \Omega_X}{\text{extr}} \quad \underset{S \in \Omega_S}{\text{extr}} \quad X_Q^0 S_Q^0,$$

где Ω_X – множество допустимых параметров, Ω_S – множество допустимых структур.

¹ Работа выполнена в рамках проекта «Разработка методов и алгоритмов параметрического синтеза стохастических систем» программы фундаментальных исследований Отделения ЭММПУ РАН № 14 «Проблемы анализа и синтеза интегрированных технических, социальных и медико-биологических систем управления».

Таким образом, синтез технических объектов включает в себя две основные части: формирование структуры объекта (структурный синтез) и выбор значений внутренних параметров (параметрический синтез) [1].

Разработка объекта начинается с формирования структуры. Для выбранного варианта структуры строится модель объекта (математическая или физическая) и выбираются исходные (начальные) значения параметров, при которых проверяются условия работоспособности. В случае их невыполнения значения внутренних параметров корректируются; при этом естественно попытаться найти такие их значения, при которых условия работоспособности выполняются с наибольшей вероятностью.

Проектирование технических объектов с учетом случайности процессов изменения их параметров связано с необходимостью решения ряда сложных и трудоемких задач. К их числу относится и задача оптимального выбора номинальных значений параметров проектируемых объектов, основные трудности решения которой обусловлены вероятностным характером критерия оптимальности и дефицитом информации о случайных закономерностях процессов изменения параметров проектируемых систем.

Высокая вычислительная трудоемкость решения оптимизационных задач со стохастическим критерием заставляет искать способы достаточно быстрого получения желаемых результатов. В работах [2, 3] предложены методы и алгоритмы, убыстряющие в известной мере процедуру поиска оп-



тимальных решений в задаче параметрического синтеза.

Сравнительный анализ и эксперименты показывают, что они позволяют сократить время решения задачи в 3...5 раз по сравнению с классическими методами, основанными на прямом методе статистического моделирования.

Вместе с тем, в последние годы активно развивается достаточно радикальный путь сокращения трудоемкости решения сложных вычислительных задач, в основе которого лежит идея распараллеливания процессов поиска конечного результата и который приводит к самым разнообразным вариантам архитектуры высокопроизводительных ЭВМ. В качестве параллельных вычислителей широко применяются открытые системы массового параллелизма, состоящие из стандартных компонентов, в том числе массовых серийных микропроцессоров. Для создания подобных кластерных компьютеров сформировался как рынок аппаратных средств, так и требуемый для распределенной обработки набор программных компонентов, состоящий из некоммерческого свободно распространяемого программного обеспечения. Естественная проблема для эффективной реализации требующих высокой производительности алгоритмов и приложений заключается в том, как наилучшим способом использовать имеющиеся аппаратные средства для конкретной вычислительной задачи.

Решение задач анализа и оптимизации подразумевает отображение всей вычислительной схемы, включая методы многовариантного детерминированного и статистического анализа, а также оптимизации по стохастическим критериям, на параллельную архитектуру ЭВМ, учитывая топологию межпроцессорных связей и обеспечивая правильность взаимодействия множества параллельно выполняющихся независимо друг от друга процессов.

ЗАДАЧА ОПТИМАЛЬНОГО ПАРАМЕТРИЧЕСКОГО СИНТЕЗА

Данная задача состоит в выборе номинальных значений внутренних параметров исследуемого устройства $x_{\text{ном}} = (x_{1\text{ном}}, \dots, x_{n\text{ном}})$, обеспечивающих максимум вероятности его безотказной работы в течение заданного времени:

$$x_{\text{ном}} = \arg \max P\{X(x_{\text{ном}}, t) \in D_x, \forall t \in [0, T]\}, \quad (1)$$

где $X(x_{\text{ном}}, t)$ – случайный процесс изменения параметров; D_x – область работоспособности (область допустимых изменений параметров); T – заданное время эксплуатации устройства.

Область допустимых вариаций внутренних параметров D_x , как правило, неизвестна, поэтому условия работоспособности обычно задаются систем-

мой неравенств $a_j \leq y(x)_j \leq b_j, j = \overline{1, m}$, где $y = \{y_j\}_{j=1}^m$ – вектор выходных параметров устройства, причем $y_j = F_j(x_1, \dots, x_n)$, а $F_j(\cdot)$ – известный оператор, зависящий от топологии исследуемого устройства.

Обычная процедура получения оценок (1) вероятности безотказной работы связана со статистическим моделированием процессов изменения параметров $X(t)$ и вычислением методом Монте-Карло вероятности $P_r(t) = P\{X(t) \in D_x, \forall t \in [0, T]\}$.

Основные трудности при параметрическом синтезе технических систем с учетом случайных вариаций их параметров связаны с высокой вычислительной трудоемкостью решения возникающих при этом задач оптимизации по стохастическим критериям. На каждом шаге оптимизации необходим статистический анализ для получения оценки критерия оптимальности. При этом на основе метода статистических испытаний (Монте-Карло) многократно рассчитывается исследуемая система (устройство) при разных случайных значениях параметров элементов. Число расчетов N , равное числу реализаций случайного вектора параметров, определяется из условия обеспечения необходимой точности оценки критерия. Обычно для обеспечения требуемой точности оценки на каждом шаге поиска необходимо полностью рассчитывать систему от нескольких сотен до тысячи раз.

Традиционно алгоритм параметрического синтеза, основанный на методе прямого статистического моделирования и технологии последовательных вычислений, выглядит следующим образом.

1. Задается исходный вектор номиналов параметров $x_{\text{ном}}^{(1)}$.

2. Исходя из заданных законов распределения параметров x_1, \dots, x_n , генерируется реализация случайного вектора параметров $x^{(k)}$.

3. Для заданной реализации значений параметров вычисляются выходные параметры $y_j = F_j(x^{(k)}), j = \overline{1, m}$.

4. Проверяются условия работоспособности $y \in D_y$, где D_y – известная область допустимых значений выходных параметров y .

Наиболее трудоемкий п. 3, поскольку вычисление выходных параметров связано нередко с решением систем дифференциальных (и не всегда линейных) уравнений.

Выполнение п. 4 позволяет отнести данную реализацию $x^{(k)}$ к числу «хороших» (обеспечивающих работоспособное состояние системы) или «плохих». Этой операцией завершается первый внутренний цикл и происходит возврат к п. 2. Генерируется следующая реализация $x^{(k+1)}$, выпол-



няются пп. 3 и 4 и т. д. Общее число итераций N определяется необходимой точностью оценки вероятности нахождения параметров в области работоспособности $\hat{P} = n_x/N$, где n_x – число «хороших», принадлежащих области работоспособности, реализаций из общего числа N испытаний.

Не останавливаясь на известных процедурах определения необходимого числа испытаний N , отметим только, что обычно $N > 100$. Для вычисления оценки вероятности безотказной работы в течение определенного времени число выполнений п. 2–4 определяется необходимым числом временных сечений случайного процесса $X(t)$, так называемых, t -сечений [1].

АНАЛИЗ ВЫЧИСЛИТЕЛЬНЫХ ОСОБЕННОСТЕЙ КОМПЛЕКСА МВС-1000/16

Прежде чем создавать параллельную программу, необходимо знать общую архитектуру параллельной машины и топологию межпроцессорных связей, которая существенна для программирования. Это связано с отсутствием средств автоматического распараллеливания, которое позволяло бы превращать любую последовательную программу в параллельную и обеспечивало бы ее высокую производительность [4]. Нужно в явном виде увязывать структуру алгоритма решаемой задачи со структурой вычислительной системы и обеспечивать правильность взаимодействия множества параллельно выполняющихся независимых друг от друга процессов.

Предлагаемые параллельные алгоритмы реализованы на имеющейся в Институте автоматики и

процессов управления ДВО РАН многопроцессорной вычислительной системе МВС-1000/16, характеристики которой представлены в таблице. Она является типичным представителем параллельных систем с массовым параллелизмом (MPP – Massively Parallel Processing). Каждый процессор имеет только свою локальную память и обменивается информацией с соседями по сети. Доступ к памяти других процессоров возможен только с помощью механизма передачи сообщений.

Основное достоинство таких систем – высокая степень масштабируемости. Для достижения необходимой производительности требуется просто собрать систему с нужным числом узлов.

Главные ограничивающие факторы вычислительного процесса, характерные для систем с массовым параллелизмом и разделенной памятью – медленная скорость межпроцессорного обмена и высокая латентность.

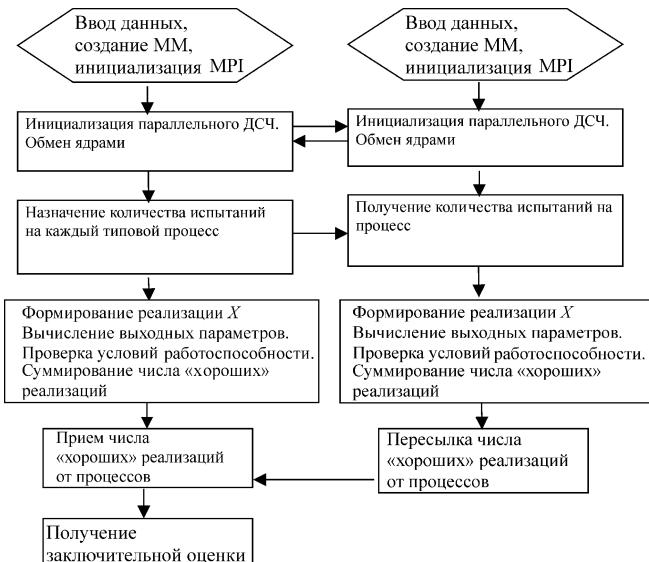
В параллельных системах данного класса особую роль играет закон Амдаля об уменьшении эффективности параллельных программ с ростом доли последовательных вычислений [4]. Кроме того, необходимо следить за общей балансированной нагрузкой процессоров и синхронизировать процессы на различных этапах выполнения алгоритмов.

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ МЕТОДА СТАТИСТИЧЕСКИХ ИСПЫТАНИЙ

Параллельные вычисления в методе статистических испытаний (метод Монте-Карло) выглядят наиболее логичными, поскольку идея параллелизма – повторения некоторого типового процесса с различными наборами данных – заложена в самой

Характеристики вычислительной системы МВС-1000/16

Архитектура	Система состоит из однородных вычислительных узлов, включающих в себя: <ul style="list-style-type: none"> • один центральный процессор (Intel Pentium III с частотой 800 МГц и кэш-памятью второго уровня 256 кБ); • локальную/оперативную память 512 МБ (прямой доступ к памяти других узлов невозможен); • сетевой адаптер; • жесткий диск HDD 10 ГБ
Межпроцессорный обмен	Связь узлов осуществляется по одной из стандартных сетевых технологий (Fast Ethernet) на базе коммутатора
Латентность (MPI)	170 мкс
Пропускная способность (MPI)	10 МБ/с
Операционная система	На каждом узле работает OS Linux Red Hat 7.2
Модель программирования	Программирование в рамках модели передачи сообщений MPI (Message Passing Interface)
Программное обеспечение	Стандартные средства Linux (GCC, G77), могут быть привязаны специализированные компиляторы (Intel C, Fortran)



Общая схема параллельного метода статистического оценивания

структуре метода. Интуитивно понятно, что применение k независимых процессоров и распределение между ними независимых испытаний уменьшает трудоемкость статистического моделирования в k раз, поскольку затраты на заключительное суммирование и осреднение результатов практически несущественны. Заключительную оценку можно получить по формуле $\hat{P} = \sum_{i=1}^k n_{gi}/N$, где n_{gi} – количество «хороших» реализаций для каждого из процессоров, N – требуемое число испытаний.

Существенное требование при распределении испытаний по процессорам заключается в отсутствии межпроцессорных корреляций при формировании реализаций векторов варьируемых параметров на различных процессорах. Для его выполнения привлекаются специальные библиотеки параллельных датчиков случайных чисел, что позволяет осуществить распределенный подход в получении последовательностей псевдослучайных чисел [5]. Применение таких приемов параллельного программирования позволяет практически избежать накладных расходов на пересылку данных и высоких временных задержек, вызванных латентностью вычислительной системы, а также серьезной разбалансировки вычислительного процесса, возникающей в ходе анализа исследуемого устройства методом критических сечений [1]. Общая схема параллельного метода статистических испытаний приведена на рисунке, где ММ – математическая модель, ДСЧ – датчик случайных чисел.

Как показывают результаты численных экспериментов, применение параллельных датчиков

случайных чисел позволяет получить на этапе статистического оценивания и анализа исследуемых устройств ускорение, близкое к линейному (для 16-ти процессоров среднее ускорение составляет около 15,6 раз).

ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ ДИСКРЕТНОЙ ПОИСКОВОЙ ОПТИМИЗАЦИИ

Универсальным и достаточно эффективным средством решения задачи (1) может стать применение параллельного аналога метода сканирования (слепого поиска).

Сущность метода заключается в том, что вся допустимая область пространства параметров разбивается на элементарные ячейки, в каждой из которых по определенному алгоритму выбирается точка: в центре ячейки, на ребрах или вершинах, случайно. Для каждой ячейки последовательно просматриваются значения целевой функции и находится среди них экстремальное. Точность метода, естественно, определяется тем, насколько плотно располагаются выбранные точки в области поиска.

Основное достоинство метода сканирования – при достаточно густом расположении точек всегда гарантируется отыскание глобального экстремума. Однако требуется значительный объем вычислений, снизить который можно распараллеливанием алгоритма. Наиболее простой алгоритм поиска экстремума методом сканирования (поиска на сетке переменных) заключается в том, что по каждой независимой переменной задаются приращения в соответствующем порядке, обеспечивающем заполнение всей исследуемой области равномерной и достаточно густой сеткой.

Поскольку номинальные значения параметров схемных элементов x_n обычно должны принадлежать ряду стандартных значений, регламентированных техническими условиями или ГОСТ, иногда предпочтительнее искать оптимальный вектор номиналов параметров на дискретном множестве номиналов, соответствующем стандартным значениям и ограниченном областью допустимых значений D_x .

Пусть известен вектор номиналов внутренних параметров $x_h^{BH} \in D_x$. Тогда в каждой точке дискретного множества $D_h^{BH} = \{x_h^{BH} / x_h \in D_x\}$ необходимо найти оценку $\hat{P}_r(x_h^{BH})$. Искомый оптимальный вектор номиналов x_h^{OPT} находим, решая задачу $x_h^{OPT} = \max_{x_h^{BH}} \hat{P}_r(x_h^{BH})$.

В простейшем случае решение задачи сводится к полному перебору элементов множества D_h^{BH} ,



для каждого из которых ищется оценка вероятности $\hat{P}_r(x_{\text{H}}^{\text{BH}})$. Формирование множества D_{H}^{BH} можно осуществить как предварительную процедуру, и его элементы поместить в базу данных исходной информации.

Оценим общую трудоемкость задачи дискретной оптимизации в случае полного перебора элементов множества D_{H}^{BH} . Если R – число элементов множества D_{H}^{BH} и для получения оценки методом Монте-Карло для каждого элемента необходимо провести N испытаний, то общее число испытаний равно RN . Соответствующий алгоритм представляет собой двухуровневый распределенный процесс, для реализации которого необходимо RN процессоров. Если $RN > k$, где k – число имеющихся процессоров, то можно представить алгоритм поиска экстремума в виде последовательно-параллельных процедур, число которых определяется количеством свободных процессоров $k_{\text{cb}} = [N/k]$, где скобки означают целую часть.

Множество D_{H}^{BH} разбивается на непересекающиеся подмножества $D_{\text{H}}^{\text{BH}} = \bigcup_{i=1}^k \{D_{\text{H}i}^{\text{BH}}\}$, каждому

j -му процессору назначается своё подмножество $D_{\text{H}j}^{\text{BH}}$ исходных данных. Таким образом, каждый из процессов последовательно находит оптимальный вектор номиналов для своей подобласти, затем результаты передаются основному процессу и выбирается оптимальный вектор по заданному критерию.

ЛИТЕРАТУРА

1. Абрамов О.В. Параметрический синтез стохастических систем с учетом требований надежности. – М.: Наука, 1992.
2. Abramov O.V., Katueva Y.V., Suponya A.A., Lazarev G.I. Computer aided design system for optimal parametric synthesis // Proc. of the 3-rd Asian Control Conference (ASCC'2000), Shanhai, 2000. – Р. 2047–2050.
3. Абрамов О.В., Катуева Я.В., Супоня А.А. Эффективные методы параметрической оптимизации по стохастическим критериям // Тр. междунар. конф. по проблемам управления. – М.: Фонд «Проблемы управления», 1999. – Т. 2. – С. 130–132.
4. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002.
5. Coddington P. Random number generators for parallel computers // NHSE Review. – 1996. – № 2.

☎(4232) 31-02-02

E-mail: abramov@iacp.dvo.ru, gloria@iacp.dvo.ru

Новая книга

Трахтенгерц Э.А., Шершаков В.М., Камаев Д.А. Компьютерная поддержка управления ликвидацией последствий радиационного воздействия. – М.: СИНТЕГ, 2004.

В первой части книги рассмотрены назначение и структура компьютерных систем поддержки управления ликвидацией последствий радиационного воздействия. Обсуждены особенности функционирования этого типа систем, в частности, необходимость принятия решений в максимально сжатые сроки при неполной информации, роль субъективных оценок и влияние фактора неопределенности. Вторая часть посвящена математическому и алгоритмическому обеспечению, в ней дано теоретическое и методологическое обоснование методов построения и эксплуатации таких систем. В третьей части дано описание конкретных алгоритмов, интерфейсов и процедур, реализованных в системе поддержки принятия решений по ликвидации последствий радиационного воздействия.

Книга предназначена для руководителей предприятий и организаций, специалистов по разработке компьютерных систем, преподавателей вузов, студентов и аспирантов, а также для широкого круга читателей, интересующихся проблемой принятия управлеченческих решений.

Э.А. Трахтенгерц – главный научный сотрудник Института проблем управления им. В.А. Трапезникова РАН, д-р техн. наук, заслуженный деятель науки РФ. Им опубликовано 11 монографий, свыше 200 печатных работ, под его руководством защищено свыше 30 докторских диссертаций. Область научных интересов – теория и методы создания и использования компьютерных систем поддержки принятия управлеченческих решений.

В.М. Шершаков – заместитель генерального директора, начальник Федерального информационно-аналитического центра НПО «Тайфун». Доктор техн. наук, автор свыше 70 научных работ. Занимается созданием и внедрением автоматизированных систем формирования и ведения банков данных, усовершенствованием и развитием систем мониторинга радиационной обстановки на территории нашей страны. Под его руководством разработана компьютерная система поддержки принятия решений управления ликвидацией последствий ядерных аварий, которая в настоящее время функционирует в системе аварийного реагирования Минатома России и ряда других министерств и ведомств.

Д.А. Камаев – начальник лаборатории математических методов Федерального информационно-аналитического центра НПО «Тайфун», канд. физ.-мат. наук. Специалист в области теории динамических систем. Им опубликовано около 60 научных работ. В настоящее время занимается разработкой методов построения систем поддержки принятия решений.



ЭФФЕКТИВНЫЕ МЕТОДЫ НАХОЖДЕНИЯ КРАТЧАЙШИХ РЕШЕНИЙ СИСТЕМ ЛИНЕЙНЫХ ЛОГИЧЕСКИХ УРАВНЕНИЙ¹

А.Д. Закревский

Объединенный институт проблем информатики, г. Минск

Предложены эффективные методы, существенно сокращающие объем перебора, производимого при поиске кратчайшего решения. Приведены результаты их программной реализации, дано оценивание конкурентоспособности разработанных программ и указаны области их применимости.

ВВЕДЕНИЕ

С задачей нахождения кратчайших решений систем линейных логических уравнений (СЛЛУ) автор встретился при разработке методов синтеза логических двухуровневых AND/EXOR-схем, обладающих некоторыми преимуществами в сравнении с традиционными AND/OR-схемами [1–3 и др.]. Например, их легче тестировать [4–6], и они оказываются более компактными при реализации симметрических булевых функций, типичных для арифметики [7]. Для решения задач их проектирования оказалось возможным привлечь хорошо развитую теорию линейных векторных пространств [8, 9]. Не менее важные приложения данная задача имеет в такой бурно развивающейся области, как защита информации [10].

В абстрактном виде задача формулируется следующим образом.

Пусть задана система из m линейных логических уравнений с n переменными:

$$\mathbf{Ax} = \mathbf{y}, \quad \bigoplus_{j=1}^n a_i^j x_j = y_i,$$

где \mathbf{A} – булева $m \times n$ матрица коэффициентов, $\mathbf{x} = (x_1, x_2, \dots, x_n)$ – булев вектор неизвестных и $\mathbf{y} = (y_1, y_2, \dots, y_m)$ – булев вектор свободных членов. Решением СЛЛУ называется такой набор зна-

чений неизвестных, который обращает каждое уравнение в тождество. В таком случае покомпонентная сумма по модулю два (называемая ниже просто суммой) тех столбцов матрицы \mathbf{A} , соответствующие которым переменные принимают в решении значение 1, будет равна столбцу \mathbf{y} .

Положим, что СЛЛУ имеет несколько решений (тогда система называется *неопределенной*, при этом обычно $n > m$). Задача заключается в выборе из них *кратчайшего решения*, т. е. такого, которое представлено значением вектора \mathbf{x} с минимальным числом единиц.

В принципе эту задачу можно решить тривиальным алгоритмом, который перебирает сочетания «из n по k » столбцов матрицы \mathbf{A} , в порядке возрастания k и начиная с $k = 1$, подсчитывает их сумму и сравнивает с вектором \mathbf{y} . Достаточно очевидно, что как только получаемая сумма оказывается равной \mathbf{y} , рассматриваемая комбинация столбцов может быть принята за искомое решение. К сожалению, данный алгоритм слишком трудоемок, поскольку число анализируемых комбинаций столбцов очень быстро растет с ростом числа неизвестных n .

В следующих разделах излагаются более эффективные комбинаторные методы решения данной задачи, ориентированные на компьютерную реализацию.

ЛЕСТНИЧНЫЙ АЛГОРИТМ

Оригинальный алгоритм поиска кратчайшего решения СЛЛУ был предложен при разработке ме-

¹ Работа доложена на конференции «Теория и практика логического управления», посвященной памяти М.А. Гаврилова (Москва, 10 ноября 2003 г.).



тодов реализации частичных булевых функций полиномами Жегалкина, что оказалось полезным при синтезе логических схем в базисе элементов AND (логическое И) и EXOR (сумма по модулю два) [11, 12].

Этот алгоритм назван лестничным, и его суть заключается в следующем. Путем соответствующего переупорядочения строк и столбцов матрица коэффициентов A приводится к ступенчатой форме, иллюстрируемой приводимым ниже примером, в котором $x = (a, b, c, d, e, f, g, h, i, j, k, l)$. При этом каждая последующая строка матрицы A не поглощается покомпонентной дизъюнкцией предыдущих (это оказывается возможным, когда конструируется компактный полином Жегалкина), а оставшиеся непоглощенными единичные элементы образуют в строке своеобразную ступеньку, ширина которой равна числу этих единиц:

$$A = \begin{matrix} \begin{array}{ccccccccccl} a & b & c & d & e & f & g & h & i & j & k & l \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} & , & \mathbf{y} = \begin{array}{c} y \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{array} \end{matrix}$$

Матрица коэффициентов A с присоединенным к ней вектором-столбцом свободных членов y образует *расширенную* матрицу системы, или *системную* матрицу $S = (A, y)$ – конкатенацию матрицы A и вектора y . Каждая ее строка представляет отдельное уравнение. Если $y = 0$, оно будет удовлетворено при включении в решение четного числа столбцов из A , отмеченных в данной строке значением 1. Если же $y = 1$, в решение потребуется включить нечетное число таких столбцов. Учитывая это требование, мы организуем поиск по дереву [13], стремясь найти минимальную совокупность столбцов, сумма которых равна вектору y . Этот процесс можно интерпретировать как бег по лестнице, в котором спуски (вниз по матрице, с целью решения последующих уравнений) чередуются с подъемами (с целью возможного упрощения решения системы в целом путем испытания альтернативных комбинаций).

Для рассматриваемого примера процесс отображается следующим деревом поиска:

$$\begin{array}{ccccccccccl} & a & b & c & d & e & f & g & h & i & j & k \\ \hline & 0 & 0 & 0 & - & 1 & 0 & 0 & - & 1 & 0 & - + \\ & | & & & & | & & & | & & & \\ * & & & & & 0 & 1 & - & * & & & \\ & & & & & | & & & | & & & \\ & & & & & 0 & 1 & 0 & - & 1 & 0 & - + \\ & & & & & | & & & | & & & \\ & & & & & 0 & 0 & 1 & - & * & & \end{array}$$

Прокомментируем его. Процесс начинается с анализа первой сверху строки и выбора комбинации значений неизвестных a, b, c, d , перечисленных в верхней ступеньке. Минимизируя число единиц, выбираем комбинацию 0000, удовлетворяющую первому уравнению. Рассматривая следующие строки и ступеньки, выбираем $(e, f, g) = 000$, $(h, i) = 10$, $(j, k) = 10$ и находим таким образом первое решение – тройку столбцов (e, h, j) , сумма которых равна вектору y . Далее пытаемся найти лучшее решение, содержащее не более двух столбцов. С этой целью идем вверх по лестнице и испытываем комбинацию 01 значений переменных h, i (с отрицательным результатом) и комбинацию 010 значений переменных e, f, g (что приводит затем к нахождению двухстолбцового решения (f, h)). Это окончательное решение, поскольку найти лучшее решение не удается.

Более подробно лестничный алгоритм описан в работе [12], а в статье [14] предложен его ускоренный вариант, основанный на минимизации ширины верхних ступенек, что существенно сокращает объем перебора комбинаций, производимого при поиске оптимального решения.

МЕТОД ГАУССА

Более перспективным оказался метод Гаусса исключения переменных. Он позволяет ограничить поиск решения рассмотрением только таких значений вектора x , которые заведомо являются корнями системы, что сокращает число рассматриваемых подмножеств в 2^m раз, в сравнении с тривиальным алгоритмом. Данный метод был предложен Гауссом для решения систем линейных уравнений с действительными переменными [15] и описывается ниже в модификации для нашего случая: переменные – булевые, сумма – по модулю два, произведение – логическое (конъюнкция).

Метод Гаусса основан на следующих положениях. Две системы (а также представляющие их системные матрицы S_i и S_j) называются эквивалентными, если множества их корней равны. Матрицы S_i и S_j эквивалентны, если каждая строка одной из них (любой) равна сумме некоторых строк другой матрицы. Любую матрицу S_i , эквивалентную данной матрице S , можно получить из S некоторой последовательностью операций сложения строк. Эта операция заключается в замене произвольной строки рассматриваемой матрицы суммой этой строки с некоторой другой строкой этой же матрицы. Можно также дублировать некоторые строки и “приводить подобные”, т.е. устранять дубли. Минимальная по числу строк матрица S_i , эквивалентная исходной матрице S , состоит из ли-

нейно независимых строк — никакая из них не равна сумме некоторых других. Число этих независимых строк называется рангом матрицы S и обозначается через $r(S)$.

Метод Гаусса приводит матрицу S к удобной для последующего анализа форме, называемой *канонической*. Делается это (для булевой матрицы) следующим образом.

- Вначале все строки и столбцы матрицы S считаются неотмеченными.
- На очередном шаге выбирается первый из столбцов (за исключением y), содержащих единицу в какой-либо из неотмеченных строк, а также первая из неотмеченных строк, содержащих единицу в этом столбце.
- Каждая строка (кроме выбранной), содержащая единицу в выбранном столбце, заменяется ее суммой с выбранной строкой. При получении строки $\mathbf{0}$ (все компоненты равны нулю) она удаляется (так устраняется избыточность в системе). Если же результат суммирования представляется строкой с единственной единицей в правой компоненте (такая строка задает уравнение $0 = 1$, что свидетельствует о противоречивости системы), то алгоритм прекращает работу.
- Выбранные таким образом строка и столбец отмечаются (полужирной единицей на их пересечении), после чего выполняется следующий шаг. Алгоритм прекращает работу, когда все строки или все столбцы становятся отмеченными.

В результате получается каноническая матрица S^* , эквивалентная исходной матрице S . Совокупность отмеченных столбцов (их число равно рангу матрицы и каждый из них содержит ровно по одной единице) назовем базисом, совокупность остальных — *остатком*. Матрица S^* представляет в компактной форме множество всех корней рассматриваемой системы уравнений, соответствующих различным подмножествам столбцов из остатка.

Например,

$$S = \begin{array}{ccccccccc} a & b & c & d & e & f & g & h & y \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{array} \rightarrow S^* = \begin{array}{ccccccccc} a & b & c & d & e & f & g & h & y \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{array}$$

Рассматриваемая в этом примере матрица S является неопределенной, $n > m$ и $m = r(S)$. Число корней такой матрицы равно 2^{n-m} , в данном случае восьми. В ходе пошагового преобразования матрицы в базис попадают столбцы, соответствующие переменным a, b, c, d, e , а остаток представляется

множеством столбцов, соответствующих переменным f, g, h . Каждое из восьми подмножеств этого множества определяет одно из решений, которое легко получить, складывая столбцы подмножества и дополняя их некоторыми столбцами из базиса так, чтобы получить в сумме столбец y . Например, включив в решение переменные f и h и сложив соответствующие им векторы 11010 и 01101 с вектором 10101, мы получим вектор 00010, откуда следует, что решение надо дополнить переменной d . Так находится одно из восьми решений системы, а именно $y = d \oplus f \oplus h$.

Чтобы решить поставленную выше задачу, достаточно перебрать таким образом все 2^{n-m} корней системы и выбрать из них кратчайший. Можно сократить перебор в процессе поиска, если уже найден некоторый корень с весом v . В этом случае достаточно ограничиться поиском лучшего решения, перебирая в остатке подмножества мощностью менее v . Эта идея реализована в описываемом ниже алгоритме.

Допустим, что рассматриваемая система линейных логических уравнений задана введенной выше системной булевой матрицей S , причем $n > m$. Тогда кратчайший корень системы можно найти следующим образом [16].

- Приведем матрицу S к канонической форме S^* , определив ее базис и остаток. Для этого используется описанный выше алгоритм, попутно проверяющий систему на совместность.
- Будем рассматривать подмножества столбцов из остатка в порядке возрастания их мощности k и взвешивать определяемые ими корни. Вес корня равен количеству выбранных столбцов плюс вес булева вектора (число единиц в нем), представляющего сумму (покомпонентную, по модулю два) этих столбцов и вектора y . По ходу будем запоминать решение с минимальным весом v . Прекратим перебор, как только k сравняется с v . В этом случае запомненное решение представит кратчайший корень системы.

Полезно оценить ожидаемый вес $\gamma(m, n)$ кратчайшего корня системы уравнений с параметрами m и n еще до его нахождения — тогда можно будет предсказать и затраты времени на его поиск.

Начнем с нахождения математического ожидания $a(m, n, k)$ числа корней с весом k . Положим, что рассматриваемая система случайна — каждый элемент булевой матрицы S принимает значение 1 с вероятностью 0,5. Тогда вероятность того, что случайно выбранное подмножество столбцов в матрице A окажется корнем, равна 2^{-m} (будучи равна вероятности совпадения двух булевых векторов с m компонентами, один из которых случаен). Отсюда следует



Утверждение 1. $\alpha(m, n, k) = C_n^k 2^{-m}$, где C_n^k – число сочетаний из n по k .

Аналогично, обозначив через $\beta(m, n, k)$ математическое ожидание числа корней, вес которых не превышает k , получим

Утверждение 2. $\beta(m, n, k) = \sum_{i=0}^k C_n^i 2^{-m}$.

Искомый вес кратчайшего корня оценим максимальным значением параметра k , при котором $\beta(m, n, k) < 1$.

Утверждение 3. $\gamma(m, n) = k$, где $\beta(m, n, k) < 1 \leq \beta(m, n, k+1)$.

Например, в табл. 1 показаны значения величин α и β для системы из 40 уравнений с 70-ю переменными, соответствующие значениям параметра k от 7 до 13. Достаточно очевидно, что вес кратчайшего решения, скорее всего, будет равен 10.

Оценивая объем вычислений при поиске кратчайшего решения, положим, что он пропорционален числу N рассматриваемых подмножеств столбцов остатка. Очевидно, значение N определяется уровнем перебора, который мы обозначим $\delta(m, n)$.

Утверждение 4. $N(m, n) = \sum_{i=0}^{\delta(m, n)} C_{n-m}^i$.

Достаточно очевидно, что объем перебора в сильной степени зависит от мощности остатка – числа столбцов в нем, которое равно разности $n - m$. В случае, когда это число оказывается меньшим веса кратчайшего корня, для гарантированного нахождения последнего приходится перебирать все 2^{n-m} подмножества столбцов в остатке, что приводит к удвоению времени поиска решения при каждом увеличении разности $n - m$ на единицу. Легко подсчитать, что при увеличении разности $n - m$ на 10 времена поиска возрастает примерно в тысячу раз.

Как показывают аналитические и экспериментальные (C++, PC COMPAC Presario – процессор Intel Pentium III, 1000 МГц) оценки эффективности рассмотренного метода, область его практического

приложения ограничивается условием $n - m < 30$. При этом условии время поиска не превышает одного часа, в то время как с дальнейшим ростом разности $n - m$ расходы времени резко возрастают. Например, при $m = 625$ и $n = 700$ данный алгоритм будет вынужден перебирать и испытывать в поисках кратчайшего решения все 2^{75} комбинаций наборов столбцов из остатка, и расчеты показывают, что на это уйдет три миллиарда четыреста миллионов лет!

РАСПОЗНАВАНИЕ КОРОТКИХ РЕШЕНИЙ

Между тем, кратчайшее решение, вес которого $\gamma(m, n)$ можно заранее оценить по формуле утверждения 3, будет содержать в остатке в среднем $\gamma(m, n) \cdot (n - m)/n$ элементов, откуда следует, что оно, скорее всего, будет найдено на уровне перебора $\delta(m, n) = \lfloor \gamma(m, n) \cdot (n - m)/n \rfloor$ – окаймляющие скобки означают «целое, ближнее снизу». В этом случае его можно распознать по весу и прекратить поиск, отказавшись от строгого доказательства оптимальности найденного решения и сэкономив тем самым время. При $m = 625$ и $n = 700$ ожидаемый вес γ кратчайшего решения равен 220, следовательно, уровень перебора $\delta = \lfloor 220 \cdot 75/700 \rfloor = 23$. На этом уровне решение будет найдено за 1 800 000 лет, примерно в 1900 раз быстрее, чем при полном переборе, однако недостаточно быстро с практической точки зрения. На более существенную экономию времени можно рассчитывать, удовлетворившись некоторым приближением к сверху.

Практический интерес представляет случай существования короткого решения СЛЛУ со случайной матрицей A и вектором y , равным сумме некоторых w столбцов этой матрицы (при этом w заметно меньше γ). Например, при $m = 625$, $n = 700$ и $w = 105$ уровень перебора δ будет в среднем равен 11, откуда следует, что затраты времени на нахождение данного решения методом распознавания составят 193 дня. Однако, если повезет, уровень перебора может оказаться и ниже, что приведет к большой экономии времени. Так, при проведении эксперимента на компьютере указанное короткое решение было обнаружено на уровне перебора 10, что соответствует затратам времени в 25 дней.

ДЕКОМПОЗИЦИОННЫЙ МЕТОД НЕПЕРЕСЕКАЮЩИХСЯ ОСТАТКОВ

При больших значениях разности $n - m$ решение рассматриваемой задачи можно существенно ускорить, применяя декомпозиционный метод не-

Таблица 1
Значения α и β при $m = 40$, $n = 70$ и $k = 7 \dots 13$

k	α	β
7	0,001	0,001
8	0,009	0,010
9	0,059	0,069
10	0,361	0,430
11	1,968	2,398
12	9,676	12,074
13	43,170	55,244

пересекающихся остатков, понижающий уровень перебора [17–19].

Предположим, что в матрице A можно найти q таких максимальных совокупностей линейно независимых столбцов, которым соответствуют взаимно непересекающиеся остатки. Заметим, что при этом должно выполняться отношение $n \geq q(n - m)$. Построим множество Q , составленное из q соответствующих канонических форм, базисы которых строятся на основе указанных совокупностей (очевидно, что это число ограничено отношением $q \leq n/(n - m)$), и будем искать оптимальное решение в рамках этих форм, последовательно увеличивая уровень перебора.

Утверждение 5. В множестве Q найдется каноническая форма, где решение с весом v можно найти на уровне перебора не выше $\lfloor v/q \rfloor$ – натурального числа, ближнего снизу к частному v/q .

Утверждение 6. Кратчайшее решение системы можно найти путем последовательного рассмотрения остатков канонических форм из множества Q , ограничивая уровень перебора подмножеств в этих остатках величиной $\lfloor (v - 1)/q \rfloor$, где v – вес кратчайшего из решений, найденных к текущему моменту времени.

Эти утверждения образуют основу предлагаемого метода. Обратим внимание на существенное понижение уровня перебора подмножеств в этом методе, в котором реализуется перебор подмножеств во всех остатках сначала на уровне 0, затем на уровне 1, и т. д., пока этот уровень не превысит $\lfloor (v - 1)/q \rfloor$.

Утверждение 7. Число $N_r(m, n)$ подмножеств столбцов, рассматриваемых при методе непересекающихся остатков, определяется следующей формулой:

$$N_r(m, n) = q \sum_{i=0}^p C_{n-m}^i, \text{ где } p = \lfloor \gamma/q \rfloor.$$

Предложенный метод позволяет существенно сократить время гарантированного нахождения кратчайшего решения, в сравнении с описанным выше методом Гаусса. Для того же примера ($m = 625$, $n = 700$) уровень перебора понизится с 220 до 23, как и у метода распознавания, с соответствующим сокращением времени поиска. Однако, в отличие от последнего, при этом гарантируется оптимальность получаемого решения.

Обратим внимание на одно весьма существенное для дальнейших рассуждений обстоятельство. Столбцы матрицы A , составляющие кратчайшее решение, распределяются по остаткам канонических форм случайно, т. е. не равномерно, а с некоторой дисперсией, благодаря чемуайдется ос-

таток с минимальным числом k вошедших туда столбцов, заметно меньшим, чем $\lfloor \gamma/q \rfloor$ (обозначим через s номер этого остатка). Это позволяет, объединив методы декомпозиции и распознавания, найти кратчайшее решение на уровне перебора k .

Наибольший эффект такое объединение дает в случае существования короткого решения с весом w . Например, при $m = 625$ и $n = 700$ число канонических форм q равно 9. При анализе на компьютере примера с данными параметрами и $w = 105$ оказалось, что $k = 7$ и именно таким числом элементов короткого решения обладает остаток третьей формы ($s = 3$). Расчеты по методу, предложенному в работе [20], показывают, что в данном случае решение будет найдено за 7,5 ч.

ОБОБЩЕННЫЙ ДЕКОМПОЗИЦИОННЫЙ МЕТОД

Условие не пересечения остатков канонических форм существенно для метода, гарантирующего оптимальность находимых решений, но не для метода распознавания короткого решения, где более важно понизить уровень перебора. С учетом этого соображения декомпозиционный метод был обобщен следующим образом. На базе различных выборов из матрицы A максимальных множеств линейно независимых столбцов строится несколько (обозначим их число через p) серий канонических форм, причем в каждой из этих серий данное условие не пересечения выполняется. Таким образом, число канонических форм увеличивается в два, три или более раз, что приводит к соответствующему росту времени на их анализ. Однако с большой вероятностью этот расход времени с лихвой компенсируется снижением значения параметра k , определяющего уровень производимого перебора.

В результате возникает выигрыш, хорошо иллюстрируемый на примере той же СЛЛУ с параметрами $m = 625$, $n = 700$, обладающей коротким решением ($w = 105$). Так, в эксперименте на компьютере при $p = 2$ (две серии, $q = 18$) решение находится на уровне перебора $k = 6$ ($s = 12$) примерно за три часа. А при $p = 3$ (три серии, $q = 27$) решение находится на уровне перебора $k = 4$ ($s = 12$), всего за 78 с!

ЭКСПЕРИМЕНТЫ

Приведенный выше пример оказался весьма выигрышным для предлагаемых методов нахождения короткого решения СЛЛУ, демонстрируя снижение уровня перебора с 10 до 4, а времени решения – с 25 дней до 78 с.



Таблица 2

Результаты испытания методов нахождения короткого решения случайных СЛЛУ

№	$p = 0$		$p = 1$		$p = 2$		$p = 3$	
	$q-s-k$	t_0	$q-s-k$	t_1	$q-s-k$	t_2	$q-s-k$	t_3
1	1-1-12	74 100 000	9-2-8	177 000	18-13-6	11 200	27-13-5	996
2	1-1-12	74 100 000	9-3-7	27 200	18-3-7	34 900	27-3-7	42 600
3	1-1-10	2 190 000	9-3-7	27 200	18-12-6	10 400	27-12-4	78
4	1-1-12	74 100 000	9-4-6	3 410	18-4-6	4 070	27-4-6	4 730
5	1-1-11	13 400 000	9-4-7	34 900	18-11-5	807	27-13-6	11 800
6	1-1-9	320 000	9-3-7	27 200	18-3-7	34 900	27-3-7	42 600
7	1-1-13	373 000 000	9-3-8	243 000	18-17-6	14 300	27-20-7	175 000
8	1-1-9	320 000	9-5-7	42 700	18-5-7	50 400	27-5-7	58 200
9	1-1-13	373 000 000	9-3-8	243 000	18-10-7	89 200	27-3-8	398 000
10	1-1-12	74 100 000	9-2-6	1 840	18-2-6	2 500	27-2-6	3 160
11	1-1-9	320 000	9-1-9	917 000	18-1-9	1 590 000	27-27-7	229 000
12	1-1-8	41 600	9-1-8	111 000	18-18-6	15 100	27-27-6	22 800
13	1-1-11	13 400 000	9-6-7	50 400	18-6-7	58 200	27-23-5	1 670
14	1-1-10	2 190 000	9-7-8	507 000	18-15-7	128 000	27-7-8	662 000
15	1-1-8	41 600	9-3-6	2 620	18-3-6	3 280	27-3-6	3 940
16	1-1-9	320 000	9-5-8	375 000	18-11-6	9 580	27-16-6	14 200
17	1-1-10	2 190 000	9-2-6	1 840	18-2-6	2 500	27-2-6	3 160
18	1-1-15	7 270 000 000	9-2-8	177 000	18-15-7	128 000	27-11-5	10 200
19	1-1-14	1 720 000 000	9-4-9	2 390 000	18-13-6	11 200	27-13-6	11 800
20	1-1-15	7 270 000 000	9-8-6	6 560	18-8-6	7 220	27-8-6	7 880
21	1-1-10	2 190 000	9-7-6	5 780	18-7-6	6 430	27-7-6	7 090
22	1-1-9	320 000	9-3-7	27 200	18-3-7	34 900	27-22-6	18 900
23	1-1-12	74 100 000	9-9-7	73 700	18-9-7	81 500	27-20-5	1 470
24	1-1-4	3	9-1-4	8	18-1-4	15	27-1-4	25
25	1-1-12	74 100 000	9-4-5	284	18-4-5	335	27-4-5	388
26	1-1-7	4 740	9-1-7	11 600	18-1-7	19 400	27-1-7	27 100
27	1-1-13	373 000 000	9-8-5	554	18-8-5	605	27-8-5	658
28	1-1-11	13 400 000	9-4-6	3 410	18-4-6	4 070	27-4-6	4 730
29	1-1-11	13 400 000	9-5-9	2 880 000	18-18-6	15 100	27-18-7	159 000
30	1-1-13	373 000 000	9-4-7	34 900	18-4-7	42 700	27-4-7	50 400

Более объективно эффективность данных методов характеризуется табл. 2, где показаны прогнозируемые результаты решения тридцати случайных СЛЛУ с теми же параметрами ($m = 625$, $n = 700$, $w = 105$) четырьмя методами. Значением $p = 0$ условно отмечен метод Гаусса, значением $p = 1$ – декомпозиционный метод на одной серии канонических форм с непересекающимися остатками, и значениями $p = 2$ и $p = 3$ – обобщенные декомпозиционные методы на двух и трех сериях. Результаты применения этих методов отображаются тройками значений параметров $q - s - k$ и временем t в секундах, затраченным на нахождение решения. Например, тройка 27-12-4 означает, что в процессе решения анализировалось 27 канонических форм и процесс успешно завершился при рассмотрении остатка формы 12 на уровне перебора 4. Полужирным шрифтом отмечены моменты, в которых рассматриваемый метод выигрывает по эф-

фективности у всех предшествующих (представленных столбцами слева).

Поскольку время приведено в секундах, заметим, что час содержит 3600 с, сутки – 86 400 с, год – 31 536 000 с. Так, пример 20 решается методом Гаусса ($p = 0$) за 230 лет, а методом декомпозиции ($p = 1$) – за два часа (в миллион раз быстрее).

ЗАКЛЮЧЕНИЕ

Нахождение кратчайшего решения неопределенной системы линейных логических уравнений является ярким примером трудных комбинаторных задач. Для их успешного решения необходимо не только привлекать компьютеры, но и разрабатывать эффективные комбинаторные методы, учитывающие особенности рассматриваемых задач. Такие методы, предложенные в настоящей статье, на порядки (в тысячи раз и более) могут сокращать время, затрачиваемое на поиск кратчайших решений.

**ЛИТЕРАТУРА**

1. *Besslich P.W.* Efficient computer method for XOR logic design // IEE Proc. on Computers & Digital Techniques. – 1983. – Vol. 130. – P. 203–206.
2. *Sasao T.* AND-EXOR expressions and their optimization // Logic Synthesis and Optimization / Kluwer Academic Publishers. – 1993. – P. 287–312.
3. *Miller J.F., Thomson P.* Highly efficient exhaustive search algorithm for optimizing canonical Reed-Muller expansions of boolean functions // Int. J. Electronics. – 1994. – Vol. 76. – P. 37–56.
4. *Reddy S. M.* Easily testable realizations for logic functions // IEEE Trans. – 1972. – C-21. – P. 1183–1188.
5. *Debany W.H., Hartmann C.R.P., Snethen T. J.* Algorithm for generating optimal tests for exclusive-OR nets // IEE Proc. – 1991. – Vol. 138. – № 2. – P. 93–96.
6. Закревский А.Д., Закревский Л.А. Диагностирование схем, построенных из элементов "сумма по модулю 2" // Докл. АН Беларуси. – 1996. – Т. 10. – № 6. – С. 7–11.
7. Suprun V.P. Fixed polarity Reed-Muller expressions of symmetric Boolean functions // Proc. IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion in Circuit Design. – Makuhari, Chiba, Japan, – 1995. – P. 246–249.
8. Кострикин А. И., Манин Ю. И. Линейная алгебра и геометрия. – М.: Наука, 1986.
9. Ланкастер П. Теория матриц. – М.: Наука, 1978.
10. Закревский А. Д. К решению систем линейных логических уравнений с искаженными правыми частями – условия корректной постановки задачи // Информ. системы и технологии (IST'2002): Mater. I Междунар. конф., Минск, 5–8 ноября 2002 г. – Минск 2002. – Ч. 2. С. 45–50.
11. Zakrevskij A.D. Looking for shortest solutions of linear logical equations: theory and applications in logic design // 2. Work-
- shop «Boolesche Probleme», 19/20 Sept., 1996. – Freiberg / Sachsen. P. 63–69.
12. Закревский А.Д., Торопов Н.Р. Полиномиальная реализация частичных булевых функций и систем. – Минск: Инт. техн. кибернетики НАН Беларуси, 2001.
13. Нильсон Н. Искусственный интеллект. Методы поиска решений. – М.: Мир, 1971.
14. Торопов Н.Р. Ускорение лестничного алгоритма решения системы линейных логических уравнений // Методы логического проектирования / ОИПИ НАН Беларуси. – Минск, 2003. – Вып. 2 – С. 103–114.
15. Gauss C.F. Beitrage zur Theorie der algebraischen Gleichungen. – Gött, 1849.
16. Закревский А.Д., Василькова И.В. Быстрый алгоритм нахождения кратчайшего решения системы линейных логических уравнений // Методы логического проектирования / ОИПИ НАН Беларуси. – Минск, 2002. – Вып. 1. – С. 5–12.
17. Закревский А.Д. Декомпозиционные методы нахождения кратчайшего решения системы линейных логических уравнений // Докл. НАН Беларуси. – 2003. – Т. 47. – № 2. – С. 58–60.
18. Закревский А.Д. Оптимизация решений в линейном булевом пространстве – методы декомпозиции // Автоматика и вычислительная техника. – 2003. – № 5. – С. 28–36.
19. Zakrevskij A.D., Zakrevski L.A. Optimizing solutions in a linear Boolean space – a decomposition method // Proc. of STI '2003, Orlando, Florida, USA, July 2003. P. 276–280.
20. Закревский А.Д., Василькова И.В. Прогнозирование затрат времени на выполнение комбинаторных алгоритмов // Методы логического проектирования / ОИПИ НАН Беларуси. – Минск, 2003. – Вып. 2.

E-mail: zakr@newman.bas-net.by

☎ (10-375-17) 288-65-99

**ПРАВИЛА ОФОРМЛЕНИЯ СТАТЕЙ
ДЛЯ ПУБЛИКАЦИИ В ЖУРНАЛЕ
«ПРОБЛЕМЫ УПРАВЛЕНИЯ»**

Статья представляется в редакцию на бумаге в 2-х экземплярах, с аннотацией и направлением организации, а также обязательно в электронном виде на диске 3,5 дюйма или по электронной почте (не более 2 МБ). Аннотация, название статьи и фамилии авторов должны быть представлены также на английском языке. Примерный объем оригинальной статьи 12, обзорной – 18 страниц текста. Текст печатается через 2 интервала с одной стороны бумаги формата А4, страницы нумеруются. В электронной форме текст должен быть в редакторе не ниже Word 97 шрифтом № 12 Times New Roman; текст не форматируется, т.е. не имеет табуляции, колонок и т.д. Рисунки должны иметь расширение, совместимое с Word 97, или в формате CorelDraw: фотографии должны быть предельно четкими, черно-белыми, на глянцевой бумаге или в электронном виде TIFF с разрешением 300 dpi. Толщина линий рисунков, представляемых в электронной форме – не менее 3 пикселей.

Все буквенные обозначения, приведенные на рисунках, необходимо пояснить в основном или подрисункочном текстах (недопустимы повторные обозначения – в подрисункочных подписях и в тексте). Нумеровать следует только те формулы, на которые есть ссылка в последующем изложении. Список литературы (только органически связанный со статьей) составляется в порядке цитирования идается в конце статьи. Ссылки на литературу в тексте отмечаются порядковыми номерами в квадратных скобках.

В конце статьи следует указать номер контактного телефона и электронный адрес. На отдельном листе дать сведения об авторах (полностью ф.и.о., ученая степень, должность и место работы, почтовый адрес и тел.)

УДК 519.27

АНАЛИЗ И ОПЕРАТИВНАЯ ДИАГНОСТИКА СИСТЕМ, ОПИСЫВАЕМЫХ НЕСТАЦИОНАРНЫМИ СЛУЧАЙНЫМИ ПРОЦЕССАМИ¹

Е.А. Гребенюк

Институт проблем управления им. В.А. Трапезникова, г. Москва

Рассмотрена задача обнаружения изменений свойств коинтегрированных нестационарных процессов. Для обнаружения нарушения коинтеграционной связи предложен алгоритм кумулятивных сумм. Эффективность предлагаемого подхода продемонстрирована решением задачи анализа группы индексов российских финансовых рынков, описываемых нестационарными временными рядами.

ВВЕДЕНИЕ

Большинство процессов в экономике, демографии и социологии представляет собой так называемые «развивающиеся процессы», анализ которых в силу их нестационарного характера зачастую невозможен традиционными методами. При определенных условиях между нестационарными процессами может существовать взаимосвязь, называемая коинтеграцией, методы анализа и исследования таких взаимосвязей относятся к коинтеграционному анализу [1, 2]. Существование коинтеграционных связей объясняется наличием «общих трендов» [3, 4]. Если связи между процессами – устойчивые, то можно предположить, что система, поведение которой описывается этими процессами, тоже устойчивая, а нарушение связей может привести к дисбалансам в системе.

Периодически в процессах происходят изменения. Изменения хотя бы в одном из процессов приводят к разрушению коинтеграционных связей. Последствия этих разрушений зависят напрямую от содержания происходящих явлений и дальнейшего поведения системы. Связи могут восстановиться через некоторое время на новом уровне, если соответственные изменения произойдут и в

остальных процессах, все компоненты могут вернуться в прежнее состояние и, наконец, нарушение связей может привести к разрушению системы. Поэтому нахождение связей, анализ их поведения и обнаружение изменений свойств нестационарных компонент, приводящих к разрушению связей, являются мощным средством анализа систем, описываемых нестационарными процессами.

В настоящей работе предлагается подход к анализу и прогнозу систем, описываемых нестационарными рядами, основанный на выделении интервалов, в которых не изменяются параметры нестационарных процессов, построении в этих интервалах моделей коинтеграционных связей и отслеживании изменений как в самих процессах, так и образуемых ими долговременных коинтеграционных связях. Для обнаружения изменений в нестационарных процессах и выделения стационарных интервалов построены алгоритмы обнаружения типа кумулятивных сумм, которые в случае точно известных параметров процесса обладают оптимальными свойствами по критерию наискорейшего обнаружения [5]. В рассматриваемом нами случае параметры процесса после изменения свойств неизвестны, могут быть сделаны только некоторые предположения относительно типа их изменений. Поэтому для обнаружения изменений мы пользуемся модификациями алгоритмов кумулятивных сумм, которые не являются оптимальными, но гарантируют приемлемое качество обнаружения.

¹ Статья рекомендована к печати Программным комитетом Второй международной конференции по проблемам управления (Москва, 2003 г.).



ПОСТАНОВКА ЗАДАЧИ

Рассмотрим систему, поведение которой описывается векторной случайной последовательностью временных рядов:

$$Y_1, Y_2, \dots, Y_t, \dots, Y_{t_a+1}, \dots,$$

где $Y_t = (y_{1t}, y_{2t}, \dots, y_{pt})^T$.

Предположим, что последовательность удовлетворяет следующим условиям.

1. Компоненты вектора Y_t описываются моделями вида:

$$y_{it} = \bar{y}_{it} + \varepsilon_{it}, \quad (1)$$

где ε_{it} – стационарный процесс, $\bar{y}_{it} = \mu_i + \bar{y}_{it-1} + v_{it}$, $v_{it} \sim N(0, \delta_v)$, v_{it} и ε_{it} – независимые процессы, μ_i – константа, определяющая размер и направление дрейфа, $i = 1, 2, \dots, p$.

2. Существует постоянная $p \times p$ -матрица C такая, что процесс

$$Z_t = CY_t = \sum_{j=0}^k A_j E_t,$$

где

$$A_j = \begin{pmatrix} \alpha_{11}^j & \dots & \alpha_{ip}^j \\ \dots & \dots & \dots \\ \alpha_{p1}^j & \dots & \alpha_{pp}^j \end{pmatrix}, \quad E_t = (\varepsilon_{1t}, \dots, \varepsilon_{pt})^T,$$

$$j = 0, 1, 2, \dots, k,$$

является стационарным.

3. В некоторый неизвестный момент времени t_a в $k \leq p$ компонентах процесса Y_t возникают изменения следующих типов:

- изменение стохастического тренда

$$v_{it} = \begin{cases} v_{it}^1, & \text{если } t < t_a \\ v_{it}^2, & \text{если } t \geq t_a \end{cases}, \quad (2)$$

где

$$\delta^2(v_{it}^1) \neq \delta^2(v_{it}^2), \quad \delta^2(v_{it}^j), j = 1, 2 - \quad (3)$$

дисперсия процесса v_{it}^j , $i = 1, 2, \dots, k$, до изменения ($j = 1$) и после изменения ($j = 2$).

- изменение дрейфа:

$$\mu_i = \begin{cases} \mu_i^1, & \text{если } t < t_a \\ \mu_i^2, & \text{если } t \geq t_a \end{cases}. \quad (4)$$

Требуется обнаружить изменение свойств процесса, т. е. определить моменты изменения и характер изменения моделей отдельных компонент и связей между компонентами.

АНАЛИЗ ПОСЛЕДСТВИЙ ИЗМЕНЕНИЙ СВОЙСТВ НЕСТАЦИОНАРНЫХ ПРОЦЕССОВ

Изменения в системе в соответствии с п. 3 постановки задачи могут возникать в одном или сразу в нескольких процессах (компонентах). Рассмотрим типы возможных изменений и влияние изменений в отдельной компоненте на коинтеграционную связь.

Нестационарный процесс, k -е разности которого стационарны, называют интегрированным k -го порядка. Если процессы y_1, y_2, \dots, y_n ($n \geq 2$) интегрированные порядка k , и их линейная комбинация является процессом, интегрированным порядка $k-1$, то они называются коинтегрированными порядка k , а связь между ними – коинтеграцией.

Пусть исходный процесс описывается моделью вида (1). Рассмотрим первые разности процесса в случае, когда в нем изменяется стохастический тренд (2):

$$\Delta y_t = \begin{cases} \mu + v_t^1 + \varepsilon_t - \varepsilon_{t-1}, & \text{если } t < t_a \\ \mu + v_t^2 + \varepsilon_t - \varepsilon_{t-1}, & \text{если } t \geq t_a \end{cases}.$$

Из выражения (3) следует, что ряд разностей Δy_t после момента t_a изменяет дисперсию. Рассмотрим первые разности процесса в случае, когда в нем изменяется дрейф (4):

$$\Delta y_t = \begin{cases} \mu^1 + v_t + \varepsilon_t - \varepsilon_{t-1}, & \text{если } t < t_a \\ \mu^2 + v_t + \varepsilon_t - \varepsilon_{t-1}, & \text{если } t \geq t_a \end{cases}$$

(индекс i здесь опущен).

Ряд разностей Δy_t после момента t_a изменяет среднее, поэтому в случае детерминированного тренда задача обнаружения сводится к обнаружению изменения среднего в стационарной последовательности Δy_t .

Рассмотренные изменения приводят к нарушению коинтеграции, если они происходят только в одном из процессов.

Пусть имеются два ряда y_{1t} и y_{2t} , коинтегрированных порядка 1, тогда существуют ненулевые константы $\beta_1 = 1$ и $\beta_2 = \beta$, для которых их линейная комбинация стационарна. Это может быть тогда и только тогда, когда

$$\bar{y}_{1t} + \beta \bar{y}_{2t} = 0. \quad (5)$$

Если условие (5) выполняется, то

$$y_{1t} + \beta y_{2t} = \bar{y}_{1t} + \beta \bar{y}_{2t} + \varepsilon_{1t} + \beta \varepsilon_{2t} = \varepsilon_{1t} + \beta \varepsilon_{2t}.$$

Предположим, что в одном из рядов изменяется стохастический тренд:

$$y_{1t} = \begin{cases} \mu_1 + \bar{y}_{1t-1} + v_{1t}^1 + \varepsilon_{1t}, & \text{если } t < t_a \\ \mu_1 + \bar{y}_{1t-1} + v_{1t}^2 + \varepsilon_{1t}, & \text{если } t \geq t_a \end{cases}.$$



Тогда в момент времени t_a

$$\begin{aligned} y_{1t} + \beta y_{2t} &= \bar{y}_{1t} + \beta \bar{y}_{2t} + \varepsilon_{1t} + \beta \varepsilon_{2t} = \\ &= (v_{1t}^2 - v_{1t}^1) + \varepsilon_{1t} + \beta \varepsilon_{2t}, \end{aligned}$$

и для любого момента времени $t \geq t_a$ линейная комбинация (5) имеет вид:

$$\begin{aligned} y_{1t} + \beta y_{2t} &= \bar{y}_{1t-1} + v_{1t}^2 + \beta \bar{y}_{2t-1} + \beta v_{2t} + \varepsilon_{1t} + \beta \varepsilon_{2t} = \\ &= \sum_{j=t_a}^t (v_{1t}^2 - v_{1t}^1) + \varepsilon_{1t} + \beta \varepsilon_{2t}, \end{aligned}$$

т. е. процесс становится интегрированным первого порядка.

- Предположим, что стохастический тренд изменяется в обоих рядах. Тогда в момент времени t_a

$$\begin{aligned} y_{1t} + \beta y_{2t} &= \bar{y}_{1t} + \beta \bar{y}_{2t} + \varepsilon_{1t} + \beta \varepsilon_{2t} = \\ &= (v_{1t}^2 - \beta v_{2t}^2) + \varepsilon_{1t} + \beta \varepsilon_{2t}, \end{aligned}$$

и для любого момента времени $t \geq t_a$ линейная комбинация (5) имеет вид:

$$\begin{aligned} y_{1t} + \beta y_{2t} &= \bar{y}_{1t-1} + v_{1t}^2 + \beta \bar{y}_{2t-1} + \beta v_{2t} + \varepsilon_{1t} + \beta \varepsilon_{2t} = \\ &= \sum_{j=t_a}^t (v_{1t}^2 - \beta v_{2t}^2) + \varepsilon_{1t} + \beta \varepsilon_{2t}. \end{aligned}$$

Если изменения таковы, что связь между ними сохраняется, то существует β' такое, что $v_{1t}^2 - \beta v_{2t}^2 = 0$.

- Предположим, что в одном из рядов изменяется дрейф:

$$y_{1t} = \begin{cases} \mu_1^1 + \bar{y}_{1t-1} + v_{1t} + \varepsilon_{1t}, & \text{если } t < t_a \\ \mu_1^2 + \bar{y}_{1t-1} + v_{1t} + \varepsilon_{1t}, & \text{если } t > t_a \end{cases}.$$

Тогда в момент времени t_a

$$\begin{aligned} y_{1t} + \beta y_{2t} &= \bar{y}_{1t} + \beta \bar{y}_{2t} = \\ &= \bar{y}_{1t-1} + \mu_1^2 + \beta \bar{y}_{2t-1} + \beta \mu_2 = (\mu_1^2 - \mu_1^1), \end{aligned}$$

и для любого момента времени $t \geq t_a$ линейная комбинация имеет вид:

$$\begin{aligned} y_{1t} + \beta y_{2t} &= \bar{y}_{1t} + \beta \bar{y}_{2t} = \\ &= \bar{y}_{1t-1} + v_{1t}^2 + \beta \bar{y}_{2t-1} + \beta v_{2t} = \\ &= (t - t_a + 1)(\mu_1^2 - \mu_1^1), \end{aligned}$$

т. е. линейная комбинация становится нестационарной и ее среднее линейно возрастает со временем.

- Предположим, что дрейф изменяется в обоих рядах. Тогда в момент времени t_a

$$\begin{aligned} y_{1t} + \beta y_{2t} &= \bar{y}_{1t} + \beta \bar{y}_{2t} = \\ &= \bar{y}_{1t-1} + \mu_1^2 + \beta \bar{y}_{2t-1} + \beta \mu_2 = (\mu_1^2 - \beta \mu_2^2), \end{aligned}$$

и для любого момента времени $t \geq t_a$ линейная комбинация имеет вид:

$$\begin{aligned} y_{1t} + \beta y_{2t} &= \bar{y}_{1t} + \beta \bar{y}_{2t} = \\ &= \bar{y}_{1t-1} + v_{1t}^2 + \beta \bar{y}_{2t-1} + \beta v_{2t} = \\ &= (t - t_a + 1)(\mu_1^2 - \beta \mu_2^2), \end{aligned}$$

Если изменения таковы, что связь между ними сохраняется, то существует β' такое, что $v_1^2 + \beta' v \mu_2^2 = 0$.

ПРОВЕРКА КОИНТЕГРАЦИИ

Методика определения коинтеграционных связей достаточно полно разработана и включает в себя следующие шаги:

- определяется порядок интегрированности процессов и их составляющих – (по критериям Дики–Фуллера, Перрона, Парка, Квятковского и др.);
- если процессы – интегрированные одного порядка, то между ними возможна коинтеграционная связь – вычисляется регрессия;
- проверяется порядок интегрированности остатков (по модифицированному критерию Дики–Фуллера).

Изменения в одном или нескольких рядах приводят к:

- изменению дисперсии ряда разностей;
- изменению порядка интегрированности линейной комбинации этих рядов и увеличению ее дисперсии;
- изменению среднего ряда разностей;
- изменению порядка интегрированности линейной комбинации этих рядов и увеличению ее среднего.

Таким образом, в случае изменений отдельного ряда коинтеграционная связь нарушается всегда, в случае изменения двух рядов она может либо сохраняться, либо нарушаться. Сохранение коинтеграционной связи означает, что система приходит в состояние равновесия после произошедших в ней изменений. Нарушение связей может привести к разрушению системы или к серьезным изменениям в ней.

Нарушение коинтеграции вследствие изменения свойств одного ряда может привести к следующим последствиям:



- формированию новой связи вследствие изменений другого ряда;
- восстановлению связи в результате возврата первого ряда в исходное состояние;
- разрушению связей между рядами в долгосрочном периоде.

ПОСЛЕДОВАТЕЛЬНЫЕ АЛГОРИТМЫ ОБНАРУЖЕНИЯ ИЗМЕНЕНИЙ

Для обнаружения изменений свойств процесса по текущим наблюдениям применим алгоритмы последовательного анализа. После получения очередного наблюдения алгоритм анализирует его и принимает решение о наличии либо отсутствии изменения свойств. Алгоритм обнаружения представляет собой последовательную процедуру проверки гипотез — проверяется гипотеза H_0 : плотность распределения процесса равна P_1 против альтернативной гипотезы H_1 : плотность распределения процесса равна P_2 .

Для условий точно известных параметров распределения θ_1 и θ_2 , соответственно до и после изменения свойств процесса $y_1, y_2, \dots, y_t, \dots$ и неизвестного момента изменения t_a , разработан алгоритм кумулятивных сумм, получивший название CUSUM [6]. Основная идея алгоритма заключается в вычислении статистики

$$s_k^t = \sum_{i=k}^t \ln \frac{f_{\theta_2}(y_i)}{f_{\theta_1}(y_i)}, \quad (6)$$

где $f_{\theta_1}(y_i)$ и $f_{\theta_2}(y_i)$ — функции правдоподобия распределения случайной величины y_t до и после изменения свойств. Момент изменения свойств определяется алгоритмом как решение оптимизационной задачи:

$$\tau = \inf \left\{ \tau \geq 1 : g = \max_{1 \leq k \leq \tau} s_k^\tau \geq h \right\}. \quad (7)$$

Решающая функция g допускает рекуррентное представление:

$$g^t = \max \left(0, g^{t-1} + \ln \frac{f_{\theta_2}(y_t)}{f_{\theta_1}(y_t)} \right), \quad g^0 = 0.$$

В работе [5] показано, что в случае точно известных параметров до и после обнаружения алгоритм (7) обладает оптимальными свойствами в смысле критерия минимизации средней задержки обнаружения в «наихудшем случае»:

$$\tau = \sup_{t_a \geq 1} [\text{esssup } E_{\theta_2}(t - t_a + 1 | y_1^{t_a}, t < t_a)]$$

при среднем времени T до ложной тревоги, удовлетворяющем условию $T = E_{\theta_1}(t | t < t_a) > \gamma$, где τ —

момент обнаружения изменений, $E_{\theta_i}(\cdot)$ — математическое ожидание величины (\cdot) при распределении $f_{\theta_i}(y_t)$, $i = 1, 2$, γ — допустимый уровень ложных тревог. Среднее время задержки в обнаружении определяется для всех действительных θ_2 выражением

$$E_{\theta_1}(\tau(\gamma)) \sim \frac{\ln(\gamma^{-1})}{K(\theta_2, \theta_1)}.$$

Величина $K(\theta_2, \theta_1)$ представляет собой информационную меру, называемую информацией Кульбака–Лейблера, которая определяется выражением

$$K(\theta_i, \theta_j) = E_{\theta_i} \left[\ln \frac{f_{\theta_2}(y_t)}{f_{\theta_1}(y_t)} \right], \quad i, j = 1, 2, \quad i \neq j. \quad (8)$$

Предположения о точно известных параметрах после момента обнаружения являются слишком жесткими. Ясно, что в случае, например, экономических индексов можно в лучшем случае высказать только предположение об увеличении, либо уменьшении параметров. В случае неизвестных параметров в [7] было предложено два подхода к решению этой проблемы. Первый заключается в замене отношения правдоподобия (6) на «взвешенное» правдоподобие, второй — в максимизации отношения правдоподобия по всей области допустимых параметров. Соответствующие модифицированные алгоритмы исследованы в работах [8, 9]. Качество обоих алгоритмов практически одинаковое. Первый из них требует вычисления достаточно сложного интеграла, второй — большого объема необходимых вычислений. Рассмотрим алгоритм обнаружения изменений, основная идея которого близка к идее второго алгоритма, но реализация значительно проще.

Изменение среднего. Рассмотрим алгоритм обнаружения изменения среднего для случая, когда наблюдения распределены по нормальному закону с параметрами $\theta_1 = (m_1, \delta_1)$, а после неизвестного момента времени t_a распределение процесса подчиняется нормальному закону с параметрами $\theta = (m > m_2, \delta_1)$, где параметр m — неизвестен, m_2 — значение среднего, превышение которого хотим обнаружить.

Изменение дисперсии. Рассмотрим алгоритм обнаружения изменения дисперсии для случая, когда наблюдения распределены поциальному закону с параметрами $\tilde{\theta}_1 = (m_1, \delta_1)$, а после неизвестного момента времени t_a распределение процесса подчиняется нормальному закону с параметрами $\tilde{\theta}_1 = (m_1, \delta \geq \delta_2)$, где $\delta_1 > \delta_2$, параметр δ неизвестен,



δ_2 – значение дисперсии, превышение которого хотим обнаружить.

Описание алгоритма. Для статистики логарифма отношения правдоподобия последовательности $y_1, y_2, \dots, y_{t_a}, y_{t_a+1}, \dots$, распределенной по нормальному закону распределения с параметрами θ_1 и θ в случае изменения среднего (θ в случае изменения дисперсии) выбираются пороги h и \tilde{h} , обеспечивающие желаемый уровень ложных тревог и задержку в обнаружении. Обозначим $\theta_2 = (m_2, \delta_1)$, $\tilde{\theta}_2 = (m_1, \delta_2)$. Среднее время между ложными тревогами и средняя задержка в обнаружении определяются по формулам [10]:

$$E_{\theta_1}(\tau) = \frac{|e^h - h - 1|}{K(\theta_1, \theta_2)}, \quad E_{\theta_2}(\tau) = \frac{|e^{-h} + h - 1|}{K(\theta_2, \theta_1)},$$

где $K(\theta_1, \theta_2)$ и $K(\theta_2, \theta_1)$ определяются выражением (8). Отсюда при заданных значениях $E_{\theta_1}(\tau)$ и $E_{\theta_2}(\tau)$ определяется порог h .

Далее при получении очередного наблюдения вычисляются статистики:

$$s_t = \ln(L(y_t)) = \ln\left(\frac{f_{\theta_2}(y_t)}{f_{\theta_1}(y_t)}\right) \quad (9)$$

при обнаружении изменения среднего и

$$\tilde{s}_t = \ln(L(y_t)) = \ln\left(\frac{f_{\tilde{\theta}_2}(y_t)}{f_{\theta_1}(y_t)}\right) \quad (10)$$

при обнаружении изменения дисперсии, где L – функция правдоподобия.

Правила остановки:

$$t = \inf \left\{ t \geq 1 : \max_{1 \leq k \leq t} \sum_{i=k}^t s_i \geq h \right\} \quad (11)$$

для обнаружения изменения среднего;

$$t = \inf \left\{ t \geq 1 : \max_{1 \leq k \leq t} \sum_{i=k}^t \tilde{s}_i \geq \tilde{h} \right\} \quad (12)$$

для обнаружения изменения дисперсии.

Свойства предлагаемых алгоритмов определяются следующей теоремой.

Теорема. Пусть $y_1, y_2, \dots, y_t, \dots$ – последовательность независимых величин, распределенных по нормальному закону с параметрами $\theta_1 = (m_1, \delta_1^2)$ до момента t_a включительно и с параметрами $\theta = (m, \delta_1) \times \tilde{\theta} = (\tilde{m}_1, \tilde{\delta})$ после момента t_a , где m и $\tilde{\delta}$ – не-

известные значения параметров, m_2 и $\tilde{\delta}_2$ – их граничные значения, причем $m > m_2$ ($\tilde{\delta} > \tilde{\delta}_2$). Тогда:

1) для алгоритмов (9), (11) и (10), (12) средняя задержка в обнаружении стремится к нулю при разнице между фактическими и граничными значениями параметров $\Delta_m = m - m_2 \rightarrow \infty$ и $\Delta_{\delta} = \tilde{\delta} - \tilde{\delta}_2 \rightarrow \infty$, соответственно;

2) для алгоритмов (9), (11) и (10), (12) средняя задержка в обнаружении стремится к средней задержке для алгоритма (6), (7) с точно известными параметрами после обнаружения при разнице между фактическими и граничными значениями параметров $\Delta_m = m - m_2 \rightarrow 0$ и $\Delta_{\delta} = \tilde{\delta} - \tilde{\delta}_2 \rightarrow 0$, соответственно.

Доказательство заключается в оценивании средней задержки в обнаружении для алгоритмов (9), (11) и (10), (12) и сравнении ее с задержкой для алгоритма (6), (7) при условиях $\Delta_m = m - m_2 \rightarrow \infty$, $\Delta_{\delta} = \tilde{\delta} - \tilde{\delta}_2 \rightarrow \infty$ и $\Delta_m = m - m_2 \rightarrow 0$, $\Delta_{\delta} = \tilde{\delta} - \tilde{\delta}_2 \rightarrow 0$, соответственно.

ПРИМЕНЕНИЕ РАЗРАБОТАННЫХ АЛГОРИТМОВ

Разработанные алгоритмы применялись для анализа экономических индексов России в период с 1995 по 2002 г. Период с 1995 г., когда появились российские биржевые индексы, характеризуется неустойчивостью процессов в российской экономике. За это время на российских финансовых рынках неоднократно возникали кризисные ситуации, кульминацией которых был кризис 1998 г.

Следуя экспертным оценкам, рассматривались следующие кризисные события:

- банковский кризис ликвидности (25 августа 1995 г.);
- социально-экономический кризис доверия правительству накануне президентских выборов (3 июня – 16 июля 1996 г.);
- обвал фондового рынка (24 октября 1997 г.);
- кризис 17 августа 1998 г., явившийся одновременно валютным, банковским, инвестиционным и кризисом внешнего долга.

Исходные данные для анализа:

- российские биржевые индексы AK&M (AKM) и PTC (PTC);
- недельная средневзвешенная доходность ГКО (GKO);
- обменный курс рубля к доллару (USD);
- однодневные объявленные ставки по размещению кредитов (MIBOR1);
- месячные объявленные ставки по размещению кредитов (MIBOR30).

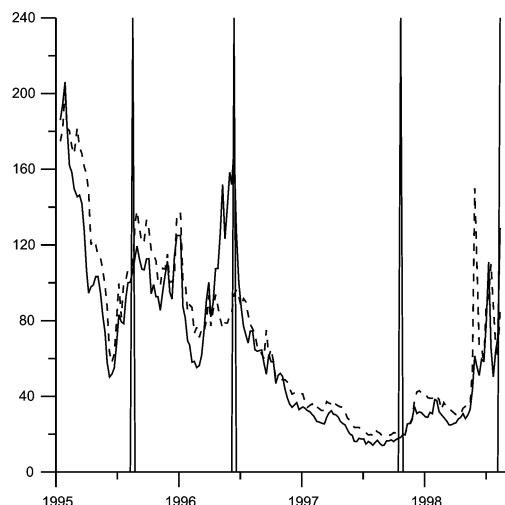


Рис. 1. Доходность *GKO* и месячные процентные ставки:

сплошная линия — недельная средневзвешенная доходность *GKO*; штриховая линия — усредненные по неделям месячные процентные ставки *MIB30W*

Методика анализа:

- проверялось наличие коинтеграционных связей в рассматриваемых рядах;
- определялись моменты изменения этих связей, в выделенных интервалах строились регрессионные модели;
- по текущим наблюдениям вычислялись отклонения реальных рядов от моделей, определялись периоды возникновения значимых отклонений, которые служили сигналами о возможности возникновения кризисов.

Рассматривалось поведение индексов *MIBOR30* и *GKO* (рис. 1). В результате статистического анализа свойств этих рядов было обнаружено, что ряды являются стационарными в первых разностях, а ряды первых разностей изменяют свои свойства после 90-й точки наблюдения (4.10.1996 г.).

Для периода 1.01.1995–4.10.1996 регрессионная модель имеет вид:

$$GKO^{M_1}(t) = 0,945 \cdot MIB30W(t) - 9,073 \quad (13)$$

Для периода 5.10.1996–16.08.1998:

$$GKO^{M_2}(t) = 0,932 \cdot MIB30W(t) - 1,316. \quad (14)$$

Если $DEL(t) = GKO(t) - GKO^{M_1}(t) > 0$, то доходность *GKO* растет быстрее, чем процентные ставки по рублевым кредитам, если $DEL < 0$, то цена на кредиты растет быстрее, чем падают цены на *GKO*. Среднее $\langle DEL(t) \rangle$ для модели (13) составляет 9,44 при стандартном отклонении $\sigma = 24,27$, а для

модели (14) — 2,42 при $\sigma = 12,31$. Статистики кумулятивных сумм остатков моделей (13) и (14) представлены на рис. 2.

Выход отклонения за нижний порог в 1995 и 1998 гг. является признаком кризисов в банковском секторе. Отклонение 1997 г. вызвано понижением доходности *GKO* относительно ставок межбанковских кредитов (из-за привлечения нерезидентов), сопровождавшегося увеличением объемов выпуска *GKO*. Выход за верхний порог (см. рис. 2, *a*) сигнализирует о резком падении цен на *GKO*; этот момент предшествует за семь недель кризису 1996 г.

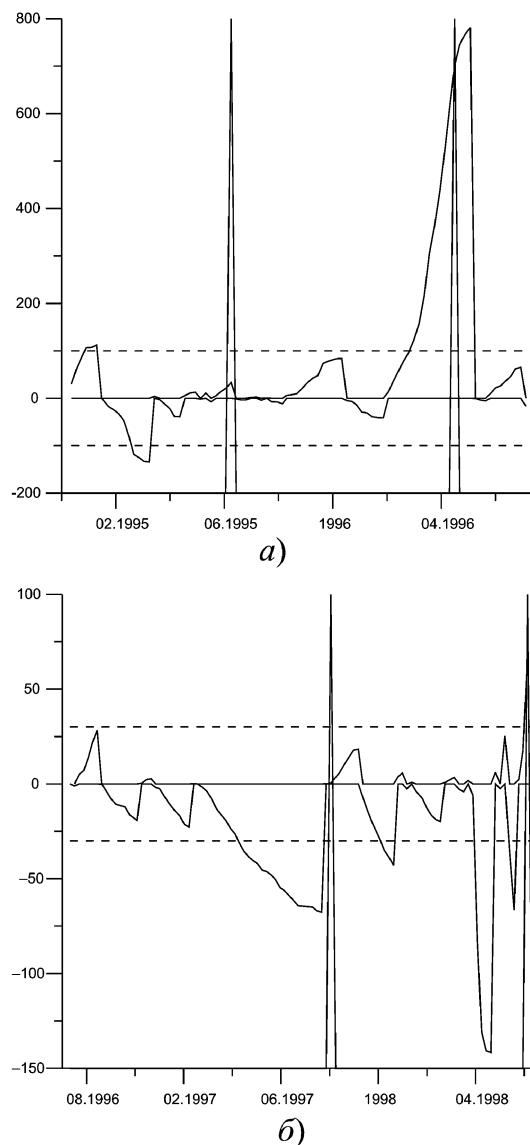


Рис. 2. Решающая функция алгоритма обнаружения отклонений доходности *GKO* от модели:

a — для периода январь 1995–октябрь 1996 гг.; *б* — для периода ноябрь 1996–август 1998 гг.; штриховые линии — пороги алгоритма

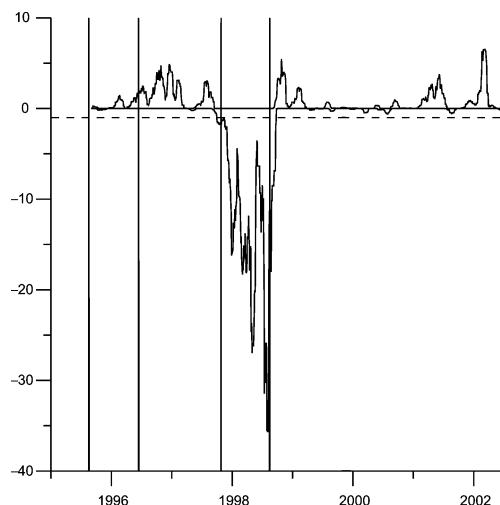


Рис. 3. Решающая функция алгоритма обнаружения отклонений обменного курса доллара от модели (штриховая линия – порог алгоритма)

Рассматривалось совместное поведение индексов *AKM*, *PTC* и обменного курса доллара (*USD*):

$$\begin{aligned} \ln \text{USD}(t) = & -0,971 \cdot \ln \text{PTC}(t) + \\ & + 0,971 \cdot \ln \text{AKM}(t) + 3,551 \\ \text{USD} = & \text{const}(\text{AKM}/\text{RTS})^{0,971}. \end{aligned}$$

Графики решающей функции, обнаруживающей отклонение *USD* от модели, и индекса *RTS* приведены на рис. 3. Рост статистики говорит о том, что курс доллара занижен относительно цены акций в долларах, падение статистики указывает на обратный эффект.

Анализ взаимосвязи процентных ставок по краткосрочным (*MIBOR1*) и среднесрочным кредитам (*MIBOR30*) показывает, что коинтеграционная связь между процентными ставками нарушилась за три недели до начала банковского кризиса 1995 г. и за два месяца до начала августовского кризиса 1998 г.

Полученные результаты показывают эффективность разработанных алгоритмов при анализе нестационарных систем, описываемых временными рядами.

ЛИТЕРАТУРА

1. Greene W. H. Econometric Analysis. — New Jersey: Prentice Hall, 1999.
2. Engle R.F., Granger C.W.J. Cointegration and error correction: representation, estimation and testing // Econometrica. — 1987. — Vol. 55. — P. 251–276.
3. Nyblom J., Harvey A. Tests of Common Stochastic Trends // Econometric Theory. — 2000. — Vol. 16. — P. 176–199.
4. Vahid F., Engle R.F. Common Trends and Common Cycles // Journal of Applied Econometrics. — 1993. — № 8. — P. 341–360.
5. Lorden G. Procedures for reacting to a change in distribution // Annals Math. Statistics. — 1971. — Vol. 42. — P. 1897–1908.
6. Page E.S. Continuous inspection schemes // Biometrika. — 1954. — Vol. 41. — P. 100–115.
7. Wald A. Sequential Analysis. — New York: John Wiley and Sons, 1947.
8. Basseville M., Nikiforov I.V. Detection of abrupt changes – theory and application. — New Jersey: Prentice Hall in Information and System Sciences, 1993.
9. Никифоров И.В. Модификация и исследование процедуры кумулятивных сумм // Автоматика и телемеханика. — 1980. — № 8. — С. 74–80.
10. Siegmund D. Sequential Analysis. Tests and Confidence Intervals. — New York: Springer–Verlag, 1985.

☎ (095) 334-76-40

E-mail: grebenuk@yahoo.com



Подписку на журнал «Проблемы управления» можно оформить с любого месяца в любом почтовом отделении (**подписной индекс 81 708** в каталоге Роспечати или **38006** в объединенном каталоге «Пресса России»), а также через редакцию (из любого места России). Отдельные номера редакция высыпает по первому же требованию. Стоимость одного номера – 440 руб.



ВИРТУАЛЬНЫЕ АНАЛИЗаторы МАРкетинговой ИНФОРМАЦИИ В СИСТЕМЕ УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ¹

Н.Н. Бахтадзе, А.В. Назин

Институт проблем управления им. В.А. Трапезникова, г. Москва

Решена одна из задач коммерческого бюджетирования. Предложены виртуальные анализаторы маркетинговой информации на базе алгоритмов и моделей параметрической идентификации и управления запасами в условиях неполной априорной информации и случайных поставок.

ВВЕДЕНИЕ

Принятие оптимальных маркетинговых решений с целью достижения конкурентного преимущества в условиях современного рынка непосредственно зависит от гибкости и оперативности учета компаниями клиентских потребностей.

Непрерывный рост ассортимента товаров и услуг делает стратегию выстраивания долгосрочных отношений с клиентом и организацию персонализированного сервиса, адаптированного под конкретного клиента, одним из важнейших факторов увеличения доходов. Бизнес-стратегии, применяемые руководством компаний для управления взаимоотношениями с клиентами, получили название Customer Relationship Management (CRM).

Выбор технологического решения для осуществления CRM-процессов определяется коммерческими потребностями фирмы. Сегодня имеется уже достаточно обширный рынок CRM-решений как крупных, так и небольших компаний-разработчиков. При этом они могут быть классифицированы как по широте охвата решаемых системой CRM задач (инструментарий для отделов продаж и маркетинга; системы для осуществления электронной коммерции, интернет-банкинг и т. д.; комплексные решения для малых и средних компаний; системы CRM, интегрированные как отдельные функцио-

нальные модули в крупные ERP-системы² – для больших компаний), так и по функциональности самих CRM. Последний тип классификации предусматривает подразделение всей совокупности CRM-решений на оперативные и аналитические.

Оперативная CRM-система осуществляет сбор маркетинговой информации о клиентах и историю взаимодействия с ними, детализированное структурирование этой информации в соответствии с заданными критериями – для последующего анализа и оперативного доступа по соответствующим запросам, например, для отдела сбыта.

Аналитические CRM-системы, прежде всего, выявляют факторы, влияющие на продажи компании, на основе статистических данных о поведении маркетинговых показателей с учетом взаимного влияния показателей друг на друга; осуществляют сегментацию и анализ сегментов рынка; статистическую обработку маркетинговой информации [1]. Далее, с помощью соответствующих алгоритмов осуществляется сегментарное прогнозирование конкурентоспособности компании, а также прогнозирование объемов продаж, спроса, других ключевых маркетинговых показателей в зависимости от изменений внешних и внутренних маркетинговых факторов, таких как цены, доходы населения, курс рубля и т.д. Такие системы, как правило, позиционируют (как в виде автономных систем [2], так и в качестве модулей ERP-систем)

¹ Рекомендована к печати Программным комитетом Второй международной конференции по проблемам управления (Москва, 17–19 июня 2003 г.).

² ERP (Enterprise Resources Planning)-системы – системы, направленные на усовершенствование таких процессов, как планирование, изготовление, учет и контроль.



как системы поддержки принятия решений в ценообразовании, планировании сбыта, инвестиционной и кредитно-денежной политике компании [3].

Сегодня можно с уверенностью говорить не только как о тенденции, но и как о характерной особенности современных CRM – о доступе к необходимой информации через Интернет [4]. Причем речь идет не только об оперативном интерактивном обмене данными, скажем, с удаленными филиалами, а и о возможности клиента самостоятельно, как минимум, осуществлять заказы товаров (не говоря уже о системах электронных магазинов и т. п.). Вопрос состоит только в ранжировании уровней доступа к информации, содержащейся в единой базе данных системы, и в обеспечении надежной защиты информации от несанкционированного доступа.

Перспективным направлением считается интегрирование CRM-решений в общую систему управления предприятием (возможно – с многофилиальной структурой и Интернетом – в качестве площадки продаж). В этой связи аналитические модули должны быть разработаны как объектно-ориентированные программные комплексы [5, 6].

В то же время, они должны содержать широкий набор инструментальных средств статистической обработки данных и объемную библиотеку алгоритмов для решения комплекса разнообразных задач, формирующих оперативные рекомендации для принятия решения пользователем системы, будь он руководителем предприятия либо клиентом, осуществляющим покупку. Алгоритмы прогнозирования в маркетинге [6] оперируют всей историей маркетинговой информации, в том числе – экспертной, и целевые показатели прогнозируются по всем косвенным параметрам. Программные комплексы, реализующие методы идентификации по косвенным параметрам (их принято называть *виртуальными анализаторами*), в последнее время широко применяются в системах управления технологическими процессами в промышленности, в частности, на предприятиях общей химии, в нефтепереработке и металлургии. Они применяются также в лечебно-диагностических и консультационных комплексах в медицине, в исследовательских системах в биохимии, в астрофизике и других многочисленных областях.

В настоящей работе аналогичный подход применяется к построению аналитической CRM-системы, оптимизирующющей процедуру очередности выплаты долгов поставщикам и прочим кредиторам. Алгоритмы оптимизации, предлагаемые в настоящей работе, представляют собой модификацию алгоритмов, применявшихся для решения других задач управления маркетингом [4]. Такие системы могут оказать существенную пользу компаниям с большим списком наименований ука-

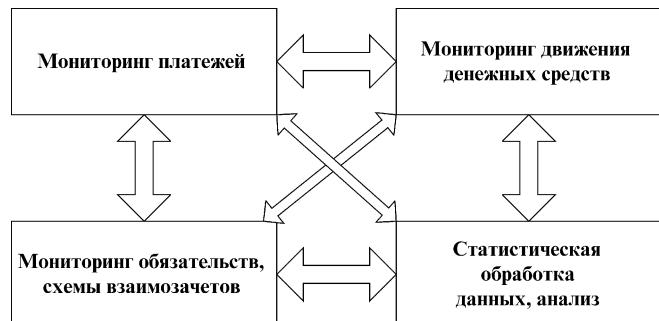
занных обязательных платежей, интерактивно взаимодействуя с различными службами компании в рамках общей системы управления предприятием. Генерируемые в режиме реального времени отчеты позволят находить резервы экономии издержек, что даст возможность компании извлекать дополнительную прибыль.

ВИРТУАЛЬНЫЕ АНАЛИЗАТОРЫ В СИСТЕМАХ УПРАВЛЕНИЯ ЗАДОЛЖЕННОСТЬЮ КОМПАНИИ

Функции системы управления и представление в ней информации. Положение компаний в плане состояния ее текущей задолженности должно быть приемлемым с точки зрения ее инвестиционной привлекательности. Вот почему оно должно быть, во всяком случае, для высшего менеджмента, прозрачным и прогнозируемым. В то же время, авторы работы [7] понимают маркетинг, как «...выражение ориентированного на рынок управленческого стиля мышления, для которого характерны творческие, систематические и, нередко, агрессивные подходы». Это неизбежно может быть отнесено и к политике ранжирования выплат по кредиторской задолженности. Однако предел такой агрессивности не только существует, но и подлежит оценке. Он определяется в соответствии с критерием невозрастания общего долга компании.

Рассматриваемая система должна представлять собой систему поддержки принятия решений менеджером. Разработанная схема сбора, анализа и оптимизации данных по долгам с активным «досье» по поставщикам отражена структурой системы, представленной на рисунке. Основные информационно-аналитические блоки, функционируя в интерактивном взаимодействии, обеспечивают выполнение следующих операций:

- расчет долгов;
- графическая интерпретация кредиторских и дебиторских задолженностей с анализом их влияния на доходы и расходы по формируемым



Структура системы управления задолженностью компании



- группам контрактеров и обязательств, по потребностям в кредитах, по периодам;
- прогноз динамики долгов;
 - выбор тактики и стратегии маневрирования структуризованными долгами;
 - моделирование вариантов покрытия долгов;
 - обработка различных видов отчетности.

Наряду с предоставлением руководителю возможности моделировать политику платежей, пользуясь структурированной информацией, в системе предусмотрена также оптимизация стратегии платежей.

Описание задачи оптимизации. Пусть компания осуществляет свою коммерческую деятельность, имея S_t рублей в t -й день на своих банковских счетах, $t = 1, 2, \dots, T$. Эти денежные средства предназначены для оплаты поставщикам продукции в соответствии с двусторонними договорами. В частности, если компания не производит оплату осуществленной поставки в указанный в договоре срок, ее долг растет, что в конечном итоге нарушает нормальное функционирование компании и приносит ей убытки.

При дефиците средств на банковских счетах компания может взять срочный кредит под определенные проценты для текущих платежей поставщикам. Очевидно, что кредиторы также могут рассматриваться как поставщики товара, в данном случае, денег. Кредит должен быть направлен на преодоление временных финансовых трудностей, и его взятие фактически представляет собой перекладывание долга с одного поставщика на другого при определенном увеличении суммы долга.

В том случае, если на банковских счетах остаются какие-то денежные средства, компания может использовать их для погашения взятых кредитов или временно положить их на депозит с целью получения определенных процентов для компенсации части потерь на кредитах и, возможно, для получения дополнительной прибыли. Таким образом, в целях исследования текущего состояния и перспектив платежеспособности компании представляет интерес решение следующей последовательности задач:

- предварительный анализ платежеспособности компании на основе планируемого поступления денежных средств и плана поставок;
- анализ различных стратегий осуществления платежей и определение оптимальной стратегии при сохранении условий имеющихся договоров с поставщиками;
- определение и анализ оптимальных стратегий осуществления платежей при различных вариантах договорных обязательств;
- разработка оптимальных договорных обязательств с поставщиками и определение соответ-

ствующей оптимальной стратегии выплат поставщикам; окончательный анализ платежеспособности компании.

Решение этих задач должно основываться на построении математической модели процесса платежей, разработке алгоритмов оптимизации и создании пакета компьютерных программ и баз данных, содержащих всю необходимую информацию (данные о поставщиках и условиях договоров с ними, планируемые поставки, текущее состояние банковских счетов, планируемые денежные поступления и др.).

Предварительный анализ платежеспособности. Поскольку известны план поставок и последовательность денежных поступлений на счета компании, можно вычислить общую сумму денег, поступивших на счета за весь рассматриваемый интервал времени, а также общую сумму платежей, которые компании необходимо осуществить (с учетом, быть может, начального долга). Сопоставление этих двух сумм характеризует платежеспособность компании и определяет необходимость и размер общего кредита. Заметим, что взятие кредита всегда связано с отсрочкой платежа, причем большего, чем сумма самого кредита. Отсюда видно, что требование невозрастания общего долга компании можно выполнить только при отсутствии суммарного дефицита на рассматриваемом интервале времени. Другими словами, кредиты могут использоваться лишь для преодоления временного дефицита с целью выполнения имеющихся договорных обязательств, но не как средство уменьшения долга компании. Уменьшение же долга или поддержание его на исходном уровне возможно лишь при отсутствии дефицита или за счет каких-либо инвестиций.

Математическая модель и постановка задачи. Пусть

$t = 1, 2, \dots, T$ – текущее время (последовательные номера дней);

T – рассматриваемый период работы компании (число дней);

S_t – общая сумма денежных средств на банковских счетах компании в день номер t , предназначенная для оплаты услуг поставщиков;

A_t – множество поставщиков Π_i в день номер t , с которыми взаимодействует компания, $i = 1, 2, \dots, m_t$, где m_t – количество этих поставщиков;

A_t^- – подмножество тех поставщиков n_i в день номер t , которые осуществили поставку и ожидают оплаты в соответствии с договорными обязательствами компании; очевидно, $A_t^- \subseteq A_t$;

τ_i – максимальное число дней ожидания оплаты по договору поставщику Π_i после осуществления им поставки;



t_i – день осуществления поставки поставщиком
 $\Pi_i \in A_t$;

C_i – стоимость последней поставки, осуществленной поставщиком Π_i ;

K_T – выплаты процентов по кредитам за рассматриваемый период времени T ;

D_T – деньги, полученные компанией в виде процентов по депозитам за рассматриваемый период времени T .

Замечание. Множество поставщиков A_t может изменяться, теряя некоторых и приобретая новых поставщиков. Могут, хотя и редко, изменяться и договорные обязательства компании перед поставщиками, что может приводить к изменению параметров τ_i , t_i и/или C_i , а также порядка оплаты поставок (погашения долгов).

Таким образом, в день номер t долг компании (до его оплаты в этот день) составляет

$$D_t = \sum_{\Pi_i \in A_t^-} C_i.$$

Потери L_T компании за рассматриваемый период времени T в результате осуществления выбранной стратегии платежей составляют разницу между выплатами процентов по кредитам и выручкой по депозитам:

$$L_T = K_T - D_T.$$

Под стратегией платежей будем понимать правило выбора сроков и размеров платежей, а также условие пересмотра договоров с поставщиками и правило определения желаемых параметров этих договоров. Тогда можно считать, что цель компании состоит в обеспечении минимальных потерь L_T за рассматриваемый период времени T при условии невозрастания долга

$$D_{t+1} \leq D_t \text{ при всех } t = 1, 2, \dots, T,$$

причем значения τ_i не должны превышаться ни для одного поставщика Π_i . Стратегию платежей, обеспечивающую достижение этой цели в рамках имеющихся договоров, назовем оптимальной.

Пусть в соответствии с принятой компанией стратегией платежей в день номер t произведена оплата некоторому подмножеству поставщиков (возможно, всем поставщикам) из множества A_t^- ; обозначим подмножество этих поставщиков через A_t^0 ; очевидно, $A_t^- \supseteq A_t^0$. Пусть, кроме того, компания в день номер t получила очередные поставки от ряда поставщиков из множества A_t ; обозначим подмножество этих поставщиков через A_t^+ , при

этом $A_t^+ \subseteq A_t$. Тогда уравнение динамики долга компании можно записать следующим образом:

$$D_{t+1} = D_t - \sum_{\Pi_i \in A_t^0} C_i + \sum_{\Pi_i \in A_t^+} C_i, \quad t = 1, 2, \dots, T.$$

Таким образом, условие невозрастания долга эквивалентно выполнению неравенств

$$\sum_{\Pi_i \in A_t^0} C_i \geq \sum_{\Pi_i \in A_t^+} C_i \text{ при всех } t = 1, 2, \dots, T-1.$$

Возможность компании производить необходимые выплаты описывается неравенствами

$$S_t \geq \sum_{\Pi_i \in A_t^0} C_i \text{ при всех } t = 1, 2, \dots, T.$$

Описание процедуры определения очередности выплаты долгов. Формально под процедурой ранжирования выплат долгов можно понимать определение совокупности переменных, значения которых указывают количество денежных средств, выплачиваемых компанией конкретным поставщикам в произвольно заданный день. Количество таких переменных примерно Tm , где m – среднее число поставщиков. Кроме того, следует добавить переменные, указывающие размер необходимого кредита для каждого дня, а также неопределенные параметры договоров (моменты времени τ_i). Очевидно, подобная формализация понятия «стратегия платежей» может привести к задаче оптимизации переборного типа с большим числом переменных и, как следствие, к невозможности ее практического решения. Однако при анализе конкретной стратегии платежей такая формализация не приводит к каким-либо трудностям и представляется естественной.

Для преодоления проблемы размерности в задаче оптимизации можно задать разумную структуру стратегии платежей, оставив неопределенными лишь некоторые «наиболее важные» переменные. Например, можно разбить множество поставщиков $A = \bigcup_t A_t$ на подмножества различной «важности» и планировать оплату поставок, исходя из имеющихся в каждый день средств в порядке убывания «важности», что определяется экспертами (с учетом возможности пересмотра договоров). При дефиците средств определяется размер необходимых кредитов и список тех поставщиков, с которыми следует пересмотреть договор. В этом случае значения неопределенных параметров и переменных вычисляются последовательно с ростом t . Так, при каждом $t = 1, \dots, T$ определяется сумма необходимых платежей поставщикам, осуществляемых компанией в этот день, и при дефиците средств



вычисляется сумма необходимого кредита на этот же день. Если кредитные средства оказались недостаточны в течение некоторого периода времени, то определяется дата взятия кредита и его размер. В противном случае, когда в течение какого-то периода времени компании для осуществления платежей хватает собственных средств, определяются размеры возможных депозитов и соответствующие даты и сроки. В результате получаем конкретную стратегию платежей, соответствующую заданной последовательности поступления денег на счета компании и запланированным поставкам.

Оптимизация процедуры выплаты долгов. Предположим, что ежедневная сумма к выплате S_t (где t – номер дня выплаты) достаточна или даже превышает уровень, необходимый для того, чтобы выплаты всем поставщикам осуществлялись без задержки.

Рассмотрим упрощенную ситуацию, когда выплаты поставщикам могут производиться по предлагаемой ниже схеме в предположении, что появление новых срочных заявок на внеочередную выплату не происходит. Предположим также, что в течение определенного промежутка времени состав поставщиков (а следовательно, и структура долгов) не изменяется.

В этом случае схема выплат может быть следующей. Все долги группируются в классы приоритетности. Отнесение поставщиков к тому или иному классу осуществляется экспертами. Вычисляется сумма необходимых выплат S_t .

Пусть r_i – момент окончания действия договорного соглашения с i -м поставщиком. Примем, что раньше этого момента долг ему не выплачивается. Пусть далее d_i – крайний срок выплаты, регламентированный контрактом. Выплаты поставщикам, отнесенными к нулевому классу приоритетности, осуществляются в порядке убывания значения $t - r_i$.

Оставшаяся после обязательных выплат S_t^{ob} сумма $S_t - S_t^{ob}$ распределяется в последующих по приоритетности классах в порядке возрастания значения $d_i - t$. В силу нашего предположения о достаточноном размере суммы S_t все платежи произойдут в срок, т. е. не позднее d_i для всех i .

Заметим, что, располагая статистикой по выплатам за предыдущий период, можно с помощью соответствующей процедуры решить обратную задачу: определить сроки d_i (для означенной упрощенной ситуации). В дальнейшем договорные соглашения оформляются с фиксацией этих сроков, возможно, с некоторым запасом. Размер этого временного резерва определяется с помощью приводимого ниже алгоритма.

Планирование и корректировка бюджета выплаты долгов в реальном времени. В реальной ситуации размер и структура долгов на каждый день

всегда носят случайный характер. Кредиторы, случайно появляющиеся каждый день (например, в виде счетов иностранных компаний с угрозой значительных штрафных санкций в случае просрочки выплаты), кардинально нарушают очередность выплат. Поэтому необходимо планировать, или, по крайней мере, определять заявку на общую сумму для выплаты долгов на следующий день за счет дополнительного кредита. Критерием формирования такой заявки служит условие минимизации долга и процентов по обслуживанию кредита в условиях случайных поставок.

Предлагается следующий алгоритм оптимизации заявки на сумму к выплате долгов на следующий день. Пусть ежедневно компания выделяет на выплату долгов сумму S_t . Введем обозначения:

$x_t = S_t - \text{«запас» (денег) на начало } t\text{-го дня}$ (представляет собой случайную последовательность, поскольку определяется ежедневным состоянием счетов компании);

$$z_t = \sum_{i=1}^m C_{ti} - \text{«спрос» (на деньги) – сумма долга}$$

всем m кредиторам к концу t -го дня плюс выплаченные за текущий день долги (также является случайной последовательностью, поскольку регламентируется поведением поставщиков);

u_t – объем поставки (возможно, за счет очередного кредита), поступающей в конце периода;

\hat{u}_t – «заявка на пополнение запаса» на следующий день. Будем формировать ее дифференцированно по отдельным кредиторам, т. е. будем полагать, что $\hat{u}_t = \sum_{i=1}^m \hat{u}_{ti}$.

Сумма, которая может быть выделена компанией для выплаты долгов на следующий день, заранее не известна, однако она с вероятностью 1 не будет превышать заявки.

Логично интерпретировать этот факт как удовлетворение «подзаявки» \hat{u}_{ti} с вероятностью q_{ti} полностью либо невыполнение ее с вероятностью $1 - q_{ti}$. Вероятности q_{ti} не известны, но очевидно, что они регламентируются приоритетностью выплаты долга i -му поставщику.

Процедура оптимизации состоит в формировании плана заявок \hat{u}_{ti} , минимизирующего невыплаченный долг и проценты по кредитам. Проценты по кредитам на каждый день можно интерпретировать как «издержки хранения» (денег), пропорциональные с коэффициентом h размеру кредита, а оставшиеся к концу t -го дня не выплаченными долги формируют «издержки дефицита» (денег) в виде выплат «неустойки», пропорциональные размеру не выплаченного долга с коэффициентом c .



Коэффициенты h_i определяются размером процентных ставок по кредитам, а коэффициенты c_i – ставками неустойки при просроченном долге (соответственно, для каждого i -го поставщика).

Таким образом, исходную задачу можно интерпретировать как задачу теории управления запасами при случайных поставках в условиях неполной априорной информации.

Динамика изменения запаса имеет вид [8]:

$$x_{t+1} = u_t + (x_t - z_t)^+.$$

Известно [8], что оптимальная по критерию минимума издержек за период (в нашем случае он равен одному календарному дню)

$$L(\hat{u}_t) = cE(z_t - x_t)^+ + hE(x_t - z_t)^+$$

стратегия выплаты долгов определяется следующим образом:

$$\hat{u}_{ti} = \begin{cases} R_{ti} - x_{ti} & \text{при } x_{ti} \leq R_{ti}, \\ 0 & \text{при } x_{ti} > R_{ti}, \end{cases} \quad (1)$$

где E – символ математического ожидания, $x^+ = \max(0, x)$, x_{ti} – доля суммы x_t , которая идет на оплату долга i -му поставщику, а величина R_{ti} определяется посредством рекуррентной процедуры:

$$\begin{aligned} R_{t+1, i} &= R_{ti} \left[1 - \gamma_{ti} \operatorname{sign}(x_{ti} - C_{ti}) - \frac{c_i}{c_i + h_i} \right]; \\ r_{t+1, i} &= r_{ti} - \gamma'_{ti} \left(\frac{c_i}{c_i + h_i} r_{ti} \operatorname{sign} u_{ti} - \right. \\ &\quad \left. - r_{ti} \operatorname{sign}(x_{ti} - C_{ti}) \operatorname{sign} u_{ti} - \theta(z_t) \operatorname{sign} u_{ti} \right) \quad (2) \\ \theta(z_t) &= \begin{cases} 0, & \text{при } z_t > R_t, y > 0; \\ z_t, & \text{при } r_t \leq z_t \leq R_t. \end{cases} \end{aligned}$$

Будем считать, что изначально экспертами определяются основные классы приоритетности, и далее оптимизационные процедуры (1) и (2) строятся для платежей одного класса. Внутри каждого класса определяются очередности выплат всем поставщикам (кредиторам) в соответствии со сроками, оговоренными в контракте. Обозначим эти приоритеты P_i и определим их следующим образом.

$$P_i = m \left(1 - \frac{d_i - t}{\sum_{i=1}^m (d_i - t)} \right)$$

Естественно, значение каждого из них не превышает m и тем больше, чем меньше срок $d_i - t$ до

крайнего дня d_i – дня выплаты i -му кредитору. Тогда $x_{ti} = x_t P_i / m$.

На конечном этапе работы алгоритма анализируется качество предложенной модели и корректируются договорные сроки.

ЗАКЛЮЧЕНИЕ

Рассмотренные методы решения оптимизационных задач в рамках регламента текущего коммерческого бюджетирования демонстрируют широту сферы применения адаптивных алгоритмов управления запасами. Конфигурируемые на базе этих алгоритмов и моделей программно-алгоритмические комплексы представляют собой пример так называемых *виртуальных анализаторов*. Они характеризуются возможностью настройки моделей в режиме реального времени на основе как текущей информации об объекте, так и ретроспективных данных (базы знаний). Ставшие традиционными сферы применения виртуальных анализаторов, для которых существует представительное множество разработок – это оперативные системы управления промышленным производством, маркетинг, социологические исследования, медицина, фармакология, военные технологии и др. Настоящая работа представляет еще один класс объектов, оперативное управление которыми осуществляется с помощью виртуальных анализаторов на базе современных методов теории управления.

ЛИТЕРАТУРА

1. Колбасова А.Б. Обзор методов изучения спроса на новые модификации товара // Экономика и мат. методы. – 1993. – Т. 29. – Вып. 1.
2. Бахтадзе Н.Н., Лотоцкий В.А., Максимов Е.М., Никулина И.В. Экспертно-статистические системы управления маркетингом // Приборы и системы управления. – 1996. – № 3.
3. Брейли Р., Майерс С. Принципы корпоративных финансов. – М.: ЗАО «Олимп-Бизнес», 1997.
4. Бахтадзе Н.Н., Лотоцкий В.А., Максимов Е.М. Информационно-статистические методы в задачах маркетинга // В сб.: «Управление большими системами». Междунар. научно-практич. конф. М., 1997. – С. 209.
5. Arthur B. Smith. Relational, Tree/Plex and Object Oriented Databases // M Computing. – 1996. – May/June – Vol. 4, N 2. – P. 8.
6. Stonebraker M. Future Trends in Database Systems // IEEE Trans. Knowledge and Data Eng. – 1989. – Vol. 1, N 1. – P. 33–44.
7. Дихтель Е., Хершген Х. Практический маркетинг. – М.: ИНФРА-М, Высшая школа, 1996.
8. Лотоцкий В.А., Мандель А.С. Модели и методы управления запасами. – М.: Наука, 1991.

☎ (095) 334-92-01

E-mail: bahfone@ipu.rssi.ru



МУЛЬТИФРАКТАЛЬНОСТЬ, ДИССИПАЦИЯ И УСТОЙЧИВОСТЬ СРЕДНЕСРОЧНЫХ ТРЕНДОВ НА ФОНДОВОМ РЫНКЕ¹

В.Г. Клепарский, В.А. Ефремов

Институт проблем управления им. В.А. Трапезникова, г. Москва

Исследованы основные закономерности переключательной активности в процессе возврата системы фондового рынка в исходное русло эволюции. Обнаружено, что характерные особенности интервалов возврата заключаются в минимизации фрактальной размерности траектории эволюции системы и в минимизации показателя перемежаемости.

ВВЕДЕНИЕ

Длительное наблюдение за графиками биржевых котировок позволяет выделить на фоне хаотичных краткосрочных (минуты, часы) ценовых изменений достаточно протяженные (до нескольких недель) среднесрочные участки траектории с установившейся усредненной скоростью роста (или спада) биржевого курса. Эволюция динамической системы фондового рынка может быть представлена поэтому в виде последовательности временно (на период реализации соответствующих среднесрочных ожиданий и прогнозов) устойчивых динамических состояний, перемежаемых достаточно короткими участками переключательной активности. Такой подход позволяет пользоваться для прогнозирования изменений биржевого курса основными положениями нелинейной динамики, связанными с понятиями русла эволюции и переключательной перемежаемости (см., например, работу [1]). Исследование особенностей превращения исходной зоны аттракции системы в зону отталкивания (и обратно) в процессе переключения тренда приобретает при этом не только большое прикладное значение, но и фундаментальную значимость, поскольку направлено на решение проблемы устойчивости (воспроизведимости) динамических систем с мультифрактальным руслом аттракции. Особый интерес представляет изучение особенностей изменения фрактальной размерности d_f зоны аттракции и потерь (диссипации) в системе для случаев выхода траектории биржевого курса на старое

русло. Такое изучение позволяет оценить «память» системы и выявить опорные точки среднесрочных трендов траектории биржевого курса.

ИСХОДНЫЕ МОДЕЛЬНЫЕ ПРЕДСТАВЛЕНИЯ

Явление переключательной перемежаемости (on-off intermittency), когда динамическая система в процессе самоподстройки под новые реалии осуществляет переход к новому этапу эволюции, обычно ассоциируется с неустойчивостью положения системы в русле решений — состоянии исходного инвариантного многообразия (invariant manifold) (см., например, работы [1, 2]). Система начинает выбрасывать (blow-out) траектории в направлении, перпендикулярном к исходному руслу эволюции. Для реальных систем, т.е. для систем с диссипацией, реализуемые траектории будут стягиваться в новую зону аттракции, которая локализована на некоторой поверхности русла решений достаточно малой размерности — поверхности инерционного многообразия (inertial manifold) по терминологии нелинейной динамики (см., например, работы [2, 3]). При этом основной физической характеристикой формирующегося русла решений должно быть, согласно основным положениям теории динамических каскадных структур А.Н. Колмогорова (см., например, работы [4, 5]), достижение минимума плотности диссипации.

На асимптотической стадии поведения системы инерционное многообразие (русло решений) минимальной размерности d_N совпадает с аттрактором системы, если сам аттрактор асимптотически устойчив и является многообразием. Определение фрактальной размерности d_f аттрактора динамической системы позволяет, следовательно,

¹ Рекомендована к печати Программным комитетом Второй международной конференции по проблемам управления (Москва, 17–19 июня 2003 г.).



получить оценку снизу минимальной размерности d_N русла решений инерционного многообразия. Увеличение фрактальной размерности d_f свидетельствует об увеличении степени хаоса в системе. Определение d_f и плотности диссипации в процессе переключения можно считать поэтому вполне достаточным для сравнительной оценки происходящих при этом изменений устойчивости русла решений.

Для интересующего нас случая хаотически самоподобного временного движения биржевого курса фрактальная размерность определяется, следя работе [6], значением масштабного показателя (self-affine scaling exponent) α самосходных изменений траектории биржевых котировок: $d_f = 2 - \alpha$.

Значение показателя α , в свою очередь, может быть определено, следя модернизированной теории динамических каскадных структур А.Н. Колмогорова [4, 5], моментами высокого порядка M_n плотности вероятности для временных изменений биржевого курса $\Delta R(\Delta t)$ в соответствии с выражением

$$M_n(\Delta t) \equiv \langle (\Delta R(\Delta t))^n \rangle \sim (\Delta t)^{\xi_n + \gamma_n}. \quad (1)$$

Здесь Δt – интервал наблюдения за графиком котировок, n – порядок (четный) момента, $\xi_n = \alpha n + \gamma_n$ – структурный показатель масштабирования, γ_n – показатель мультифрактальности.

Значение показателя γ_n определялась, следя работам [4, 5], с помощью выражения $\gamma_n = \mu n(3 - n)/18$, где μ – показатель перемежаемости, значение которого обусловлено временной неоднородностью (перемежаемостью) локальной диссипации в динамической структуре.

РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ И ИХ ОБСУЖДЕНИЕ

Для экспериментального выявления основных закономерностей переключательной активности в процессе выхода системой на старое русло эволюции были изучены статистические характеристики стохастической траектории изменения курса акций Сбербанка России в период с августа 2002 г. по июль 2003 г. (рис. 1). В работе использовался поток внутридневной биржевой информации по торгу акциями Сбербанка на ММВБ с разбиением на интервалы 15 мин. Основные вычисления выполнены с помощью разрабатываемой нами оригинальной программы исследования рынков GAZOVIK.

Для каждого дня исследуемых этапов эволюции определялись моменты $M_n(\Delta t) = \langle (\Delta R(\Delta t))^n \rangle$ плотности вероятности распределения изменений биржевого курса $\Delta R(\Delta t)$ при $n = 2, 4, 6$ и при длительности интервалов наблюдения $\Delta t = 15 \dots 120$ мин. При расчете моментов по формуле (1) применялась средняя цена для каждого зарегистрированного временного интервала. Затем строились соответствующие (для $n = 2, 4, 6$) графики (в двой-



Рис. 1. Изменение стоимости акций (в рублях) Сбербанка России с июля 2002 г. по июль 2003 г.

ном логарифмическом масштабе) временных зависимостей $M_n(\Delta t)$, по наклону которых определялись значения структурного показателя ξ_n для заданных порядков n (см., например, работу [7]). Построение графиков $\lg M_n(\lg \Delta t)$ для заданных порядков n позволяло выделить монофрактальные сегменты, подтверждавших наличие в динамической системе фондового рынка краткосрочных динамических подсистем с различным временем эффективного существования. Длительность монофрактальных сегментов графиков $\lg M_n(\lg \Delta t)$ интерпретировалась как длительность $\Delta\theta$ эффективного существования обнаруживаемых краткосрочных динамических структур.

Первый период роста биржевого курса (начало августа – конец декабря 2002 г.) имеет усредненную скорость роста ~23,4 руб./день. При этом в пределах канала роста имели место колебания курса в пределах $4\sigma = 352$ руб. Затем (с конца декабря 2002 г. и до начала мая 2003 г.) имел место следующий участок эволюции рынка с усредненной скоростью роста биржевого курса ~14,6 руб./день и колебаниями внутри канала $4\sigma = 352$ руб. Однако 12 мая 2003 г. в связи с изменением кредитного рейтинга эмитента произошел резкий положительный выброс курса Сбербанка (7350...8000 руб. за акцию в течение двух торговых дней). После этого система трижды (15.05, 28.05 и 16.06) выходила на осевую линию первоначального (август–декабрь 2002 г.) канала. При этом каждый раз в момент выхода на осевую линию канала имело место существенное снижение как фрактальной размерности d_f , так и показателя перемежаемости μ .

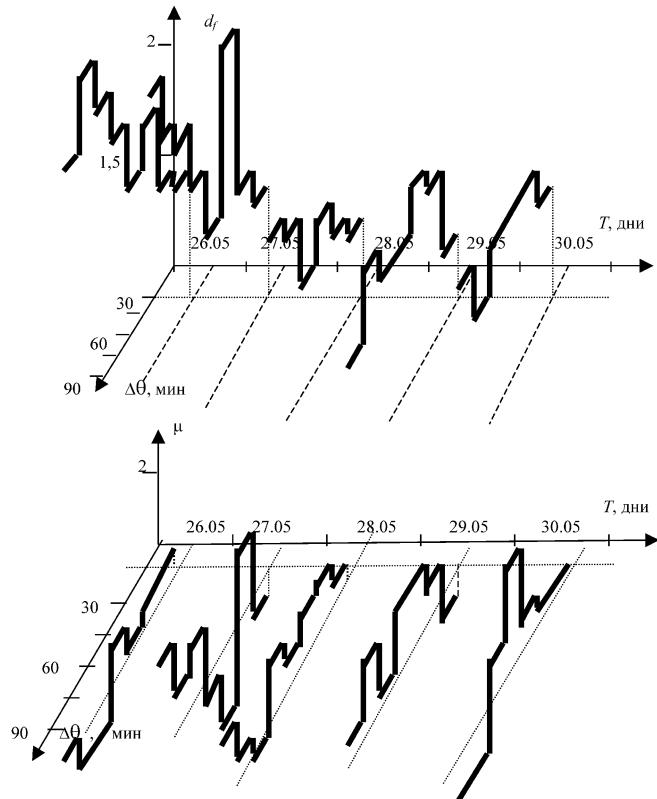


Рис. 2. Деформация мультифрактальной зоны аттракции в процессе выхода траектории на ось исходного среднесрочного тренда эволюции

В качестве примера на рис. 2 представлены результаты экспериментального выявления особенностей формирования канала аттракции биржевого курса в период с 26.05.03 по 30.05.03 (второй выход на ось исходного (август–декабрь 2002 г.) канала роста).

Представленные на рис. 2 графики позволяют заметить, что 26.05 и 27.05 динамическая система находилась в неустойчивом состоянии (рост фрактальной размерности d_f с нарастанием длительности $\Delta\theta$ эффективного существования динамических подструктур, существенный разброс значений показателя перемежаемости μ). 28.05 биржевый курс практически весь день находился на уровне оси исходного канала аттракции. В этот же день было отмечено значительно снижение фрактальной размерности d_f . Одновременно показатель перемежаемости μ снизился до минимальных значений, что позволяет предполагать снижение до минимума плотности диссипации в системе. Следуя работам [4, 5], можно говорить о выходе самоподстраивающейся динамической системы биржевого курса на наиболее узкие уровни русла решений малой размерности (inertial manifold), в режим, достаточно близкий к режиму «инерционного» интервала каскадных динамических структур по мо-

дели А.Н. Колмогорова. Состояние это оказалось, однако, устойчивым лишь в течение одного дня 28.05 (для $\Delta\theta > 105$ мин значение d_f заметно нарастает). Уже в первые часы 29.05 началось падение курса, сопровожданное соответствующим нарастанием значений фрактальной размерности и нарушением однородности диссипации в системе.

Аналогичное состояние «дна» канала аттракции (с минимальными значениями фрактальной размерности d_f и минимальными и однородными значениями показателя перемежаемости μ) было достигнуто 16.06.03, когда траектория биржевого курса вновь вышла на осевую линию исходного (август–декабрь 2002 г.) русла эволюции системы.

Таким образом, как 28.05, так и 16.06 – траектория биржевого курса дважды выходила в опорные точки (с минимальной фрактальной размерностью зоны аттракции и минимальной плотностью диссипации) – на осевую линию исходного (август–декабрь 2002 г.) канала эволюции системы.

Максимум биржевого курса (8700 руб./акция) был достигнут в начале июля 2003 г., после чего начался спад. Как видно из графика биржевых цен (см. рис. 1), спад этот оказался необратимым, сигнализируя о переходе на следующий этап развития системы.

ЗАКЛЮЧЕНИЕ

Обнаружено возвращение траектории биржевого курса акций Сбербанка на исходное русло развития по истечении достаточно длительного периода нахождения системы на другом среднесрочном тренде эволюции. Характерные особенности точек возврата (опорных точек сложной траектории) заключаются в минимизации фрактальной размерности d_f , достижении минимума и однородности для значений показателя перемежаемости μ . Резкое нарастание значений d_f и выбросы показателя перемежаемости μ , свидетельствующие о переходе от диссипации к аккумуляции, соответствуют участкам эволюции с высокой переключательной активностью.

ЛИТЕРАТУРА

1. Малинецкий Г.Г., Потапов А.Б. Современные проблемы нелинейной динамики. – М.: Эдиториал УРСС, 2000.
2. Temam R. Infinite-dimensional dynamical systems in mechanics and physics. – N.-Y.: Springer, 1988.
3. Ott E., Sommerer J.C. Blowout bifurcations: the occurrence of riddled basins and on-off intermittency // Phys. Letters. A. – 1994. – Vol. 188. – P. 39–47.
4. Фриш У. Турублентность. Наследие А.Н. Колмогорова. – М.: Фазис, 1998.
5. Nelkin M. Universality and scaling in fully developed turbulence // Adv. Phys. – 1994. – Vol. 43. No. 2. – P. 143–181.
6. Федор Е. Фракталы. – М.: Мир, 1991.
7. Клепарский В.Г. Мультифрактальность и самоподстройка канала аттракции фондового рынка // Автоматика и телемеханика. – 2001. – № 4. – С. 109–119.

☎ (095) 334-92-50

E-mail: kleparvg@ipu.rssi.ru

УДК 681.3-1.92

САМОДИАГНОСТИРОВАНИЕ ЦИФРОВЫХ СИСТЕМ С РЕКОНФИГУРАЦИЕЙ

В.А. Ведешенков

Институт проблем управления им. В.А. Трапезникова, г. Москва

Предложен метод самодиагностирования компонентов (модулей и линий связи) цифровых систем с реконфигурацией, допускающий устойчивые кратные отказы ограниченного числа компонентов, в основу которого положен принцип расширяющихся областей. Дан пример самодиагностирования отказавших компонентов в 2-отказоустойчивой системе.

ВВЕДЕНИЕ

Многопроцессорные вычислительные системы (МВС) широко применяются в автоматизированных системах управления различными объектами. Одно из обязательных свойств управляющих систем реального времени состоит в их отказоустойчивости, т. е. способности правильно функционировать в течение заданного времени в условиях возникновения отказов и сбоев в работе их компонентов [1, 2]. Для обеспечения необходимой отказоустойчивости МВС часто применяется стратегия динамической избыточности, включающая в себя этапы контроля работоспособности системы, поиска места нахождения отказавшего компонента, реконфигурации системы для его замены исправным компонентом из числа резервных, восстановления функционирования системы. Методы синтеза графа структуры отказоустойчивых систем, предложенные в работе [2], позволяют построить граф с заданной степенью m отказоустойчивости (m -ОУ граф) для целевого (исходного неизбыточного) графа достаточно общего вида. Инвариантно-групповой подход к синтезу приводит к получению структур m -ОУ графов регулярного вида, что упрощает процедуры их анализа в задачах диагностирования и реконфигурации. Но работа по диагностике отказавших компонентов и реконфигурации в отказоустойчивых системах немного [3–5]. В настоящей статье основное внимание удалено разработке метода самодиагностирования (СД) отказавших компонентов

в цифровых системах с реконфигурацией, построенных с помощью алгоритмов синтеза [2].

Методы системного диагностирования отказов в цифровых системах (моделях МВС) основаны на том, что, ограничивая максимальное число одновременно отказавших модулей (процессоров МВС), можно реализовать необходимые диагностические функции с помощью оставшихся исправных модулей. Модели системного диагностирования, максимально абстрагированные от конкретной реализации МВС, позволяют применять общие методы для диагностирования отказов в системах, построенных на основе процессоров и связей различной физической природы. Взаимодействие отдельных частей системы (процессоров, модулей) при реализации диагностических процедур в данной работе описывается с помощью модели Барси – Грандони–Маестрини (сокращенно, модели БГМ) [6]. Она основана на предположении, что каждый модуль, на которые разбивается исходная МВС, может в одиночку проверить любой другой модуль, имеющий с ним непосредственные связи. Для этого первый модуль (проверяющий, в дальнейшем сокращенно МПщ) подает на входы проверяемого модуля (сокращенно МПм) последовательность тестовых воздействий (тест), получает ответные реакции МПм на этот тест, сравнивает их с эталонными, а результаты сравнения оценивает в двоичной форме: 0 – результаты правильные, МПм исправен, 1 – результаты неправильные, МПм неисправен. Предполагается: оценка результатов теста всегда достоверна, если исправен МПщ; она равна 1, если отказали оба модуля; недостоверна

(0 или 1) только в том случае, когда МПщ отказал, а МПм – исправен. Предполагается также, что неисправности модулей – устойчивы, а тесты полные, т. е. обнаруживают всевозможные неисправности модулей.

Подчеркнем, что техническое состояние МПм (компонент) достоверно определяется двоичным результатом его тестовой проверки: 0 – исправен, 1 – неисправен, если исправный МПщ проверял только один модуль (компонент). Если же в данном teste одновременно проверялись несколько компонентов, то по единичному результату нельзя однозначно указать отказавший компонент, нужны дополнительные проверки (тесты). Выбору необходимых проверок и разработке методов обработки их результатов посвящены методы системного диагностирования.

В пользу модели БГМ говорит повышенная сложность модулей (процессоров, машин, устройств связи) современных МВС. Поэтому ничтожно мала вероятность того, что при тестировании неисправного МПм также неисправный МПщ выдаст оценку 0: МПм исправен. Другими словами, нулевой результат, выдаваемый МПщ после проверки МПм, с большой вероятностью подтверждает исправность МПм. Такая предпосылка существенно упрощает процедуру диагностирования по сравнению с моделью ПМЧ [7], так как для идентификации исправных модулей достаточно 0-путей единичной длины (в терминологии работы [8]). Правомочность модели БГМ подтверждают и результаты двухлетних экспериментов с алгоритмом распределенного диагностирования Robust [9], показавших, что отказавший модуль очень редко выдает какие-либо данные. Значительно чаще он прекращает выполнение диагностического алгоритма, не выдает никакой информации и потому идентифицируется исправным МПщ как отказавший.

Суммарный результат проверки модулей системы образует двоичный вектор R , который называется реальным синдромом [6, 7]. Поскольку далее тестовая информация будет представляться в табличном виде, то будем говорить об обработке тестовой информации, а не о дешифрации реального синдрома. Задача обработки тестовой информации заключается в том, чтобы по ней и известной структуре тестовых связей цифровой системы определить подмножество отказавших компонентов (модулей, связей). В большинстве алгоритмов системного диагностирования предполагается (или констатируется), что результаты тестов собираются и обрабатываются внешним устройством.

Другой подход основан на СД состояния компонентов системы путем реализации необходимых алгоритмов, включая сбор и обработку результатов тестирования, на исправных процессорах самой системы (см. например, работы [9–12]). Общие

черты большинства подобных алгоритмов, получивших также название распределенных, заключаются в использовании модели ПМЧ [7], наличии полносвязного диагностического графа, исправности линий связи. Последняя предпосылка частично объясняется тем, что в системах, где реализованы распределенные алгоритмы [11, 12], отказы магистральных линий связи между модулями плохо диагностируются логическими методами.

Ниже предлагается метод СД цифровых систем с реконфигурацией, в основу которого положен принцип расширяющихся областей. Выполнение необходимых процедур начинается с исправного модуля и последовательно передается другим модулям, исправность которых подтверждена предшествующими проверками. Начальный модуль процесса СД определяется по результатам работы системы контроля [13].

Реализация того или иного способа диагностирования зависит от структуры тестовых связей системы, числа тестов, проверяющих каждый модуль, а также количества одновременно отказавших компонентов, топологии их расположения. Максимальное число одновременно отказавших компонентов, однозначно диагностируемых по результатам тестирования системы, называется степенью диагностируемости t системы. Большинство оценок значения t получено в предположении, что отказывают только модули, а связи исправны. Так, в работе [6] для модели БГМ доказано, что в однократно диагностируемых системах из n модулей, обладающих полносвязным диагностическим графом, для степени диагностируемости t справедлива оценка $t \leq (n - 2)$. Предположение о возможности одновременного отказа модулей и связей существенно затрудняет получение подобных оценок.

1. ХАРАКТЕРИСТИКА ОБЪЕКТА ДИАГНОСТИРОВАНИЯ

Структура неизбыточной МВС представляется графом G_0^n , вершинам которого соответствуют однородные группы ресурсов (процессоры, модули), а дугам – линии связи между ресурсами. Предполагается, что отказывать могут как вершины, так и дуги, причем отказ вершины приводит к отказу всех инцидентных ей (входящих и выходящих) дуг.

Алгоритмы синтеза [2] позволяют построить для n -вершинного целевого графа G_0^n и заданной степени m отказоустойчивости искомый объемлющий граф G_R^{n+m} . Наличие гамильтонова цикла в целевом графе G_0^n является необходимым условием существования отказоустойчивого графа G_R^{n+m} . Отметим некоторые свойства графа G_R^{n+m} , необходимые для последующих построений.



Граф G_R^{n+m} m -ОУ системы содержит $(n + m)$ вершин, из которых n вершин соответствуют рабочим модулям, а m вершин – резервные. Он симметричен относительно $(n + m)$ вращений (сдвигов) вокруг своего центра на угол $dg_1 = 360^\circ/(n + m)$, т. е. при повороте на угол dg_1 каждая вершина вместе с инцидентными ей дугами совмещается с соседней вершиной и инцидентными ей дугами. Такая симметрия обеспечивает замену отказавшей вершины и инцидентных ей дуг на исправную вершину и соответствующие дуги путем поворота на соответствующий угол, кратный dg_1 . Добавим, что график G_R^{n+m} после подобной реконфигурации сохраняет логическую структуру задач, решавшихся на МВС до реконфигурации, так как старые логические имена компонентов, совпадавшие с их физическими номерами, будут заменены новыми, зависящими от выполненного числа сдвигов.

Для примера на рис. 1 показан исходный целевой граф G_0^8 , представленный в циркулянтном виде: внешние дуги образуют гамильтонов цикл между его восемью вершинами, а остальные дуги – внутренние. Двукратное применение алгоритмов синтеза [2] к графу G_0^8 привело к построению 2-ОУ графа G_R^{10} (рис. 2); подробности применения алгоритмов синтеза к графу G_0^8 можно найти в работе [5].

Для исправной m -ОУ системы, структура которой представлена графиком G_R^{n+m} , существует C_{n+m}^m конфигураций (структур), включающих в себя n рабочих и m резервных процессоров (вершин графа). Для упрощения будем считать, что исходная рабочая конфигурация (РК) содержит n смежных рабочих вершин, номера которых совпадают с номерами целевого графа, а добавленные (резервные) вершины также являются смежными. Последовательно сдвигая $(n + m)$ вершин на одну позицию, например, против часовой стрелки, получим остальные $(n + m - 1)$ РК.

Для последующих действий представим информацию о графике G_R^{n+m} в виде таблицы, которую назовем первичной [5]. В ее левом столбце запишем номера $(n + m)$ РК, в следующих $(n + m)$ столбцах – составы этих РК, а затем – номера дуг, соответствующих номерам дуг исходных РК (т. е. номерам дуг целевого графа). Пример первичной таблицы, построенной для графа G_R^{10} , представлен табл. 1. Она содержит 8 столбцов для номеров рабочих вершин, 2 столбца – для номеров резервных вершин и 5 столбцов – для номеров дуг целевого графа (подробнее о составе этих дуг – при обсуждении примера).

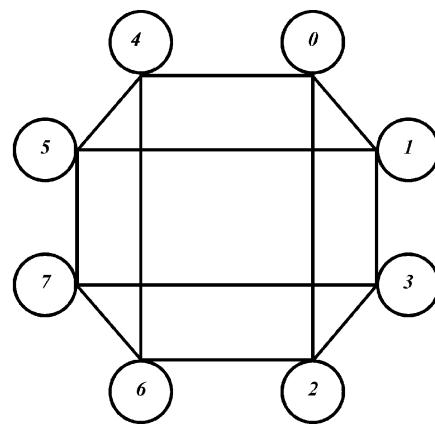


Рис. 1. Граф 3-куба

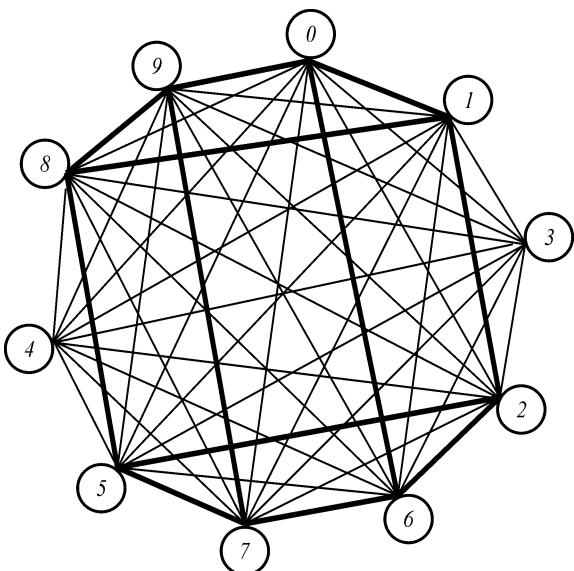


Рис. 2. Диагностический график 2-ОУ системы

По мере функционирования m -ОУ системы и накопления в ней отказавших компонентов для их устранения будут использоваться РК с меньшим числом резервных вершин. Для систематического перебора таких РК нужна совокупность вторичных таблиц первого, второго, ..., m -го уровней, подобных первичной таблице, но содержащих меньше на одну, две, ..., m резервных вершин [5].

2. ПОСТАНОВКА ЗАДАЧИ

Пусть анализируемая m -ОУ система представлена диагностическим графиком G_R^{n+m} , n вершин которого представляют n рабочих модулей (процессоров) системы, а m вершин – резервные и (или) отказавшие модули. По построению, график G_R^{n+m} –

симметричный и однородный, но не обязательно полный.

Будем считать, что допускаются устойчивые кратные отказы ограниченного числа вершин и дуг. Предположим также, что во время выполнения СД новые отказы не возникают, а существующие неисправности таковы, что не приводят к несанкционированному запуску процесса СД в отказавших модулях и, следовательно, не искажают разработанную организацию СД; при реализации диагностических процедур используется модель БГМ [6].

Предполагается, что в исходной МВС есть диагностический монитор (часть центральной операционной системы), который инициирует процессы контроля, самодиагностирования и осуществляет реконфигурацию по результатам диагностирования технического состояния компонентов очередной РК.

Требуется разработать такой метод контроля и СД компонентов отказоустойчивой системы, чтобы после выполнения СД каждый исправный модуль имел информацию, достаточную для идентификации технического состояния остальных компонентов (модулей, линий связи) РК системы.

3. МЕТОД САМОДИАГНОСТИРОВАНИЯ ОТКАЗАВШИХ КОМПОНЕНТОВ

Контроль цифровой системы состоит в проверке каждым исправным модулем технического состояния своих преемников и соответствующих линий связи. Для оптимальной реализации последовательного выполнения различных проверок одного МПм разными МПш на диагностическом графе системы построим гамильтонов путь, который, по определению, заходит в каждую вершину точно один раз. Если начальная и конечная вершины такого пути совпадают, то построенный путь

является гамильтоновым циклом. Для однородного диагностического графа исправной системы такой цикл всегда существует. Отметим, что для излагаемого ниже метода достаточно выбрать любой гамильтонов цикл, который обеспечивает обход всех вершин диагностического графа РК исправной системы. Такой цикл выбирается один раз при проектировании системы контроля.

Информацию о составе выбранной РК №_w m-ОУ системы представим в виде таблицы T_{S_w} . Она состоит из n строк $T_{i_w}, \dots, T_j, \dots, T_{n_w}$, в каждой из которых для модуля m_j ($j = i_w, \dots, n_w$, где $0 \leq i_w, \dots, n_w = n + m - 1$) записаны номера модулей — преемников m_{k1}, m_{k2} , (смежных с модулем m_j), в ней также есть поля для записи результатов проверки модулем m_j этих модулей $r_{j, k1}, r_{j, k2}$, и значений их ранга при ранжировании графа. Пример такой таблицы T_{S_0} , построенной для РК № 0 (из табл. 1), содержащей 8 строк, представлен в табл. 2. Как видно из табл. 2, в каждой ее строке заполнены только клетки для МПш m_j и его преемников МПм — m_{k1}, \dots, m_{k3} .

Для отображения местонахождения модулей в гамильтоновом цикле будем записывать в строке T_j преемников модуля m_j так, что именно модуль m_{k1} является соседом справа модуля m_j в построенном цикле. Для РК № 0 выбран такой гамильтонов цикл: $0 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 6 \rightarrow 7 \rightarrow 5 \rightarrow 4 \rightarrow 0$, именно в таком порядке в табл. 2 записаны строки T_j .

Обычно процесс контроля запускается диагностическим монитором через определенные интервалы времени. Это предотвращает накапливание отказавших компонентов в системе и способствует безошибочности решения целевых задач.

Назначим начальной вершиной процесса контроля одну из вершин выбранного гамильтонова

Первичная таблица

№ п/п	Номера рабочих вершин								Резервные вершины		Номера дуг				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	3	2	6	7	5	4*	8	9	4 – 0	0 – 2	3 – 7	6 – 4*	5 – 1*
1	1	3	2	6	7	5	4*	8	9	0	8 – 1	1 – 6	2 – 5	7 – 8	4 – 3
2	3	2	6	7	5	4*	8	9	0	1	9 – 3	3 – 7	6 – 4*	5 – 9	8 – 2
3	2	6	7	5	4*	8	9	0	1	3	0 – 2	2 – 5	7 – 8	4 – 0	9 – 6
4	6	7	5	4*	8	9	0	1	3	2	1 – 6	6 – 4*	5 – 9	8 – 1	0 – 7
5	7	5	4*	8	9	0	1	3	2	6	3 – 7	7 – 8	4 – 0	9 – 3	1 – 5*
6	5	4*	8	9	0	1	3	2	6	7	2 – 5	5 – 9	8 – 1	0 – 2	3 – 4
7	4*	8	9	0	1	3	2	6	7	5	6 – 4*	4 – 0	9 – 3	1 – 6	2 – 8
8	8	9	0	1	3**	2	6	7	5	4*	7 – 8	8 – 1	0 – 2	3 – 7	6 – 9**
9	9	0	1	3**	2	6	7	5	4*	8	5 – 9	9 – 3	1 – 6	2 – 5	7 – 0



цикла, обозначим ее m_n . Тогда по сигналу диагностического монитора МПш m_n должен выполнить следующие действия:

- из таблицы T_{Sw} выделить строку T_n , найти в ней номера преемников m_{k1}, m_{k2}, \dots ; проверить их техническое состояние и в клетки $r_{n, k1}, r_{n, k2}$, строки T_n записать результаты проверок в двоичном коде: 0 – исправен, 1 – подозревается в отказе;
- оценить результаты проверок:
 - если $r_{n, k1} = 0$, то передать управление процессом контроля модулю m_{k1} ;
 - если $r_{n, k1} = 1$, но один из других результатов (например, $r_{n, k2}$) – нулевой, то передать управление процессом контроля соответствующему модулю (модулю m_{k2});
 - если же проверки всех преемников m_n дали единичные результаты, то процесс контроля нужно начать с другого модуля (например, с модуля m_{n2} , находящегося в «середине» гамiltonова цикла).

Модуль $m_1 (m_{k1}, m_{k2}, \dots)$ и каждый из последующих модулей выполняет аналогичные действия. Этот процесс проверок и «хождения» по выбранному циклу заканчивается, когда управление возвращается к начальному модулю $m_n (m_{n2})$. Такой детерминированный путь обхода вершин графа возможен только при исправном состоянии всех компонентов проверяемой РК. Если же на одном из этапов проверки получен единичный результат, свидетельствующий об обнаружении одного или нескольких отказавших компонентов, то процесс контроля прерывается, и диагностический монитор переходит к процессу СД, который включает в себя такие процедуры:

- исполнение исправным модулем функций МПш;
- нахождение пути от одного исправного модуля к другому, возможно, с помощью процедуры

ранжирования соответствующих исправных модулей;

- обработка и дешифрация результатов выполненных проверок.

Эти процедуры реализуются совокупностью исправных модулей РК № w . Необходимость исполнения всех или части этих процедур определяется очередным МПш на основе результатов выполненных им проверок и информации, полученной от предыдущих модулей.

Для сохранения информации, полученной в процессе контроля, диагностический монитор начинает процесс СД с того исправного модуля $m_j (\neq m_n, m_{n2})$, который первым получил единичный результат. Начальный модуль m_j должен выполнить следующие действия:

1) образовать промежуточную таблицу PT_j и переписать в нее строку T_j с результатами проверок, полученными в процессе контроля (в индексе символа данной таблицы указан номер начального МПш);

2) по нулевым результатам проверок в строке T_j выделить подмножество номеров исправных преемников $\{q_j\}$ и одному из преемников передать таблицу PT_j и функции МПш на следующем шаге СД.

Если среди преемников m_j нет ни одного исправного, то процесс СД не может быть продолжен, и для получения диагностических результатов монитор должен запустить процесс контроля с другого модуля.

Очередной исправный модуль m_u , получив управление процессом СД, должен выполнить следующие действия:

1) из таблицы состава T_{Sw} выделить строку T_u , переписать ее в таблицу PT_j , полученную от предшественника $m_j (m_{u-1})$;

Таблица 2

Таблица состава T_{S0} для РК № 0

Символ строки	Проверяющий модуль и ранг		Преемники m_j , результаты их проверок и ранг								
	m_j	ранг m_j	m_{k1}	$r_{j, k1}$	ранг m_{k1}	m_{k2}	$r_{j, k2}$	ранг m_{k2}	m_{k3}	$r_{j, k3}$	ранг m_{k3}
T_0	0	–	1	–	–	2	–	–	4	–	–
T_1	1	–	3	–	–	5	–	–	0	–	–
T_3	3	–	2	–	–	7	–	–	1	–	–
T_2	2	–	6	–	–	0	–	–	3	–	–
T_6	6	–	7	–	–	4	–	–	2	–	–
T_7	7	–	5	–	–	3	–	–	6	–	–
T_5	5	–	4	–	–	1	–	–	7	–	–
T_4	4	–	0	–	–	6	–	–	5	–	–



2) в строке T_u найти номера своих преемников m_{k1}, m_{k2}, \dots ; проверить их состояние и в клетки $r_{u, k1}, r_{u, k2}$, строки T_u записать результаты проверок;

3) по нулевым результатам проверок в строке T_u выделить подмножество исправных преемников $\{q_u\}$, сравнить номера $\{q_u\}$ с номерами модулей в левом столбце PT_j :

- если один из номеров подмножества $\{q_u\}$ отсутствует в списке номеров левого столбца PT_j , то передать ему (модулю m_{u+1}) таблицу PT_j и функции МПЩ на следующем шаге; при этом модуль m_{u+1} вернется к п. 1 процесса СД;
- если два (или больше) номера из подмножества $\{q_u\}$ отсутствуют в списке номеров левого столбца PT_j , то одному из них (например, m_{k1}) передать таблицу PT_j и функции МПЩ на следующем шаге;
- если все номера $\{q_u\}$ в списке номеров левого столбца PT_j , то перейти к п. 4;

4) найти кратчайший путь от модуля m_u к одному из исправных модулей $m_{u+1}, m_{u+2}, \dots, m_{u+v}$, еще не исполнявших функции МПЩ, с помощью процедуры ранжирования, передать найденному модулю m_{u+1} (m_{u+2}, \dots) таблицу PT_j и функции МПЩ на следующем шаге; если при выполнении процедуры ранжирования выяснится, что все обнаруженные исправные модули исполняли функции МПЩ, перейти к п. 5;

5) передать информацию, собранную в таблице PT_j модуля m_{u+v} (у которого совпали списки исправных и проверяющих модулей), в направлении, обратном тому, по которому эта информация собиралась; получив идентичную информацию в виде таблицы PT_j от модуля m_{u+v} , каждый из модулей

выполнит процедуру ее обработки (приведенную ниже) и сформирует одинаковый диагноз.

Для нахождения кратчайшего пути от модуля m_u к одному из исправных модулей $m_{u+1}, m_{u+2}, \dots, m_{u+v}$, не являющихся его преемниками, каждому исправному модулю РК № w присваивается номер (ранг), равный минимальному числу линий связи, соединяющих данный модуль с модулем m_u , который ищет этот путь. Процедура ранжирования (присвоения номеров) носит «расширяющийся» характер: модуль m_u присваивает ранг 1 своим исправным преемникам, затем ранг 2 – исправным преемникам модулей с рангом 1, и т. д., пока некоторый ранг (например, d) не получит один из исправных модулей, еще не исполнявших функции МПЩ. Тогда поиск нужного пути заключается в выборе линий, связывающих модули с рангами d и $(d-1)$, затем модули с рангами $(d-1)$ и $(d-2)$ и т. д. до линии, связывающей модули с рангами 1 и 0 (m_u). Процедура формального описания действий модуля m_u при ранжировании графа и поиске нужного пути приведена в работе [14].

Для обработки и дешифрации информации, собранной в таблице PT_j модулем m_{u+v} , каждый исправный модуль переписывает ее в диагностическую таблицу T_{Dw} , построенную заранее для РК № w . В строке D_k ($k = i_w, \dots, n_w$) таблицы T_{Dw} для проверяемого модуля m_k записаны номера его предшественников (МПЩ) m_{j1}, m_{j2}, m_{j3} ; есть также клетки для записи результатов $r_{j1, k}, r_{j2, k}, r_{j3, k}$ его проверки этими модулями и клетки для записи диагностических заключений о состоянии модуля m_k и инцидентных ему линий связи, получаемых после обработки результатов $r_{j1, k}, r_{j2, k}, r_{j3, k}$. Пример такой таблицы, построенной для РК № 0 (из табл. 1), содержащей 8 строк, представлен в табл. 3. Нетрудно заметить, что преемники модуля m_j , по-

Диагностическая таблица T_{D0} для РК № 0

Символ строки	Предшественники m_k			Проверяемый модуль m_k	Результаты проверок			Диагностическое заключение			
	m_{j1}	m_{j2}	m_{j3}		$r_{j1, k}$	$r_{j2, k}$	$r_{j3, k}$	s_k	$a_{j1, k}$	$a_{j2, k}$	$a_{j3, k}$
D_0	1	2	4	0	0	0	—	0	0	0	—
D_1	3	5	0	1	0	1	0	0	0	1	0
D_3	2	7	1	3	0	0	0	0	0	0	0
D_2	6	0	3	2	0	0	0	0	0	0	0
D_6	7	4	2	6	0	—	0	0	0	—	0
D_7	5	3	6	7	0	0	0	0	0	0	0
D_5	4	1	7	5	—	1	0	0	—	1	0
D_4	0	6	5	4	1	1	1	1	z	z	z



казанные в строке T_j табл. 2, совпадают с предшественниками модуля m_k , приведенными в строке D_k табл. 3. Это совпадение – не случайность, а следствие использования в системе двунаправленных связей, поэтому модули m_{k1} , m_{k2} , m_{k3} – преемники модуля m_j – являются также его предшественниками m_{j1} , m_{j2} , m_{j3} .

Процедура обработки исправным модулем информации из таблицы PT_j модуля m_{u+v} заключается в следующем:

1) из строк $T_j, \dots, T_u, \dots, T_{u+v}$ таблицы PT_j выделяются результаты проверок $r_{j, k1}, r_{j, k2}, \dots, r_{u, k1}, r_{u, k2}, \dots, r_{u+v, k1}, r_{u+v, k2}$ и записываются в клетки строк диагностической таблицы T_{Dw} ;

2) анализируются результаты проверок, содержащиеся в строке D_k , и в клетки $s_k, a_{j1, k}, a_{j2, k}, a_{j3, k}$ этой строки записывается диагностическое заключение, полученное согласно правилам:

1) если в клетках результатов проверок строки D_k есть хотя бы один нуль (пусть $r_{j1, k} = 0$), то в клетки на пересечении строки D_k со столбцами s_k и $a_{j1, k}$ записываются нули;

a) если в клетках результатов этой же строки D_k есть еще нули, то записываются нули в клетки на пересечении строки D_k со столбцами переменных соответствующих линий;

b) если в клетках результатов этой же строки D_k есть единицы, то записываются единицы в клетки на пересечении строки D_k со столбцами переменных соответствующих линий;

2) если в клетках результатов проверок строки D_k записаны только единицы, то записывается 1 в клетку на пересечении строки D_k со столбцом s_k , а в клетки на пересечении строки D_k со столбцами переменных соответствующих линий записываются буквы z .

Таким образом, дешифрация результатов проверок из таблицы PT_j модуля m_{u+v} сводится к замене 0, 1 и z , записанных в клетках $s_k, a_{j1, k}, a_{j2, k}, a_{j3, k}$ строки D_k таблицы T_{Dw} , состояниями модуля m_k и инцидентных линий связи согласно правилам:

- 0 → «модуль исправен», «линия исправна»;
- 1 → «модуль подозревается в отказе», «линия подозревается в отказе»;
- z → неопределенное состояние линии связи, так как отказавший модуль «маскирует» состояние инцидентной линии связи.

4. ПРОЦЕДУРА РЕКОНФИГУРАЦИИ

Дешифрация результатов проверок из таблицы PT_j модуля m_{u+v} определяет для РК № w подмножество компонентов, подозреваемых в отказе. Процедура реконфигурации m -ОУ графа сводится к нахождению в первичной таблице (или в одной из вторичных таблиц) такой строки, в которой компоненты, подозреваемые в отказе, не входят в состав РК № w^* – т.е. отказавшая вершина попадет на место одной из резервных, отказавшая дуга не будет соединять пару смежных рабочих вершин и не будет входить в состав дуг, перечисленных в правой части этой строки. Начало поиска нужной работоспособной РК зависит от того, к какой из таблиц – первичной или вторичной – принадлежит анализируемая РК № w .

После нахождения новой работоспособной РК № w^* диагностический монитор модифицирует таблицу состава T_{Sw} и диагностическую таблицу T_{Dw} , заменяя номера рабочих модулей прежней РК № w на соответствующие номера новой РК № w^* . Затем монитор запускает процесс контроля РК № w^* . Если контроль РК № w^* подтвердит ее работоспособность, то m -ОУ система возвращается к выполнению своих функциональных задач. В противном случае монитор вновь запускает процесс СД для идентификации тех компонентов, которые не участвовали в РК № w , а потому их отказы не влияли на ее работоспособность.

После идентификации нового подмножества подозреваемых компонентов процедура реконфигурации повторяется, но новая РК выбирается так, чтобы в ней отсутствовали не только впервые обнаруженные компоненты, подозреваемые в отказе, но и все отказавшие компоненты, ранее обнаруженные и выведенные из состава РК. Таким образом, может оказаться, что после отказа некоторого компонента в ПТ не останется ни одной работоспособной РК. В этом случае нужно перейти к РК, включающим в себя на 1, 2, ..., m вершин меньше. По определению, в каждую из таких РК с меньшим числом резервных вершин можно вложить целевой граф G_0^n . Но нужно проверить, что ранее отказавшие компоненты (вершины, дуги) не мешают работоспособности выбранной РК.

Несколько слов о стратегии перебора первичной и вторичных таблиц. Очевидно, что идентификация процессом СД отказавшей вершины вынуждает переходить к РК, не содержащим эту вершину; для этого может потребоваться переход к поиску во вторичных таблицах. Что же касается устранения влияния отказавшей дуги, то здесь целесообразно просмотреть все оставшиеся строки таблицы, РК которой функционировала в момент

обнаружения отказа. И если среди них нет ни одной работоспособной (вследствие влияния предыдущих отказов), то следует переходить к вторичной таблице с уменьшенным на единицу числом резервных вершин. При этом желательно исключить из дальнейшего рассмотрения ту вершину, которой инцидентно наибольшее число отказавших дуг.

Возможность возникновения неразличимых кратных отказов модулей и их входных линий связи [14] не является серьезным ограничением для восстановления работоспособности систем с реконфигурацией. Действительно, смена РК выводит из числа рабочих не только подозреваемый модуль, но и все его входные и выходные линии связи. Поэтому для таких систем диагностирование отказавших компонентов с точностью до модуля или его входной линии связи представляется не нужным.

5. ПРИМЕР

Особенности процесса СД и реконфигурации покажем на примере 2-ОУ системы, построенной для целевой системы со структурой в виде трехмерного куба. Гиперкубовые структуры МВС обладают рядом достоинств, среди которых отмечают простоту алгоритмов маршрутизации при межпроцессорных обменах информацией, регулярность алгоритмов реконфигурации систем при отказах их компонентов, хорошо formalизованные модели их представления [15, 16]. В примере рассматривается 3-куб, в вершинах которого расположены процессорные элементы (на рис. 1 они обозначены кружками, в которых указаны их номера).

Считая граф 3-куба (см. рис. 1) в качестве целевого графа G_0^8 , путем двукратного применения алгоритмов синтеза [2] был построен 2-отказоустойчивый граф G_R^{10} (см. рис. 2). Первичная таблица РК графа G_R^{10} (см. табл. 1) состоит из 10-ти строк, соответствующих 10-ти РК и обозначенных номерами 0, 1, ..., 9, клетки столбцов 1, 2, ..., 8 в каждой строке содержат номера рабочих вершин данной РК, клетки столбцов 9 и 10 – номера резервных вершин. В столбцах 12–15 приведены номера дуг, соответствующие номерам четырех дуг целевого графа G_0^8 . Начальная РК № 0 показана в строке 0: рабочие 0, 1, ..., 5, 4 и резервные 8, 9 вершины, рабочие дуги (0–2), (3–7), (6–4) и (5–1). В столбце 11 показана дуга (4–0): она замыкает гамильтонов цикл, в который входят рабочие вершины 0, 1, ..., 5, 4. Дуги между смежными вершинами этого цикла в табл. 1 отдельно не показаны, так как их номера однозначно определя-

ются номерами соседних столбцов: (0–1), (1–3) и т. д. Номера вершин РК, показанные в строках 1–9, получены циклическим сдвигом справа налево номеров вершин строки 0. Номера дуг, показанные в клетках столбцов 11–15 в каждой строке, получены путем замены номеров рабочих вершин строки 0 номерами вершин, записанными в тех же столбцах очередной РК. Поэтому в столбце 12 дуга (0–2) из строки 0 последовательно заменяется дугой, соединяющей вершины с номерами (1–6), (3–7) и т. д., стоящими в соответствующих столбцах ниже вершин 0 и 2. Таким же образом получены номера других рабочих дуг.

В статье [15] предложен алгоритм A2 оптимального обхода и тестирования вершинных процессорных элементов, состоящий из $2n$ раундов параллельных проверок смежных вершин; n -размерность куба. Таким образом, все 24 теста, необходимые для контроля состояния 3-куба, будут выполнены за 6 раундов, т.е. в 4 раза быстрее последовательного выполнения этих тестов. Но в реализации этих тестов участвуют все модули n -куба, как исправные, так и отказавшие. Поэтому таблица проверок n -куба с отказавшими компонентами будет содержать недостоверные результаты, что усложнит процедуру дешифрации.

Подчеркнем, что алгоритм A2 разработан для проверки работоспособности вершин и дуг неизбыточного n -куба. Поэтому его применение для проверки 2-ОУ системы, построенной на основе 3-куба, возможно только для отдельных РК, содержащих 8 вершин и 12 дуг. Нетрудно видеть, что 2-ОУ граф G_R^{10} (см. рис. 2) содержит большое число таких РК. Поэтому вместо перебора всевозможных РК будем использовать описанный процесс последовательного контроля компонентов РК № 0 и тех РК, в которые переходит система после обнаружения и устранения очередного подмножества отказавших компонентов.

Итак, пусть 2-ОУ система работает в начальной РК № 0 (см. табл. 1). Предположим, что из-за большого интервала между циклами контроля в 2-ОУ системе накопились отказы модуля 4 и линий связи (5–1) и (6–4) (эти компоненты отмечены знаком * в клетках тех РК табл. 1, в состав которых они входят). Контроль работоспособности 2-ОУ системы начинается с модуля, соответствующего начальной вершине гамильтонова цикла для выбранной РК. Так, для РК № 0 выберем такой цикл: $0 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 6 \rightarrow 7 \rightarrow 5 \rightarrow 4 \rightarrow 0$.

По сигналу диагностического монитора начальный модуль процесса контроля m_0 (вершина 0) выполнит следующие действия:

- из таблицы T_{S0} состава РК № 0 (см. табл. 2) выделит строку T_0 , найдет в ней номера преемни-



ков m_{k1} (1), m_{k2} (2), m_{k3} (4), проверит их техническое состояние и в клетки $r_{j, k1}, r_{j, k2}, r_{j, k3}$ строки T_0 запишет результаты проверок: $r_{0,1} = 0$, $r_{0,2} = 0$, $r_{0,4} = 1$ которые показаны в строке T_0 табл. 4;

— оценит результаты проверок: хотя $r_{0,4} = 1$, но модуль 0 начальный, поэтому модуль 0 передаст управление процессом контроля соседнему модулю 1.

Модуль 1 (m_{k1}), получив управление от модуля 0, выполнит следующие действия:

— из таблицы T_{S0} выделит строку T_1 , найдет в ней номера преемников 3, 5, 0, проверит их и в соответствующие клетки строки T_1 запишет результаты проверок: $r_{1,3} = 0$, $r_{1,5} = 1$, $r_{1,0} = 0$ (для сокращения числа таблиц будем записывать результаты

проверок, находящиеся в разных модулях, в общую табл. 4);

— оценив результаты проверок и обнаружив $r_{1,5} = 1$, достоверно исправный модуль 1 передаст к процессу СД.

Для сохранения информации, полученной в процессе контроля, монитор начнет процесс СД с модуля 1, который выполнит следующие действия:

1) образует промежуточную таблицу PT_1 (см. табл. 4), в которую перепишет строку T_1 с результатами проверок, полученными в процессе контроля (в индексе символа данной таблицы указан номер 1 начального МПЩ);

2) по нулевым результатам проверок в строке T_1 выделит номера 3 и 0 исправных преемников $\{q_i\}$ и согласно следующему пункту процедуры пере-

Таблица 4

Информация к Примеру

Символ строки	Проверяющий модуль		Преемники m_j , результаты их проверок и ранг								
	m_j	ранг m_j	m_{k1}	$r_{j, k1}$	ранг m_{k1}	m_{k2}	$r_{j, k2}$	ранг m_{k2}	m_{k3}	$r_{j, k3}$	ранг m_{k3}
Строки с результатами контроля РК № 0											
T_0	0	—	1	0	—	2	0	—	4	1	—
T_1	1	—	3	0	—	5	1	—	0	0	—
Таблица PT_1											
T_1	1	—	3	0	—	5	1	*	0	0	—
T_0	0	—	1	0	—	2	0	—	4	1	—
T_2	2	—	6	0	—	0	0	—	3	0	—
T_6	6	2	7	0	1	4	1	*	2	0	3
T_7	7	1	5	0	0	3	0	2	6	0	2
T_5	5	0	4	1	*	1	1	*	7	0	1
T_3	3	2	2	0	—	7	0	—	1	0	—
Строки с результатами контроля РК № 8											
T_8	8	—	9	0	—	1	0	—	7	0	—
T_9	9	—	0	0	—	6	1	—	8	0	—
Таблица PT_9											
T_9	9	—	0	0	—	6	1	—	8	0	—
T_0	0	—	1	0	—	2	0	—	9	0	—
T_1	1	—	3	1	—	8	0	—	0	0	—
T_8	8	—	9	0	—	1	0	—	7	0	—
T_7	7	—	8	0	—	3	1	—	6	0	—
T_6	6	—	7	0	—	9	1	—	2	0	—
T_2	2	—	6	0	—	0	0	—	3	1	—

даст таблицу PT_1 и функции МПщ на следующем шаге СД, например, модулю 0.

Модуль 0 выполнит следующие действия:

1) из таблицы T_{S0} выделит строку T_0 и перепишет ее в таблицу PT_1 (см. табл. 4);

2) в строке T_0 найдет номера преемников 1, 2, 4; проверит их и в соответствующие клетки строки T_0 запишет результаты проверок: $r_{0,1} = 0$, $r_{0,2} = 0$, $r_{0,4} = 1$;

3) по нулевым результатам проверок в строке T_0 выделит номера исправных преемников $\{q_0 = 1, 2\}$, сравнил их с номерами 1, 0 в левом столбце таблицы PT_1 и обнаружит, что модуль 2 не исполнял функции МПщ, поэтому передаст ему таблицу PT_1 с результатами проверок и функции МПщ на следующем шаге СД, при этом модуль 2 вернется к п. 1 (см. § 3) процесса СД.

Аналогичные действия в роли МПщ поочередно выполнят модули 2, 6, 7 и 5, результаты их проверок своих преемников показаны в строках T_2 , T_6 , T_7 , T_5 табл. 4. В отличие от предыдущих МПщ модуль 5 обнаружит, что его исправный преемник – модуль 7 – уже исполнял функции МПщ и передаст к поиску пути к одному из исправных модулей, еще не исполнявших функции МПщ, с помощью процедуры ранжирования.

Из результатов проверок, записанных в строках T_1, \dots, T_5 табл. PT_1 , модуль 5 определит, что только исправный модуль 3 не исполнял функции МПщ, из табл. T_{S0} выделит строку T_3 и перепишет ее в табл. PT_1 . Затем, выполняя процедуру ранжирования, модуль 5 присвоит ранг 0 модулю 5, ранг 1 модулю 7, затем ранг 2 модулям 3 и 6. Ранжирование закончено, так как модуль 3 попал в число ранжированных. (Результаты ранжирования показаны в строках табл. PT_1 ; в некоторых клетках столбцов «ранг m_{k1} » и «ранг m_{k2} » стоит знак *, означающий, что соответствующие модули m_{k1} и m_{k2} не участвуют в формировании искомого пути, так как они признаны отказавшими по единичным результатам их проверок). Теперь модуль 5 передаст табл. PT_1 и функции МПщ на следующем шаге СД модулю 3 через цепочку модулей: $5 \rightarrow 7 \rightarrow 3$.

Модуль 3, получив управление от модуля 5, выполнит следующие действия:

1) в строке T_3 табл. PT_1 найдет номера преемников 2, 7, 1; проверит их и в соответствующие клетки строки T_3 запишет результаты: $r_{3,2} = r_{3,7} = r_{3,1} = 0$;

2) по нулевым результатам проверок в строке T_3 выделит номера исправных преемников $\{q_3 = 2, 7, 1\}$,

обнаружит, что все они входят в список номеров левого столбца PT_1 , что других исправных модулей, не исполнявших функции МПщ, нет, поэтому процесс сбора тестовой информации закончен. Теперь модуль 3 отправит табл. PT_1 в обратный путь по цепочке: $3 \rightarrow 7 \rightarrow 5 \rightarrow 7 \rightarrow 6 \rightarrow 2 \rightarrow 0 \rightarrow 1$, чтобы каждый из исправных модулей имел одинаковую информацию.

Получив табл. PT_1 от модуля 3, каждый исправленный модуль выделит из ее строк результаты проверок и запишет их в клетки строк диагностической таблицы T_{D0} (см. табл. 3). Затем на основании правил, приведенных в основном тексте статьи, сформирует и запишет в клетки $s_k, a_{j1,k}, a_{j2,k}, a_{j3,k}$ восьми строк табл. T_{D0} соответствующие диагностические заключения. Их дешифрация позволяет сделать следующие выводы:

- подозреваются в отказе модуль 4 и линии связи ($5 - 1$) и ($1 - 5$);

- нет информации о состоянии линий ($0 - 4$), ($6 - 4$) и ($5 - 4$), так как отказавший модуль 4 «маскирует» отказы своих входных линий, поэтому линия ($6 - 4$) не попала в число подозреваемых в отказе;

- также нет информации о состоянии линий ($4 - 0$), ($4 - 6$) и ($4 - 5$): отказавший модуль 4 не использовался в качестве МПщ;

- в РК № 0 исправны 7 модулей ($0, 1, \dots, 7, 5$) и остальные линии связи, кроме перечисленных.

Сравнение полученного диагноза с составом исходной отказовой ситуации показывает, что вместо линии ($5 - 1$) подозревается в отказе пара линий ($5 - 1$) и ($1 - 5$); причина такого диагноза в том, что логическими методами нельзя различить отказы линий, участвующих в инверсных проверках.

Для устранения влияния подозреваемых компонентов – модуля 4 и дуг ($5 - 1$) и ($1 - 5$) – монитор должен перевести 2-ОУ систему из РК № 0 в РК, в которую не входят эти компоненты. Обратившись к первичной таблице (см. табл. 1), нетрудно обнаружить, что модуль 4 входит в состав восьми РК с номерами от 0 до 7 и не входит только в РК № № 8 и 9, где он записан в числе резервных модулей; в эти РК не входит и дуга ($5 - 1$). Для определенности, выберем в качестве следующей РК № 8.

После нахождения новой работоспособной РК диагностический монитор должен модифицировать таблицу состава T_{S0} и диагностическую таблицу T_{D0} , заменив номера рабочих модулей в соответствии с подстановкой: (89013267) вместо (01326754) и получив таблицу T_{S8} (табл. 5) и диагностическую таблицу T_{D8} (табл. 6). Затем монитор



запустит процесс контроля РК № 8, который подтвердит ее работоспособность.

Пусть спустя некоторое время работы системы в РК № 8 возникнут отказы модуля 3 и линии (6 – 9), которые в табл. 1 отмечены знаком **.

Как отмечалось, контроль работоспособности 2-ОУ системы начинается с модуля, соответствующего начальной вершине гамильтонова цикла для выбранной РК. После смены РК и замены номеров рабочих вершин меняется и состав гамильтонова цикла: так, для РК № 8 получим такой цикл: $8 \rightarrow 9 \rightarrow 0 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 6 \rightarrow 7 \rightarrow 8$. Теперь по сигналу ДМ начальный модуль 8 процесса контроля выполнит действия, аналогичные действиям модуля 0 в РК № 0, но искомую строку T_8 возьмет из табл. T_{S8} , передаст управление контролем модулю 9, который, обнаружив единичный результат проверки модуля 6, переведет ДМ в режим выполнения процесса СД (результаты проверок, выпол-

ненных модулями 8 и 9, показаны в строках T_8 и T_9 табл. 4).

Начальный модуль процесса СД – модуль 9 – начнет построение промежуточной таблицы PT_9 , переписав в нее строку T_9 с результатами проверок, выполненных в процессе контроля, затем по нулевым результатам проверок выделит номера 0 и 8 исправных преемников, и одному из них, например, модулю 0 передаст таблицу PT_9 и функции МПш на следующем шаге СД.

Действия модуля 0 и последующих МПш аналогичны действиям, ранее описанным в процессе СД в РК № 0, за исключением того, что необходимые строки будут извлекаться из табл. T_{S8} (см. табл. 5); результаты выполненных ими проверок показаны в табл. PT_9 (см. табл. 4). Модуль 2, завершив ее построение, обнаружит, что больше нет исправных модулей, не выполнявших функции МПш, и передаст табл. PT_9 по обратной цепочке:

Таблица 5

Таблица состава T_{S8} для РК № 8

Символ строки	Проверяющий модуль		Преемники m_j , результаты их проверок и ранг								
	m_j	ранг m_j	m_{k1}	$r_{j, k1}$	ранг m_{k1}	m_{k2}	$r_{j, k2}$	ранг m_{k2}	m_{k3}	$r_{j, k3}$	ранг m_{k3}
T_8	8	–	9	–	–	1	–	–	7	–	–
T_9	9	–	0	–	–	6	–	–	8	–	–
T_0	0	–	1	–	–	2	–	–	9	–	–
T_1	1	–	3	–	–	8	–	–	0	–	–
T_3	3	–	2	–	–	7	–	–	1	–	–
T_2	2	–	6	–	–	0	–	–	3	–	–
T_6	6	–	7	–	–	9	–	–	2	–	–
T_7	7	–	8	–	–	3	–	–	6	–	–

Таблица 6

Диагностическая таблица T_{D8} для РК № 8

Символ строки	Предшественники m_k			Прове-ряемый модуль m_k	Результаты проверок			Диагностическое заключение			
	m_{j1}	m_{j2}	m_{j3}		$r_{j1, k}$	$r_{j2, k}$	$r_{j3, k}$	s_k	$a_{j1, k}$	$a_{j2, k}$	$a_{j3, k}$
D_8	9	1	7	8	0	0	0	0	0	0	0
D_9	0	6	8	9	0	1	0	0	0	1	0
D_0	1	2	9	0	0	0	0	0	0	0	0
D_1	3	8	0	1	–	0	0	0	–	0	0
D_3	2	7	1	3	1	1	1	1	z	z	z
D_2	6	0	3	2	0	0	–	0	0	0	–
D_6	7	9	2	6	0	1	0	0	0	1	0
D_7	8	3	6	7	0	–	0	0	0	–	0

$2 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 1 \rightarrow 0 \rightarrow 9$. Каждый из этих модулей перепишет из PT_9 результаты проверок в диагностическую таблицу T_{D8} (см. табл. 6), обрабатывает их и придет к следующим диагностическим заключениям:

- подозреваются в отказе модуль 3 и линии связи ($6 - 9$) и ($9 - 6$);
- нет информации о состоянии линий ($2 - 3$), ($7 - 3$) и ($1 - 3$), так как отказавший модуль 3 «маскирует» отказы своих входных линий;
- также нет информации о состоянии линий ($3 - 1$), ($3 - 2$) и ($3 - 7$): отказавший модуль 3 не использовался в качестве МПш;
- в РК № 8 исправны семь модулей ($9, 0, 1, 3, 2, 6, 7$) и остальные линии связи, кроме перечисленных.

Сравнение полученного диагноза с составом исходной отказовой ситуации показывает, что вместо линии ($6 - 9$) подозревается в отказе пары линий ($6 - 9$) и ($9 - 6$); причина – та же, что и при диагностировании линии ($5 - 1$) в РК № 0.

Для устранения влияния подозреваемых компонентов – модуля 3 и дуг ($6 - 9$) и ($9 - 6$) – монитор должен перевести 2-ОУ систему из РК № 8 в РК, в которую не входят эти компоненты, а также ранее обнаруженные отказавшие модуль 4 и линии ($5 - 1$) и ($1 - 5$). Отказ двух модулей оставляет в составе 2-ОУ системы восемь исправных модулей, т. е. система оказывается неизбыточной для отказов любого рабочего модуля. Удаление из графа G_R^{10} (см. рис. 2) отказавших вершин 4 и 3 и инцидентных им дуг преобразует его в восемь вершинный полно связный граф $G_{R^*}^8$, показанный на рис. 2 более толстыми линиями. В статьях [4, 5] показано, что избыточное число дуг длины 3 в графике $G_{R^*}^8$ по сравнению с числом дуг длины 3 в целевом графике G_0^8 предоставляет дополнительные возможности для устранения их отказов с помощью реконфигурации. Действительно, в графике G_0^8 таких дуг – четыре, их номера в РК № 0 – ($0 - 2$), ($3 - 7$), ($6 - 4$) и ($5 - 1$); а график $G_{R^*}^8$ содержит восемь дуг длины 3. Анализ расположения этих восемь дуг показывает, что они позволяют построить два РК из восемь вершин с непересекающимися подмножествами дуг длины 3. На рис. 2 показана одна из этих РК, использующая дуги длины 3 с номерами ($2 - 5$), ($7 - 9$), ($8 - 1$) и ($0 - 6$). Вторая РК использует дуги длины 3 с номерами ($6 - 8$), ($5 - 0$), ($9 - 2$), ($1 - 7$). Нетрудно проверить, что одно подмножество дуг переходит в другое (и обратно) при сдвиге рабочих вершин графа $G_{R^*}^8$ на 1 шаг, 3, 5, 7 шагов (2, 4 шага, 6, 8 шагов) по га-

милтонову циклу для графа $G_{R^*}^8$: $2 \rightarrow 6 \rightarrow 7 \rightarrow 5 \rightarrow 8 \rightarrow 9 \rightarrow 0 \rightarrow 1 \rightarrow 2$. Следовательно, наличие двух работоспособных РК в графике $G_{R^*}^8$ позволяет устранить отказ одной дуги, входящей (или отказы 2...4 дуг, входящих) в одно из приведенных подмножеств, путем реконфигурации и таким образом продлить рабочее функционирование 2-ОУ системы без ручной замены отказавших линий связи.

ЗАКЛЮЧЕНИЕ

Предложен метод самодиагностирования состояния компонентов (модулей и линий связи) цифровых систем с реконфигурацией, для построения графов структур которых применены алгоритмы синтеза, предложенные М.Ф. Караваем. В основу разработанного метода СД положен принцип расширяющихся областей: выполнение необходимых процедур начинается с исправного модуля и последовательно передается другим модулям, исправность которых подтверждена предшествующими проверками. Начальный модуль процедуры СД не является фиксированным, а определяется по результатам работы системы контроля. Предполагается, что допускаются устойчивые кратные отказы ограниченного числа компонентов, причем за время диагностирования новые отказы не возникают, а существующие неисправности таковы, что не приводят к несанкционированному запуску процесса СД в отказавших модулях. Диагностические процедуры реализуются в соответствии с моделью БГМ.

Предполагается, что в исходной отказоустойчивой системе есть диагностический монитор (часть центральной операционной системы), который инициирует процессы контроля, СД и осуществляет реконфигурацию по результатам диагностирования очередной РК.

В соответствии с рассмотренным методом СД выполняются процедуры: исполнения исправным модулем функций проверяющего для своих преемников; нахождения кратчайшего пути от исправного модуля к ближайшему, исправному модулю, еще не исполнявшему функций проверяющего; обработки и дешифрации исправным модулем результатов выполненных проверок.

Для устранения влияния компонентов, подозреваемых в отказе, путем смены РК отказоустойчивого графа применяется способ реконфигурации, представляющий собой вариант скользящего резервирования. В цифровых системах возможны такие сочетания отказавших смежных компонентов (модулей и линий связи), потенциальные синдромы которых неразличимы для однократного



диагностирования. Однако для отказоустойчивых систем с реконфигурацией неразличимость отказов смежных компонентов не препятствует восстановлению их работоспособности.

ЛИТЕРАТУРА

1. Авиженис А. Отказоустойчивость – свойство, обеспечивающее постоянную работоспособность цифровых систем // ТИИЭР. – 1978. – Т. 66. – № 10. – С. 5–25.
2. Каравай М.Ф. Инвариантно-групповой подход к исследованию k -отказоустойчивых структур // Автоматика и телемеханика. – 2000. – № 1. – С. 144–156.
3. Maehle E., Moritz K., Wirl K. A graph model for diagnosis and reconfiguration and its application to a fault-tolerant multiprocessor system // 16-th Int. Conf. on Fault-Tolerant Computing FTCS-16, 1986. – Р. 292–297.
4. Ведешенков В.А. Об использовании избыточных хорд k -отказоустойчивого графа для устранения влияния отказавших компонент // Автоматика и телемеханика. – 2003. – № 4. – С. 158–166.
5. Ведешенков В.А. Процедура восстановления работоспособности отказоустойчивых цифровых систем с динамической избыточностью // Автоматика и телемеханика. – 2003. – № 5. – С. 167–179.
6. Barsi F., Grandoni F., Maestrini P. A theory of diagnosability of digital systems // IEEE Trans. Comput. – 1976. – Vol. C-25. – № 6. – Р. 585–593.
7. Preparata F.P., Metze G., Chien R.T. On the connection assignment problem of diagnosable systems // IEEE Trans. Electr. Comput. – 1967. – Vol. 16. – № 6. – Р. 848–854.
8. Пархоменко П.П. Определение технического состояния многопроцессорных вычислительных систем путем анализа графа синдромов // Автоматика и телемеханика. – 1999. – № 5. – С. 126–135.
9. Buskens R.W., Bianchini R.P. Distributed on-line diagnosis in the presence of arbitrary faults // Proc. of the 23-rd Int. Symp. on Fault Tolerant Computing, 1993. – Р. 470–479.
10. Hosseini S.H., Kuhl J.G., Reddy S.M. A diagnosis algorithm for distributed computing systems with dynamic failure and repair // IEEE Trans. Comput. – 1984. – Vol. C-33. – № 3. – Р. 223–233.
11. Bianchini R.Jr., Goodwin K., Nydick D.S. Practical application and implementation of distributed system-level diagnosis theory // Proc. of the 20-th Int. Symp. on Fault Tolerant Computing, 1990. – Р. 332–339.
12. Bianchini R.Jr., Buskens R.W. An adaptive distributed system-level diagnosis algorithm and its implementation // Proc. of the 21-st Int. Symp. on Fault Tolerant Computing, 1991. – Р. 222–229.
13. Ведешенков В.А. Организация самодиагностирования технического состояния цифровых систем // Автоматика и телемеханика. – 2003. – № 11. – С. 165–182.
14. Ведешенков В.А. О диагностировании отказавших модулей и связей в цифровых системах с использованием модели БГМ // Автоматика и телемеханика. – 2002. – № 2. – С. 159–171.
15. Пархоменко П.П. О системном диагностировании вершинных и реберных гиперкубовых структур многопроцессорных вычислительных систем // Автоматика и телемеханика. – 1996. – № 11. – С. 165–174.
16. Ведешенков В.А. О системном диагностировании отказавших модулей и связей в гиперкубовых структурах многопроцессорных вычислительных систем // Автоматика и телемеханика. – 2001. – № 12. – С. 83–94.

☎ (095) 334-75-90

E-mail: vva@l9.ipu.rssi.ru

Новая книга

Новиков Д.А. Стимулирование в организационных системах. – М.: СИНТЕГ, 2003. – 312 с. (серия «Управление организационными системами»).

Монография посвящена описанию формальных моделей стимулирования в организационных системах. Ее целью, помимо ознакомления читателя с современным состоянием дел в данной области теории управления, является демонстрация необходимости и целесообразности применения моделей стимулирования для повышения эффективности работы организаций.

Рассмотрены модели индивидуального и коллективного стимулирования, установлена взаимосвязь между результатами моделирования и применяемыми на практике формами и системами оплаты труда, приведены результаты экспериментальных исследований, обсуждена возможность использования решений задач стимулирования при оптимизации состава организационных систем.

Книга адресована студентам вузов, аспирантам и специалистам (теоретикам и практикам) в области экономики труда и управления социально-экономическими системами.

Д.А. Новиков – д-р техн. наук, профессор кафедры «Проблемы управления» Московского физико-технического института, ведущий научный сотрудник лаборатории активных систем Института проблем управления им. В.А. Трапезникова. Автор многих работ по теории управления социально-экономическими системами, в том числе по теории игр, принятию решений и механизмам управления организационными системами.



АНАЛИЗ НЕШТАТНЫХ СИТУАЦИЙ И КРИТИЧНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ПРОЕКТЕ МЕЖДУНАРОДНОЙ КОСМИЧЕСКОЙ СТАНЦИИ

Е.А. Микрин*, В.П. Пелихов**

*Ракетно-космическая корпорация «Энергия», г. Королев

**Институт проблем управления им. В.А. Трапезникова, г. Москва

Дана классификация основных типов нештатных ситуаций, возникающих при функционировании долговременной орбитальной станции. Описаны основные характеристики критических функций, выполняемых автоматикой и программным обеспечением станции. Приведена классификация программного обеспечения по категориям опасности.

ВВЕДЕНИЕ

Безопасность космического полета долговременной орбитальной станции (ДОС) обеспечивается организацией работ по следующим направлениям:

- достижение при создании модулей ДОС и поддержание в процессе их эксплуатации требуемого уровня надежности функционирования бортовых систем;
- сведение риска возникновения возможных опасностей к приемлемому уровню мероприятиями по их предупреждению, парированию и локализации;
- анализ расчетных нештатных ситуаций (НШС) модулей и ДОС в целом, а также выбор способов выхода из них.

В основу решения проблемы обеспечения приемлемого уровня риска возникновения возможных опасностей положен принцип анализа функциональных возможностей и конструкции в целом на устойчивость к проявлению возможных опасностей, связанных с угрозой здоровью и жизни экипажа, с угрозой утраты одного из модулей ДОС, транспортного корабля или всей станции в целом. Перечень таких опасностей определяется в результате исследования программы функционирования от момента запуска до завершения существования станции на орбите.

ОПРЕДЕЛЕНИЕ И КЛАССИФИКАЦИЯ НЕШТАТНЫХ СИТУАЦИЙ

Нештатная ситуация – это состояние космического комплекса, его составных частей и привлекаемых средств, а также условий полета, не предусмотренное программой штатного функцио-

нирования, или отклонение состояния здоровья космонавтов от нормального. В качестве причин возникновения НШС могут рассматриваться отказы в бортовых системах и агрегатах, изменение условий эксплуатации и ошибки экипажа или наземного персонала управления, которые приводят к невыполнению функций бортовыми системами или к изменениям характеристик их функционирования, выходящими за допустимые пределы.

Рассмотренные НШС (см. рисунок) – это нештатные ситуации в космическом полете, которые были выявлены и рассмотрены в процессе создания космического комплекса и предполетного анализа и внесены в конструкторскую документацию.

Нерассмотренные НШС – это нештатные ситуации, анализ которых не мог быть проведен в предполетный период и которые не содержатся в конструкторской документации.

Расчетные НШС – это нештатные ситуации в космическом полете, способы и средства выхода из которых предусмотрены и внесены в конструкторскую документацию.

Нерасчетные НШС – это нештатные ситуации, причиной возникновения которых являются отказы, приводящие к нерасчетным или неопределенным условиям эксплуатации. Выходы из таких



Классификация нештатных ситуаций



НШС не гарантируются с помощью разработанных способов и средств, но для них могут быть предусмотрены мероприятия, снижающие риск и повышающие вероятность выполнения задач полета.

Аварийные ситуации (AC) – это нештатные ситуации, приводящие к полному или частичному разрушению, выходу из строя хотя бы одной из составных частей космического комплекса, а также к угрозе жизни или ухудшению здоровья хотя бы одного из членов экипажа.

ОСНОВНЫЕ ПОЛОЖЕНИЯ, ПРИНЯТЫЕ ПРИ РАЗРАБОТКЕ ПЕРЕЧНЯ НШС

По каждой опасности оформляется отчет по форме NASA, который представляется в Комиссию NASA по безопасности полета для защиты на соответствие установленным требованиям. В состав отчета включаются мероприятия по предупреждению опасностей, их парированию и локализации, а также данные по верификации этих мероприятий. Работы по анализу нештатных ситуаций выполняются в соответствии с нормативной документацией и государственными стандартами Российской Федерации.

Анализ НШС осуществляется для различных уровней структуры ДОС (отдельный агрегат, бортовая система, блок). Для этого используются возможности различных алгоритмов управления функционированием и полетом ДОС: от конкретных циклограмм, формализующих порядок функционирования отдельных агрегатов и систем, до алгоритмов верхнего уровня управления – алгоритмов программы полета ДОС.

Анализ возможных отказов и нештатных ситуаций осуществляется в рамках кооперации смежников по принадлежности. Результаты такого анализа и мероприятия, связанные с предупреждением, парированием и локализацией возможных отказов, НШС и их последствий, отражаются в соответствующей документации.

Возможные отказы и НШС, последствия которых не выходят за пределы отдельной бортовой системы, анализируются разработчиком данной системы. Результаты такого анализа отражаются соответствующими смежниками в их конструкторской документации на систему. Парирование возможных отказов и НШС рассматриваемого уровня осуществляется для всех этапов полета внутри циклограмм и алгоритмов функционирования агрегатов и систем с помощью встроенной автоматики или конечных исполнителей программно-математического обеспечения бортовых вычислительных средств.

ОСНОВНЫЕ ПОЛОЖЕНИЯ МЕТОДИКИ АНАЛИЗА НШС

Нештатные ситуации выявляются по результатам построения моделей их развития и анализа

возможных опасностей, отказов, потерь функций бортовых систем и ошибок операторов. Цель анализа НШС состоит в определении возможных последствий, методов идентификации и мероприятий по выходу из таких ситуаций.

Выходы из НШС могут осуществляться либо автоматически, либо по управляющим воздействиям наземного комплекса управления (НКУ), либо экипажем посредством ручных органов управления бортовыми системами, а также при выполнении ремонта бортовых систем и конструкций.

Анализ НШС и их отбор для рассмотрения производятся по следующим критериям:

- влияние на выполнение программы полета;
- влияние на работоспособность модулей ДОС;
- необходимость ремонтных работ, выполняемых космонавтами.

Нештатные ситуации рассматриваются для всех типов полета. В ряде случаев НШС анализируются для фиксированного момента, когда возможность их проявления связана с работой системы или агрегата в этот момент.

Результаты анализа расчетных и нерасчетных НШС, а также АС представляются в форме таблиц, содержащих следующие сведения:

- наименование НШС, которое должно содержать физический смысл ситуации с привязкой к потере или угрозе потери функции;
- причина возникновения НШС и модель ее развития;
- участок, на котором рассматривается возникновение НШС;
- средства идентификации НШС (автоматика, бортовой комплекс управления, НКУ); приоритетное средство идентификации указывается первым;
- последствия НШС, в качестве которых указываются возможные состояния бортовых систем ДОС при неприятии мер по выходу из НШС;
- мероприятия по выходу из НШС, включающие в себя действия бортовой автоматики, НКУ и экипажа;
- примечания, в которых дается дополнительная информация.

Обобщенный перечень НШС включает в себя перечни расчетных и нерасчетных НШС, а также АС для различных участков полета.

АНАЛИЗ КРИТИЧНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Причины и обстоятельства, приводящие к появлению НШС, порождают множество критических функций, связанных с программным обеспечением (ПО) [1, 2].

Анализ критичности ПО бортовой вычислительной системы (БВС) должен проводиться с учетом предварительной оценки архитектуры ПО и размещения его по вычислительным средствам



Классификация программного обеспечения по категориям опасности

Категория ПО	Определение
A	<ul style="list-style-type: none"> • Программное обеспечение, повреждение которого может привести к катастрофическим последствиям, когда никакие адекватные средства вмешательства, имеющиеся в системе, и никакое время недостаточны, чтобы эффективно вмешаться для предотвращения последствий события
B	<ul style="list-style-type: none"> • Программное обеспечение, повреждение которого может привести к катастрофическим последствиям, но в системе есть адекватные средства вмешательства и имеется достаточно времени для эффективного вмешательства. • Программное обеспечение, повреждение которого может привести к критическим последствиям, когда никакие адекватные средства вмешательства, имеющиеся в системе, и никакое время недостаточны, чтобы эффективно вмешаться для предотвращения последствий события
C	<ul style="list-style-type: none"> • Другое бортовое ПО, повреждение которого может привести к: <ul style="list-style-type: none"> – критическим последствиям, и в системе имеются адекватные средства вмешательства для предотвращения последствий события; – потере системы, и в системе имеются адекватные средства вмешательства для предотвращения последствий события; – невыполнению задания полета. • Все другое наземное ПО, которое функционирует вместе с бортовым ПО, летной материальной частью или которое используется для генерации бортового ПО
D	<ul style="list-style-type: none"> • Все другое наземное ПО

БВС в соответствии с техническими спецификациями на ПО.

На основании технических спецификаций на интегрированное ПО и анализа реализуемости маневра ДОС осуществляется декомпозиция интегрированных функций и их размещение по компьютерам БВС.

Далее на основании предварительной оценки архитектуры программного обеспечения БВС осуществляется декомпозиция элементов конфигурации ПО на программные компоненты, реализующие те или иные критические функции во времени. Составляются соответствующие матрицы участия каждого программного компонента в реализации критических функций. В результате такого анализа выявляются программные компоненты, участвующие в реализации критических функций.

Выделяют следующие категории опасности.

Катастрофическая:

- гибель людей, угроза их жизни, ранения, приводящие к инвалидности, или временная профессиональная болезнь;
- потенциальная потеря модуля ДОС;
- потеря государственной или частной собственности.

Критическая:

- временный ущерб, не угрожающий жизни, или временная профессиональная болезнь;
- значительное повреждение модуля ДОС;
- потеря или значительное повреждение государственной или частной собственности;
- длительное, вредное влияние на окружающую среду.

В соответствии с техническими требованиями Европейского космического агентства принята классификация ПО по категориям опасности (см. выше).

Результаты проведенного анализа, с учетом принципа максимальной критичности, показыва-

ют, что программные компоненты, размещаемые на компьютерах БВС, должны иметь достаточно высокие категории – классов В и С. Проектирование интегрированного программного обеспечения БВС должно отвечать соответствующим требованиям обеспечения качества программного продукта. Данным требованиям должны отвечать все аспекты проектирования ПО на всех его стадиях, от разработки технических спецификаций до его валидации (проверке правильности) и верификации.

Отработочные, интеграционные и верификационные испытания на наземном комплексе отработки должны включать в себя моделирование как nominalного полета, так и НШС в односторонних и совместных испытаниях, а также проверку правильности взаимодействия с реальными данными на односторонних и совместных тестах. Особое внимание следует обратить на отработку взаимодействия ПО и системных функций в НШС, моделирующих отказы тех или иных программных компонентов.

В заключение отметим, что применение formalизованных методов анализа НШС представляется перспективным направлением исследований процессов возникновения и распространения НШС и способов их предупреждения.

ЛИТЕРАТУРА

1. Микрин Е.А., Пелихов В.П. Оценивание риска элементов программного обеспечения как фактор управления безопасностью функционирования долговременных орбитальных станций // Междунар. конф. по пробл. упр. (29 июня – 2 июля 1999 г.) / Ин-т пробл. упр.: Тез. докл. М., 1999.
2. Микрин Е.А. Управление производством программно-математического обеспечения при реализации проекта МКС «Альфа» // Междунар. научн.-практ. конф. «Теория активных систем» (15–17 ноября 1999 г.) / Ин-т пробл. упр.: Тез. докл. М., 1999.

☎ (095) 334-90-09

E-mail: kulba@ipu.rssi.ru





УДК 681.5

ОТКРЫТЫЕ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ С РАСПОЗНАВАНИЕМ РЕЧИ

В.А. Жожикашвили, Р.В. Билик, В.А. Вертиб, А.В. Жожикашвили,
Н.В. Петухова, М.П. Фархадов

Институт проблем управления им. В.А. Трапезникова, г. Москва

Обсуждена проблема доступа широких масс населения к информации. Введено понятие «открытые системы массового обслуживания» и рассмотрено применение технологии распознавания речи для организации голосового интерфейса информационных систем. Описана действующая система с телефонным доступом к базам данных.

ВВЕДЕНИЕ

Понятие *автоматизированные системы массового обслуживания* (СМО) впервые было предложено в трудах Института проблем управления (ИПУ) в 1965 г. и стало в дальнейшем общепринятым. Первой крупной СМО в СССР стала система «Сирена», предназначенная для бронирования мест и продажи билетов на самолеты Аэрофлота. В ней были реализованы наиболее передовые идеи вычислительной техники того времени, а именно, интерактивная удаленная база данных, сеть передачи данных с адаптивной маршрутизацией пакетов, экранные пульты с алфавитно-цифровой клавиатурой.

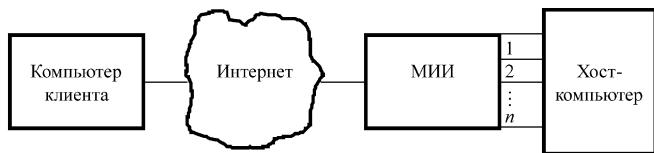
Вместе с тем «Сирена» по современным понятиям была закрытой системой в том смысле, что доступ к «своей» базе данных поддерживался только для узкого круга специально обученных операторов, в то время как доступа к чужим базам данных не было вовсе. С появлением Интернета «Сирена» стала эволюционировать в сторону большей открытости. Под открытостью системы понимается ее общедоступность.

Однако в настоящее время трансформация СМО из закрытой системы в открытую доступна только разработчикам или эксплуатационникам (только владельцам) СМО, которые в этом смысле являются монополистами, так как этот процесс требует серьезных усилий. Такое положение можно считать пережитком закрытого статуса СМО.

Несколько лет назад в ИПУ были начаты работы в области открытых систем обслуживания и разработаны принципы применения элементов естественного языка в человеко-машинном общении, позволяющие привлечь к системе большие массы клиентов, которые не проходили специального обучения [1, 2].

Структура СМО показана на рисунке. Клиент, не владеющий специальным языком запросов, обращается к системе на естественном языке. Перед хост-компьютером СМО, где размещена база данных, устанавливается компьютер, выполняющий функции модуля интеллектуального интерфейса (МИИ). Получив вызов клиента, МИИ выделяет из естественного языка смысл обращения (например, по ключевым словам) и подсказывает клиенту, что именно он должен передать в сторону хост-компьютера. При этом МИИ предлагает клиенту по возможности передать за один раз все необходимые для оформления сделки сведения. Получив их, МИИ формализует их в соответствии с требованиями хост-компьютера и располагает в необходимом порядке. Такой алгоритм позволяет достичь следующих результатов:

- обеспечивается доступ к услугам СМО массам необученных клиентов в среде Интернет, минуя агентов-операторов (телефонистов, диспетчеров или кассиров);
- не требуется никакого вмешательства в программы хост-компьютера;



Структура СМО с использованием элементов естественного языка

- минимизируется количество переключений «туда» и «обратно» в среде Интернет. В частном случае достигается одно обращение от клиента к базе данных хост-компьютера. Оно порождает n обращений от МИИ к базе данных, которые выполняются с высокой скоростью и не замедляют процесс заключения сделки.

Отметим еще две принципиальные особенности открытых СМО, получивших развитие в работах ИПУ:

- максимизация доходов от функционирования СМО (технология MAX);
- применение технологии распознавания речи.

Технология MAX начала применяться в странах Запада еще в 1990-х годах, но до настоящего времени практически не применяется в России. Ее компоненты:

- оптимальное управление ценами и тарифами,
- управление маршрутизацией товаров и услуг,
- сверхбронирование.

В наиболее развитой форме эти компоненты технологии MAX получили воплощение в бизнесе авиаперевозок [3–5].

Однако, хотя технология MAX и характерна для СМО нового поколения, она не вполне вписывается в понятие «открытые» системы и поэтому подробно не рассматривается в настоящей статье.

ПРИМЕНЕНИЕ ТЕХНОЛОГИИ РАСПОЗНАВАНИЯ РЕЧИ

Пожалуй, самым новым, оригинальным и перспективным стало применение в СМО технологии распознавания речи [6]. На сегодня она представляет собой наиболее естественный и многообещающий путь развития СМО, но одновременно это одна из самых сложных проблем в области человека-машинного интерфейса. Распознавание речи – это ключ к максимальной открытости СМО, ибо оно доступно практически всем. В отличие от упомянутого алгоритма (см. рисунок), технология распознавания речи не требует наличия у клиента под рукой никакого компьютера, а МИИ дополняется программой распознавания речи. Технология распознавания речи, с одной стороны, работает в интересах клиентов, упрощая для них доступ к информации, с другой – в интересах фирм-провайдеров услуг и товаров, позволяя им сокращать

численность телефонистов, операторов и диспетчеров, занятых обслуживанием клиентов, и этим увеличивать доходность своего бизнеса.

Области применения открытых СМО с распознаванием речи настолько обширны и разнообразны, что подробно перечислять их нет необходимости. Укрупненно можно указать на следующие применения:

- всевозможные справочные системы;
- системы заказа и бронирования услуг и товаров;
- экспертные системы;
- управление механизмами, агрегатами и технологическими процессами;
- помочь людям с нарушениями зрения и двигательных функций рук.

Все известные алгоритмы распознавания речи работают на базе словарей, выделяя отдельные слова в речевом сигнале и преобразуя их в текст. Системы с распознаванием речи подразделяются на дикторозависимые и дикторонезависимые. Первые настраиваются на определенный голос и могут иметь словари объемом в несколько десятков тысяч слов, которые распознаются при слитном произношении. Вторые, напротив, настраиваются на различное произношение и способны содержать много меньшее количество слов, однако при большем разнообразии тематики. Они воспроизводят образцы речи сотен и даже тысяч носителей языка. Системы именно такого типа пригодны для открытых СМО.

Все компании пользуются примерно одинаковой базовой методикой, основанной на выполняемой в четыре этапа процедуре:

- *предпроцессор* выполняет предварительную обработку сигнала – автоматическую регулировку усиления, подавление эха и посторонних шумов, обнаружение начала и конца речи, обнаружение интонации конца фразы – и в результате на следующий этап передает сигнал высокого качества;
- *экстрактор* выполняет частотный анализ сигнала, акустико-фонетический поток разбивается на короткие кадры (векторы) продолжительностью примерно 10 мс;
- *компаратор* сравнивает каждый кадр (вектор) с имеющимися фонетическими образцами, для каждой пары вычисляется функция правдоподобия;
- *интерпретатор* решает задачу динамического программирования с целью найти наилучшее разбиение полученного от компаратора алфавитного потока на слова и фразы, в зависимости от объема словаря и действующих синтаксических правил применяется та или иная стратегия поиска.



В результате распознавания речи выдается поток слов и предложений. Чтобы некоторое приложение могло воспользоваться этим продуктом, необходим прикладной программный интерфейс API.

Далее излагаются результаты, достигнутые в ИПУ РАН по тематике распознавания речи в СМО и существенно повышающие их открытость.

ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ ОТКРЫТЫХ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ С РАСПОЗНАВАНИЕМ РЕЧИ

Главные отличительные особенности таких систем обслуживания, серьезно влияющие на процесс проектирования, заключаются в ориентации открытых систем на необученных пользователей с самым широким спектром вариантов их поведения и применение телефонов в качестве средства общения между ними и системой.

Процесс проектирования конкретного приложения, в котором предполагается автоматическое распознавание речи, состоит из этапов:

- изучения объекта и постановки задачи;
- проектирования приложения;
- реализации проекта;
- тестирования и настройки.

На первом этапе выявляются цели проекта, изучается технология, обследуется существующая техническая база.

Как правило, заказчик проекта имеет свою базу данных и операторов, которые принимают звонки клиентов, обращаются в базу данных за информацией и отвечают клиенту также по телефону. Во многих случаях такая система обслуживания уже бывает оборудована центрами обслуживания вызовов (Call Center, Contact Center) – программно-аппаратными комплексами, автоматизирующими процесс приема и обработки вызовов [7]. Call-центры включают в себя, как правило, интерактивную систему голосовых ответов IVR (Interactive Voice Response), позволяющую автоматизировать выдачу некоторых справок без обращения к оператору. Для этого система IVR зачитывает меню и предлагает выбрать нужный пункт путем нажатия соответствующей клавиши на клавиатуре тонального телефона. Абоненты с телефонными аппаратами без тонального набора не могут воспользоваться этой возможностью.

Отметим, что в нашей стране у большинства стационарных телефонных аппаратов пока нет опции тонального набора и, следовательно, они не годятся для самообслуживания в IVR-системах.

На этапе обследования объекта рассчитывается необходимое число портов, требуемая производительность обрабатывающего комплекса, срок окупаемости инвестиций, составляется бизнес-план. Если у заказчика уже есть IVR-система, то добав-

ление распознавания речи не потребует новых затрат на дооборудование существующих IVR-портов. Однако может потребоваться увеличение числа портов, тогда соответствующие затраты должны быть включены в расчет.

По оценкам западных источников стоимость обслуживания звонков с помощью IVR на порядок меньше, чем стоимость обслуживания звонков операторами. Практически то же соотношение имеет место для портов с возможностью распознавания речи. Поэтому даже при достаточно большой стоимости разработки проекта инвестиции окупаются за срок от шести месяцев до года. В наших условиях соотношение между стоимостью обслуживания звонков людьми и автоматом будет не таким впечатляющим по причине значительно меньших затрат на оплату работы операторов, однако и разработка проекта, и эксплуатационные расходы на поддержку работы оборудования тоже значительно меньше, чем на Западе, и по нашим оценкам инвестиции в проекты с распознаванием речи будут окупаться достаточно быстро.

На этом же этапе следует составить представление о клиентах: кто будет звонить в систему и как они будут реагировать на применение голосовой технологии. Это необходимо для последующего проектирования пользовательского интерфейса, хотя вполне возможно, что по мере дальнейшего знакомства с задачей представление о клиентах может трансформироваться. Следует также составить представление об имидже фирмы, для которой разрабатывается проект. Это может потребоваться, например, для решения вопроса какой нужен диктор для представления пользователю системы наиболее адекватным бренду фирмы способом.

Результат первого этапа – согласованный обеими сторонами (заказчиком и разработчиком) План проекта.

Этап проектирования приложения, пожалуй, самый важный, поскольку здесь фактически разрабатывается технологический процесс. Хорошо спроектированная технология позволяет достичь вполне удовлетворительных результатов даже при ошибках в распознавании, которые неизбежны как вследствие недостаточного качества телефонной связи, так и по причине особенностей речи пользователей.

На этапе проектирования:

- пишутся сценарии обработки всех типов вызовов;
- определяется список грамматик;
- составляется спецификация промптов (текстов, произносимых системой) и намечается их содержание;
- определяется интерфейс с базой данных;
- выявляется необходимость модулей преобразования текста в речь и верификации голоса;



- составляются структурные схемы программ;
- записываются примеры аудиоклипов, демонстрирующие звучание голоса диктора.

На данном этапе необходима плотная работа с персоналом заказчика, владеющим всеми вопросами технологии, и исследование существующих диалогов между клиентами и операторами.

Результат этапа проектирования — спецификация пользовательского интерфейса, спецификация грамматик, структурные схемы программ обработки вызовов, примеры и спецификация промптов, спецификация интерфейса с базой данных, техническое задание на программирование.

На этапе реализации проекта намеченные задачи воплощаются в программные коды. Характерная особенность этого этапа заключается в создании специальной базы вызовов (речевого корпуса) для тестирования и настройки параметров.

На этапе реализации проекта:

- кодируется приложение;
- разрабатываются грамматики и создаются пользовательские словари;
- записываются все промпты;
- отлаживается интерфейс с базой данных;
- создается корпус для тестирования и настройки;
- настраивается модуль преобразования текста в речь, если предполагается его применение.

Этап реализации проекта заканчивается созданием системы, которая готова к тестированию.

Этап тестирования и настройки совершенно необходим для открытых систем с распознаванием речи, которые ориентированы именно на работу с весьма широким кругом пользователей. Прежде чем они начнут звонить в систему, разработчики должны сделать все возможное, чтобы пользователи приняли систему и оставались довольны ее работой. Для этого предназначен аппарат внутреннего тестирования и настройки параметров. Однако в практике эксплуатации таких систем широко применяются методы получения «обратной связи», суть которых состоит в накоплении записей о совершившихся диалогах и в специальной обработке этих записей, а также в получении прямой информации о работе системы от ряда клиентов, привлекаемых к таким опросам.

На этапе тестирования и настройки:

- осуществляется пилотное тестирование, выполняемое разработчиками, и настраиваются параметры;
- составляется программа ввода в опытную эксплуатацию;
- настраиваются параметры и дорабатываются сценарии и грамматики по результатам опытной эксплуатации;
- система вводится в эксплуатацию.

Особенности пользовательского интерфейса

Пользовательский интерфейс должен предоставлять пользователю как смысловую информацию, так и информацию о самом интерфейсе, а также принимать информацию от пользователя.

В рассматриваемых нами системах связь с пользователями осуществляется чаще всего по телефону, хотя в общем случае можно рассматривать в дополнение к таким клиентам еще и пользователей, обращающихся за тем же сервисом со своих компьютеров. В этом случае интерфейс комбинированный: голосовой и традиционный графический.

В голосовом пользовательском интерфейсе поставка информации со стороны системы и реакция пользователя производятся в форме речи. Речь заменяет или дисплей, или кнопки, или индикаторы. Клиент пользуется телефоном — аппаратом, который изначально предназначался для взаимодействия с человеком на другом конце провода, поэтому он интуитивно ждет от компьютера такого же поведения, как от человека, да и сам часто разговаривает с компьютером, как с человеком.

Разумеется, идеалом была бы поддержка со стороны компьютера естественной речи человека. Работы в этом направлении ведутся в течение уже долгого времени, однако пока разработчики приложений имеют в своем распоряжении технологии с ограниченными возможностями, поэтому приближение диалога к естественному достигается в основном алгоритмическим путем.

Голосовой пользовательский интерфейс — относительно новая разновидность интерфейса «человек — машина», и для него пока нет установившихся правил, как для графического интерфейса [8], гарантирующих успех. Однако можно назвать несколько принципов, которые следует соблюдать и которые базируются на особенностях восприятия информации на слух, влиянии формы вопроса на ответ и на возможностях современных платформ распознавания речи.

Первый принцип, носящий самый общий характер и относящийся к проектированию любого интерфейса «человек-машина», состоит в том, что интерфейс должен быть ориентированным на человека и учитывать его возможности и слабости. Разработчик должен понимать трудности, с которыми будут сталкиваться люди при общении с создаваемым приложением. При разработке голосового интерфейса соблюдение этого принципа, может быть, даже важнее, чем в тех случаях, когда в цепочку общения включено зрение человека.

Другие принципы относятся уже непосредственно к голосовому интерфейсу:



- самый первый промпт должен содержать краткое объяснение, что и как делает данная система — клиент должен убедиться, что он позвонил в нужную ему систему, и понять, что сейчас он будет говорить с компьютером;
- при конструировании последующих промптов надо учитывать, что пользователь довольно быстро начинает понимать, как организован диалог, и усваивает терминологию; по мере обучения пользователя промпты-вопросы могут быть короче, а сообщаемая информация, наоборот длиннее;
- промпты должны быть однозначными, ясными и простыми по структуре;
- в промпте следует сообщать ровно столько информации, сколько необходимо пользователю для ее восприятия и однозначного ответа на задаваемый вопрос.

Помимо учета особенностей пользователя, разработчик интерфейса должен учитывать возможности машинной стороны. Что же предоставляют разработчику современные платформы распознавания речи?

Прежде всего отметим, что они обеспечивают достаточно высокий уровень распознавания: 95% и выше, что вполне достаточно для развертывания приложений. Неизбежные ошибки должны выявляться и корректироваться в ходе диалога. Для этого служит механизм подтверждения *Да/Нет*, однако включать его надо очень продуманно. Если задавать этот вопрос на каждом этапе получения данных от клиента, то диалог удлиняется и клиент быстро начинает раздражаться. С другой стороны, клиент будет обеспокоен, если поймет, что он неправильно понят или что он дал неправильный ответ на заданный вопрос, и будет торопиться исправить ошибку вместо того, чтобы заняться ответом на следующий вопрос. Однако это уже вопросы, относящиеся к задачам управления диалогом.

Далее, неплохим инструментом для сокращения продолжительности диалога служит программно-аппаратная поддержка функции *Barge-In*, когда клиент может отвечать на задаваемый компьютером вопрос, не дожидаясь окончания машинного промпта. О наличии такой возможности приложение должно сообщать клиенту в начале своей работы. Пользователь-новичок вряд ли воспользуется этой опцией. Но как только он поймет, что это такое, он будет ее использовать, тем самым значительно сокращая время диалога. Клиенты обучаются довольно быстро и, тем более, таким вещам, которые позволяют им сэкономить время. Функция *Barge-In* может быть реализована в самом хосте, но гораздо предпочтительнее голосовые платы с поддержкой функции *Barge-In*, чтобы освободить хост-компьютер от этой работы. Функция *Barge-In*

работает вместе с функциями определения начала и конца речи и эхоподавления. Для ее эффективной работы необходимо учитывать некоторые ограничения на грамматику, применяемую в сочетании с этой функцией. Грамматика должна оперировать по возможности простыми словами и фразами и не быть иерархической.

Еще один инструмент для конструирования интерфейса представляют собой открытые грамматики, предлагаемые разработчиками ряда платформ распознавания речи. Открытые грамматики дают пользователю значительно большую свободу высказывания в ответ на вопрос системы. Для системы важно, чтобы в ответе присутствовали ожидаемые ключевые слова, а все другие слова игнорируются. Открытые грамматики позволяют клиенту говорить более естественно, что очень важно для интерфейса открытых систем, ориентированных на массового пользователя. Однако такие грамматики требуют настройки и тренинга, что весьма трудоемко.

Конструкторы грамматик предлагают и другие механизмы для приема более натуральной речи от клиента. Они состоят в том, например, что в грамматику включается несколько предвидимых вариантов ответов, допускаются разные варианты порядка слов в ответах, заранее устанавливается более низкий порог распознавания для необязательных слов.

Можно назвать и другие средства систем распознавания, которые разработчик может применить при конструировании интерфейса, но тут важно отметить одно важное обстоятельство, а именно, что базовое матобеспечение платформ распознавания речи постоянно развивается и что эти усовершенствования в значительной своей части направлены именно на создание более натурального голосового интерфейса «человек — машина».

Управление диалогом в открытых системах

Цель управления диалогом — обеспечение успешного обслуживания вызова за кратчайшее время.

Управление диалогом необходимо даже в простейших системах, поскольку надо предвидеть возможные ошибки в распознавании и различные варианты поведения пользователя. В сложных системах с продолжительным диалогом и с выводом большого объема информации клиенту количество точек, где необходимо управлять процессом, бывает значительным.

Задача минимизации времени обслуживания вызова должна решаться еще на этапе планирования сценария. Для этой цели применяются вероятностные модели, которые описывают состояния

процесса, характеризующиеся временами пребывания в них и вероятностями переходов между состояниями.

Такой подход позволяет выбрать наиболее оптимальный сценарий, но это будет планирование на макроуровне. Для приложений со сложным и продолжительным диалогом со многими разветвлениями этих действий недостаточно.

Для таких систем помимо решения задачи оптимизации на этапе планирования сценария закладывается также возможность динамического управления диалогом в ходе его выполнения. Управление реализуется обычно в точках получения результата от модуля распознавания и в точках подготовки к выводу информации, полученной из базы данных.

Модуль распознавания может выдать результат распознавания и уверенность (вероятность), с которой этот результат получен, либо в случае неуспеха ошибку распознавания и ее тип. Если используется опция *N-best*, то выдается не один результат, самый вероятный, а *N*, каждый со своей гипотезой и ее вероятностью.

Если уверенность в распознавании основной (первой) гипотезы больше некоторого заданного значения (не путать с порогом уверенности, который разграничивает успех и неудачу в распознавании), то модуль управления продвигает диалог к следующему шагу. В противном же случае предпринимаются некоторые действия, нацеленные на то, чтобы отметить, что распознавание в этой точке было выполнено с недостаточно высокой уверенностью. Эти данные могут быть использованы далее, а пока управление может быть возвращено в ту же точку, после чего диалог также продвигается к следующему шагу. Если используется опция *N-best*, то управление может быть более сложным. Например, при низкой уверенности основной гипотезы и достаточно высокой уверенности второй гипотезы необходимо проанализировать, на каком слове какой грамматики это произошло. Дело в том, что часто не удается избежать включения в грамматику похожих по звучанию слов, например, *июнь* и *июль* в грамматике месяцев, или очень коротких слов со свистящими звуками, которые распознаются как правило хуже. При разработке стратегии управления можно учитывать такие факторы и если обнаруживается низкая уверенность распознавания основной гипотезы, а также есть и вторая гипотеза с относительно высокой вероятностью, то диалог может быть направлен на веточку с промптом типа «Вы сказали *июнь*, да или нет?».

Другой тип управления можно назвать ситуационным. Суть его состоит в том, что в расчет принимаются такие факторы, как история диалога,

наличие ошибок на предыдущих фазах, объем предназначенный для вывода информации. К примеру, если произошла ошибка в распознавании, то необходимо проанализировать, первая ли она на этом шаге диалога или нет, и в зависимости от этого направить клиенту промпт того или иного содержания. Или если объем данных, полученных из базы, велик, то в промпте надо предупредить клиента об этом и предложить выбрать способ передачи ему этой информации.

Существуют и другие виды управления диалогом, при умелом применении которых можно значительно повысить удовлетворенность клиента работой системы.

Пути улучшения качества распознавания

Базовые soft-средства, используемые в системах распознавания, обычно позволяют выполнить настройку системы с тем, чтобы повысить надежность и достоверность распознавания. К сожалению, настройка системы не приводит к коренному улучшению распознавания, а позволяет лишь улучшить поведение системы в отношении одного из показателей качества за счет ухудшения другого.

Один из основных настраиваемых параметров – порог уверенности распознавания. Он представляет собой такое значение вероятности некоторой гипотезы, которое является пороговым для принятия решения о том, принять данную гипотезу или отвергнуть ее.

При выборе порога уверенности во внимание принимаются следующие соображения:

- если установить слишком высокий порог уверенности, то будут частыми случаи, когда слово или фраза распознаны верно, но отвергнуты модулем распознавания;
- если порог уверенности занижен, то будет много случаев, когда сказанная клиентом фраза распознана неверно, но модуль распознавания выдал свою гипотезу в качестве верной.

Рекомендуемый разработчиками базовых систем порог уверенности равен 45 (при шкале от 1 до 100). Считается, что при таком пороге обеспечивается достаточно уверенное распознавание с небольшим количеством ошибочного распознавания и отказов в распознавании.

Порог распознавания может быть задан как по отношению ко всему приложению, так и по отношению к конкретной грамматике, группе слов и кциальному слову.

Другой важный фактор, влияющий на качество распознавания, – сама грамматика. Понятие грамматики в системах распознавания речи отличается от принятого в лингвистике. Здесь под граммати-



кой понимается последовательность слов, которые модуль распознавания способен понять на данной фазе диалога. Грамматики могут быть простыми, содержащими перечни слов или групп слов, соединенных логическими союзами И или ИЛИ, но могут быть и сложными, включающими в себя другие грамматики или рекурсию.

Известно, что чем проще грамматика, тем лучше будет распознавание, однако не всегда это возможно обеспечить, хотя всегда следует к этому стремиться.

Каждой грамматике соответствует свой словарь, содержащий транскрипцию произношения всех слов грамматики. Этот словарь создается компилятором на основе базового словаря, а транскрипцию слов, отсутствующих в нем, разработчик должен написать сам. Хорошо известно, что следует избегать использования в одной грамматике слов, имеющих схожее произношение, но этого тоже не всегда удается достичь. Разработчик должен очень внимательно отнести к содержимому словарей, поскольку часто некорректное описание произношения приводит к серьезным ошибкам в распознавании.

Базовое матобеспечение содержит обычно несколько наиболее распространенных грамматик, которые рекомендуется использовать в приложениях, а также базовый словарь достаточно большого объема.

СОСТОЯНИЕ ДЕЛ В МИРЕ

В настоящее время на Западе уже сотни компаний внедрили у себя системы распознавания речи, чтобы автоматизировать обслуживание различного рода вызовов. Одними из первых применили у себя эту технологию авиакомпании, среди которых «American Airlines», «United Airlines» и др. Широко применяется распознавание речи банками, страховыми компаниями, правительственными учреждениями, операторами связи. Инвестиции в технологии распознавания речи составили в 2002 г. 695 млн. дол. США и ожидается, что они возрастут к концу 2004 г. до 1,4 млрд. дол. [9].

В нашей стране работы по распознаванию речи велись в течение многих лет, и достигнутые результаты были на очень высоком теоретическом уровне. Однако систем, пригодных для практического применения, создано не было. В настоящее время в России нет ни одной действующей системы массового обслуживания населения с распознаванием русской речи.

СИСТЕМА С ГОЛОСОВЫМ ИНТЕРФЕЙСОМ

В Институте проблем управления РАН создана система доступа к информации СМО «Сирена», которая демонстрирует возможности применения технологии распознавания речи в открытых системах массового обслуживания [10]. Любой человек с любого телефона может набрать нужный номер. Компьютер снимет трубку, ответит ему и задаст несколько вопросов, чтобы узнать, какая информация нужна позвонившему. Ответы распознаются, и на основании полученной информации компьютер формирует запрос в «Сирену». Затем ответ, полученный из «Сирены», озвучивается и сообщается клиенту. Система может распознавать числа, названия улиц, номера телефонов, цифровые коды.

В процессе разработки системы были созданы грамматики, фонетические словари, записаны звуковые файлы, построен диалог. Пример диалога приводится далее.

Система построена на компьютере Pentium 4, дополненном платой Dialogic, к которой подключены две телефонные линии. Операционная система – Windows NT. Применены платформы распознавания речи компаний «Nuance Communication» и «Philips».

Пример диалога с системой «Сирена»

Система приветствует клиента и предлагает выбрать одну из тем: месяцы, числа, расписание рейсов.

Клиент. Расписание рейсов.

Система. Назовите город отправления.

Клиент называет город.

Система (*для проверки произносит название города*). Назовите город назначения.

Клиент называет город.

Система (*повторяет город*). Назовите месяц вылета.

Клиент называет любой месяц.

Система (*повторяет месяц*). Назовите число месяца.

Клиент называет любое число в пределах от 1 до 31 (можно говорить «второе» или «второго»).

Система. Ваш запрос содержит: (*перечисляет параметры запроса*). Это правильно или нет?

Клиент говорит «нет» или «неправильно» или «нет, неправильно», если была ошибка, или «правильно», «да, правильно», если все верно.

В случае, если клиент говорит «нет»:

Система. Что вы хотите изменить: город отправления, город назначения, месяц вылета или число месяца?

Клиент (*например*). Число месяца.

И т. д.

Этот диалог используется далее для обращения в реальную базу данных действующей системы «Сирена». После получения ответа из «Сирены» система озвучивает его, называя количество выполняемых в данный день рейсов, номера рейсов, времена вылета и прилета, базовый тариф и другую информацию.

Эксперименты показали, что система вполне удовлетворительно распознает речь клиентов, а также демонстрирует высокую реактивность.

Из приведенного диалога ясно, что в этой разработке использовались грамматики, обеспечивающие распознавание часто употребляемых слов: чисел, названий месяцев, городов. Поскольку создан ряд и других общеупотребительных грамматик (номера телефонов, цифровые коды, улицы Москвы и др.), можно утверждать, что данная система достаточно универсальна и может найти применение в различных областях.

ЗАКЛЮЧЕНИЕ

Информационные системы давно уже вошли в повседневную жизнь человека. Информации становится все больше, и все более актуальной становится проблема доступа к ней. Современное состояние информационных технологий позволяет находить новые подходы к решению проблемы доступа. Одно из таких направлений – применение технологии распознавания речи, весьма перспективной и способствующей эволюции информационных систем в сторону открытых систем массового обслуживания.

ЛИТЕРАТУРА

1. Жожикашвили В.А., Жожикашвили А.В., Фархадов М.П. Принципы построения системы общения с использованием естественного языка в больших автоматизированных системах массового обслуживания нового поколения // Массовое обслуживание. Потоки, системы, сети: Сб. материалов 14 Белорус. зимней школы-семинара по теории массового обслуживания (BWWQT-98). – Минск, 1998. – С. 190–194.
2. Жожикашвили В.А., Жожикашвили А.В., Петухова Н.В. и др. Построение структуры человека-машинного интерфейса для автоматизированных систем массового обслуживания // Автоматизация и современные технологии. – 2000. – № 3.
3. Жожикашвили В.А., Билик Р.В. Петухова Н.В., Ребортович Б.И. Управление доходами в автоматизированных системах массового обслуживания. // Тр. Ин-та пробл. упр. РАН. – 1999. – Т. IV. – С. 27–35.
4. Зутлер И.А., Фархадов М.П. Стратегия управления сверхбронированием и сверхпродажами // Тр. Ин-та пробл. упр. РАН. – 2000. – Т. IX. – С. 54–58.
5. Зутлер И.А. Управление ограниченным ресурсом на конечном промежутке времени в автоматизированных системах массового обслуживания // Автоматика и телемеханика. – 2002. – № 2. – С. 179–184.
6. Жожикашвили В.А., Жожикашвили А.В., Петухова Н.В., Фархадов М.П. Применение распознавания речи в автоматизированных системах массового обслуживания // Автоматизация и современные технологии. – 2003 (в печати).
7. Росляков А.В., Самсонов М.Ю., Шибаева И.В. Центры обслуживания вызовов (Call Centre). – М.: Эко-Трендз, 2002.
8. Raskin J. The Human Interface: New Directions for Designing Interactive Systems. – Addison-Wesley: Pearson Education, Inc., 2000.
9. <http://www.datamonitor.com>
10. <http://www.ipu.ru/labs/lab17/frame17.htm>

☎ (095) 334-87-10



Читайте в следующем номере нашего журнала

- ❖ **Эпштейн В.Л.** Предвидимое будущее научных журналов (обзор)
- ❖ **Бабаян Р.Р., Морозов В.П.** Надо ли возрождать аналоговую технику?
- ❖ **Нижегородцев Р.М.** Логическое моделирование экономической динамики. Ч. I.
- ❖ **Лебедев В.Н.** Управление синхронизацией распределенных разнородных баз данных на основе организации трехэлементных очередей
- ❖ **Кульба В.В., Темкин В.М., Рывкин Д.Б.** Сбор и обработка данных в системах производственной экологической безопасности предприятий нефтегазовой промышленности

УДК 621.854:62-5

УПРАВЛЕНИЕ СВЕРТЫВАНИЕМ ДВУХМОДУЛЬНОЙ КОСМИЧЕСКОЙ ТРОСОВОЙ СИСТЕМЫ

Л.М. Калашников, Г.В. Малышев, А.П. Свотин

Московский авиационный институт (государственный технический университет), г. Москва

Рассмотрена задача свертывания космической тросовой системы путем управления натяжением троса в предположении, что система движется по круговой орбите, состоит из двух модулей, связанных невесомым и неупругим тросом и находится в состоянии либрации либо вертикального устойчивого равновесия. Предложена программа управления натяжением, предусматривающая наличие опорной траектории и многопараметрического управления относительно нее.

ВВЕДЕНИЕ

В отечественной и мировой космонавтике существуют проблемы, остающиеся втуне и якобы не влияющие на общий уровень техники, однако не менее существенные, чем, например, повышение удельной тяги двигателей или миниатюризация электронных устройств.

К таковым относится разработка космических тросовых систем (КТС), теоретические основы которых заложены около двадцати лет назад, в чем заслуга и российских механиков¹.

Управление динамикой развертывания и свертывания, разделением связанной системы двух и более тел на орбите с переменными начальными условиями по-новому решает энергетические проблемы движения каждого тела в пространстве и относительно центра масс. Использование электромагнитодинамических свойств троса-кабеля в гравитационном и магнитном полях позволяет создавать космические электрогенераторы и движители нового типа.

Обогащая арсенал конструктора, повышая энергетический потенциал систем процентов на десять и более без расхода рабочего тела и электроэнергии, тросовые технологии безусловно перспективны. Достаточно сказать, что станция «Мир», обслуживаемая 105-ю челночными аппаратами типа «Союз» и «Прогресс» и четырьмя аппаратами типа

«Шаттл» могла бы существовать бесконечно долго без расхода топлива и тенденции снижения орбиты, если бы применялась соответствующая тросовая технология спуска в атмосферу возвращаемых аппаратов. Каждый космический аппарат, сопутствующий отработавшей последней ступенью носителя, мог бы получать дополнительный импульс в несколько процентов от энергетики этой ступени.

Начиная с классических задач механики двойной связанной системы в гравитационном поле, разработаны прикладные тросовые технологии (развертывание и свертывание без потери натяжения связи, либрационные и ротационные режимы, орбитальные переходы концевых масс при разделении системы, способы их стабилизации при изменении точек подвеса), ориентированные на инженеров-механиков и конструкторов перспективных космических систем.

Наибольший интерес, с точки зрения реализации большинства перспективных проектов КТС, представляют задачи управления их развертыванием и свертыванием. Первые из них проработаны достаточно хорошо, о чем свидетельствуют свыше двадцати реализованных проектов. Вторые исследованы недостаточно и решение большинства из них основано на применении двигателей, установленных на одном из модулей, и информации о фазовых координатах модулей. Однако для простоты реализации и, как следствие, более высокой надежности предпочтительно управлять натяжением троса, используя информацию о длине, скорости движения и минимум информации о фазовых координатах модулей.

¹ Белецкий В.В., Левин Е.М. Динамика космических тросовых систем. — М.: Наука, 1990.



ЗАКОН УПРАВЛЕНИЯ СВЕРТЫВАНИЕМ

Исходная система уравнений, описывающая поведение невесомой неупругой тросовой системы с двумя концевыми массами, центр масс которой расположен на круговой орбите (рис. 1), имеет вид:

$$\ddot{l} = l(\dot{\alpha}^2 + 2\omega\dot{\alpha} + 3\omega^2 \sin^2 \alpha) - N$$

$$\ddot{\alpha} = -2\frac{\dot{l}}{l}(\omega + \dot{\alpha}) + 3\omega^2 \sin \alpha \cos \alpha,$$

где l – длина троса, α – угловое положение троса, m_1 и m_2 – массы модулей, $m^* = \frac{m_1 m_2}{m_1 + m_2}$ – приведенная масса системы, $N = \frac{F}{m^*}$, F – сила натяжения троса, ω – угловая скорость движения центра масс по орбите.

Линеаризованная система дифференциальных уравнений относительно произвольной программной траектории свертывания (развертывания) представляется в виде:

$$\Delta \ddot{l} = [\dot{\alpha}_{\text{пр}}^2 + 2\omega\dot{\alpha}_{\text{пр}} + 3\omega^2 \sin^2 \alpha_{\text{пр}}] \Delta l + 3\omega^2 \sin 2\alpha_{\text{пр}} \cdot l_{\text{пр}} \Delta \alpha + 2l_{\text{пр}} [\dot{\alpha}_{\text{пр}} + \omega] \Delta \dot{\alpha} - \Delta N$$

$$\Delta \alpha_{\text{пр}} = -2 \left[\Delta l - \frac{\dot{l}_{\text{пр}} \Delta l}{l_{\text{пр}}} \right] (\omega + \dot{\alpha}_{\text{пр}}) - 2 \frac{\dot{l}_{\text{пр}}}{l_{\text{пр}}} \Delta \dot{\alpha} + 3\omega^2 \cos 2\alpha_{\text{пр}} \cdot \Delta \alpha,$$

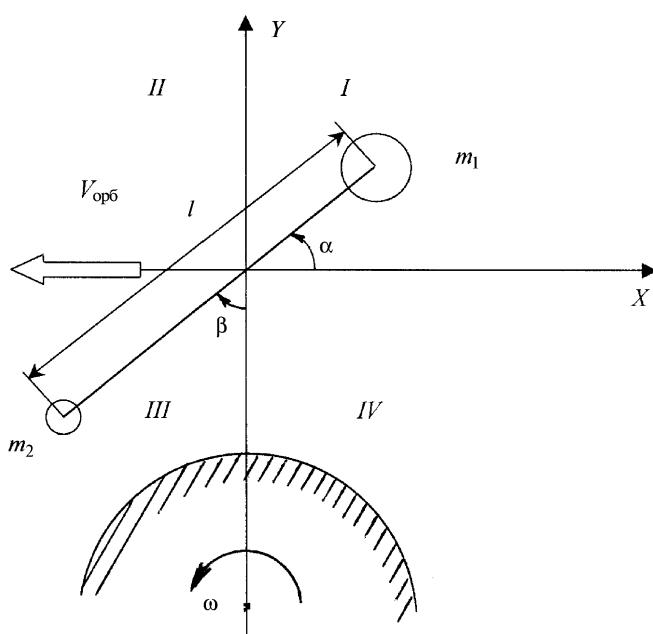


Рис. 1. Схема тросовой системы с двумя концевыми массами

где $\alpha_{\text{пр}}$ и $l_{\text{пр}}$ – координаты программной траектории, $\Delta l = l - l_{\text{пр}}$, $\Delta \alpha = \alpha - \alpha_{\text{пр}}$, $\Delta \dot{l} = \dot{l} - \dot{l}_{\text{пр}}$.

В ряде случаев вместо переменной $\Delta \alpha$ удобнее пользоваться параметром $\Delta x = l_{\text{пр}} \Delta \alpha$. Тогда связь производных $\Delta \dot{x}$ и $\Delta \dot{\alpha}$ записывается как

$$\Delta \dot{\alpha} = \frac{1}{l_{\text{пр}}} \left(\Delta \dot{x} - \frac{\dot{l}_{\text{пр}}}{l_{\text{пр}}} \Delta x \right),$$

а уравнения принимают вид:

$$\Delta \ddot{l} = [\dot{\alpha}_{\text{пр}}^2 + 2\omega\dot{\alpha}_{\text{пр}} + 3\omega^2 \sin^2 \alpha_{\text{пр}}] \Delta l + 3\omega^2 \sin 2\alpha_{\text{пр}} \Delta x + 2(\dot{\alpha}_{\text{пр}} + \omega) \left(\Delta \dot{x} - \frac{\dot{l}_{\text{пр}}}{l_{\text{пр}}} \Delta x \right) - \Delta N$$

$$\Delta \ddot{x} = -2 \left(\Delta l - \frac{\dot{l}_{\text{пр}} \Delta l}{l_{\text{пр}}} \right) (\omega + \dot{\alpha}_{\text{пр}}) + \left[\left(\frac{\dot{l}_{\text{пр}}}{l_{\text{пр}}} \right)^2 + \frac{d}{dt} \left(\frac{\dot{l}_{\text{пр}}}{l_{\text{пр}}} \right) + 3\omega^2 \cos 2\alpha_{\text{пр}} \right] \Delta x.$$

Для выбора управления натяжением троса, обеспечивающего устойчивый переходный процесс свертывания связки, предлагается программная траектория стягивания, представляющая собой прямолинейное движение концевых масс в начало координат. Данная траектория обеспечивается выполнением следующих условий:

$$\frac{\dot{l}_{\text{пр}}}{l_{\text{пр}}} = \frac{3}{4} \omega \sin 2\alpha_{\text{пр}},$$

$$\dot{\alpha}_{\text{пр}} = 0,$$

$$N_{\text{пр}} = 3\omega^2 l_{\text{пр}} \left(\sin^2 \alpha_{\text{пр}} - \frac{3}{16} \sin^2 2\alpha_{\text{пр}} \right),$$

где $N_{\text{пр}}$ – программное ускорение, вызванное силой натяжения.

В зависимости от знака отношения $\dot{l}_{\text{пр}}/l_{\text{пр}}$, когда осуществляется программное развертывание системы, угол $\alpha_{\text{пр}}$ лежит в I или III квадрантах (см. рис. 1). В случае программного свертывания угол $\alpha_{\text{пр}}$ находится во II или IV квадрантах.

В случае свертывания закон управления, удовлетворяющий условию устойчивого переходного процесса, должен содержать не менее трех параметров (l , \dot{l} и α), в отличие от процесса развертывания, когда можно обойтись двумя параметрами l и \dot{l} .

Программа натяжения троса: $N = N_{\text{пр}} + \Delta N$; $\Delta N = \omega^2 a \Delta l + \omega b \Delta l + \omega^2 c \Delta x$, где a , b и c – коэффициенты усиления, $a = N_{\text{пр}}/l_{\text{пр}}$. Коэффициенты b и c выбираются из условия устойчивости переходного процесса.



Для произвольного процесса свертывания система уравнений относительно невязок Δl и Δx окончательно записывается в виде:

$$\begin{aligned}\ddot{\Delta l} &= A\Delta l + B\dot{\Delta l} + C\Delta x + D\dot{\Delta x} \\ \ddot{\Delta x} &= E\Delta l + F\dot{\Delta l} + G\Delta x,\end{aligned}$$

где

$$\begin{aligned}A &= -\frac{9}{16}\omega^2 \sin^2 2\alpha_{\text{пп}}, \quad B = -b\omega, \\ C &= -\left(c - \frac{3}{2}\sin 2\alpha_{\text{пп}}\right)\omega^2, \\ D &= 2\omega, \quad E = \frac{3}{2}\omega^2 \sin 2\alpha_{\text{пп}}, \quad F = -2\omega, \\ G &= \omega^2 \left(3\cos 2\alpha_{\text{пп}} + \frac{9}{16}\sin^2 2\alpha_{\text{пп}}\right).\end{aligned}$$

Характеристическое уравнение данной системы дифференциальных уравнений имеет вид:

$$\lambda^4 - B\lambda^3 - \lambda^2[G + FD + A] - \lambda[FC - GB + DE] - (EC - AG) = 0.$$

Для устойчивости ее решения необходимы одинаковые знаки коэффициентов при степенях λ^n . Отсюда приходим к следующим условиям:

$$\begin{aligned}b &> 0; 4 - \left(\frac{9}{8}\sin^2 2\alpha_{\text{пп}} + 3\cos 2\alpha_{\text{пп}}\right) > 0; \\ c + b\left(\frac{3}{2}\cos 2\alpha_{\text{пп}} + \frac{9}{32}\sin^2 2\alpha_{\text{пп}}\right) &< 0; \\ c - \frac{3}{8}\sin 2\alpha_{\text{пп}} \left[4 - \left(3\cos 2\alpha_{\text{пп}} + \frac{9}{16}\sin^2 2\alpha_{\text{пп}}\right)\right] &< 0.\end{aligned}$$

Второе из этих условий выполняется для произвольных $\alpha_{\text{пп}}$. Наиболее подходящим в смысле быстродействия является значение $\alpha_{\text{пп}} = -\pi/4 + k\pi$, где $k = 0$ при свертывании в IV квадранте, $k = 1$ при свертывании во II квадранте.

При этом $i = -\frac{3}{4}\omega l$; $N_{\text{пп}} = \frac{15}{16}\omega^2$.

В данном случае сила управления натяжением, отнесенная к единице приведенной массы, имеет вид:

$$N = \frac{15}{16}\omega^2 + b\left(i + \frac{3}{4}\omega l\right) + c\left(\frac{\pi}{4} + \alpha - k\pi\right).$$

Коэффициенты b и c выбираются из условий $b > 0$, $c < -0,281b$, $c < -1,289$.

Расчеты показывают, что для обеспечения устойчивости $b \approx 4$, $c = -2...-3$.

Однако полученный закон управления при начальных фазовых состояниях, лежащих вдали от программной траектории, не всегда обеспечивает устойчивый переход в начало координат (стягивающие массы закручиваются вокруг начала координат).

По мере приближения к значению $|\alpha_{\text{пп}}| = \pi/2$ степень устойчивости возрастает, но резко увеличивается время сближения. Компромиссным вариан-

том является переход со значением $\alpha_{\text{пп}} = -\pi/3 + k\pi$. При этом $i = -0,65l\omega$, $N_{\text{пп}} = 1,83\omega^2l$, $b > 0$, $c < 0,54b$, $c < -1,65$.

Окончательное управление подчинено закону:

$$\begin{aligned}N &= 1,83\omega^2l + 4\omega(i + 0,65\omega l) - \\ &\quad - 2,8\omega^2(\pi/3 + \alpha - k\pi)l.\end{aligned}$$

Близким к предельному можно считать случай, когда $\alpha_{\text{пп}} = -5\pi/12 + k\pi$. При этом

$$\begin{aligned}i &= -0,37\omega l, \quad N_{\text{пп}} = 2,66\omega^2l, \quad b > 0, \quad c < -1,005, \\ N &= 2,66\omega^2l + 4\omega(i + 0,37\omega l) - \\ &\quad - 2,8\omega^2(5\pi/12 + \alpha - k\pi)\omega^2l.\end{aligned}$$

РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

Конструктивные параметры системы свертывания базируются на параметрическом анализе реальной связки двух концевых масс модулей, первоначально либрующей относительно вертикали с заданными углами β ($0 \leq \beta \leq 65^\circ$), центр масс которой движется по круговой орбите с угловой скоростью ω (см. рис. 1).

Интегрируются уравнения движения концевых модулей в системе координат, начало которой совпадает с центром масс. Варьируется угол либрации, опорный угол программы $45^\circ \leq |\alpha_{\text{пп}}| \leq 75^\circ$ и коэффициенты усиления в программе натяжения троса: $b = -4...-2$; $c = 2...4$. Этот диапазон получен при дальнейшем исследовании устойчивости системы по достаточному условию на базе критерия Руяса-Гурвица.

Анализируются динамические характеристики модельной задачи: субспутник массой $m_2 = 150$ кг отделяется от транспортного корабля «Прогресс» массой $m_1 = 7250$ кг на рабочей орбите (высотой ~ 350 км). Тросовая связка вертикализируется (например, по закону Раппа). Затем трос сматывается лебедкой транспортного корабля (рис. 2) при опорном угле программы $\alpha_{\text{пп}} = -60^\circ$. Представлена траектория концевой массы — две полуволны в IV квадранте. Аналогичное по характеру движение осуществляется и тяжелый модуль во II квадранте с масштабом координат, равным соотношению масс $M = 1:(7250/150)$. Процесс продолжается около 8 000 с с практически монотонным убыванием усилия в тросе от 5,5 до 0 Н при монотонном убывании длины троса между сближающимися модулями. Последнее упрощает конструкцию лебедки, осуществляющей намотку без реверсивных режимов.

Рассмотрена динамика свертывания системы при начальном либрационном движении с амплитудой $\pm 45^\circ$ и $\pm 65^\circ$:

- из крайнего левого положения (рис. 3) при амплитуде либрации $\pm 65^\circ$;

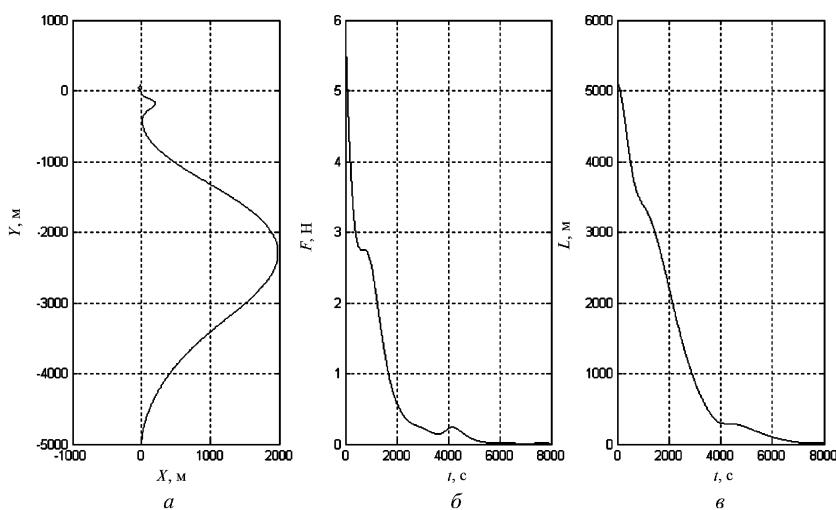


Рис. 2. Свертывание из положения вертикального равновесия:
а – траектория движения концевых масс; б – натяжение троса; в – длина троса

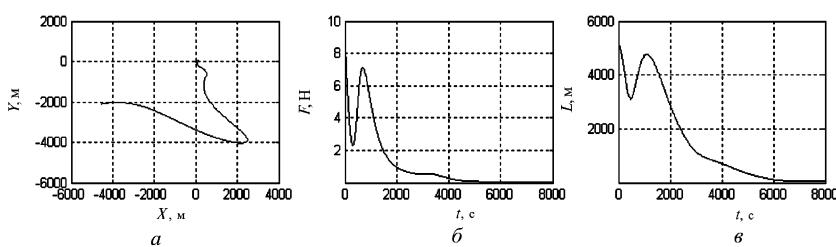


Рис. 3. Свертывание из крайнего левого положения при максимально возможном угле либрации:

а – траектория движения концевых масс; б – натяжение троса; в – длина троса

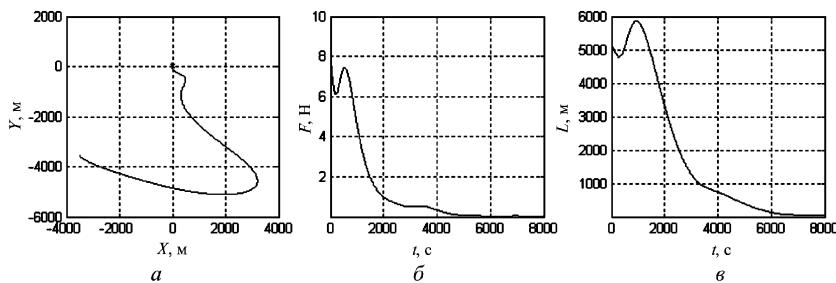


Рис. 4. Стягивание начинается в момент прохождения угла -135° против хода орбитального движения с максимальным углом либрации:

а – траектория движения концевых масс; б – натяжение троса; в – длина троса

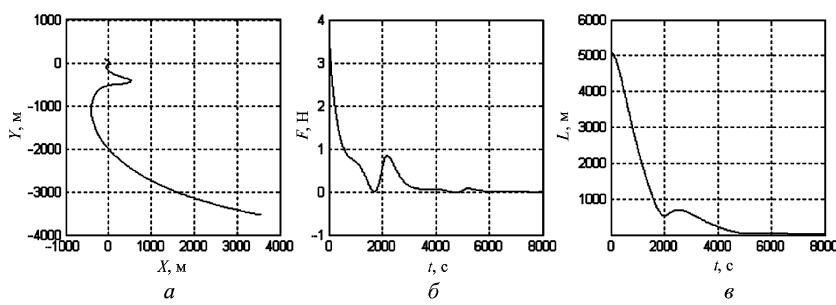


Рис. 5. Стягивание из крайнего правого положения при угле либрации 45° :

а – траектория движения концевых масс; б – натяжение троса; в – длина троса

- из положения $\alpha = -135^\circ$ против хода орбитального движения (рис. 4) при меньшей амплитуде либрации ($\pm 45^\circ$);
- при стягивании из крайнего правого положения (рис. 5).

С увеличением программного угла $\alpha_{\text{пр}}$ увеличивается время процесса стягивания, однако понижается тенденция ввода системы в ротацию на заключительном этапе.

Уменьшение угла либрации делает процесс более мягким (по амплитуде колебаний усилия в трофе и его длины).

Предпочтительно начинать стягивание, когда нижняя масса движется в III квадранте.

В большинстве случаев при начальном удалении масс около 5 000 м длина трофе в процессе движения не превышает этого значения.

ЗАКЛЮЧЕНИЕ

Предложен трехпараметрический закон управления свертыванием двухмодульной космической троевой системой только за счет управления натяжением трофе. Задача решалась в предположении, что центр масс системы движется по круговой орбите, трофе невесомый и неупругий, система находится в состоянии либрации либо вертикального равновесия.

В качестве измеряемых параметров были приняты текущая длина, скорость смотки трофе, а также угол между визирной линией, связывающей концевые массы, и вектором, обратным вектору скорости орбитального движения центра масс связки.

Результаты моделирования показали хорошую работу полученного закона управления натяжением в широком диапазоне углов либрации. В случае стягивания из состояния вертикального равновесия полностью отсутствует участок вытравливания трофе, что позволяет значительно упростить весь механизм свертывания.

Стягивание за счет только управления натяжением трофе позволяет значительно упростить и повысить надежность систем развертывания (свертывания) двухмодульных троевых систем, расширяя тем самым области их возможного применения.

☎ (095) 158-42-61

E-mail: malyshev@mail.ru

УДК 629.78 (092)

МИХАИЛ АЛЕКСАНДРОВИЧ ГАВРИЛОВ

(к 100-летию со дня рождения)

В ноябре 2003 года исполнилось 100 лет со дня рождения выдающегося ученого, яркой личности, удивительного человека, доктора технических наук, профессора, члена-корреспондента АН СССР Михаила Александровича Гаврилова. Чтобы оценить роль и место М.А. Гаврилова в науке управления, позволим себе некоторый экскурс в историю становления автоматики и телемеханики в нашем Отечестве.

Несмотря на то, что автоматические устройства управления применяются в технике с незапамятных времен, устройства автоматического регулирования были «обласканы» научными исследованиями в значительно большей степени, чем устройства релейного действия. Действительно, регуляторы Дж. Уатта, И.И. Ползунова, братьев В. и Э. Сименс через несколько десятков лет после их изобретения привлекают внимание крупнейших физиков, механиков и математиков (П.Л. Чебышева, Дж. Максвелла, И.А. Вышнеградского и многих других). Вместе с тем устройства релейного действия, например, золотник, переключающий струю пара высокого давления от полости к полости, без которого паровая машина так же неосуществима, не удостаиваются внимания исследователей как элементы динамической системы. В принципе в этом и не было необходимости, поскольку реализуемая этими устройствами функция была тривиальна. В начале XX века, с развитием телефонной связи, релейная техника занимает значительные позиции в мире технических систем, и в их проектировании и эксплуатации проявляются значительные проблемы. Не случайно физик П. Эренфест в 1910 г. в предисловии к книге Л. Кутюра «Алгебра логики» указывает на потенциальную возможность этого раздела математики в анализе схем телефонии вместо «пробования на графике». Однако эта ремарка осталась незамеченной. Но уже в 1920-х годах в технических системах появляется объект, в управлении которым релейные устройства – устройства телемеханики – и, что важно, логические методы в принятии управлений решений становятся преобладающими, а ошибки в управлении влекут за собой огромные потери. Этот объект – энергосистемы. Первые из них появляются в США, затем в Европе и, как правило, ограничиваются рамками одной производящей компании. В Советской



России по мере воплощения плана ГОЭРЛО создаются крупные энергосети, использующие преимущества свойственной социализму централизации в управлении хозяйством. И, несмотря на общую техническую отсталость России, энергосети по сложности и масштабам были вполне соизмеримы с крупнейшими энергосистемами США и Европы.

В это время в Москве и Подмосковье вводится в действие огромное количество вновь сооружаемых энергетических объектов (электростанций, энергосетей с резервными подсетями, коммутирующими подстанциями, с понижающими подстанциями и питающимися от них потребителями с различной степенью важности – от рядовых коммунальных потребителей до оборонных объектов и вплоть до Кремля и его обитателей).

Во главе всего этого сложного хозяйства стоит диспетчер, который в течение нескольких секунд должен (только лишь по тенденциям изменения частоты в некоторых подсетях) предугадать, что будет дальше, и отключить часть (менее важных) потребителей, чтобы энергосеть не развалилась, и при первых же признаках появления резерва мощности включить этих потребителей, пока они еще не успели нажаловаться. При этом в голове нужно держать десятки, а возможно, и сотни вариантов отключения и помнить обратные цепочки включения.

И вот этим «дирижером» в центральной диспетчерской МОГЭС был молодой человек, среднего роста, интеллигентной внешности, коренным москвичем, совсем не пролетарского происхождения, вчерашний выпускник МВТУ – Михаил Александрович Гаврилов.

Михаил Александрович Гаврилов родился 11 ноября 1903 г. в Москве. Отец его Александр Митрофанович был артистом балета Большого театра, мать Елизавета Сергеевна – зубной врач, здесь же, при Большом.

В семье была дружная и теплая атмосфера, не редкая в среде творческой интеллигенции тех лет. Часто музицируют, Миша тонко чувствует музыку и, как сам с гордостью вспоминал, «был лучшим перевертышателем нот в Москве». Хотя в семье техникой никто не занимался, по словам Нины Александровны – сестры Михаила Александровича, он буквально с



двуухлетнего возраста проявлял интерес к технике. Михаил Александрович очень любил своих близких и дорожил отношениями с ними. Забегая несколько вперед, отметим, что лучшую свою книгу «Теория релейно-контактных схем» он посвятил памяти отца.

Михаил Александрович получил среднее образование, окончив в 1920 г. 109-ю трудовую школу в Москве. Еще будучи учащимся, в 1918 г., он начал трудовую деятельность.

Вот как он сам пишет об этом в своей автобиографии «По найму я работаю с 1918 года, когда еще до окончания трудовой школы я принужден был поступить на работу в качестве конторщика в продотдел Московского Совета РК и К депутатов». Как он вспоминал, «театры не работали, у отца почти не было заработка, артисты разъехались, и у мамы — зубного врача при театре — работы тоже практически не стало», и он взял на себя часть тягот по обеспечению семьи продуктами и всем необходимым.

Постепенно жизнь налаживается, и в 1920 г. Михаил Александрович становится курсантом Высших железнодорожных курсов. В 1921 г. Михаил Александрович перевелся в Московское высшее техническое училище, которое окончил в декабре 1925 г. по электротехническому факультету, получив звание инженера-электрика.

После окончания МВТУ Михаил Александрович с 1926 по 1935 г. работал в Правлении МОГЭС (затем Управление Мосэнерго), где занимал ряд должностей, начиная с инженера технического отдела и дежурного диспетчера до начальника цеха экспериментальной лаборатории.

М.А. Гаврилов очень гордился своей работой в Мосэнерго, здесь он впервые столкнулся с проблемами телемеханики и диспетчеризации. Он считался лучшим диспетчером. В 1975 г. отмечалось 50-летие диспетчерской службы Мосэнерго, Михаил Александрович был приглашен на торжества, был в президиуме торжественного заседания, получил памятные награды и потом с удовольствием рассказывал, что его вспоминали как лучшего и одного из первых диспетчеров МОГЭС. Выше уже говорилось о работе диспетчера, но напомним, что основным инструментом диспетчерской тех лет были журналы распоряжений, рапорты от дежурных специалистов станций и подстанций и собственная голова, а основным техническим средством — телефон. Здесь М.А. Гаврилов впервые проявил свои способности к комбинаторике и логике, способность мгновенно оценить ситуацию, предугадать сценарий развития, оценить последствия и в реальном масштабе времени принять решение и добиться его неукоснительного выполнения. Работая сменным инженером в диспетчерской, он на практике остро осознал потребность в устройствах автоматики и телемеханики. Этот период его жизни описан в первых главах книги писателя Юрия Вебера «Когда приходит ответ». Как известно, Михаил Александрович Гаврилов был прообразом главного героя книги инженера Григория Мартынова.

Работая в системе Мосэнерго в экспериментальной лаборатории, Михаил Александрович начал заниматься научно-исследовательской деятельностью. Заметим, что отраслевых институтов еще нет, в Академии наук занимаются в основном высокой наукой, а проблемами техники и энергетики в частности за-

нимаются в основном инженеры в «заводских лабораториях». Экспериментальная лаборатория МОГЭС была одной из таких «заводских лабораторий», а фактически мозговым центром нарождающейся советской энергетики. Здесь ведут собственные технические разработки, проводят экспертизу предложений по закупке техники за рубежом, готовят методические материалы для региональных технических служб энергосистем, организуют всевозможные курсы для ИТР, готовят обзорные материалы по состоянию советских и зарубежных энергетических объектов и сетей.

Михаил Александрович Гаврилов разрабатывал регламенты эксплуатационных режимов больших энергосистем. В 1932 г. он создал первое отечественное устройство телеуправления-телесигнализации для энергосистем. В течение нескольких десятилетий он систематически публиковал статьи и книги по проблемам телемеханики.

Первые публикации М.А. Гаврилова по телемеханике относятся к 1928—1935 гг. Он исследовал вопросы применения телемеханики в энергосистемах, изучал и обобщал имеющийся опыт в области телеуправления и телеметрии, подходя к нему критически, выполняя новые разработки. Начиная с 1933 г. М.А. Гаврилов создает методы проектирования и наладки телемеханических устройств. Задачи проектирования в дальнейшем послужили базой для развития теории релейных устройств.

Для энергетического хозяйства страны телемеханика представляла собой новую малоизученную область, недоставало специалистов. М.А. Гаврилов по поручению Мосэнерго занимался систематизацией сведений из этой области. Для институтов и комиссий АН СССР составлял и публиковал аналитические обзоры по принципам построения устройств телемеханики и дистанционного диспетчерского управления производством, изучал состояние промышленной телемеханики и освещал этапы и тенденции ее развития.

Еще в Мосэнерго Михаил Александрович начал и педагогическую деятельность: сначала выступая с лекциями на различных курсах повышения квалификации инженеров, а затем в МЭИ в качестве ассистента и доцента.

В 1934 г. он основал в МЭИ кафедру автоматики и телемеханики (первую в Москве) и был ее первым заведующим, а в 1935 г. перешел туда на постоянную работу.

Одна из первых студенток по кафедре автоматики и телемеханики МЭИ М.И. Карлинская вспоминает: «Он пришел к нам, когда мы закончили курс общетехнического факультета, и встал задача выбора специализации в 1936 г. Он стал рассказывать про телемеханику. У него была лаборатория в Мосэнерго. После его рассказа о новой специальности «телемеханика» к нему на специализацию записалось 22 очника и 22 заочника, а когда он начал читать лекции — это был ужас, он читал так быстро и сумбурно, что мы не давали ему покоя своими вопросами. На экзамене вся группа получила двойки. Летом вышла его книжка, мы готовились по ней к экзамену и все получили 4 и 5. Это была прекрасная книга — мы поняли, что такое телемеханика».

М.А. Гаврилов практически всю свою активную научную жизнь преподавал. К работе он активно



привлекал своих сотрудников, особенно молодежь, тем самым предоставляя возможность как проявить себя, так и поддержать материально. Он очень бережно относился к своим талантливым ученикам. Так, М.И. Карлинская, одна из лучших студенток Михаила Александровича, после окончания МЭИ по его рекомендации была распределена на работу в систему Мосэнерго, где он продолжал консультировать ее и всю команду своих вчерашних студентов. Он фактически руководил их работой при создании первой в Москве системы телевещания освещения города, которая затем проработала 26 лет, выдигаясь на Сталинскую премию. Затем М.И. Карлинская стала его первой защитившейся аспиранткой.

В 1934 г. для подготовки подраздела «Автоматика и телемеханика» в раздел по развитию науки в плане очередной пятилетки при АН СССР был организован общественный Комитет по автоматике, телемеханике и диспетчеризации. Комитет был создан по решению объединенного заседания по диспетчерскому управлению в народном хозяйстве, проведенного в 1934 г. под председательством вице-президента АН СССР, академика Г.М. Кржижановского. В этот комитет Михаил Александрович был включен по рекомендации Главэнерго и готовил раздел по диспетчеризации. Комитет подготовил аргументированный материал, хорошо себя зарекомендовал, и было принято решение о его сохранении вначале при ВСНITO, а затем в виде подразделения Академии наук. Так был в 1935 г. создан Комитет (затем преобразованный в Комиссию) по телемеханике и автоматике – КТА. М.А. Гаврилов был приглашен работать в него вначале вне штата, а затем уже на штатной основе, где возглавил лабораторию по дискретной телемеханике. Он активно участвует в подготовке обоснований необходимости создания Института автоматики и телемеханики. Комиссия в 1939 г. преобразуется в Институт автоматики и телемеханики (ИАТ) и, таким образом, Михаил Александрович является одним из его «отцов-основателей». Позднее ИАТ был переименован в Институт проблем управления, в котором Михаил Александрович трудился до последнего дня своей жизни.

Итогом десяти лет работы в телемеханике было присуждение в 1938 г. М.А. Гаврилову без защиты диссертации, по совокупности опубликованных работ, на заседании Ученого совета Ленинградского индустриального института степени кандидата технических наук и на том же заседании ему было предложено по опубликованным работам представить докторскую диссертацию. Это было официальным признанием его вклада в телемеханику.

Не будем характеризовать основные результаты Михаила Александровича в области телемеханики, это хорошо сделал его ученик В.А. Жожикашвили в своей статье в сборнике, посвященном памяти М.А. Гаврилова (1982 г.).

Однако заметим, что к этому времени (1938 г.) он опубликовал уже более 40 научных работ по телемеханике, три монографии, среди них одна из первых в Советском Союзе монографий по телемеханике «Телемеханизация диспетчерского управления энергетическими системами» (1938 г.). Он публикует ставшие широко известными аналитические обзоры по состоянию работ в области телемеханики, выступает на всесоюзных совещаниях с проблемными докладами.

После образования Института автоматики и телемеханики АН СССР (1939 г.) он возглавил коллектив лаборатории, занимавшейся в то время проблемами телеуправления и выполнившей ряд практических работ по автоматизации различных производственных процессов.

М.А. Гаврилов занимается телемеханикой еще более 20 лет. Ряд новых разработок был внедрен в различных отраслях народного хозяйства страны. Это системы и устройства телемеханики и релейной автоматики для:

- сверхмощных энергосистем;
- нефте- и газопроводов;
- мелиоративного хозяйства;
- угольных шахт;
- кранового и транспортного хозяйств предприятий;
- сооружений подземной газификации угля;
- водопроводных станций;
- метрополитена и железнодорожных сортировочных горок.

Поддерживая непрерывный контакт с широким кругом специалистов-практиков, через систему преподготовки специалистов и сложившиеся договорные отношения, Михаил Александрович оказывает ощущимое влияние на образ мыслей специалистов и на техническую политику соответствующих организаций в области телемеханизации и автоматизации производства, а его идеи находят конкретные воплощения.

Однако справедливости ради отметим, что пути распространения его идей не были гладкими, предлагаемые им направления исследований и разработок пробивали себе дорогу в полемике с другими видными специалистами, которым представлялись более правильными иные направления. Полемика возникала, главным образом, по принципиальным вопросам:

- какая теоретическая научная проблематика вытекает из конкретных инженерных задач телемеханизации в народном хозяйстве;
- каким инженерным решениям следует отдать предпочтение для обеспечения более полного соответствия имеющемуся и ожидаемому уровню развития техники (например, при выборе способов уплотнения каналов, видов модуляции сигналов).

Во многих случаях время и практика подтвердили правоту М.А. Гаврилова в этих спорах.

С 1948 г. в работах М.А. Гаврилова выкристаллизовалось одно из важнейших для телемеханики направлений исследований: комбинированное использование разнообразных импульсных признаков в телемеханических сигналах, создание соответствующей аппаратуры, обеспечивающей высокую информационную емкость сигналов и надежную защиту от искаажений. В дальнейшем это направление привело к появлению теории построения телемеханических сигналов, базирующейся на принципах помехоустойчивого кодирования. Аналогичный подход к рассмотрению неисправностей дискретных устройств привел к разработке теории и методов построения устройств с заданной степенью безотказности, обеспечивающей автоматической коррекцией сигналов, поступающих с отказавших элементов устройства.

Итог своих работ по телемеханике он подводит в докладе на сессии АН СССР в 1957 г. «Основные научные проблемы, вытекающие из задач телемеханизации в народном хозяйстве».

Позднее разнообразие проводимых его лабораторий исследований привело к выделению новых научных направлений, вокруг которых группировались ученые, и в результате в составе Института проблем управления образовались пять новых самостоятельных лабораторий:

- лаборатория «Системы массового обслуживания», руководимая д-ром техн. наук, профессором В.А. Жожикашвили (первым из его ИАТ'овских защитившихся аспирантов);
- лаборатория «Техническая диагностика», долгое время руководимая д-ром техн. наук, проф., членом-корреспондентом РАН П.П. Пархоменко;
- лаборатория «Параллельные вычислительные системы», руководимая д-ром техн. наук, проф., академиком Грузинской АН И.В. Прангишвили (ныне директор ИПУ);
- лаборатория «Методы управления дискретными процессами», руководимая д-ром техн. наук, профессором О.П. Кузнецовым;
- лаборатория (ныне сектор) «Автоматизация проектирования дискретных управляющих устройств», руководимая д-ром техн. наук, профессором В.В. Девятковым.

С переходом на научно-исследовательскую работу Михаил Александрович не прекращает педагогическую деятельность. С 1936 по 1949 г. он заведовал кафедрой автоматики и телемеханики Заочного политехнического института, организовав заочное обучение по этой специальности. В 1947 г. он организовал кафедру автоматики и телемеханики и стал ею заведовать (в течение 15 лет) в новом Всесоюзном заочном энергетическом институте.

Работая в области телемеханики, Михаил Александрович особое внимание уделял вопросам проектирования различных устройств, а, как известно, основными устройствами диспетчерского управления и всей телемеханики того времени были релейные устройства. Основной методологический прием тех лет в создании релейных устройств заключался в изобретении устройства (придумывании схемы, как говорил Михаил Александрович), а затем в его совершенствовании и исследовании. При этом основные исследования и дискуссии велись вокруг способов функционирования систем телемеханики: с разделением времени или с разделением частот. Михаила Александровича интересовали совсем другие вопросы: он все время пытался понять, почему устройства разной структуры, следовательно, разной проводимости, работают одинаково?

Напомним, что первыми автоматчиками и телемеханиками были инженеры-электрики. Как известно, основные исследования цепей в электротехнике основываются на изучении динамики потока электронов (тока). При этом электрическая цепь рассматривается как некоторый двухполюсник, соответственно, с входом в одном полюсе и выходом в другом. Релейное устройство с точки зрения электрической проводимости вне зависимости от числа контактов также является двухполюсником. Однако динамика тока ничего не проясняет в его функционировании. Более того, в некоторых случаях устройства с различным числом контактных элементов с разной проводимостью одинаково реагируют на одни и те же переключения контактных элементов.

Возвращаясь к сравнению устройств автоматического регулирования и релейных устройств, видим, что последним опять не повезло. В одной из своих публикаций по теории автоматического регулирования А.М. Летов отмечал, что теории автоматического регулирования очень повезло, поскольку математика XIX века создала прекрасную математическую теорию, благодаря успехам которой были решены основные задачи автоматического регулирования. Что же касается дискретной техники, то все попытки воспользоваться известным математическим аппаратом не привели к успеху.

И вот в 1938 г. М.А. Гаврилов случайно оказывается на семинаре Московского математического общества, на котором выступал физик из МГУ В.И. Шестаков с докладом о применении алгебры Буля для описания структуры релейных схем. Это посещение красиво описано в статье Д.А. Поспелова о Школе М.А. Гаврилова. Сам Михаил Александрович работу В.И. Шестакова считал революционной, но ее значение, как он полагал, не было понято даже автором. Все рассматривали релейно-контактные схемы с точки зрения проводимости (даже В.И. Шестаков и К. Шенон), а исчисление высказываний (алгебра логики) применялось исключительно для описания структуры. А Михаил Александрович в то время контактные элементы классифицирует как приемные и исполнительные (читай – входные и выходные), т.е. рассматривается движение не электрического тока, а информации от входов к выходам, что совсем не совпадает с движением тока.

Второе обстоятельство, послужившее тормозом в принятии предлагаемых методов проектирования релейных устройств, это отсутствие понятий, отображающих реальное время в функционировании схемы. Михаил Александрович как практический блестящий проектировщик релейно-контактных схем и не менее блестящий аналитик понимает, что в таблице истинности и формулах алгебры логики теряется последовательность срабатывания элементов. В то время как любой инженер-проектировщик пользуется последовательностью срабатывания как мощнейшим инструментом в интуитивном синтезе (придумывании схемы).

Напомним, что это тридцатые годы. Еще нет работ по конечным автоматам, выполненных С. Клини, Дж. Нейманом и Э. Муром и вошедших в знаменитый сборник «Автоматы» (1956 г.); еще и нет работы У. Маккаллока и У. Питса (1943 г.) по модели сети нейронов. Это в модели конечного автомата появляется дискретное время, как заменитель реального. А Михаил Александрович уже знает про такты, но вот проблема: как их связать с таблицей истинности, которой безразлично, что в какой последовательности срабатывает?

В начале сороковых годов М.А. Гаврилов задался целью создать научную методику проектирования релейно-контактных схем. Анализируя развитие методов создания релейно-контактных схем, М.А. Гаврилов писал: «Первые существенные достижения по пути развития научного обоснования методики построения схем были получены только тогда, когда был найден математический аппарат, отображающий соотношения, существующие в этих схемах».



Таким аппаратом стала алгебра логики. Исходя из работ В.И. Шестакова и К. Шеннона, М.А. Гаврилов создал стройную теорию анализа и синтеза одно- и многотактных релейно-контактных схем. За первой его работой (1943 г.) последовало много статей этого направления, которые были положены в основу докторской диссертации (1946 г.).

В первых работах упор делался на создание теории (описание и построение) многотактных (последовательностных) схем – релейных устройств с памятью. Михаил Александрович предложил язык «таблиц включений», по которым можно было получить структурную формулу многотактных схем. Он разработал методику преобразования не только параллельно-последовательных, но и мостиковых схем (так называемых схем класса Н), как с релейно-контактными элементами общего вида, так и со специализированными элементами (искателями, поляризованными и амплитудными реле и т. п.). О защите докторской диссертации Михаила Александровича и по сей день ходят легенды. По рассказу М.И. Карлинской, она длилась 8 часов, выступали философ С.В. Яновская и логик С.П. Новиков, была группа недоброжелателей – инженеров-практиков из Ленинграда, но чашу весов в пользу Михаила Александровича склонил академик А.И. Берг. Он сказал: «Мы присутствуем при величайшем открытии современности, которое совершил революцию в технике». После этого было голосование, оказалось всего два или три голоса «против», а все остальные – «за».

Все результаты исследований, представленные в диссертации, были им систематизированы и оформлены в виде монографии «Теория релейно-контактных схем», выпущенной в свет в 1950 г. издательством АН СССР.

Об этой книге академик В.С. Кулебакин отзывался следующим образом: «Книга доктора технических наук М.А. Гаврилова является первым фундаментальным трудом, в котором на базе практических достижений и теоретических разработок как самого автора, так и других лиц созданы научные основы построения релейно-контактных схем».

Сам Михаил Александрович в предисловии к книге писал: «Автор далек от мысли, что в настоящем его труде дано исчерпывающее решение основных вопросов теории релейно-контактных схем. Однако даже в том объеме, в котором она изложена в настоящей книге, эта теория, как показал опыт практического ее применения в некоторых проектных и научно-исследовательских организациях, уже приносит существенный эффект как в отношении сокращения времени, необходимого для разработки схем, так и в отношении получения более совершенных решений. Внедрение теории релейно-контактных схем в практику несомненно выдвигает новые вопросы и вызывает необходимость в ее дальнейшем развитии».

Опубликование монографии привлекло внимание широких кругов инженеров и исследователей к вопросам этой теории и послужило толчком к ее дальнейшему развитию. Книга была переведена на ряд иностранных языков.

В середине 1950-х гг. М.А. Гавриловым были разработаны методы минимизации мостиковых контактных и контактно-вентильных схем, а также метод минимизации булевых функций, получивший из-

вестность как «метод проб М.А. Гаврилова». В связи с большой трудоемкостью минимизации возникла необходимость в разработке методов приближенной минимизации, годных для инженерной практики. М.А. Гаврилов предложил общий подход к построению методов синтеза такого типа, получивший название направленного поиска минимальных реализаций. За развитие этой теории, методов расчета и принципов построения релейных схем в 1958 г. Президиум АН СССР присудил М.А. Гаврилову премию им. П.М. Яблочкива.

В 1963 г. Михаил Александрович избирается членом-корреспондентом АН СССР по теории управления.

Работы М.А. Гаврилова 1960–1970-х гг. были посвящены развитию метода направленного поиска применительно к сложным базисам логических элементов: мажоритарных, пороговых и элементов с произвольной структурой, однородных сред и др., а также применительно к сложным формам задания дискретных устройств большой размерности – интервальным и скобочным формам.

Важный вклад в теорию дискретных автоматов составили предложенные М.А. Гавриловым формулировка проблемы полноты и непротиворечивости описания поведения релейных устройств, методы минимизации элементов памяти, блочный подход к описанию и синтезу автоматов.

М.А. Гаврилов ввел операции композиции над таблицами переходов, позволяющие описывать различные виды взаимодействия автоматов и устанавливать эквивалентность автомата некоторой сети автоматов. Тем самым был сформулирован подход к блочному синтезу, при котором сложное устройство представляется системой автоматов, каждый из которых описывается таблицей переходов. Он исследовал зависимость числа состояний автомата от числа состояний блоков, составляющих этот автомат, при различных видах взаимодействия блоков. Работы М.А. Гаврилова по блочному синтезу явились толчком к развитию целой серии работ по методам композиции и декомпозиции автоматов.

Интересной и плодотворной оказалась высказанная М.А. Гавриловым идея применить к дискретным автоматам для повышения их надежности разработанные в теории передачи информации коды, позволяющие обнаруживать и исправлять ошибки – сбои в работе отдельных релейных элементов или цепей. Это положило начало новому направлению в развитии теории дискретных автоматов – надежностному синтезу дискретных последовательностных автоматов.

В последние годы М.А. Гавриловым был разработан метод синтеза многовходовых комбинационных устройств большой размерности, описываемых системой булевых функций. Этот метод развивал и обобщал накопленный опыт применения идеи направленного поиска.

М.А. Гаврилов неоднократно подчеркивал важность доведения предлагаемых методов синтеза до уровня инженерной практики. Именно поэтому в последние 20 лет своей жизни он уделяет основное внимание вопросам автоматизации проектирования дискретных устройств, начав с создания специализированных логических устройств – машин для анализа и



синтеза релейных устройств. Это направление перешло в разработку отдельных программ для ЭВМ, а затем – в создание интегрированных систем автоматизированного проектирования – человеко-машинных систем, работающих в режиме диалога.

М.А. Гаврилов принимал личное активное участие в организации работ по автоматизации проектирования. Под его научным руководством создан и утвержден государственный стандарт «Проектирование автоматизированное. Термины и определения» (ГОСТ 22487–77). Михаил Александрович активно участвовал и в разработке «Общеотраслевых руководящих методических материалов по созданию систем автоматизированного проектирования (САПР)».

В последние годы жизни М.А. Гаврилов вел большую научно-организационную и координационную деятельность:

- в области технической кибернетики в Научном совете по комплексной проблеме «Кибернетика» АН СССР, возглавляемом академиком А.И. Бергом, он председатель Секции технической кибернетики;
- в области автоматизированного проектирования в Комитете системного анализа АН СССР он председатель Научного совета по проблеме «Автоматизация проектирования».

В составе Секции технической кибернетики Михаил Александрович в 1962 г. организовал Комиссию по теории релейных устройств и через неё координировал научно-исследовательские работы в этой области. Эта комиссия рассматривала также вопросы, связанные с организацией всесоюзных конференций и международных семинаров, формированием их программ. Более десяти лет при комиссии работал Общемосковский семинар по теории релейных устройств.

Наиболее плодотворной формой координации и руководства научно-исследовательскими работами по тематике комиссии явилась Школа по теории дискретных устройств, организованная Михаилом Александровичем. Почти все ее созывы были проведены под его председательством. Первая школа состоялась в 1964 г. в г. Томске, а последняя, проходившая под председательством Михаила Александровича, была юбилейной, двадцатой, и состоялась в Москве в марте 1979 г. Гавриловские школы проходят и по сей день. Школа помогла сконцентрировать внимание исследователей на наиболее важных для практики задачах, объединить усилия и предоставила возможность научным работникам ознакомить достаточно широкий круг специалистов с полученными ими результатами. Школа по теории дискретных устройств была детищем М.А. Гаврилова, и ей по праву присвоено его имя. О Школе как явлении в научной жизни Михаила Александровича и всех «школьников» написано ряд интересных статей Д.А. Поспеловым, О.П. Кузнецовым и А.А. Шалыто, поэтому, чтобы не повторяться, отправляем читателей к ним.

М.А. Гаврилов на протяжении всей своей жизни растил научные кадры, собирая вокруг себя и воспитывая талантливую молодежь. Под его руководством около 50 человек защитили кандидатские диссертации, более десяти из них присвоена учченая степень доктора наук. Отношение Михаила Александровича

к аспирантам было удивительным, он был неимоверно строг, но всегда вставал на защиту своих учеников.

М.А. Гавриловым за время его научной деятельности опубликовано 256 работ. Первая, посвященная сетям электропередач, была опубликована в 1928 г., последние работы, посвященные системам автоматизации проектирования, – в 1979 г.

М.А. Гаврилов выполнял огромную работу по редактированию сборников, книг, трудов многих все-союзных и международных совещаний, периодических и переводных изданий. Под его редакцией издавался периодический сборник «Абстрактная и структурная теория релейных устройств», он являлся членом редколлегии «Энциклопедии по автоматике», был членом редакционной коллегии международного журнала «Проблемы управления и теория информации» с момента его основания, членом редакционных коллегий ряда других журналов.

Говоря о редакционно-издательской деятельности М.А. Гаврилова, нельзя не сказать о той огромной работе, которую он вел в области терминологии релейных устройств. Первая его статья по терминологии появилась в 1950 г. в «Бюллетене Комитета технической терминологии АН СССР». Впоследствии работы М.А. Гаврилова по терминологии были опубликованы в 1953, 1960, 1961 гг., переведены на английский язык и изданы в США. Редактируя различные издания, Михаил Александрович проявлял себя как чуткий, доброжелательный, но вместе с тем строгий критик, как внимательный и мудрый редактор.

За свою активную научную и организационную деятельность Михаил Александрович был награжден тремя орденами Трудового Красного Знамени и четырьмя медалями, а также «Медалью Республики Румыния» в ознаменование 25-летия Республики (1969 г.) и медалью АН ГДР в ознаменование 20-летия научных связей АН СССР и ГДР (1977 г.).

Большое место в жизни М.А. Гаврилова занимали международные научные контакты. Он читал лекции в научных организациях мира, пропагандировал достижения советской науки и техники во многих странах мира, был участником всех конгрессов ИФАК по дискретным системам и многих симпозиумов. М.А. Гаврилов работал в тесном содружестве с учеными из социалистических стран. Он был одним из организаторов, а затем и руководителем темы «Теория управления», работа в рамках которой позволила сформировать международный коллектив ученых, охватывающий самый широкий спектр задач теории управления, а регулярный обмен информацией позволял повысить эффективность труда ученых.

Можно смело сказать, что фундаментальные результаты М.А. Гаврилова, его оригинальные идеи оказали большое влияние на развитие теории и автоматизации проектирования дискретных управляемых устройств в мире.

М.А. Гаврилов скончался 29 апреля 1979 года, возвращаясь из ГДР с советско-германского семинара по теории автоматов. Уже минуло 24 года с того трагического дня, но память о нем жива, согревает сегодня нас, его учеников.

А.А. Амбарцумян

✉ (095) 334-87-89

E-mail: ambar@ipu.rssi.ru





Конференция «Теория и практика логического управления», посвященная памяти М.А. Гаврилова

10–11 ноября 2003 г. в Институте проблем управления им. В.А. Трапезникова РАН состоялась международная конференция, посвященная 100-летию со дня рождения выдающегося ученого, доктора технических наук, профессора, члена-корреспондента АН СССР Михаила Александровича Гаврилова – одного из создателей и лидеров теории логических схем и конечных автоматов, которая в 1960-х годах была наиболее продвинутой и бурно развивающейся ветвью кибернетики.

В конференции приняли участие ученики и коллеги М.А. Гаврилова, участники знаменитых Гавриловских школ по теории дискретных устройств и конечных автоматов, приехавшие из Санкт-Петербурга, Киева, Минска, Томска и других городов.

С докладом о жизненном пути М.А. Гаврилова и его вкладе в мировую науку выступил *А.А. Амбарцумян*. *О.П. Кузнецов* рассказал о Гавриловских школах, их традициях и незабываемой научной и человеческой атмосфере, царившей на них. С воспоминаниями о М.А. Гаврилове выступили директор ИПУ *И.В. Прангшивили*, *П.П. Пархоменко*, *Е.В. Бабичева*, *А.Е. Янковская*. С теплыми словами обратилась к участникам вдова М.А. Гаврилова *Н.Н. Потапова*.

Представленные на конференции доклады с некоторой долей условности можно классифицировать следующим образом.

Логическое управление

Амбарцумян А.А. Логическое управление на основе событийного моделирования.

Бибilo П.Н. Декомпозиция частичных булевых функций на основе решения логических уравнений.

Закревский А.Д. Эффективный метод нахождения кратчайшего решения систем линейных логических уравнений.

Лазарев В.Г., Пиль Е.И. Применение теории автоматов и логического управления при проектировании телекоммуникационных систем.

Потехин А.И. Аналитическое описание технологической структуры объектов с потоковой технологией.

Потосин Ю.В., Томашев В.Ф., Шестаков Е.А. Мультиплексная реализация булевой функции по ее секционному разложению.

Выхованец В.С., Малюгин В.Д. Аппаратная и программная реализация мультиплексивных форм.

Редькин Н.П. Минимальные самокорректирующиеся схемы для оператора поразрядного сравнения булевых наборов.

Финько О.А. Модульные формы арифметической логики.

Логическое проектирование

Черемисинов Д.И. Конвертер структурных описаний для интеграции САПР.

Кононенко И.А., Захарова Г.Б., Титов В.Г. Программно-алгоритмический комплекс для компи-

лятивного проектирования систем логического управления и вычислительных структур.

Чеботарев А.Н. Язык логической спецификации реактивных систем (синтаксис и семантика).

Черемисинова Л.Д. Комплекс программ синтеза логических сетей на ПМЛ.

Янковская А.Е. К вопросу оптимизации и надежностного проектирования и перепроектирования устройств логического управления.

Системный анализ

Прангшивили И.В. Системный подход, системное мышление и энтропизация знаний.

Теория программирования

Абрамова Н.А. О практических возможностях формализованных спецификаций.

Искусственный интеллект

Кузнецов О.П., Кулинич А.А., Марковский А.В. Применение аппарата нечетких отношений к анализу ситуаций.

Постолов Д.А., Шустер В.А. Нормативное поведение интеллектуальных систем.

Случайные процессы

Гребенюк Е.А. Анализ и оперативная диагностика систем, описываемых нестационарными случайными процессами, применение разработанных алгоритмов для прогноза кризисных событий в российской экономике.

Техническая диагностика

Правильщиков П.А. Законы технической диагностики как обоснование новой архитектуры вычислительных средств для решения логических и других уравнений.

Прикладная дискретная математика

Выхованец В.С. Алгебраическая декомпозиция дискретных функций в аддитивной алгебре.

Пархоменко П.П. Построение гамильтоновых циклов в двоичных гиперкубах с неисправными ребрами.

Шоломов Л.А. Формальные методы построения, агрегирования и декомпозиции отношений в порядке моделях многокритериального выбора.

Новые информационные технологии

Григорян А.К. Современные технологии управления предприятиями.

Девятков В.В. Мультиагентное иерархическое распознавание на основе нечеткого ситуационного исчисления.

Научная тематика конференции оказалась весьма разнообразной, показав, насколько широк круг областей информатики и управления, в которых работают люди, начинавшие свой научный путь с теории логических схем и конечных автоматов.

О.П. Кузнецов

✉ (095) 334-91-19

E-mail: olkuznes@ipu.rssi.ru

Указатель статей, опубликованных в журнале «Проблемы управления» в 2003 г.

Абрамов О.В., Катуева Я.В. Технология параллельных вычислений в задачах анализа и оптимизации. № 4.

Бахтадзе Н.Н., Назин А.В. Виртуальные анализаторы маркетинговой информации в системе управления предприятием. № 4.

Беляков А.Г., Мандель А.С., Борзенко Н.И. и др. Экспертно-статистические системы прогнозирования временных рядов и имитационно-оценочное моделирование. № 3.

Бернацкий Ф.И., Диго Г.Б., Диго Н.Б. Параллельные вычисления в задачах управления. № 3.

Борис Николаевич Петров (к девяностолетию со дня рождения). № 1.

Бутковский А.Г. О единой геометрической теории управления. № 1.

Ведененков В.А. Самодиагностирование цифровых систем с реконфигурацией. № 4.

Волгин Л.И., Климовский А.Б. Имплекативная алгебра выбора как основа информационных технологий и систем управления в континуальной области. № 4.

Вторая международная конференция по проблемам управления. № 2.

Гребенюк Е.А. Анализ и оперативная диагностика систем, описываемых нестационарными случайными процессами. № 4.

Гуляев С.В., Черепова Т.И., Шубладзе А.А., Шубладзе А.М. Оптимальные по степени устойчивости системы управления динамическими объектами. № 3.

Дартау Л.А. Теоретические аспекты управления здоровьем и возможности его реализации в условиях Российской Федерации. № 2.

Дургарян И.С., Пашченко Ф.Ф. Сложные статистические критерии и модели, оптимальные на классе критериев. № 2.

Жожикашвили В.А., Билик Р.В., Вертиб В.А. и др. Открытые системы массового обслуживания с распознаванием речи. № 4.

Закревский А.Д. Эффективные методы нахождения кратчайших решений систем линейных логических уравнений. № 4.

Затуливетер Ю.С., Фищенко Е.А. Организация распределенных вычислений в системе программирования ПАРСЕК на примере сжатия цифрового видео. № 4.

Калашников Л.М., Малышев Г.В., Свотин А.П. Управление свертыванием двухмодульной космической троцовой системы. № 4.

Кара-Мурза С.Г. Проблемы управления наукой на новом этапе реформ. № 1.

Клепарский В.Г., Ефремов В.А. Мультифрактальность, диссипация и устойчивость среднесрочных трендов на фондовом рынке. № 4.

Конференция «Восток–Запад: Проектирование и диагностирование цифровых устройств» (East–West Design & Test Conference). № 3.

Конференция по теории управления, посвященная памяти академика Б.Н. Петрова. № 1.

Конференция «Теория и практика логического управления», посвященная памяти М.А. Гаврилова. № 4.

Копин М.Ю., Кульба В.В., Микрин Е.А. Применение языка сетей Петри в системах сетевого планирования и управления при дефиците ресурсов. № 2.

Крапухина Н.В., Пастухова К.М., Свиридов П.А. Методы искусственного интеллекта в задачах оперативного управления и оптимизации сложных технологических комплексов. № 3.

Кузнецов Л.А. Системное представление финансово-хозяйственной деятельности предприятия. № 3.

Кульба В.В., Малюгин В.Д., Шубин А.Н. Информационное управление (предпосылки, методы и средства). № 1.

Микрин Е.А., Пелихов В.П. Анализ нештатных ситуаций и критичности программного обеспечения в проекте международной космической станции. № 4.

Михаил Александрович Гаврилов (к 100-летию со дня рождения). № 4.

Михальский А.И., Яшин А.И. Биологическая регуляция и продолжительность жизни. № 3.

Пашченко Ф.Ф. Технопарковые структуры и инновационное развитие. № 1.

Попков Ю.С. Макросистемы и GRID-технологии: моделирование динамических стохастических сетей. № 3.

Портнов-Соколов Ю.П. Краткий очерк некоторых работ по космонавтике Института проблем управления им. В.А. Трапезникова. № 3.

Прангвишили И.В. Системный подход, системное мышление и энтропизация фундаментальных знаний. № 1.

Прангвишили И.В., Амбарцумян А.А., Полетыкин А.Г. и др. Состояние уровня автоматизации энергетических объектов и системотехнические решения, направленные на его повышение. № 2.

Прангвишили И.В., Островский Ю.И. Увеличение платежеспособного спроса населения на товары отечественного производства путем использования внутренних платежных средств. № 2.

Реймаров Г.А., Грицук Р.К., Ионов В.В. Программный комплекс «Персона» как средство анализа и управления человеческим капиталом предприятий. № 3.

Рутковский В.Ю., Земляков С.Д., Суханов В.М., Глумов В.М. Некоторые новые направления развития теории и применения адаптивного координатно-параметрического управления. № 2.

Смирнов С.В., Тюкавкин Д.В. Геоинформационная система для поддержки принятия решений в органах управления социально-образовательной сферой. № 3.

Трахтенберг Э.А. Компьютерные системы поддержки принятия управленческих решений. № 1.

Третий семинар ИФАК «Автоматические системы для создания инфраструктуры в развивающихся странах (Знания и техника)». № 3.

Цвиркун А.Д. Управление развитием крупномасштабных систем в новых условиях. № 1.

Шубин А.Н., Пелихов В.П. Задачи оптимизации медиапланирования в информационном управлении. № 2.

Эштейн В.Л. Антропоцентрическое информационное взаимодействие (вопросы терминологии). № 1.



Index of Papers published in 2003

Abramov O.V., Katueva Y.V. Multivariant analysis and stochastic optimization using parallel processing techniques. No 4.

Bakhtadze N.N., Nazin A.V. Virtual analyzers of marketing information in an enterprise management system. No 4.

Belyakov A.G., Mandel' A.S., Borzenko N.I., et. al. Expert statistical systems for time series prediction and the estimating simulation. No 3.

Bernatsky F.I., Digo G.B., Digo N.B. Parallel computations in control problems. No 3.

Boris Nikolaevich Petrov (The Ninetieth Anniversary). No 1.

Butkovsky A.G. On the unified geometrical theory of control. No 1.

Conference on Control Theory in memory of academician B.N.Petrov. No 1.

Dartau L.A. Theoretical aspects of health control and its implementation possibilities in Russian Federation. No 2.

Durgarian I.S., Pashchenko F.F. Combined statistical criteria and the models optimal subject to a class of criteria. No 2.

East-West Design & Test Conference. No 3.

Epstein V.L. Anthropocentric informational interaction: terminology issues. No 1.

Grebnyuk E.A. Analysis and on-line diagnostics of the systems described as nonstationary stochastic processes. No 4.

Gulyaev S.V., Cherepova T.I., Shubladze A.A., Shubladze A.M. Dynamic object control systems optimal with respect to the degree of stability. No 3.

Kalashnikov L.M., Malyshev G.V., Svitin A.P. The control of two-module space rope system rolling. No 4.

Kara-Murza S.G. Scientific policy in the new stage of reform in Russian Federation. No 1.

Kleparsky V.G., Efremov V.A. Multifractality, dissipation and stability of the middle-term stock market trends. No 4.

Kopnin M.Yu., Kul'ba V.V., Mikrin E.A. Application of Petri net tools in network planning and management systems under resource deficit conditions. No 2.

Krapukhina N.V., Pastukhova K.M., Sviridov P.A. Artificial intelligence techniques in on-line control and optimization of complex process systems. No 3.

Kul'ba V.V., Malyugin V.D., Shubin A.N. Informative control: premises, techniques and tools. No 1.

Kuznetsov L.A. System representation of financial and economic activity of an enterprise. No 3.

Michalski A.I., Yashin A.I. Biological regulation and longevity. No 3.

Mikhail Alexandrovich Gavrilov (The Centenary). No 4.

Mikrin E.A., Pelikhov V.P. Contingency and software criticality analysis in international space station projects. No 4.

Pashchenko F.F. Innovative centers and technopark structures. No 1.

Popkov Yu.S. Macrosystems and grid technologies: dynamic stochastic network modeling. No 3.

Portnov-Sokolov Yu.P. Some works in astronautics performed at the Institute of Control Sciences – a brief survey. No 3.

Pranghishvili I.V. System approach, system thought and entropization of fundamental knowledge. No 1.

Pranghishvili I.V., Ambartsumyan A.A., Poletykin A.G., et. al. Automation level of power objects: current state analysis and improvement opportunities. No 2.

Pranghishvili I.V., Ostrovsky Yu.I. Improving population's effective demand for domestics by using internal means of payment. No 2.

Reymarov G.A., Gritsuk R.K., Ionov V.V. Persona software system as a tool for enterprise human assets analysis and management. No 3.

Rutkovsky V.Yu., Zemlyakov S.D., Sukhanov V.M., Glumov V.M. New directions of adaptive coordinate-parametric control theory and applications. No 2.

Shubin A.N., Pelikhov V.P. On media planning optimization problems in information management. No 2.

Smirnov S.V., Tyukavkin D.V. A geoinformation system for decision-making support in social and educational authorities. No 3.

The 3rd IFAC Workshop "Automatic systems for infrastructure formation in developing countries (Knowledge and Technology)". No 3.

The Second International Conference on Control Problems. No 2.

Theory and Practice of Logical Control Conference in memory of M.A.Gavrilov. No 4.

Trachtenhertz E.A. Computer-aided management decision-making support systems. No 1.

Tsvirkun A.D. Management of large-scale system development in new conditions. No 1.

Vedeshenkov V.A. System-level self-diagnosis of reconfigurable systems. No 4.

Volgin L.I., Klimovsky A.B. Implicative algebra of choice as a base of information technologies and control systems in continuum domain. No 4.

Zakrevsky A.D. Effective techniques for finding the shortest solution of linear logic equations. No 4.

Zatuliveter Y.S., Fischenko E.A. Organization of distributed calculations in PARSEC programming system with the example of digital video compression. No 4.

Zhozhikashvili V.A., Bilik R.V., Vertlieb V.A., et al. Open queuing systems with speech recognition. No 4.

CONTENTS & ABSTRACTS

IMPLICATIVE ALGEBRA OF CHOICE AS A BASE OF INFORMATION TECHNOLOGIES AND CONTROL SYSTEMS IN CONTINUUM DOMAIN 2

Volgin L.I., Klimovsky A.B.

The paper considers new lines of investigation in theoretical information and mathematical cybernetics (continual algebraic logics – special algebras and general logics) and in analog computing machinery (relator circuitry) aimed at solving computational control tasks on the basis of universal logical-and-algebraic tool of implicative choice algebra.

ORGANIZATION OF DISTRIBUTED CALCULATIONS IN PARSEC PROGRAMMING SYSTEM WITH THE EXAMPLE OF DIGITAL VIDEO COMPRESSION 6

Zatuliveter Y.S., Fischenko E.A.

The paper examines a new approach to the distributed calculations organization in the Internet, based on the mathematically complete and closed model of tree-type structures calculus, realized by means of PARSEC programming system. The PARSEC language is supplemented with the operators for resources administration and bulk data transfer. The effectiveness of the proposed net resources control technique is shown by the example of real-time compression encoding for H.261 videoconferences and for MPEG digital video.

MULTIVARIANT ANALYSIS AND STOCHASTIC OPTIMIZATION USING PARALLEL PROCESSING TECHNIQUES 11

Abramov O.V., Katueva Y.V.

The problem of choosing optimal parameter values for analog devices and systems is considered. Special attention is paid to the algorithms that reduce the computation cost of optimization problem. Several parallel techniques for multivariate analysis and stochastic optimization are discussed.

EFFECTIVE TECHNIQUES FOR FINDING THE SHORTEST SOLUTION OF LINEAR LOGIC EQUATIONS 16

Zakrevsky A.D.

The paper offers effective techniques that considerably reduce the amount of searching performed for finding the shortest solution of linear logic equations. It presents software implementation results, estimates the competitiveness of the programs developed and discusses their application areas.

ANALYSIS AND ON-LINE DIAGNOSTICS OF THE SYSTEMS DESCRIBED AS NONSTATIONARY STOCHASTIC PROCESSES 23

Grebnyuk E.A.

The problem of detecting the changes in the properties of cointegrated nonstationary processes is discussed. A cumulative sum algorithm is offered for detecting the cointegration loss. The effectiveness of the approach proposed is demonstrated with the solution to the problem of analyzing a group of Russian market indices described as nonstationary time series.

VIRTUAL ANALYZERS OF MARKETING INFORMATION IN AN ENTERPRISE MANAGEMENT SYSTEM 30

Bakhtadze N.N., Nazin A.V.

The paper solves a commercial budgeting problem. It offers virtual analyzers for marketing information based on parametric identification and inventory control algorithms and models under incomplete a priori information and random supplies.

MULTIFRACTALITY, DISSIPATION AND STABILITY OF THE MIDDLE-TERM STOCK MARKET TRENDS 36

Kleparsky V.G., Efremov V.A.

The paper studies basic regularities of the switching activity in process of returning the stock-market to the initial channel of the system's evolution. It reveals two distinctive features of the return intervals. The first feature is the minimization of the fractal dimension of the evolution trajectory, while the second one is the minimization of intermittency index.

SYSTEM-LEVEL SELF-DIAGNOSIS OF RECONFIGURABLE SYSTEMS 39

Vedeshenkov V.A.

The paper presents a system-level self-diagnosis (SD) method for the components (modules and communication lines) of reconfigurable systems with multiple hard failures of several components. The method is based on the expanding domain assumption. An example of 2 failed components SD in a 2-fault-tolerant system is adduced.

CONTINGENCY AND SOFTWARE CRITICALITY ANALYSIS IN INTERNATIONAL SPACE STATION PROJECTS 52

Mikrin E.A., Pelikhov V.P.

The paper classifies basic types of contingencies, which might take place at the long-functioning space station. It also defines and describes key features of critical functions performed by space station's automatics and software. The software is categorized by danger classes according to NASA and ESA requirements.

OPEN QUEUING SYSTEMS WITH SPEECH RECOGNITION 55

Zhozhikashvili V.A., Bilik R.V., Vertlieb V.A., et al.

The problem of wide audience access to information is discussed. The paper introduces the idea of an open queuing system and discusses the application of speech recognition technology to organize a voice interface for information systems. The existing system with the access to databases via the phone is described.

THE CONTROL OF TWO-MODULE SPACE ROPE SYSTEM ROLLING 63

Kalashnikov L.M., Malyshev G.V., Svitin A.P.

The problem of rolling a space rope system by controlling the rope tension is considered based on the following assumption: the system is revolving along the circular orbit; it consists of 2 modules connected with an imponderable and nonelastic rope and is in the state of either libration or vertical stable equilibrium. A tension control program is proposed that presumes the availability of basic trajectory and multiparameter control relative to it.