

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

12(196)
2012

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Бобков С. Г. Архитектура микропроцессоров суперЭВМ эксафлопсного диапазона . 2
Егоров Г. А., Шяудкулис В. И., Финотти М. Проблемы построения современных архивных хранилищ данных 9

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Левин В. И. Упрощенная методика оптимизации систем в условиях интервальной неопределенности 19
Саушев А. В. Метод и алгоритмы параметрического синтеза электротехнических систем по критерию запаса работоспособности 24
Борчик Е. М., Якимов А. И., Степанов А. И., Башаримов В. В. Построение плотностей распределения Пирсона для многомодальных выборок с применением кластеризации 30

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

- Щеголева Н. Л., Кухарев Г. А., Форчманский П. Простой алгоритм распознавания изображений лиц с проблемами освещения. 36
Шабунин А. Б., Кузнецов Н. А., Скобелев П. О., Бабанин И. О., Кожевников С. С., Симонова Е. В., Степанов М. Е., Царев А. В. Разработка онтологии для мультиагентной системы управления ресурсами ОАО "РЖД". 42
Нечаев В. В. Раскрытие неопределенности системной задачи, представленной концептуальной моделью. 46
Волобуев М. Ф., Демчук В. А., Замыслов М. А., Мальцев А. М., Михайленко С. Б. Управление резервированной с помощью мажоритарных элементов системой с применением методов нечеткой логики. 53

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Перегула А. И., Тимашов Д. А. Математическая модель надежности информационных систем с эшелонированной защитой от несанкционированного доступа 61
Указатель статей, опубликованных в журнале "Информационные технологии" в 2012 г. 67
Указатель приложений к журналу "Информационные технологии", опубликованных в 2012 г. 70
Contents 71

- Приложение. Кухаренко Б. Г. Аккумулирующий анализ главных компонент потоков тензорных данных

Главный редактор
НОРЕНКОВ И. П.

Зам. гл. редактора
ФИЛИМОНОВ Н. Б.

Редакционная
коллегия:

АВДОШИН С. М.
АНТОНОВ Б. И.
БАРСКИЙ А. Б.
БОЖКО А. Н.
ВАСЕНИН В. А.
ГАЛУШКИН А. И.
ГЛОРИОЗОВ Е. Л.
ДОМРАЧЕВ В. Г.
ЗАГИДУЛЛИН Р. Ш.
ЗАРУБИН В. С.
ИВАННИКОВ А. Д.
ИСАЕНКО Р. О.
КОЛИН К. К.
КУЛАГИН В. П.
КУРЕЙЧИК В. М.
ЛЬВОВИЧ Я. Е.
МАЛЬЦЕВ П. П.
МЕДВЕДЕВ Н. В.
МИХАЙЛОВ Б. М.
НЕЧАЕВ В. В.
ПАВЛОВ В. В.
ПУЗАНКОВ Д. В.
РЯБОВ Г. Г.
СОКОЛОВ Б. В.
СТЕМПКОВСКИЙ А. Л.
УСКОВ В. Л.
ФОМИЧЕВ В. А.
ЧЕРМОШЕНЦЕВ С. Ф.
ШИЛОВ В. В.

Редакция:

БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.
Журнал включен в систему Российского индекса научного цитирования.
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

УДК 004.382.2

С. Г. Бобков, д-р техн. наук,
ст. науч. сотр., нач. отделения,
НИИ системных исследований РАН, г. Москва,
e-mail: s_g_bob@mail.ru

Архитектура микропроцессоров суперЭВМ эксафлопсного диапазона

Рассматриваются новые подходы создания высоко-надежных микропроцессоров суперЭВМ эксафлопсного диапазона.

Ключевые слова: суперЭВМ, суперкомпьютерные технологии, архитектура вычислительной системы

Состояние работ по созданию суперЭВМ эксафлопсного диапазона

В последние десятилетия одним из основных признаков высокого технического развития страны являлось наличие в стране собственной суперЭВМ. И это понятно почему. Во-первых, создание суперЭВМ требует наличия высоких технологий в самых различных областях: инженерных, электронных, программных. Во-вторых, такие ЭВМ нацелены на решение сложных технических задач; если страна не способна ставить и решать такие задачи, то и суперЭВМ не нужна. К концу XX века потенциал инженерных технологий теорий микроскопического взаимодействия в рамках моделей сплошной среды был практически исчерпан. Дальнейшее развитие ядерной и термоядерной энергетики, электроники, авиастроения, биотехнологий и т. д. стало невозможным без проведения полномасштабных инженерных расчетов таких сложных технических и биологических систем с учетом атомно-молекулярного взаимодействия, требующих уже в среднесрочной перспективе (до 2020 г.) использования суперЭВМ эксафлопсного класса (10^{18} оп/с). Только те страны, которые будут иметь такие ЭВМ и соответствующее программное обеспечение, способны будут создавать принципиально новые изделия [1].

Из обширного перечня областей человеческой деятельности, задачи которых требуют применения вычислительных систем эксафлопсной производительности, можно выделить следующие:

- атомная промышленность (включая ядерно-оружейный комплекс);
- сложные промышленные системы, прежде всего, в авиационной и космической отрасли;

- энергетика и энергосбережение;
- медицина;
- биология;
- информационные технологии.

В США в 2009 г. по указанию президента США в целях сохранения лидирующих позиций в области суперкомпьютерных технологий, их внедрения в науку, промышленное производство и социальные сферы разработан план действий под названием Exascale Initiative [2]. Министерством энергетики США из специалистов ядерных лабораторий создан специальный комитет Exascale Initiative Steering Committee (EISC), целями которого являются:

- определение особо значимых научных проблем, требующих наибольшего привлечения вычислительных ресурсов в течение следующего десятилетия;
- определение оптимального пути повышения производительности вновь создаваемых вычислительных систем с учетом потребностей прикладной и фундаментальной науки;
- организация обмена междисциплинарными идеями в интересах развития суперкомпьютерной базы.

Объем ежегодного финансирования исследовательских работ по этому эксафлопсному проекту превышает \$100 млн.

Кроме того, Министерство энергетики США является одним из основных финансистов международного проекта International Exascale Software Project (IESP) [3]; соучредителем проекта является также национальный научный фонд (NSF) США. Цель проекта — создание программной базы для суперкомпьютеров эксафлопсного уровня, появление которых ожидается в 2018—2020 гг. В проекте, кроме американских, широко задействованы европейские и японские фирмы и организации.

Аналогичные работы, предусматривающие достижение эксафлопсной производительности, выполняются под патронажем Министерства обороны США через агентство DARPA [4] — в частности, проекты HPSC, UHPC и OHPC. В них объединены ведущие фирмы — Intel, Cray, Nvidia, университеты и национальные лаборатории.

Создание систем производительностью 10 Пфлоп/с предусматривается в США уже с 2012 г. В Европе эксафлопсные технологии реализуются в проекте PRACE (он предусматривает появление эксафлопсного компьютера в 2019 г.) [5] и финансируемых Евросоюзом проектах European Exascale Software Initiative (EESI) и Towards Exascale Applications (TEXT), направленных на разработку программ-

ных технологий, предназначенных для использования в экскалальных компьютерах.

Китай планы не публикует, но, по мнению экспертов, способен создать суперЭВМ экскафлопсного класса на собственной элементной базе в 2020 году.

Все изложенное свидетельствует об интенсивных усилиях ведущих государств мира по созданию экскафлопсных технологий.

Применение экскафлопсных технологий, реализующих принципиально различные дисциплины вычислений на качественно более высоком уровне параллелизма, сложности и неоднородности вычислительных систем требует:

- фундаментальных исследований математических методов, алгоритмов и архитектур, позволяющих достигнуть необходимых значений производительности;
- разработки аппаратных и системных программных средств экспериментальных вычислительных систем, удовлетворяющих требованиям по энергопотреблению, надежности и другим физическим параметрам;
- создания соответствующего прикладного программного обеспечения.

Основные проблемы создания микропроцессоров для суперЭВМ экскафлопсного класса

Микропроцессоры требуемой области применения должны иметь производительность не менее единиц терафлопс. Достижение таких производительностей требует использования предельных технологических норм изготовления кристаллов и частот их функционирования. На этом пути стоят две основные проблемы: **низкая надежность и высокая потребляемая мощность** микросхем и, соответственно, всего компьютера. Необходимость эффективного функционирования $10^8...10^9$ процессорных ядер такой суперЭВМ и преодоление физических ограничений, обусловленных энергопотреблением, надежностью и конструктивными размерами, предполагают иные подходы к архитектуре, схемотехнике и программному обеспечению микропроцессоров.

При использовании предельных технологических норм изготовления кремниевых пластин, необходимых для создания экскафлопсных суперЭВМ, невозможно добиться высокой надежности функционирования микросхем без использования всего комплекса мер повышения надежности. Это обусловлено:

- снижением напряжения питания и соответствующим уменьшением возможного разброса порогов логических уровней, повышением требований к качеству питания;
- уменьшением толщин диэлектриков и соответствующим снижением надежности, повышением токов утечки, снижением значений паразитных зарядов (порогов), приводящим к сбою или отказу микросхем;
- уменьшением сечений проводников, что приводит к уменьшению длительности функционирования

микросхем за счет эффекта электромиграции, снижением механической прочности, прежде всего, при снижении температур окружающей среды до предельно низких значений с последующим повышением до предельно высоких;

- большим числом элементов, статистически приводящим к повышению вероятности сбоя микросхем;
- увеличением поражающего фактора для одиночных и локальных радиационных эффектов, связанного с существенно меньшими топологическими размерами активных элементов в сравнении с размерами трека поражающей частицы;
- существенным влиянием фактора микродозовых радиационных эффектов на надежность и долговечность схем, обусловленным структурными изменениями в приборных областях схем при воздействии высокоэнергетичных частиц и потоков атмосферных нейтронов.

Для достижения требуемой надежности функционирования микросхем предлагается применить комплекс мер:

- конструктивно-технологических;
- схемотехнических;
- программных и микропрограммных.

К **технологическим мерам** следует отнести использование специальных процессов изготовления кремниевых пластин. Так, компания IBM идет по пути развития процесса КНИ (кремний на изоляторе), приводящего к снижению потребляемой мощности и повышающего рабочую частоту до 30 %.

Следующим фактором, приводящим к существенному повышению надежности, является создание специальных **радиационно-стойких библиотек**. Работы компании Боинг показали, что даже при использовании КМОП-технологий при наличии таких библиотек возможно достижение стойкости на уровне нескольких сотен килорад (при использовании КНИ — единицы мегарад). Следующим конструктивным решением повышения надежности микросхем является разработка правил топологического проектирования, учитывающих требования по защитным кольцам, дополнительным контактам, сетке земли и питания, блоку питания, внутреннему (для ионизирующих частиц) разнесению активных элементов и пр. Факторы радиационной стойкости начнут неизбежно проявляться при большом числе микросхем даже в обычные дни, а при наличии солнечных вспышек работа рассматриваемой суперЭВМ может полностью блокироваться в силу огромного числа сбоев [6]. Необходимо также отметить, что начало тиристорного эффекта в КМОП-схемах можно отследить резким повышением тока в месте прохождения высокоэнергетичной частицы (в КНИ КМОП-схемах тиристорный эффект практически отсутствует [6]).

Схемотехнические приемы повышения надежности микросхем следует разбить на две группы: создание высоконадежной памяти и создание на-

дежных вычислительных и управляющих блоков. Это связано с различными способами реализации таких схем и различными методиками повышения их надежности. В свою очередь, методики создания высоконадежной памяти тоже следует разбить на две группы: создание быстродействующего ОЗУ небольшого объема (до нескольких десятков килобайт), прежде всего, кэш-памяти 1-го уровня, и ОЗУ объемом свыше 100 Кбайт. Это связано с тем, что для небольшого ОЗУ можно применить дополнительные схемотехнические решения, приводящие к повышению надежности, но и увеличивающие площадь ОЗУ в несколько раз. Для ОЗУ большого объема необходимо применение элементов и блоков помехоустойчивого кодирования, обеспечивающих контроль и исправление не только одиночных, но и множественных ошибок.

Повышение надежности вычислительных узлов с использованием схемотехнических решений также возможно за счет применения различного рода дублирований и троирований. Возможно также использование многофазной специальной логики, функционирующей аналогично коду Хэмминга для памяти, позволяющей определить сбой при выполнении арифметической операции.

Еще один вариант — использование асинхронной или строго самосинхронной логики. Однако все коммерческие САПР рассчитаны на синхронную логику. Необходимы большие вложения в создание инструментария разработки таких микросхем. Возможным вариантом является первоначальное создание проекта на синхронной логике с последующим его переводом на строго самосинхронную логику.

Для схем управления возможны использование кодирования долгоживущих регистров с помощью ЕСС (error-correcting code) и контроль корректности машин состояния.

Дальнейшим повышением надежности является **создание встроенного тестирования**, включая тестирование на аппаратном и микропрограммном уровнях. И наконец, предлагается предоставление максимальных возможностей для тестирования на **программном уровне**, сбора статистики по сбоям, возможностям отключения/включения дополнительных ядер микропроцессора и изменения частоты его функционирования в процессе выполнения программы.

Таким образом, для создания требуемых микропроцессоров можно выделить следующий комплекс мер по повышению их надежности.

1. Разработка радиационно-стойкой библиотеки.
2. Разработка правил топологического проектирования.
3. Создание датчиков тока и температуры для каждого функционального блока в целях выявления нештатных ситуаций косвенными методами и принятия мер по их ликвидации на аппаратном и программном уровнях.

4. Создание ОЗУ емкостью до нескольких десятков килобайт на базе ячеек Disc с разнесением половинок ячеек памяти в шахматном порядке и перемешиванием данных в целях уменьшения вероятности сбоя обеих половинок одновременно.

5. Создание ОЗУ емкостью сотни килобайт — единицы мегабайт с кодом Хэмминга (Хсяо). Встроенное помехоустойчивое кодирование данных должно учитывать факторы образования множественных ошибок и сбоев.

6. Дублирование вычислительных узлов (ядер). Ядра могут функционировать или независимо, или в режиме дублирования. В режиме дублирования при отличии результатов выполнения инструкций проводится их выполнение заново. При повторной (третьей) фиксации отличия проводится внутреннее тестирование для выявления отказавшего ядра и исключения его из процесса вычислений.

7. Выполнение схем контроля дублирования и части схем управления на более сбоеустойчивых радиационно-стойких элементах (например кольцевых транзисторах), использование также кодирования долгоживущих регистров с помощью ЕСС и контроль корректности машин состояния.

8. Контроль времени выполнения отдельных частей программы, включая тестовые программы, сторожевыми таймерами. Должно быть достаточное число таких таймеров с различными временными диапазонами и точностью.

9. Сетки земли и питания для ядер делают разными.

10. Накопление статистики по сбоям в специальных регистрах с возможностью выдачи этой информации, например, в локальную сеть Ethernet.

11. Создание аппаратных возможностей внутреннего самотестирования функциональных блоков и, прежде всего, памяти.

12. Наличие нескольких независимых ОЗУ для групп ядер с контролем по коду Хэмминга.

13. Наличие собственных коммуникационных каналов у каждой группы ядер процессора. Каналы имеют внутренние средства самотестирования и могут отключаться аналогично ядрам микропроцессора.

14. Возможность раздельного питания ядер.

15. Возможность дублирования генератора микропроцессора.

16. Постепенный переход на логику, приводящую к снижению потребления питания и повышению надежности (например на строго самосинхронную логику).

17. Создание программных средств тестирования узлов микропроцессора (прежде всего, памяти) в режимах ожидания.

18. Создание системы сбора статистики сбоев и отказов с выходом, например, на локальную сеть.

Использование обратимой логики

Одним из направлений в области схемотехники снижения потребляемой мощности является использование обратимой логики.

Для формирования логического состояния требуется потратить некоторое количество энергии. Установлено, что минимальное количество энергии, необходимое для формирования логического уровня, имеет порядок энергии теплового шума [7]:

$$E = K \cdot T \cdot \ln 2,$$

где K — константа Больцмана; T — абсолютная температура вычислительного устройства.

Именно это минимальное количество энергии рассеивается при потере одного логического состояния. Прогнозируется, что традиционная статическая КМОП-схемотехника приблизится к этому фундаментальному пределу к 2015 г. Преодоление данного фундаментального предела осуществимо, если при совершении любой логической операции не происходит потеря логических значений входных сигналов и возможно их восстановление на основе выходных. То есть для снижения рассеивания энергии при вычислениях необходимо избежать потери логических состояний сигналов.

Обратимая архитектура содержит совокупность методов по снижению потери информации, которые можно применять на разных уровнях организации вычислений. Например, в программировании некоторые операции являются логически обратимыми ($R1 = R1 + R2$), а некоторые являются логически необратимыми ($R1 = 0$). Логически необратимая операция приводит к потере бита информации в регистре и, соответственно, к дополнительному рассеиванию энергии. Эта проблема решается оптимизацией алгоритма работы компилятора, который по возможности старается избегать в машинном коде таких операций.

Необратимым является процесс рассеивания энергии при потере битов информации в ходе совершения логических операции или записи в элементы памяти. Можно преодолеть этот предел, если в вычислительном устройстве использовать обратимые логические элементы (т. е. логические элементы, в которых происходящие процессы должны быть обратимыми). Эти элементы могут реализовать обратимую булеву функцию и сохранять полное число битов информации, поэтому энтропия при работе таких элементов остается неизменной. При реализации прямой булевой функции на выходе обратимого элемента, кроме информационных битов, формируются так называемые мусорные (*garbage*) биты, с помощью которых можно реализовать обратную булеву функцию и восстановить значения входных сигналов.

Особенностью всех обратимых логических элементов является возможность однозначно определить состояние входных сигналов по состоянию выходных. Простейшим примером обратимого логического элемента является вентиль НЕ. По состоянию выхода этого элемента можно однозначно определить состояние входа. Вентиль НЕ реализует обратимую логическую функцию, и информация при его работе не теряется.

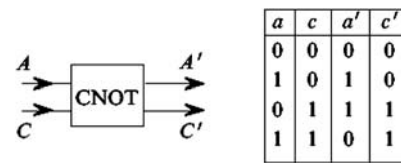


Рис. 1. Схематическое изображение вентиля УПРАВЛЯЕМОЕ НЕ и его таблица истинности

Другой пример обратимого логического элемента — это УПРАВЛЯЕМОЕ НЕ (рис. 1) [8].

Состояние выхода C' всегда совпадает с входом C , а состояние выхода A' инвертируется при $C = 1$. Как видно из таблицы истинности, этот элемент является обратимым, поскольку каждому выходному состоянию соответствует определенное состояние входов. Логическая функция этого элемента может быть записана следующим образом: $a' = a \times \bar{c} + \bar{a} \times c$. Стоит отметить, что под свойством консервативности элемента понимается сохранение количества информации при ее прохождении через обратимый логический элемент или цепочки из таких вентилях. Под свойством обратимости элемента также понимается возможность замены его входов на выходы вместе с заменой его логической функции на обратную. При этом число входов у обратимого элемента должно совпадать с числом выходов.

Вычислительно-универсальным обратимым элементом является вентиль Фредкина [9], который можно понимать как вентиль УПРАВЛЯЕМЫЙ ОБМЕН, реализующий следующую функцию:

$$c' = c; y_1 = c \times x_1 + \bar{c} \times x_2; y_2 = \bar{c} \times x_1 + c \times x_2.$$

С точки зрения оптической реализации рассматривались два логических обратимых и вычислительно-универсальных логических элемента: переключатель Прайса и вентиль ВЗАИМОДЕЙСТВИЕ (Interaction Gate) [10].

В зависимости от технологического базиса возможен обоснованный выбор той или иной схемы обратимого вентиля. Перечень схем обратимых элементов не ограничивается элементами, представленными выше. Были предложены и другие схемы обратимых элементов: вентиль Тоффли [11], вентиль Кернтонфа [12], вентиль Переса [13], вентиль Магроласа [14] и несколько вентилях Де Васа [15].

В работе [16] была предложена методика построения основных последовательных элементов на обратимых вентилях Фредкина. Согласно описанию, триггеры, спроектированные по предложенной методике, имеют лучшие показатели, чем триггеры, предложенные в работе [17]. Триггеры сравнивали по числу "мусорных" выходов и по числу используемых вентилях. При этом были приведены схемы триггеров без привязки к какой-либо элементной базе.

В работе [18] предложена схема сумматора на обратимых вентилях Фредкина. В статье [19] описывается программируемое логическое устройство,

построенное на обратимых элементах Фредкина и обратимых элементах ВЗАИМОДЕЙСТВИЕ (вентиль Фейнмана): Reversible Programmable Logic Array (RPLA).

Таким образом, направление обратимой логики имеет достаточно хорошую проработку и может стать одним из основных направлений при создании микропроцессоров для суперЭВМ эксафлопсного уровня. Однако для потоковых машин с большим числом вычислительных узлов, где сегодня достигается максимальная производительность, эффективность обратимой логики может снижаться в силу сложности организации сохранения логических состояний сигналов и заметного увеличения площади схем на кристалле.

Опыт использования самосинхронной логики

Другим направлением снижения потребляемой мощности и повышения надежности является самосинхронное (СС) проектирование. Это направление является областью активных исследований с середины прошлого века. В последние годы интерес к СС-схемотехнике повышается по мере нарастания проблем в синхронной схемотехнике. Во многом это происходит благодаря следующим уникальным свойствам СС-схем [20]:

- устойчивая работа без сбоев в максимально возможной области эксплуатации, определяемой только физическим сохранением переключаемых свойств транзисторов;
- естественная устойчивость к параметрической деградации, вызываемой старением элементов и изменением их параметров;
- прекращение всех переключений в момент появления любой константной неисправности элементов, естественной стопроцентной самопроверяемости и самодиагностируемости по отношению к множественным константным неисправностям, при которых сигнал в какой-либо цепи схемы "залипает" в одном состоянии.

Однако в полной мере данными свойствами обладают только не зависящие от задержек (*Delay Insensitive*, DI) схемы, причем не зависящие от задержек как в элементах, так и в цепях.

К DI-схемам, имеющим практическую значимость, относятся NCL-схемы [21]. Но они имеют слишком большую аппаратную избыточность. С этой точки зрения более практичным является подкласс схем, не зависящих от задержек в элементах (*Element's Delay Insensitive*, EDI). В пределах эквихронной зоны [20] они обладают всеми свойствами действительно DI-схем.

Перечисленные выше свойства СС-схем определяют высокую эффективность создания надежных изделий, в том числе и отказоустойчивых.

В России развитие самосинхронной схемотехники во многом ведется коллективом ИПИ РАН при участии НИИСИ РАН [22, 23]. Созданы микроконтроллер и ряд вычислительных узлов с техно-

логическими нормами 65...180 нм. Рассмотрим основные различия между самосинхронной и синхронной логикой в контексте создания микропроцессоров для ЭВМ эксафлопсного диапазона.

В синхронных схемах интерфейс между физическим и логическим (системным) временем определяется системными часами, удаляющими физическое время из поведения модели. События во внешних часах отделены от модели системного поведения и не имеют заверщенного причинно-следственного отношения к событиям в системе. Все события в синхронной системе инициируются метками физического времени. Действительная длительность инициированных событий никак не отслеживается. Чтобы синхронизируемая аппаратура работала корректно, период синхроимпульсов выбирают из расчета на наихудший случай — максимально возможное время переключения отдельных элементов при неблагоприятных сочетаниях условий функционирования (напряжения питания, температуры, параметров, характера распределения нагрузки и т. п.). Таким образом, цена корректной работы синхронной аппаратуры — недоиспользование ее возможностей по быстродействию. Другим недостатком синхронной схемотехники является существенное повышение потребления питания при переходе на предельные технологические нормы и предельные частоты функционирования. Клоковое дерево в современных микропроцессорах требует до 30 % от общей потребляемой мощности микропроцессора.

Механизмы, обеспечивающие системное время в СС-подходе, включены в модель системного поведения и должны быть разработаны вместе с созданием начальной поведенческой спецификации. Корректные СС-системы базируются на механизме фиксации действительного окончания переходных процессов, т. е. на определении момента завершения переключений элементов схемы. При этом обеспечивается правильное функционирование таких схем независимо от задержек элементов, их составляющих. Таким образом, СС-схемы потенциально должны выигрывать по быстродействию и приводить к снижению потребляемой мощности до 30 %.

Однако при моделировании и исследовании тестовых кристаллов СС-схем, спроектированных и изготовленных с нормами 65...180 нм, выяснилось, что имеется ряд ограничивающих факторов. СС-схемы весьма чувствительны к перекрестным наводкам, схемы правильно отрабатывают свои функции, однако вследствие перекрестных наводок падает быстродействие на десятки процентов, несмотря на то что размер блоков составлял единицы квадратных миллиметров. Таким образом, проектирование СС-схем требует дополнительных усилий по уменьшению различного рода паразитных составляющих, в противном случае заметно падает быстродействие. Представим теперь, что будет происходить на схеме с площадью 100...300 мм² — площади современного высокопроизводительного

микропроцессора. В синхронных схемах принципиальная электрическая схема клакового дерева микропроцессоров очень проста, однако ее топология делается по специальному маршруту [24]. Ряд ведущих компаний, включая компанию Intel, топологию клакового дерева делают вообще заказным образом для достижения максимальной частоты, вкладываются огромные ресурсы для достижения максимальных частот. В случае СС-схем подобные работы провести невозможно в силу сложности логических уравнений. То есть потери в быстродействии на больших размерах кристалла будут катастрофические. Проведенные оценки показывают снижение быстродействия в несколько раз на размерах порядка 10 мм с нормами 65 нм. Для норм 45 нм и ниже проблемы с перекрестными наводками и импульсными помехами становятся особенно критичными. Кроме того, для таких норм возникают проблемы разброса параметров транзисторов и существенного возрастания токов утечки. В результате, для достижения предельных параметров становится необходимо оптимизировать каждый проект под конкретный технологический маршрут. Диапазон вариации электрических параметров внутри одного и того же кристалла составляет 1...3 мВ. Если линейный размер ядра микропроцессора не превышает 2 мм, внутри ядра сохраняются корреляции статических параметров элементов и возможна схемотехническая компенсация технологических вариаций. Для СС-логики эти разбросы должны приводить к снижению быстродействия. Указанные проблемы выдвигают необходимость снижения размера ядер и отдельных функциональных узлов. Именно поэтому многие компании ведут активные работы по созданию микропроцессоров с десятками ядер и сети на кристалле. Для СС-логики это, пожалуй, единственный вариант развития высокопроизводительных микропроцессоров. То есть проект должен разбиваться на многие функционально законченные узлы с небольшой площадью на кристалле (порядка 1 мм²).

Архитектура высокопроизводительного микропроцессора

В настоящее время высокопроизводительные микропроцессоры имеют два основных вектора развития — модернизация архитектур универсальных микропроцессоров и создание потоковых, прежде всего графических, процессоров. Именно на графических процессорах сегодня достигаются предельные производительности. Однако программирование графических процессоров принципиально иное, и зачастую проще подготовить новых программистов, чем переqualифицировать старых. Задача компилятора потоковой машины состоит в переводе программ с внешнего языка в их эквиваленты на язык программ графов.

В НИИСИ РАН создана архитектура микропроцессора 1890ВМ7Я, где находятся одно универсаль-

ное ядро и несколько вычислительных ядер, ориентированных на задачи обработки изображений и сигналов. Производительность микросхемы с технологическими нормами 0,18 мкм составляет 8 Гфлопс, но главное в том, что производительность на реальных задачах превышает 70 % пиковой производительности, в то время как на графических процессорах производительность на тех же задачах — порядка 10 %, а на универсальных — единицы процента. Выделяя области применения, можно добиться существенного увеличения реальной производительности [25]. Учитывая то, что СС-схемы имеют существенно большую надежность, предлагается развитие архитектуры осуществлять с учетом полученных результатов и разбиением проекта на отдельные функционально законченные СС-узлы. Рассматривая задачу горения как одну из наиболее перспективных задач для решения на компьютерах эксафлопсного диапазона, можно выделить ряд базовых функций, аппаратная поддержка которых существенно поднимет производительность системы. К таким функциям можно отнести экспоненту, сложную степень и др. Должен быть также достигнут баланс числа аппаратно поддерживаемых функций и своевременная подкачка данных из памяти — необходимость обеспечения параллельного доступа функциональных блоков к банкам памяти большого объема.

В такой архитектуре вычислительная задача представляется в виде потокового графа, каждый узел которого реализуется на соответствующих вычислительных узлах. Необходимо дать однозначное соответствие каждого узла графа и вычислительного узла и передачу информации между узлами для организации потока вычислений. Для каждой посылки данных должна однозначно быть определена стадия выполнения на вычислительном графе задачи. Поскольку необходимо в общем случае передать данные с выхода одного вычислительного узла потоковой машины на вход любого другого, нужно организовать систему коммутации. Как и в любой системе коммутации, в пересылаемых данных должны быть выделены поля, определяющие направления коммутации. Кроме того, необходимо определить набор данных, передающихся на один и тот же вычислительный узел одновременно для исключения сбоя процесса вычислений. Вычислительные операции (программы узла) активизируются при появлении операндов на всех входах узла — модель вычислений, основанная на управлении данными (*data driven computation*). Нет необходимости использовать счетчик команд, который обязателен в универсальных ЭВМ. Выполнение команды активизируется самими данными, что позволяет осуществлять одновременное исполнение многих команд в машине. Таким образом, во всей цепочке вычислений имеется причинно-следственная связь, которая естественным образом реализуется в СС-логике. В такой постановке задачи при разбиении на отно-

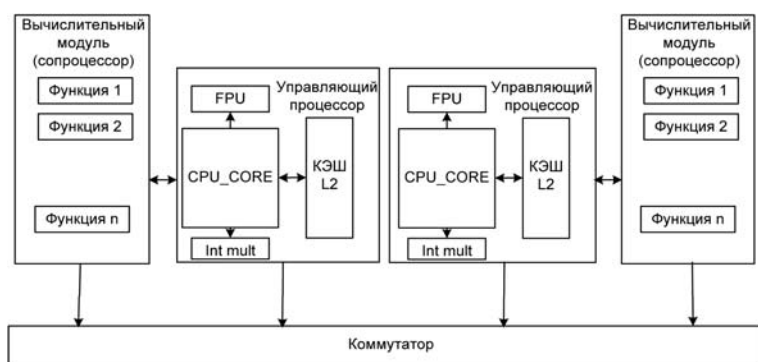


Рис. 2. Структурная схема гибридного процессора

сительно мелкие вычислительные узлы (функции) снижается зависимость СС-логики от наличия различных паразитных элементов, перекрестных наводок и прочих, свойственных современным кристаллам с предельными технологическими нормами. Исходя из этого предложена схема гибридного процессора (рис. 2).

На рис. 2 для примера показана структурная схема процессора только с двумя узлами, при создании микропроцессора с технологическими нормами 28 нм число узлов достигает 16. Для удобства управления и программирования в качестве управляющего процессора используется универсальный процессор с усовершенствованным ядром КОМДИВ64 [26]. Вычислительный модуль организован как потоковая машина на базе СС-узлов. Данная архитектура процессора является переходной, когда используются как синхронные, так и самосинхронные узлы. Однако поскольку вычислительные модули занимают большую часть кристалла по сравнению с универсальными ядрами, а процесс вычислений во многом определяется вычислительными ядрами, надежность такого процессора приближается к надежности полностью СС-процессора. На рис. 2 не показаны коммуникационные каналы и внешние интерфейсы, поскольку в данном рассмотрении они не принципиальны.

Конструктивные и архитектурные решения для рассматриваемого компьютера приведены в работе [25].

Заключение

В работе показано, что создание компьютеров эксафлопсного диапазона является одной из наиболее приоритетных задач развития вычислительной техники. Создание такого компьютера невозможно при использовании существующих подходов, поскольку его надежность была бы минимальной и не позволяла бы вести реальные расчеты, а потребление энергии потребовало бы построить рядом с ним небольшую электростанцию. Решить задачу можно, только применив комплекс мер: необходимо изменить конструкцию компьютеров, перейти на менее энергоемкую среду передачи (оптику), в том числе и внутри крейтов, создать высоконадежные

и низкопотребляющие микропроцессоры. Предложена архитектура гибридных микропроцессоров, сочетающих универсальные микропроцессоры с потоковыми сопроцессорами, реализованные на самосинхронной логике. Ориентация микропроцессора на выделенные задачи позволяет существенно поднять реальную производительность и тем самым уменьшить число вычислительных ядер более чем в 5 раз. В качестве такой задачи выбрана задача горения. Решение этой задачи позволит полностью промоделировать работу двигателей автомобилей, летательных аппаратов и пр.

Список литературы

1. Велихов Е. П., Бетелин В. Б., Кушниренко А. Г. Промышленность, инновации, образование и наука в России. М.: Наука, 2009. 141 с.
2. Stevens R., White A. A DOE Laboratory plan for providing exascale applications and technologies for critical DOE mission needs. URL: http://computing.ornl.gov/workshops/SCIDAC2010/r_stevens.pdf
3. International Exascale Software Project. URL: <http://www.exascale.org>
4. The Defense Advanced Research Projects Agency DARPA [Электронный ресурс]. URL: <http://www.darpa.mil>
5. PRACE. URL: <http://prace-project.eu>
6. Бетелин В. Б., Баранов С. В., Бобков С. Г., Краснюк А. А., Осипенко П. Н., Стенин В. Я., Черкасов И. Г., Чумаков А. И., Яненко А. В. Перспективы использования субмикронных КМОП СБИС в сбоеустойчивой аппаратуре, работающей под воздействием атмосферных нейтронов. // Микроэлектроника. 2009. Т. 38, № 1. С. 1—5.
7. Landauer R. Irreversibility and heat generation in the computing process // IBM J. Res. Develop. 1961. Vol. 5. P. 183—191.
8. Fredkin E., Toffoli T. Conservative logic // Int. J. of Theor. Phys. 1982. Vol. 21. P. 219—253.
9. Haghparast M., Navi K. A Novel Fault Tolerant Reversible Gate For Nanotechnology Based Systems // American Journal of Applied Sciences 5. 2008. N 5. P. 519—523.
10. Maimistov A. I. Reversible logic elements as a new field of application of optical solutions // Quantum Electronics. 1995. Vol. 25, Iss. 10. P. 1009—1013.
11. Toffoli T. Reversible computing. Tech memo MIT/LCS/TM-151, MIT Lab for Comp. Sci, 1980.
12. Kerntopf P. A comparison of logical efficiency of reversible and conventional gates // In International Workshop on Logic Synthesis. 2000. P. 261—269.
13. Peres A. Reversible logic and quantum computers // Physical Review A. 1985. N 32. P. 3266—3276.
14. Margolus N. Feynman and Computation, chapter Crystalline Computation. Perseus Books, 1999.
15. De Vos A. Towards reversible digital computers. In European Conference on Circuit Theory and Design. Budapest, Hungary, 1997. P. 923—931.
16. <http://nthur.lib.nthu.edu.tw/bitstream/987654321/13644/1/2030238030013.pdf>
17. Thapliyal H., Srinivas M. B. A beginning in the reversible logic synthesis of sequential circuits // Proc. of MAPLD. 2005.
18. Bruce J. W., Thornton M. A., Shivakumaraiah L., Kokate P. S. Efficient adder circuits based on a conservative reversible logic gate // Proc. IEEE Computer Society Annual Symposium on VLSI, 2002. P. 74—79.
19. <http://www1.ucmss.com/books/LFS/CSREA2006/CDE5277.pdf>
20. Kishinevsky M., Kondratyev A., Taubin A., Varshavsky V. Concurrent Hardware: The Theory and Practice of Self-timed Design // J. Wiley. 1994.

21. Fant K. M. Logically determined design: clockless system design with NULL convention logic // J. Wiley, 2005.

22. Степченко Ю. А., Дьяченко Ю. Г., Петрухин В. С., Плеханов Л. П. Самосинхронные схемы — ключ к построению эффективной и надежной аппаратуры долговременного действия // Системы высокой доступности. 2007. Т. 3, № 1—2. С. 73—88.

23. Дьяченко Ю. Г., Степченко Ю. А., Бобков С. Г. Квазисамосинхронный вычислитель: методологические и алгоритмические аспекты // В сб. трудов МЭС—2008. М.: ИППМ РАН, 2008. 6 с.

24. Бобков С. Г. Методика проектирования микросхем для компьютеров серии "Багет" // Информационные технологии. 2008. № 3. С. 2—7.

25. Бобков С. Г. Архитектурные и конструктивные решения вычислительных систем с производительностью 1—100 Тфлопс // Информационные технологии. 2011. № 10. С. 26—30.

26. Бобков С. Г., Аряшев С. И., Барских М. Е., Бычков К. С., Зубковский П. С. Микропроцессор гибридный. Патент № 2359315 от 20.06.2009.

УДК 004.6

Г. А. Егоров, д-р техн. наук,
зам. ген. директора по научной работе,
ОАО "Институт электронных
управляющих машин им. И. С. Брука",
e-mail: egorov_q@ineum.ru,

В. И. Шяудкулис, канд. техн. наук,
вед. архитектор-разработчик,
QStar Technologies, Inc.,
e-mail: v.chiaoudkoulis@qstar.com,

М. Финотти, PhD, технический директор,
QStar Technologies, Inc.,
e-mail: m.finotti@qstar.com

Проблемы построения современных архивных хранилищ данных

Рассматриваются проблемы построения хранилищ данных и их решение в современных системах хранения данных. Концепцию архивных облачных хранилищ предлагается использовать в качестве основы для создания новых типов архивных хранилищ данных. Приводится пример реализации архитектуры современной архивной системы хранения данных.

Ключевые слова: хранилище данных, архив данных, облачное хранилище данных

Введение

В последние годы в распределенных системах обработки данных появилась новая тенденция, связанная со стремлением к большей открытости. Открытость проявляется в разных формах, но в основном она выражена в стремлении к унификации интерфейсов операционных систем, интерфейсов и протоколов файловых систем и баз данных, независимости от аппаратуры и др. Так, операционная система UNIX построена на идее независимости от компьютерной архитектуры, а Linux является вершиной реализации такой открытости, протокол HTTP представляет собой пример многоцелевого доступа к неограниченным ресурсам в сети Интер-

нет. Наиболее свежим примером стремления к открытости в настоящее время являются облачные вычисления (cloud computing), которые предоставляют дополнительный уровень независимости, связанной с возможностью виртуализации операционных систем и аппаратного обеспечения.

Одним из ресурсов облачных вычислений, предоставляемых пользователям, являются хранилища данных (файлы и записи баз данных), получившие название "облачные хранилища". Облачные хранилища являются примером высокого уровня виртуализации обычных хранилищ данных. В общем случае обработка данных, включая облачные вычисления, требует следующих ресурсов: процессоров, памяти, программных ресурсов, коммуникационных устройств, устройств хранения данных. В большинстве применений результаты обработки данных должны быть сохранены в каком-то виде в хранилище данных. Только постоянно сохраняемые (persistent) данные предоставляют возможность повторно использовать результаты предыдущей обработки как ссылочные данные в будущем. Например, данные, собранные по телеметрии от спутников, могут многократно обрабатываться с помощью различных алгоритмов для выявления новых свойств исследуемого космического объекта. Таким образом, хранение данных в среде облачных вычислений является неотъемлемой частью обработки данных и, в отличие от других ресурсов облачных вычислений, обладает важной особенностью — постоянством сохранения.

Хранение данных, архивные и облачные хранилища

Общий термин — хранилище данных — включает различные виды памяти: локальные диски, распределенные файловые системы и др. Средства хранения данных имеют свои характеристики, наиболее важными из которых являются:

- объем хранимой информации (мегабайты, гигабайты и т. д.);
- производительность, определяемая скоростью выполнения операций ввода-вывода на устройстве;

- протокол доступа. В настоящее время нет единого универсального протокола доступа к данным, хотя имеется множество протоколов физического и логического уровней.

Эти характеристики хранилищ данных должны учитываться при создании прикладных программ. С практической точки зрения можно выделить быстрые (локальные) хранилища (чаще небольшого объема) для файлов системы, хранения временных файлов, таблиц баз данных и др. В то же время архивные хранилища данных — сравнительно медленные, но большего объема, где хранение информации реализуется на основе сетевой архитектуры. Такие хранилища используются как разделяемый ресурс, и здесь требуется уделять особое внимание защите данных как от сбоев, так и от несанкционированного доступа.

Несмотря на различие структур современных систем хранения данных все они могут быть представлены в виде пятиуровневой архитектуры, приведенной на рис. 1.

Доступ к системе хранения данных обеспечивается с помощью интерфейса файловой системы (FSI — File System Interface) или других протоколов доступа. FSI определяется разработчиком файловой (или операционной) системы и обеспечивает пользователю доступ к хранилищу данных в соответствии с этим протоколом. Например, в UNIX-подобных операционных системах протокол доступа к файловой системе является давно установившимся и состоит из набора системных вызовов (open/close, read/write, fcntl и др.).

Как правило, ядро операционной системы содержит в качестве одного из основных компонентов диспетчер файловой системы (FSM — File System

Manager). FSM работает как сервер запросов от FSI и обеспечивает реализацию различных сервисов файловой системы (поддержание структуры каталогов, управление файлами, управление правами доступа к файлам и др.). Практически виртуальная модель файловой системы реализуется модулем FSM.

Внешнее представление файловой системы — каталоги и файлы (или объекты хранения) — являются виртуальными объектами, которые должны быть отображены на постоянной физической памяти устройства хранения данных. В случае использования дисков такие объекты хранения должны быть расположены на простых секторах диска. В случае, когда постоянная память выполнена на объектной памяти, виртуальные объекты файловой системы должны располагаться на устройствах с объектной структурой (OSD — Object Storage Device).

С точки зрения операционной системы (пользователя) все файлы являются идентичными, но механизм доступа к ним различается. Хранилища данных можно классифицировать по следующим категориям:

- *первичные хранилища* обеспечивают по возможности самый быстрый доступ к объектам данных ограниченного объема. Такие хранилища содержат чаще всего файлы операционной системы, временные (рабочие) файлы, таблицы баз данных. Как правило, данные в первичных хранилищах принадлежат локальным пользователям и характеризуются требованиями большой частоты обращения и высокой производительности. Аппаратные интерфейсы к первичной памяти поддерживают стандарты ATA/PI, SATA, FC (Fibre Channel). Практически первичные хранилища выполняются на локально подсоединенных устройствах (DAS — Direct Attached Storage);
 - *архивные хранилища* обычно разделяются между несколькими пользователями. Эти хранилища характеризуются небольшой частотой обращения к ним, но большим объемом, защищенностью хранимых данных с использованием механизмов копирования и репликации, учитывая их высокую ценность. Архивные хранилища обеспечивают удаленный доступ к ним, что предполагает использование сети Интернет;
 - *автономные (offline) хранилища* содержат данные, которые не требуются пользователям немедленно, или доступ к которым специально ограничивается для защиты от техногенных катастроф (пожаров, наводнений, терактов). Автономные хранилища используют сменные носители, такие как магнитные ленты, Blu Ray-диски, USB-диски и др.
- Настоящая статья связана с проблемами архивных хранилищ с точки зрения их преломления в области облачных хранилищ. Архивные хранилища имеют следующие особенности, в отличие от традиционных первичных файлов:
- обычно они предоставляют доступ только в режиме чтения (read-only);

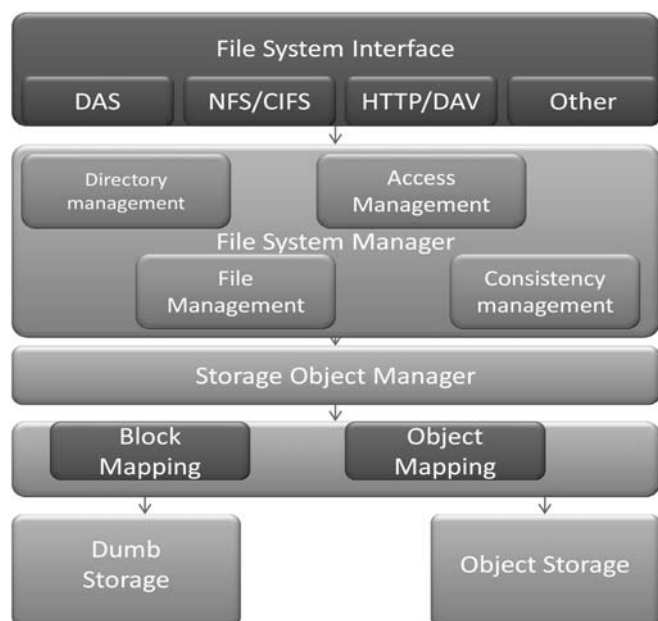


Рис. 1. Многоуровневая архитектура систем хранения данных

- частота обращения к архивным данным небольшая, и она снижается по мере увеличения срока хранения данных;
- должен обеспечиваться постоянный контроль достоверности (аутентичности) хранимых данных, так как со временем устройства хранения данных (носители) теряют надежность;
- архивные данные должны быть надежно защищены от удаления и модификации на период их сохранности. Например, файлы с налоговой информацией должны храниться по законам США семь лет;
- архивные данные должны быть надежно удалены по истечении срока их сохранности;
- во многих странах требуется обеспечить секретность данных, что достигается, как правило, использованием технологий криптографии;
- архивные данные требуются надежно защищать, копируя и дублируя их на других типах носителей;
- архивные данные должны быть устойчивы к изменению технологий. Например, файлы, хранимые на восьмидюймовых носителях 20 лет назад, сегодня не имеют смысла, потому что практически нельзя найти такие устройства. То же самое можно сказать о магнитных лентах того поколения. Следовательно, одной из наиболее важных особенностей архивных хранилищ является возможность постоянной миграции данных на устройства, отвечающие новым технологиям хранения;
- архивные хранилища должны обеспечивать возможность поиска данных по имени и другим атрибутам файла, а также по его содержанию.

Очевидно, что для выполнения перечисленных требований система архивного хранения должна быть в полной мере независима от используемых операционных систем и методов хранения данных, и в этом смысле облачные хранилища данных во многом обеспечивают такую независимость. Идея независимости, скорее всего, должна являться основой при построении архивных хранилищ.

Технология облачных хранилищ предоставляет следующие аспекты независимости, необходимые при создании архивов:

- *независимость от аппаратных интерфейсов.* Облачные хранилища основаны на использовании стека протоколов TCP/IP и HTTP, что обеспечивает независимость от SCSI-кабелей, SAS-интерфейсов, разъемов для USB-кабелей и других средств, специфичных для каждого производителя и подверженных изменениям в будущем;
- *независимость от структуры файловых систем.* Облачные хранилища не связаны с файловой системой. Это позволяет решить две важные задачи:
 - обеспечить неограниченное неструктурированное поле имен объектов;
 - обеспечить возможность построения развитых файловых систем над таким полем объектов;
- *технология миграции.* Так как прикладные программы не взаимодействуют непосредственно с физическими устройствами облачных хранилищ

данных, то внутренние методы хранения данных в облачных хранилищах могут изменяться без влияния на внешние интерфейсы. Это позволяет переходить на новые технологии хранения данных в будущем без изменения прикладных программ. Новые диски большей емкости и производительности могут по мере необходимости заменять старые, так как жизненный цикл дисков не превышает 3—5 лет. Одним из свойств независимости является то, что архивные данные могут храниться на несовместимых устройствах, например на дисках и лентах. Это обеспечивает возможность миграции данных на более дешевые устройства по мере их старения;

- *скрытая внутренняя структура.* Облачные архивы позволяют создавать архивные системы с необходимым уровнем сложности при неизменности внешних протоколов и интерфейсов взаимодействия. При этом отслеживается возможность добавления дополнительных операций с данными, связанных с их репликацией, сжатием (уплотнением), шифрованием, индексацией содержимого данных.

Реализация архивных хранилищ на базе облачных технологий связана с рядом технических проблем:

- хранилища данных должны быть постоянными (persistent), в отличие от других ресурсов их нельзя сделать мобильными;
- по мере увеличения объемов хранилищ перемещение их из одной виртуальной среды в другую требует значительно большего времени;
- доступ к хранилищам должен быть совместим со стандартами операционных систем. Не может быть виртуальной операционной системы, есть конкретно Windows, Linux, UNIX, MAC. Каждая из этих операционных систем поддерживает собственные стандарты, структуры данных, методы доступа и специальные расширения, хотя внешне все эти операционные системы имеют иерархическую систему каталогов и файлов;
- защита доступа к данным также имеет свои особенности реализации в различных операционных системах (учет пользователей, группы пользователей, сетевые домены и др.).

Традиционные файловые системы не могут быть основой для создания архивных систем, так как они не предоставляют специфических функциональных возможностей для архивных хранилищ. В то же время облачные хранилища могут быть использованы как устройства хранения, но они не интегрированы в стандартные операционные системы. Интерфейсы файловых систем не поддерживают важных функциональных возможностей, таких как сжатие, шифрование и сохранность данных.

В связи с этим необходимо выделить ряд задач, требующих решения при реализации архитектуры системы хранения, приведенной на рис. 1, на базе технологии облачности:

- как создать облачные хранилища, удовлетворяющие требованиям архивных хранилищ;

- какие проблемы надо решить при реализации интерфейсов к таким системам архивных хранилищ;
- как интегрировать такие архивные хранилища со стандартными операционными системами.

Чтобы дать ответ на эти вопросы, необходимо провести анализ современного состояния хранилищ данных.

Современное состояние систем хранения данных

Первичные хранилища (DAS), как правило, строятся на базе следующих типов устройств внешней памяти:

- жесткие диски или дисковые массивы жестких дисков (RAID);
- твердотельные диски (SSD), эта технология пока весьма дорогая и только начинает использоваться;
- CD/DVD, Blu Ray-диски, магнитооптические диски (MO) — практически все дисковые носители являются съемными;
- магнитные ленты.

В приведенный список устройств внешней памяти не включены накопители на голографической памяти и другие нетрадиционные устройства хранения, так как они пока не получили большого распространения.

Наибольшее распространение в системах хранения данных находят диски, объем и производительность которых постоянно увеличиваются. Благодаря последним разработкам в области технологии магнитных головок и перпендикулярного метода записи данных, связанного с увеличением плотности записи, емкость одного диска достигает нескольких терабайтов, а емкость RAID-массивов — нескольких десятков терабайтов.

Диски являются достаточно простыми устройствами с точки зрения файловой системы. Практически диск представляет набор физических блоков (как правило, длиной 512 байт), пронумерованных от 1 до некоторого числа N . При большом числе блоков возникают следующие проблемы, связанные со структурой файловых систем:

- отказ дисков большой емкости влечет за собой существенный ущерб, так как теряется много пользовательских данных. Простое копирование всех данных больше не имеет смысла, и требуются новые методы резервного копирования;
- файловая система управляет локальным диском. Одновременный доступ нескольких пользователей к данным, расположенным на одном диске, невозможен без специального программного обеспечения;
- RAID-массивы могут разделяться в сетях SAN, но это не означает, что файлы могут разделяться между узлами SAN, потому что они управляются на локальном узле;
- хотя RAID-массивы обеспечивают значительное увеличение емкости хранилищ, при выходе из строя даже одного из дисков массива его замена связана с восстановлением системы с потерей ее производительности;

- как правило, RAID-массивы полностью зависят от производителей контроллеров. Это приводит к тому, что стоимость одного диска, входящего в RAID-массив, может значительно превышать стоимость такого отдельного диска;
- дисковые системы являются самым большим потребителем электроэнергии в составе вычислительного комплекса. Обычно RAID-контроллеры не позволяют останов дисков, так как постоянно выполняется проверка целостности RAID-массива. Даже если диск в RAID-массиве может остановиться или перейти в "спящий" режим, RAID-контроллер или файловая система вернут его в рабочее состояние. То есть эти диски не обеспечивают режим уменьшения энергопотребления;
- для обслуживания больших хранилищ данных на базе дисков и RAID-массивов требуется квалифицированный персонал, что еще больше увеличивает стоимость таких систем.

Развитие дисковой памяти в настоящее время идет в направлении "больше — быстрее", в то время как функционально диски продолжают представлять собой простой набор блоков. Все это приводит к еще большей нагрузке на процедуру отображения (см. Block Mapping на рис. 1), которая должна надежно управлять все возрастающим объемом данных.

Магнитные ленты. Магнитные ленты до настоящего времени остаются предпочтительным средством хранения больших резервных копий данных, хотя технология лент насчитывает уже 50 лет и много раз раздавались призывы заменить их дисками, обеспечивающими сравнимые с лентами емкости. Такое положение связано с рядом особенностей магнитных лент, среди которых необходимо выделить следующие:

- магнитные ленты являются наиболее "зеленым" методом хранения данных, потому что после их выемки из устройства управления они не потребляют электроэнергию;
- емкость магнитных лент достигает от 1,5 Тбайт (LTO-5) до 5 Тбайт (T10000C), а при использовании операции сжатия объем хранимых данных на магнитной ленте может увеличиться, по крайней мере, в два раза. В многотерабайтных системах магнитная лента сегодня является единственным практическим способом хранения данных;
- магнитная лента является съемным носителем, и ее можно просто перемещать. Хотя диски также могут быть съемными, но они гораздо более чувствительны к физическим нагрузкам и ударам;
- последние технологические достижения позволяют с высокой степенью надежности использовать магнитные ленты для хранения данных. Здесь, прежде всего, можно выделить продольный, поперечный и диагональный методы коррекции, практически исключающие потерю данных;
- форматы данных на магнитной ленте более мобильные (portability), чем на дисках. Например, формат "tar" пережил несколько поколений

ленточных носителей и используется до настоящего времени. На смену ему приходит формат IBM LTFS, который учитывает последние требования к архивации данных на магнитных лентах.

Очевидным недостатком магнитных лент как носителей с последовательным методом доступа является большое MTTD (Mean Time To Data) — среднее время доступа к первому байту данных. Это время достигает трех минут, после чего данные могут читаться со скоростью 200...300 Мбайт.

Другим недостатком магнитных лент является невозможность доступа к чтению данных при ошибке программного обеспечения. Так, если при программной ошибке новые данные будут записаны в начало ленты, то все старые данные, записанные на магнитной ленте, становятся недоступными.

Файловые системы и разделение данных. Файловая система обеспечивает организацию файлов и управление доступом к хранящимся в них данным. Иерархическая структура современных файловых систем предоставляет пользователям удобные механизмы хранения файлов, обращения к ним, разделения и обеспечения их безопасности. Файловые системы должны обладать аппаратной независимостью, предоставляя возможность пользователям обращаться к файлам по символьным, а не по физическим именам. Пользователю при этом не надо знать, как хранятся данные на физическом носителе, форму и физические методы хранения и передачи данных с накопителей. Файловая система может быть просто виртуальной.

До настоящего времени файловые системы вполне удовлетворяли требованиям простых пользователей. Однако при значительном росте емкости накопителей и числа хранимых объектов (файлов и каталогов) архитектура файловых систем не удовлетворяет требованиям обеспечения целостности и восстанавливаемости. Учитывая, что даже самые надежные диски могут отказывать, проблема резервного копирования данных становится очень острой. Решение проблем, связанных с очень большими файловыми системами, обеспечивается использованием различных технологий. Например, концепция "файлового сервера" предполагает выделение специального компьютера (иногда специального изготовления) для решения проблем производительности, расширяемости и копирования.

Технология SAN, основанная на использовании оптоволоконных каналов связи, решает проблему разделенного доступа к данным, но требует при этом разделяемых файловых систем, таких как Quantum Stor-Next, IBM GPFS, FalconStor Hyper FS, и не решает проблем, связанных с большим числом объектов.

Новый технический подход был реализован в файловой системе Lustre [1]. Целью этой системы является поддержка нескольких миллиардов объектов с предоставлением к ним разделяемого доступа более чем из 10 000 узлов. Для достижения такой

цели разработчики Lustre предложили использовать, в отличие от простых дисков, модель объектно-ориентированных устройств хранения данных (OSD) [2].

Таким образом, дальнейшее развитие файловых систем будет продолжаться в направлении их специализации, ориентированной на поддержку объектов хранения и отображения этих объектов на реальные объектные хранилища с возможностью разделения данных между множеством пользователей.

Объектно-ориентированные устройства хранения данных. Концепция объектно-ориентированных устройств хранения данных (OSD-устройства) была предложена техническим комитетом T10 для решения двух основных проблем:

- в отличие от простых блоков, предоставить более управляемые (интеллектуальные) методы отображения объектов файловой системы на хранилище данных;
- предоставить простые методы хранения данных, ориентированных, прежде всего, на компьютер, а не на пользователя.

OSD-устройства предоставляют услуги через SAN или с использованием протокола TCP/IP, расширяют протокол SCSI путем добавления новых команд и не связаны с какой-либо специальной аппаратурой.

OSD-устройства — это скорее сервис, чем собственно устройство, что позволяет реализовать различные дополнительные требования. Например, виртуализация сервиса OSD-устройств может быть использована для удаленного дублирования и копирования, верификации, шифрования, локальной защиты данных и др. Одной из важных функций OSD-устройств является возможность обеспечения контроля доступа к данным. В отличие от блочных устройств (дисков или RAID-массивов), которые практически полностью открыты для случайного доступа, OSD-устройства обеспечивают пользователям доступ к объектам в соответствии с предоставляемыми правами доступа и полномочиями.

SAN и архивные хранилища. Сеть хранения данных (SAN) является высокопроизводительной коммуникационной архитектурой системы хранения данных на базе оптоволоконных каналов связи, позволяющей подключать удаленные внешние устройства хранения данных (дисковые массивы, ленточные библиотеки и оптические устройства) к серверам таким способом, что устройства хранения рассматриваются как локально подключенные.

Первоначально с помощью архитектуры SAN предполагалось решить многие проблемы хранения данных, однако конечный продукт, который сегодня можно купить, содержит только средства, позволяющие в основном объединить сервер и внешние устройства хранения данных. С точки зрения программного обеспечения здесь практически ничего не изменилось. Конечно, скорость каналов связи значительно увеличилась, но появилась новая проблема. Так как теперь все устройства хранения стали "видны" всем серверам, необходимо решать

проблему владения устройством. SAN также не обеспечивает какие-либо дополнительные сервисы, связанные с разделением файловой системы. Скорее всего, преимущества SAN будут видны, когда устройства хранения станут более интеллектуальными и практически будут обеспечивать обмен не блоками, а объектами.

Требования к файловым системам, необходимые для создания архивных хранилищ

Каждый файл в файловой системе характеризуется определенными свойствами. Эти свойства составляют так называемые стандартные метаданные, связанные с файлом, например, имя, размер, дата и время создания, расположение, тип и др. Дополнительно к этим метаданным в некоторых файловых системах добавляются расширенные атрибуты файлов:

- атрибуты доступа к файлу, например, ACL (Access Control List) в Windows-системах, POSIX ACL в UNIX-системах. Эти атрибуты обеспечивают контроль доступа к файлам как отдельным пользователям, так и групп пользователей;
- потоки метаданных в ОС Windows позволяют связать с данными файла неограниченное число дополнительной информации (характеристик). Например, для видеофайла можно дополнительно определить имя режиссера, исполнителей ролей, описание сценария, а также некоторую изменяющуюся информацию (отзывы о фильме и др.) Примерно такое же назначение для поддержки дополнительных метаданных имеет ветвь ресурсов (resource forks) в ОС MAC.

В UNIX-системах для этих целей используются расширенные атрибуты файла.

Возможность расширения атрибутов файлов в определенных системах обеспечивается с помощью специальных системных вызовов. К сожалению, эти вызовы являются системно зависимыми, что значительно усложняет разработку и поддержку разделяемых файловых систем.

Стандартные и расширенные атрибуты, предоставляемые стандартными файловыми системами, не обеспечивают все требования, необходимые для построения архивных хранилищ, среди которых необходимо, прежде всего, выделить следующие.

- *Методы верификации содержимого файла.* Стандартные файловые системы предполагают, что контроль данных обеспечивается аппаратурой и протоколами (например, NFS, CIFS). В настоящее время только файловая система ZFS (Oracle) обеспечивает полный доступ к содержимому данных с использованием механизма подсчета контрольных сумм при записи/чтении данных. Архивная файловая система должна обеспечивать механизм поддержки контрольных (конфигурируемых) сумм, таких как SHA1, MD5, SHA256, для каждого файла или его частей (например, объектов). Особенно это важно, если

будет осуществляться копирование данных на различные устройства хранения в период жизненного цикла файла.

- *Электронные подписи.* Наряду с верификацией содержимого файла в некоторых применениях для файлов требуется наличие подписи, выполненной программным методом. Такая электронная подпись подтверждает, что файл не изменялся за период его хранения (контрольная сумма осталась неизменной), а также то, что файл был подписан (и существовал) в определенное время. Многие госучреждения (в частности, медицинские) в Европе требуют наличия электронной подписи в хранимых файлах (документах).
- *Атрибуты шифрования данных.* Для многих применений не допускается хранение данных открытым способом. Обычно файловые системы не содержат средств шифрования данных (за исключением NTFS). Сама задача шифрования данных решена давно, например, алгоритм AES 256 предоставляет достаточно средств для решения этой задачи в гражданских применениях. Проблема здесь связана с хранением ключей шифрования. Стандартные файловые системы не предоставляют средств создания, хранения и обновления ключей шифрования. Необходимо при этом отметить, что ключи шифрования по определению должны храниться отдельно от самих файлов с данными, так же как нельзя оставлять ключ в двери автомобиля.
- *Сжатие данных.* Способ и параметры сжатия данных должны являться частью атрибутов файлов. Сжатие данных позволяет уменьшить размер файла и тем самым обеспечить экономию требуемой памяти на носителях хранения данных. Большинство типов данных, за исключением некоторых уже сжатых (например, с расширениями .JPG, .ZIP, .MPG и др.), поддаются процедуре сжатия. В последнее время широко обсуждаются алгоритмы уменьшения требуемой памяти для хранения повторяющихся данных (data dedup). Эта технология, скорее всего, не дает эффекта при ее применении в архивных системах.
- *Время сохранности файла.* В соответствии с новыми требованиями не разрешается удалять или модифицировать файлы в течение времени их сохранности. По истечении этого времени файлы должны быть обязательно удалены автоматически или по согласию администратора.
- *Арест данных.* Так как архив может содержать данные, имеющие особое значение (юридическое), должна обеспечиваться возможность запрета на удаление какого-либо файла до снятия с него ареста.
- *Метод уничтожения содержимого файла.* Обычно при выполнении операции удаления файла файловая система не стирает данные файла, а только освобождает занятое им место на носителе. Архивные системы должны обеспечивать возмож-

ность стирания данных файла в целях невозможности последующего восстановления. Для разных целей могут использоваться определенные алгоритмы, например, запись нулей и единиц, бегающих единиц и др. В крайних случаях носители могут размагничиваться.

- *Требования по копированию файлов.* Архивная система должна позволять специфицировать число требуемых копий файлов, типы носителей, используемых для копирования файлов, а также перемещение файла на носителях в течение его жизненного цикла. Использование удаленного копирования требуется для сохранения файлов при техногенных событиях.
- *Требования к версиям файла.* Некоторые файловые системы позволяют хранить несколько версий файла по мере их модификации. В архивных системах модификация файла не допускается, но если она разрешена, то для возможности восстановления старой версии файла это требование является желательным. В будущем, возможно, понадобятся более специфические требования к хранению файлов или наборов файлов на время их жизненного цикла.

С одной стороны, использование концепции файловых систем для создания архивных хранилищ является весьма привлекательной идеей, что позволило бы использовать стандартные средства манипуляции с файлами и другие услуги. С другой стороны, стандартные средства не позволяют управлять расширенными атрибутами файлов. Очевидно, что развитие файловых систем направлено в основном на решение задачи повышения производительности, и каких-либо серьезных изменений в интерфейсах прикладного программирования (API) в ближайшее время не предвидится.

Архивные хранилища данных должны обеспечивать стандартные методы доступа к данным через стандартные интерфейсы файловой системы (FSI) и предоставлять специальный интерфейс API для управления атрибутами файлов, требуемых в архивных хранилищах. К сожалению, в настоящее время работы в области стандартов в этом направлении не ведутся. В конце 1990-х годов была предложена концепция управления иерархической памятью (HSM), которая преследовала основную цель расширения емкости дисковой памяти за счет копирования файлов на другие носители (в основном магнитные ленты). Однако в связи со значительным снижением стоимости дисковой памяти концепция HSM потеряла свою актуальность. В последнее время концепция HSM получила новое развитие, но для другой задачи, связанной с защитой данных. Вместо того чтобы выполнять долгие процедуры копирования всех данных, HSM поддерживает заданное число копий файлов по мере их появления. К сожалению, для создания высокопроизводительных HSM-систем требуются специальные интерфейсы (типа DMAP), которые сегодня поддерживаются только некоторыми файловыми системами.

Основные направления развития архивных хранилищ

Как показано выше, стандартные файловые системы не удовлетворяют требованиям архивных хранилищ данных, так как не предоставляют средств для расширения их возможностей, необходимых для архивных применений. В большинстве случаев это обусловлено коммерческими и лицензионными соображениями. В этих условиях разработчикам архивных систем остается единственное решение, связанное с разработкой специализированных программных продуктов для поддержки архивных решений.

Рассмотрим основные базовые требования, которые должны учитываться при разработке архивных систем.

- *Время жизни данных больше, чем время жизни аппаратуры.* По мере старения аппаратуры данные (файлы, файловые системы) должны переноситься на новые устройства хранения. Каждое такое перемещение данных связано со снижением вероятности их целостности. Поэтому современная архивная система должна содержать средства миграции данных с необходимыми методами их верификации. Одновременно это означает, что нельзя при разработке архивных хранилищ ориентироваться на специфические особенности аппаратуры. В этом смысле объектный подход приобретает важный смысл — если устройство хранения не рассчитано на хранение объектов, то, скорее всего, потребуется специфическая программная поддержка, и в дальнейшем возможность миграции данных может стать проблемой.
- *Архивная информация должна быть всегда доступной.* Архивные системы не рассчитаны на использование в режиме реального времени, но должны обеспечивать постоянный в течение суток доступ к ним. Практически это означает, что архивные системы должны быть отказоустойчивыми и строиться на основе кластерной архитектуры с постоянным дублированием и верификацией данных. Как правило, должны обеспечиваться резервные коммуникационные пути между серверами и устройствами хранения данных. Архивная система должна быть самовосстанавливаемой, с возможностями автоматического обнаружения неисправностей и обхода неработающих узлов системы. Особое внимание должно уделяться возможности предоставления пользователю конкретной и достоверной информации о неисправностях и ошибках и рекомендаций по их устранению.
- *Аудит информации.* В отличие от простых файловых систем архивные системы должны обеспечивать постоянный контроль данных. Это касается как файлов, так и их копий и других объектов хранения в системе. Такой аудит должен выявлять слабые места в системе и предотвращать потерю данных и отказ системы в целом. Результатом аудита является отчет о состоянии и работоспо-

способности системы, рекомендации по расширению емкости хранилища, списки файлов/объектов, период сохранности которых истекает, и др. Контроль каждого хранимого объекта основывается на формальных методах верификации. Необходимо учитывать, что аудит может значительно нагрузить систему и поэтому должен выполняться в фоновом режиме и с низким приоритетом.

- **Потребление электроэнергии.** Постоянное поддержание архивных хранилищ в рабочем состоянии связано со значительным энергопотреблением. Архивная система должна обеспечивать режим автоматического отключения некоторых устройств в системе без изменения требований к надежности и производительности системы. Например, в ночное время, когда нагрузка на архивную систему значительно снижается, имеет смысл отключить некоторые дисковые устройства и даже серверы. При увеличении нагрузки должен обеспечиваться режим автоматического включения этих устройств в работу. Таким образом обеспечивается значительная экономия потребляемой электроэнергии.
- **Независимый от производителей доступ к информации.** Жизненный цикл архивной информации превышает жизненный цикл не только аппаратного обеспечения, но и самих разработчиков аппаратного и программного обеспечения архивных систем. Отсюда понятно стремление к использованию открытых, общепризнанных протоколов (NFS, SCSI, OSD, HTTP). Стремление к захвату рынка рождает много внутренних (проприетарных) протоколов, применения которых, скорее всего, следует избегать в архивных системах.
- **Интеллектуальные устройства хранения данных.** При разработке архивных систем следует учитывать постоянно растущую интеллектуальность архивных систем. Современные дисковые и ленточные контроллеры содержат все более совершенное встроенное программное обеспечение

(firmware), что позволяет обеспечивать большой набор объектно-ориентированных услуг. Это дает возможность некоторые функции файловых систем переносить в аппаратуру. Однако разработчики файловых систем должны всегда осторожно использовать такие возможности, чтобы не попасть в зависимость от производителей аппаратуры.

- **Предотвращение катастроф.** Потеря архивных данных приводит к развалу бизнеса. Обеспечение удаленной копии данных является требованием по закону для банков США, при этом резервный удаленный центр должен находиться на расстоянии не менее 300 км от основного центра хранения данных и должен быть полностью продублирован как по составу оборудования, так и по обслуживающему персоналу. Очевидно, что при этом производительность каналов связи должна в полной мере обеспечивать нормальное функционирование резервного центра с возможностью проведения необходимого аудита данных. Периодические учебные тревоги должны поддерживаться соответствующим программным обеспечением архивных систем.
- **Расширяемость.** Так как архивные системы в основном рассчитаны на длительный период эксплуатации, возможность их расширения является обязательной составляющей их архитектуры. Расширение архивной системы не связано с требованием полной замены оборудования, но постоянное его обновление должно обеспечивать решение всех задач, связанных с расширением архивной системы.
- **Цена.** Цена архивной системы складывается из таких составляющих, как стоимость аппаратного и программного обеспечения, потребляемой электроэнергии, каналов связи, помещений, удаленного резервного центра, а также эксплуатационные и другие расходы. Так как программное обеспечение архивных систем не относится

Требования к архивным системам хранения	Стандартные файловые системы на базе дисков и RAID-массивов	Ленточные системы	Архивные системы на базе объектно-ориентированных и облачных систем
Долгосрочное хранение данных	Нет	Да	Да — за счет постоянной миграции на новые технологии
Скорость доступа	Очень высокая	Высокая, с большим временем поиска	Высокая
Аудит информации	Да	Затруднен	Да
Потребление электроэнергии	Большое	Малое	Управляемое
Независимость интерфейсов	Нет	Нет	Да
Интеллектуальность устройств хранения	Нет	Нет	Да
Предотвращение катастроф	Затруднено	Да	Да
Расширяемость	Зависит от производителя	Да	Да
Цена	Высокая	Низкая	Умеренная
Верификация данных	Нет	Нет	Да
Шифрование	Нет	Поддержаны в LTO-4,5	Да
Электронные подписи	Нет	Нет	Да
Сжатие данных	Нет	Да	Да
Арест данных	Нет	Нет	Да
Уничтожение данных	Нет	Нет	Да
Встроенная защита и копирование	Нет	Нет	Да
Поддержка версий	Нет	Нет	Да

к продуктам массового потребления (как, например, ОС Windows), рассчитывать на его низкую цену не приходится. Такие системы, как правило, нельзя купить на рынке. Успешная реализация таких проектов может быть достигнута только путем кооперации заказчика-владельца системы и ее разработчиков. Следует быть готовым к длительному времени внедрения такой системы, учитывая обязательные этапы согласования ее архитектуры, постоянного уточнения всех требований к ней и длительной опытной эксплуатации. Обучение персонала является весьма важной составляющей внедрения таких систем, так как у всех пользователей должно появиться уважение к данным.

Таблица дает представление о возможностях обеспечения приведенных выше требований в различных типах архивных систем.

В каждом конкретном случае этим требованиям можно присвоить определенный вес и получить наиболее подходящее решение для конкретной области применения. Объектно-ориентированные и облачные системы являются наиболее перспективным направлением развития архивных систем, связанным, прежде всего, с долгосрочностью и сохранностью данных. Для файловых систем и RAID-массивов определяющим является повышение производительности.

Архитектура архивных систем QStar Technologies, Inc.

В настоящее время на рынке существует ряд фирменных продуктов, позволяющих с их помощью строить архивные хранилища на основе модульной организации. Как правило, "под ключ" могут быть разработаны только очень крупные и дорогие архивные системы. Поэтому многие пользователи используют модульный подход, когда аппаратное обеспечение для архивных систем приобретается под выбранное программное обеспечение. Такая интеграция позволяет легче переносить финансовые трудности, связанные с созданием архивной системы, так как требует незначительных начальных капиталовложений при сохранении возможности дальнейшего расширения архивной системы.

В наибольшей степени рассматриваемому подходу для построения архивных систем удовлетворяют программные средства компании QStar Technologies [3]. Эта компания работает на рынке архивных систем уже 25 лет и имеет большой опыт в этом направлении.

В настоящее время компания предоставляет три основных программных модуля (рис. 2), позволяющих создавать современные архивные системы хранения данных:

- модуль управления архивным хранилищем (QStar HSM);
- модуль управления объектным хранилищем (QStar OSM);
- модуль управления файлами (QNM).

В данной статье ограничимся кратким описанием этих модулей, так как каждый из них является достаточно сложным набором программных компонентов, выполняющих вполне автономные функции. Модули программного обеспечения QStar практически покрывают все уровни, необходимые для построения архивных систем.

QStar HSM (HSM) представляет собой специализированную файловую систему, которая одновременно обладает всеми функциями стандартной файловой системы и предоставляет специализированные средства для построения архивных хранилищ. Этот программный модуль по своей сути является шлюзом для доступа к архивным данным, может быть установлен на большинстве операционных систем (Solaris, AIX, Linux, MAC, Windows) и предоставляет следующие возможности:

- интерфейс с базовой операционной системой (FSI) обеспечивает транспортировку запросов к файловой системе из ядра операционных систем на пользовательский уровень. Это открывает широкие возможности для дальнейшего развития функциональности и услуг системы, так как реализация и поддержка сложных систем, включая облачные системы, практически затруднены в ядре ОС;
- управление файловой системой осуществляется с использованием локального дискового кэша и поддержкой носителей различного типа. Файловая система поддерживает как стандартные, так и расширенные атрибуты файлов для Windows, Posix и других операционных систем. Управление кэшем осуществляется в соответствии с пользовательскими правилами (политиками) с последующей передачей данных на сменные носители или объектные хранилища;
- управление различными устройствами хранения обеспечивается набором "мигрантов". Основной задачей миграторов является поддержка форматов и особенностей хранения объектов на накопителях (CD, DVD, Blu Ray-диски, магнитные ленты, CAS, облачные хранилища). Практически любой новый накопитель данных — от

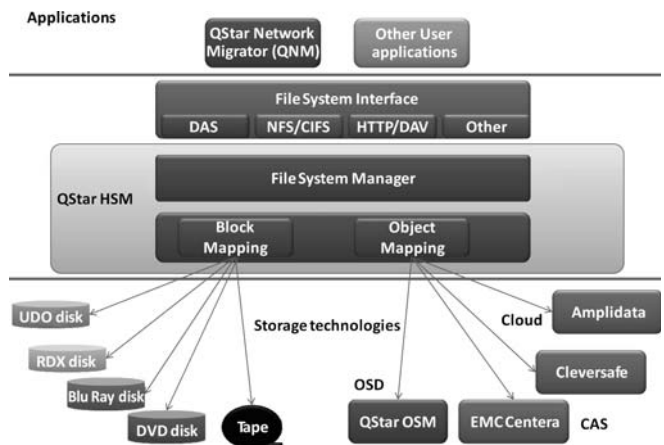


Рис. 2. Структура архивной системы QStar Technologies, Inc.

простого блочного устройства до объектно-ориентированного (в том числе и другие файловые системы) — может быть включен в QStar HSM за счет написания соответствующего мигратора;

- имеется широкий набор средств обработки архивных данных, включая сквозной контроль корректности данных, сохранность и уничтожение данных, сжатие и шифрование данных, поддержку до четырех копий (реплик) на устройствах разного типа на уровне файловой системы, аудит и различные методы восстановления файлов.

QStar OSM (OSM) является программным средством для создания кластеров хранилищ, доступ к которым осуществляется из многих удаленных компьютеров. По сути OSM является высокопроизводительным, расширяемым и отказоустойчивым облачным хранилищем данных индивидуального пользования, построенного специально для создания архивных систем. Увеличение емкости и производительности OSM достигается за счет добавления дисков и компьютерных узлов в кластер. Необходимый уровень надежности обеспечивается за счет репликации объектов. OSM позволяет создавать копии объектов как на локальных, так и на удаленных устройствах хранения, которые могут быть доступны через протоколы NFS или CIFS и, таким образом, являться другой системой OSM. OSM обеспечивает режим оптимизации энергопотребления путем изменения производительности процессоров, уменьшения частоты вращения дисков, отключения узлов в кластере.

QStar Network Migrator (QNM) является программным продуктом, состоящим из двух функциональных частей — системы управления (мастера) и набора агентов. Агенты устанавливаются на компьютерах пользователя и обеспечивают сбор информации о пользовательских файлах в рамках локальной стандартной файловой системы. Эта информация пересылается в центральную базу данных на мастере, где осуществляется ее обработка с подготовкой отчетов о состоянии файлов (число различных типов файлов, объем занимаемой ими памяти, время последних обращений к файлам и др.). Эти отчеты обрабатываются оператором в целях определения политики обработки файлов. Пользователь может определять достаточно сложные правила выделения наборов файлов для обработки с последующими возможными действиями, а именно:

- копировать файл из пользовательской файловой системы в архив. Практически это является процедурой инкрементного резервного копирования;
- мигрировать пользовательский файл в архив. Миграция позволяет освободить "дорогую" память на первичных устройствах хранения за счет пересылки пользовательских файлов в более дешевую память архивных хранилищ. Во время миграции файл копируется в архивное хранилище (например, QStar HSM), сохраняя в пользовательской файловой системе специальную ссылку на этот

файл (stub). Эта ссылка в последующем позволяет пользователю прозрачно открывать такой файл, при этом QNM в зависимости от выбранной пользователем политики или перемещает файл обратно в файловую систему пользователя, или направляет запрос на открытие файла в архивное хранилище;

- переместить файл в архив. Эта операция обеспечивает копирование файла в архивное хранилище и удаляет его на первичном устройстве хранения;
- удалить файл. В этом случае файл удаляется на первичном устройстве хранения без создания архивной копии.

Интеграция трех программных продуктов QStar Technologies, Inc. позволяет создавать достаточно гибкие архивные системы с высокой эффективностью за счет предоставления широких возможностей конфигурирования программных средств и независимости от используемых аппаратных средств.

Заключение

Хранение данных является одним из самых консервативных разделов общей технологии обработки данных. Основным отличием хранилищ данных является их постоянная сохраняемость (живучесть), что не позволяет динамично изменять форматы хранимых данных и стандарты. Показано, что облачные хранилища, предъявляя дополнительные требования к хранению данных, в основном базируются на принципах построения традиционных систем хранения данных.

Неудержимый рост объемов хранимой информации, в особенности архивной, требует новых методов организации и хранения, которые отсутствуют в традиционных файловых системах. По сравнению с первичными системами хранения, где основным критерием является производительность, архивные системы на передний план выдвигают такие требования, как длительность хранения данных, защита, достоверность и независимость от аппаратных средств.

В работе на основе анализа особенностей хранения архивных данных сформулированы требования к современным архивным системам, а также рассмотрена перспектива их развития. Приведен пример практической реализации одной из современных архивных систем.

Список литературы

1. [HITACHI] URL: [http://www.hitachigst.com/tech/techlib.nsf/techdocs/F47BF010A4D29DFD8625716C005B7F34/\\$file/PMR_white_paper_final.pdf](http://www.hitachigst.com/tech/techlib.nsf/techdocs/F47BF010A4D29DFD8625716C005B7F34/$file/PMR_white_paper_final.pdf)
2. [ORACLE] URL: <http://www.oracle.com/us/corporate/press/302409>
3. [Lustre] URL: <http://www.sun.com/software/products/lustre/>
4. [OSD] URL: <http://www.tl0.org>
5. [SAN] URL: <http://www.scribd.com/doc/82157625/Storage-Area-Network>
6. [ZFS] URL: <http://features.techworld.com/storage/2744/zfs-the-future-of-file-systems/>
7. [KEYS] URL: http://csrl.nist.gov/publications/drafts/800-130/second-draft_sp-800-130_april-2012.pdf
8. [QStar HSM] URL: <http://www.qstar.com/products/qstar-software-products/hsm/>

В. И. Левин, д-р техн. наук, проф.,
e-mail: vilevin@mail.ru,
Пензенская государственная
технологическая академия

Упрощенная методика оптимизации систем в условиях интервальной неопределенности

Предлагается и обосновывается методика условной оптимизации функций с интервальными параметрами, отличающаяся тем, что интервальная задача сводится к одной задаче оптимизации с точно известными параметрами, а не к двум, как это было ранее.

Ключевые слова: система, оптимизация, неопределенность, интервальная математика, метод детерминизации

Введение

В настоящее время существует множество различных методов решения задач оптимизации как универсальных, решающих задачи различных классов, так и специализированных, рассчитанных лишь на решение отдельных узких классов задач [1–6]. Все эти методы, однако, применимы только к оптимизации точно известных (детерминированных) функций. Но в практических задачах оптимизируемые функции часто задаются не точно, а с некоторой степенью неопределенности (т. е. недетерминированы). Существует несколько подходов к нахождению оптимума таких функций, они имеют свои достоинства и недостатки [7]. В работах автора [8–12] предложен детерминизационный подход к оптимизации неполностью определенных функций. Этот подход принципиально отличен от уже известных тем, что: 1) оптимизация неполностью определенной функции проводится с учетом всего множества возможных значений ее недетерминированных параметров; 2) для произвольной функции $\tilde{f}$, неопределенность которой связана с неопределенностью ее параметров, которые задаются лишь с точностью до интервалов их возможных значений, нахождение оптимума сводится к нахождению одноименных оптимумов двух полученных из $\tilde{f}$ детерминированных функций. Таким образом, важным преимуществом детерминизационного подхода является то, что для нахождения оптимума неполностью оп-

ределенных (недетерминированных) функций становится возможным применять хорошо известные и эффективные методы нахождения оптимума полностью определенных (т. е. детерминированных) функций. Однако недостаток данного подхода заключается в том, что вместо решения одной недетерминированной задачи оптимизации приходится решать две одноименные детерминированные задачи оптимизации.

Настоящая работа показывает, что для некоторого, достаточно широкого класса неполностью определенных функций $\tilde{f}$ детерминизационный подход позволяет свести оптимизацию функции $\tilde{f}$ к нахождению соответствующего оптимума одной полученной из $\tilde{f}$ детерминированной функции. Тем самым частично снимается указанный выше недостаток подхода.

1. Постановка задачи

Рассмотрим задачу условной оптимизации неполностью определенной функции, параметры которой заданы лишь с точностью до интервалов возможных значений [8]. Пусть задана непрерывная функция n переменных

$$y = F(x_1, \dots, x_n). \quad (1)$$

Далее, пусть функция существует в некоторой области, определяемой следующей системой ограничений:

$$\Phi_i(x_1, \dots, x_n) \leq b_i, \quad i = \overline{1, m}, \quad (2)$$

причем все параметры (коэффициенты) явного представления функции известны точно. Тогда для функции (1) можно сформулировать такую полностью определенную задачу условной оптимизации:

$$F(x_1, \dots, x_n) = \max \\ \text{при } \Phi_i(x_1, \dots, x_n) \leq b_i, \quad i = \overline{1, m}. \quad (3)$$

Сегодня известно много различных методов эффективного решения задач типа (3), привязывающихся к типам функций F и Φ_i , $i = \overline{1, m}$.

Пусть теперь параметры p_k , $k = \overline{1, l}$, явного представления целевой функции F известны не точно, а в виде интервалов значений $\tilde{p}_k = [p_{k1}, p_{k2}]$, $k = \overline{1, l}$. Аналогичным образом неточно заданы параметры q_s явного представления функций Φ_i в левых частях ограничений и параметры b_i в правых частях:

$$\tilde{q}_{si} = [q_{si1}, q_{si2}], \quad s = \overline{1, t}, \quad \tilde{b}_i = [b_{i1}, b_{i2}], \quad i = \overline{1, m}.$$

Тогда функции F и $\Phi_i, i = \overline{1, m}$, становятся функциями-интервалами вида $\tilde{F} = [F_1, F_2], \tilde{\Phi}_i = [\Phi_{i1}, \Phi_{i2}], i = \overline{1, m}$, равно как параметры $b_i, i = \overline{1, m}$, принимающие вид $\tilde{b}_i = [b_{i1}, b_{i2}], i = \overline{1, m}$. В результате полностью определенная задача условной оптимизации (3) переходит в неполностью определенную (интервальную) задачу условной оптимизации

$$\begin{aligned} \tilde{F}(x_1, \dots, x_n) &= \max \\ \text{при } \tilde{\Phi}_i(x_1, \dots, x_n) &\leq \tilde{b}_i, i = \overline{1, m}. \end{aligned} \quad (4)$$

Требуется построить усовершенствованную методику решения оптимизационной задачи (4), отличающуюся от прежней методики [13] тем, что решение задачи (4) сводится к решению не двух, а одной полностью определенной задачи условной оптимизации вида (3).

2. Усовершенствованная математика сравнения интервалов

Для построения усовершенствованной математики сравнения интервалов, которая позволяет решить поставленную задачу, нам потребуются основные сведения из разработанной ранее обычной математики сравнения интервалов [13]. Эти сведения приводятся ниже.

Рассмотрим два интервала $\tilde{a} = [a_1, a_2]$ и $\tilde{b} = [b_1, b_2]$. Сравним интервалы по величине, рассматривая их как интервальные числа. Реализуем сравнение на теоретико-множественном уровне, т. е. рассматривая интервалы как множества, не подлежащие дальнейшему дроблению. Поэтому операции взятия максимума $\vee$ и минимума $\wedge$ двух интервалов $\tilde{a}$ и $\tilde{b}$ введем в виде следующих теоретико-множественных конструкций:

$$\begin{aligned} \tilde{a} \vee \tilde{b} &= \{a \vee b \mid a \in \tilde{a}, b \in \tilde{b}\}, \\ \tilde{a} \wedge \tilde{b} &= \{a \wedge b \mid a \in \tilde{a}, b \in \tilde{b}\}. \end{aligned} \quad (5)$$

Операции $\vee, \wedge$, введенные в формулах (5), существуют всегда, т. е. для любой пары интервалов $\{\tilde{a}, \tilde{b}\}$. Чтобы пару интервалов $\{\tilde{a}, \tilde{b}\}$ можно было сравнить по величине, установив отношения ($\tilde{a} \geq \tilde{b}$ или $\tilde{a} \leq \tilde{b}$), необходимо, чтобы: 1) операции $\vee, \wedge$ над указанными интервалами давали своим результатом один из операндов — $\tilde{a}$ или $\tilde{b}$; 2) операции $\vee$ и $\wedge$ были согласованы, т. е. была справедлива эквивалентность: $(\tilde{a} \vee \tilde{b} = \tilde{a}) \Leftrightarrow (\tilde{a} \wedge \tilde{b} = \tilde{b})$. Условие 2 выполняется всегда (т. е. для любой пары $\{\tilde{a}, \tilde{b}\}$), а условие 1 — не всегда, поэтому не любые два интервала $\tilde{a}, \tilde{b}$ сравнимы.

Теорема 1. Для того чтобы интервалы $\tilde{a} = [a_1, a_2]$ и $\tilde{b} = [b_1, b_2]$ были сравнимы по величине и находились в отношении $\tilde{a} \geq \tilde{b}$, необходимо и доста-

точно, чтобы одноименные границы интервалов удовлетворяли условиям

$$a_1 \geq b_1, a_2 \geq b_2, \quad (6)$$

а для того, чтобы они были сравнимы по величине и находились в противоположном отношении $\tilde{a} \leq \tilde{b}$, необходимо и достаточно выполнения условий

$$a_1 \leq b_1, a_2 \leq b_2. \quad (7)$$

Итак, теорема 1 сводит сравнение двух интервалов и выбор большего или меньшего из них к простой задаче сравнения двух одноименных границ интервалов, являющихся вещественными числами. При этом оказывается, что интервалы $\tilde{a}$ и $\tilde{b}$ находятся в отношении $\tilde{a} \geq \tilde{b}$, когда интервал $\tilde{a}$ сдвинут обеими своими границами вправо относительно интервала $\tilde{b}$, и в отношении $\tilde{a} \leq \tilde{b}$, когда интервал $\tilde{a}$ сдвинут обеими границами влево относительно $\tilde{b}$.

Теорема 2. Для того чтобы интервалы $\tilde{a} = [a_1, a_2]$ и $\tilde{b} = [b_1, b_2]$ были не сравнимы по величине, т. е. не находились в отношениях $\tilde{a} \geq \tilde{b}$ либо $\tilde{a} \leq \tilde{b}$, необходимо и достаточно, чтобы выполнялись условия

$$\{a_1 < b_1, a_2 > b_2\} \text{ или } \{b_1 < a_1, b_2 > a_2\}. \quad (8)$$

Итак, теорема 2 показывает, что интервалы $\tilde{a}$ и $\tilde{b}$ не сравнимы по отношениям $\geq$ и $\leq$ только тогда, когда один из интервалов "накрывает" другой. Из теоремы 1 вытекают теоремы 3, 4.

Теорема 3. Для того чтобы существовал максимальный (т. е. находящийся в отношении $\geq$ со всеми остальными интервалами) интервал в некоторой системе $\tilde{a}(1) = [a_1(1), a_2(1)], \tilde{a}(2) = [a_1(2), a_2(2)], \tilde{a}(3) = [a_1(3), a_2(3)], \dots$ и этим интервалом был $\tilde{a}(1)$, необходимо и достаточно, чтобы его границы были расположены относительно одноименных границ всех остальных интервалов согласно условиям

$$\begin{aligned} a_1(1) &\geq a_1(2), a_1(1) \geq a_1(3), \dots; \\ a_2(1) &\geq a_2(2), a_2(1) \geq a_2(3), \dots \end{aligned} \quad (9)$$

или, в сокращенном виде,

$$\begin{aligned} a_1(1) &\geq \max\{a_1(2), a_1(3), \dots\}; \\ a_2(1) &\geq \max\{a_2(2), a_2(3), \dots\}. \end{aligned}$$

Теорема 4. Для того чтобы существовал минимальный (т. е. находящийся в отношении $\leq$ со всеми остальными интервалами) интервал в некоторой системе $\tilde{a}(1) = [a_1(1), a_2(1)], \tilde{a}(2) = [a_1(2), a_2(2)], \tilde{a}(3) = [a_1(3), a_2(3)], \dots$ и этим интервалом был $\tilde{a}(1)$, необходимо и достаточно, чтобы его границы были расположены относительно одноименных границ всех остальных интервалов согласно условиям

$$\begin{aligned} a_1(1) &\leq a_1(2), a_1(1) \leq a_1(3), \dots; \\ a_2(1) &\leq a_2(2), a_2(1) \leq a_2(3), \dots \end{aligned} \quad (10)$$

или, в сокращенном виде,

$$\begin{aligned} a_1(1) &\leq \min\{a_1(2), a_1(3), \dots\}; \\ a_2(1) &\leq \min\{a_2(2), a_2(3), \dots\}. \end{aligned}$$

Из теорем 3 и 4 понятно, что интервал является максимальным (минимальным) в системе интервалов, только если максимальны (минимальны) его нижняя граница — среди нижних границ всех интервалов и верхняя граница — среди верхних границ всех интервалов.

Перейдем теперь к построению усовершенствованной математики сравнения интервалов. При этом будем рассматривать лишь множества интервалов, сравнимых между собой по величине в указанном выше смысле.

В соответствии с поставленной в п. 1 задачей оптимизации нам необходимо попытаться свести сравнение интервалов по двум параметрам — нижней и верхней границам, согласно теореме 1, к сравнению интервалов по их какому-нибудь подходящему одному параметру.

Базовый результат в этом направлении исследований интервальных чисел содержит нижеследующая теорема 5.

Теорема 5. Пусть $\tilde{a} = [a_1, a_2]$ и $\tilde{b} = [b_1, b_2]$ — сравнимые по величине интервалы, $M(\tilde{a}) = (a_1 + a_2)/2$, $M(\tilde{b}) = (b_1 + b_2)/2$ — центры интервалов. Тогда справедливы следующие эквивалентности:

$$\begin{aligned}(\tilde{a} \geq \tilde{b}) &\Leftrightarrow (M(\tilde{a}) \geq M(\tilde{b})), \\(\tilde{a} \leq \tilde{b}) &\Leftrightarrow (M(\tilde{a}) \leq M(\tilde{b})).\end{aligned}\quad (11)$$

Доказательство. Докажем первую эквивалентность (11). Учитывая результат теоремы 1, имеем

$$\begin{aligned}\tilde{a} \geq \tilde{b} &\Rightarrow a_1 \geq b_1, a_2 \geq b_2 \Rightarrow a_1 + a_2 \geq b_1 + b_2 \Rightarrow \\&\Rightarrow (a_1 + a_2)/2 \geq (b_1 + b_2)/2 \Rightarrow M(\tilde{a}) \geq M(\tilde{b});\end{aligned}$$

далее, с учетом того, что для сравнимых интервалов $\tilde{a}$ и $\tilde{b}$ по теореме 1 справедливо $a_1 \geq b_1, a_2 \geq b_2$ либо $a_1 \leq b_1, a_2 \leq b_2$, получаем

$$\begin{aligned}M(\tilde{a}) \geq M(\tilde{b}) &\Rightarrow (a_1 + a_2)/2 \geq (b_1 + b_2)/2 \Rightarrow \\&\Rightarrow a_1 + a_2 \geq b_1 + b_2 \Rightarrow a_1 \geq b_1, a_2 \geq b_2 \Rightarrow \tilde{a} \geq \tilde{b},\end{aligned}$$

что и требовалось доказать. Вторая эквивалентность формулы (11), в которой стоит знак $\leq$, доказывается аналогично.

Результат теоремы 5 показывает, что сравнение интервалов (с последующим нахождением большего или меньшего интервала) по двум их параметрам — нижней и верхней границам, как это предписывается теоремой 1, можно свести к сравнению их по одному параметру — центру интервала. Однако здесь необходимо отметить, что теорема 5 неверна для не сравнимых по величине интервалов, для которых из того или иного соотношения между центрами двух интервалов не вытекает никакого соотношения между величинами этих интервалов.

Пример 1. Рассмотрим тройку интервалов: $\tilde{a} = [1, 11]$, $\tilde{b} = [2, 4]$ и $\tilde{c} = [8, 10]$. Сравним от-

ношения между этими интервалами с отношениями между их центрами: $M(\tilde{a}) = (1 + 11)/2 = 6$, $M(\tilde{b}) = (2 + 4)/2 = 3$, $M(\tilde{c}) = (8 + 10)/2 = 9$. Таким образом, отношения между центрами интервалов

$$M(\tilde{b}) < M(\tilde{a}) < M(\tilde{c}).$$

Но при дальнейшем рассмотрении значений верхней и нижней границ данных интервалов видим, что интервал $\tilde{a}$ покрывает как интервал $\tilde{b}$, так и интервал $\tilde{c}$, так что $\tilde{a}$ не сравним ни с $\tilde{b}$, ни с $\tilde{c}$. Следовательно, приведенные выше отношения между точными числами — центром $M(\tilde{a})$ интервала $\tilde{a}$ и центрами $M(\tilde{b})$, $M(\tilde{c})$ соответственно интервалов $\tilde{b}$ и $\tilde{c}$, ничего не говорят об отношении самого интервала $\tilde{a}$ с интервалами $\tilde{b}$ и $\tilde{c}$. Однако интервалы $\tilde{b}$ и $\tilde{c}$ в соответствии с теоремой 1 сравнимы по величине. Поэтому отношение величин этих двух интервалов и отношение их центров эквивалентны в смысле (11), т. е. имеем

$$(\tilde{b} \leq \tilde{c}) \Leftrightarrow (M(\tilde{b}) \leq M(\tilde{c})).$$

Этот пример наглядно показывает, что теореме 5 возможно применять только в случае сравнимости интервалов по теореме 1. Из теоремы 5 вытекают следующие теоремы 6 и 7.

Теорема 6. Для того чтобы в системе из n интервалов $\tilde{a}(1) = [a_1(1), a_2(1)]$, $\tilde{a}(2) = [a_1(2), a_2(2)]$, $\tilde{a}(3) = [a_1(3), a_2(3)]$, ..., $\tilde{a}(n) = [a_1(n), a_2(n)]$, попарно сравнимых между собой, существовал максимальный интервал (находящийся со всеми остальными $n - 1$ интервалами в отношении $\geq$) и, кроме того, этим интервалом был $\tilde{a}(1)$, необходимо и достаточно, чтобы центр $M[\tilde{a}(1)]$ данного интервала был расположен относительно центров $M[\tilde{a}(2)]$, $M[\tilde{a}(3)]$, ..., $M[\tilde{a}(n)]$ всех остальных интервалов согласно условиям

$$\begin{aligned}M[\tilde{a}(1)] &\geq M[\tilde{a}(2)], M[\tilde{a}(1)] \geq M[\tilde{a}(3)], \dots, \\M[\tilde{a}(1)] &\geq M[\tilde{a}(n)],\end{aligned}\quad (12)$$

или, в сокращенном виде,

$$M[\tilde{a}(1)] \geq \max\{M[\tilde{a}(2)], M[\tilde{a}(3)], \dots, M[\tilde{a}(n)]\}.$$

Теорема 7. Для того чтобы в системе из n интервалов $\tilde{a}(1) = [a_1(1), a_2(1)]$, $\tilde{a}(2) = [a_1(2), a_2(2)]$, $\tilde{a}(3) = [a_1(3), a_2(3)]$, ..., $\tilde{a}(n) = [a_1(n), a_2(n)]$, попарно сравнимых между собой, существовал минимальный интервал (находящийся со всеми остальными $n - 1$ интервалами в отношении $\leq$) и, кроме того, этим интервалом был $\tilde{a}(1)$, необходимо и достаточно, чтобы центр $M[\tilde{a}(1)]$ данного интервала был расположен относительно центров $M[\tilde{a}(2)]$, $M[\tilde{a}(3)]$, ..., $M[\tilde{a}(n)]$ всех остальных интервалов согласно условиям

$$\begin{aligned}M[\tilde{a}(1)] &\leq M[\tilde{a}(2)], M[\tilde{a}(1)] \leq M[\tilde{a}(3)], \dots, \\M[\tilde{a}(1)] &\leq M[\tilde{a}(n)],\end{aligned}\quad (13)$$

или, в сокращенном виде,

$$M[\tilde{a}(1)] \leq \min\{M[\tilde{a}(2)], M[\tilde{a}(3)], \dots, M[\tilde{a}(n)]\}.$$

Из теорем 6 и 7 видно, что интервал является максимальным (минимальным) среди имеющихся попарно сравнимых интервалов, только если его центр является максимальным (минимальным) среди центров всех имеющихся интервалов. Это открывает возможность упрощенного нахождения максимального (минимального) интервала путем замены всех интервалов их центрами и нахождения максимального (минимального) центра.

Однако, согласно теоремам 6, 7, для того чтобы в системе интервалов максимальный (минимальный) интервал совпадал с интервалом, который имеет максимальный (минимальный) центр, приходится выдвигать дополнительное требование к указанной системе — попарная сравнимость всех интервалов. Если это не выполнено, то теоремы 6, 7 могут оказаться не пригодны для нахождения максимального (минимального) интервала.

Пример 2. Рассмотрим интервалы примера 1: $\tilde{a} = [1, 11]$, $\tilde{b} = [2, 4]$, $\tilde{c} = [8, 10]$ и найдем максимальный и минимальный интервалы. В примере 1 было показано, что отношение между центрами данных интервалов имеет вид цепочки неравенств $M(\tilde{b}) < M(\tilde{a}) < M(\tilde{c})$. Таким образом, если бы теоремы 6, 7 были здесь применимы, то максимальным и минимальным были бы соответственно $\tilde{c}$ и $\tilde{b}$. Однако в примере 1 было выяснено, что сравнима лишь пара интервалов $\tilde{b}$, $\tilde{c}$, причем $\tilde{b} \leq \tilde{c}$. Так что теоремы 6 и 7 здесь не применимы. Будем искать максимальный и минимальный из интервалов $\tilde{a}$, $\tilde{b}$, $\tilde{c}$ перебором. Проверим сначала интервал $\tilde{a}$. Согласно теореме 2 этот интервал не сравним ни с $\tilde{b}$, ни с $\tilde{c}$ (накрывает их). Поэтому он не является ни максимальным, ни минимальным из интервалов $\tilde{a}$, $\tilde{b}$, $\tilde{c}$. Проверим интервал $\tilde{b}$. Согласно теореме 1 этот интервал сравним с $\tilde{c}$, однако, по теореме 2 он не сравним с $\tilde{a}$. Так что интервал $\tilde{b}$ также не является ни максимальным, ни минимальным среди интервалов $\tilde{a}$, $\tilde{b}$, $\tilde{c}$. Аналогично находим, что и $\tilde{c}$ не является ни максимальным, ни минимальным из интервалов $\tilde{a}$, $\tilde{b}$, $\tilde{c}$. Итог: заданная система интервалов $\tilde{a}$, $\tilde{b}$, $\tilde{c}$ не имеет ни максимума, ни минимума. При этом система центров интервалов $M(\tilde{a})$, $M(\tilde{b})$, $M(\tilde{c})$ имеет максимум и минимум: $M_{\max} = M(\tilde{c})$, $M_{\min} = M(\tilde{b})$.

Этот пример наглядно показывает, что распространенная на практике методика решения задач оптимизации в условиях неопределенности путем замены интервалов возможных значений параметров задачи их центрами в общем случае не обоснована и может привести не просто к неверному числовому результату, но и к результату, не имеющему отношения к исходной задаче. Например, можно получить конкретное числовое решение для задачи, которая теоретически вообще не имеет решений.

Но теоремы 6, 7 могут быть существенно обобщены путем замены содержащегося в них требования попарной сравнимости всех имеющихся интервалов более слабым требованием сравнимости одного — "подозреваемого" в экстремальности интервала — со всеми остальными интервалами. Получаем теоремы 8 и 9.

Теорема 8. Для того чтобы существовал максимальный интервал в системе n интервалов $\tilde{a}(1) = [a_1(1), a_2(1)]$, $\tilde{a}(2) = [a_1(2), a_2(2)]$, ..., $\tilde{a}(n) = [a_1(n), a_2(n)]$ и им был интервал $\tilde{a}(1)$, необходимо и достаточно, чтобы этот интервал был сравним со всеми остальными интервалами $\tilde{a}(2)$, ..., $\tilde{a}(n)$ этой системы и, кроме того, центр интервала $M[\tilde{a}(1)]$ был расположен относительно центров всех остальных интервалов $M[\tilde{a}(2)]$, ..., $M[\tilde{a}(n)]$ системы согласно условиям (12).

Теорема 9. Для того чтобы существовал минимальный интервал в системе n интервалов $\tilde{a}(1) = [a_1(1), a_2(1)]$, $\tilde{a}(2) = [a_1(2), a_2(2)]$, ..., $\tilde{a}(n) = [a_1(n), a_2(n)]$ и им был интервал $\tilde{a}(1)$, необходимо и достаточно, чтобы этот интервал был сравним со всеми остальными интервалами $\tilde{a}(2)$, ..., $\tilde{a}(n)$ этой системы и, кроме того, центр интервала $M[\tilde{a}(1)]$ был расположен относительно центров всех остальных интервалов $M[\tilde{a}(2)]$, ..., $M[\tilde{a}(n)]$ системы согласно условиям (13).

Из теорем 8 и 9 видно, что интервал является максимальным (минимальным) среди интервалов, только если он сравним со всеми остальными интервалами, а его центр является максимальным (минимальным) среди центров имеющихся интервалов. Таким образом, использование при отыскании оптимального интервала теорем 8 и 9 вместо теорем 6 и 7 позволяет уменьшить число требуемых сравнимых интервалов с C_n^2 до $n - 1$, т. е. в $n/2$ раз.

Приведенные теоремы 8, 9 дополняются теоремой 10, дающей необходимое условие существования оптимального (максимального или минимального) интервала, и теоремой 11, устанавливающей необходимое и достаточное условие существования оптимального интервала.

Теорема 10. Для того чтобы в системе n интервалов существовал оптимальный интервал, необходимо наличие в системе хотя бы одного интервала, сравнимого со всеми остальными интервалами.

Теорема 11. Для того чтобы в системе n интервалов существовал максимальный интервал, необходимо и достаточно, чтобы в данной системе имелся хотя бы один интервал, сравнимый со всеми остальными интервалами и являющийся большим в каждом из таких сравнений, а для существования в данной системе минимального интервала необходимо и достаточно, чтобы в ней имелся хотя бы один интервал, сравнимый со всеми остальными интервалами и являющийся меньшим в каждом из таких сравнений.

Доказательства теорем 10 и 11 вытекают непосредственно из понятия сравнимости двух интервалов, введенного выше.

На базе теорем 10, 11 строится следующий алгоритм отыскания максимального $\tilde{a}_{\max}$ или минимального $\tilde{a}_{\min}$ интервала в заданной системе интервалов $A = \{\tilde{a}(1), \tilde{a}(2), \dots, \tilde{a}(n)\}$ (ниже приведен только алгоритм отыскания $\tilde{a}_{\max}$; алгоритм отыскания $\tilde{a}_{\min}$ аналогичен).

Шаг 1. Сравниваем интервал $\tilde{a}(1)$ со всеми остальными интервалами системы $\tilde{a}(2), \dots, \tilde{a}(n)$. Если $\tilde{a}(1)$ сравним с $\tilde{a}(2), \dots, \tilde{a}(n)$ и $\tilde{a}(1) \geq \tilde{a}(2), \dots, \tilde{a}(1) \geq \tilde{a}(n)$, то $\tilde{a}(1)$ и есть максимальный интервал $\tilde{a}_{\max}$. *Конец процедуры.* В противном случае переход к шагу 2.

Шаг 2. Сравниваем теперь интервал $\tilde{a}(2)$ со всеми остальными интервалами системы $\tilde{a}(1), \tilde{a}(3), \dots, \tilde{a}(n)$. Если $\tilde{a}(2)$ сравним с $\tilde{a}(1), \tilde{a}(3), \dots, \tilde{a}(n)$, причем $\tilde{a}(2) \geq \tilde{a}(1), \tilde{a}(2) \geq \tilde{a}(3), \dots, \tilde{a}(2) \geq \tilde{a}(n)$, то интервал $\tilde{a}(2)$ — максимальный интервал $\tilde{a}_{\max}$. *Конец процедуры.* Иначе — переход к шагу 3.

Последующие шаги алгоритма по содержанию аналогичны описанным шагам. В частности, на шаге 3 алгоритма проверяется возможная максимальность интервала $\tilde{a}(3)$, на шаге 4 — возможная максимальность $\tilde{a}(4)$ и т. д. Работа алгоритма останавливается, как только выявлен максимальный интервал $\tilde{a}_{\max}$. Если же все n шагов алгоритма, т. е. проверка возможной максимальности всех n интервалов, дала отрицательный результат, т. е. не выявила максимального интервала, это означает, что максимума не существует.

Алгоритм существенно упрощается, если интервалы попарно сравнимы. Тогда в соответствии с теоремами 6, 7 алгоритм состоит из двух шагов.

Шаг 1. Все имеющиеся интервалы $\tilde{a}(1), \tilde{a}(2), \dots, \tilde{a}(n)$ заменяем их центрами, соответственно, $M[\tilde{a}(1)], M[\tilde{a}(2)], \dots, M[\tilde{a}(n)]$.

Шаг 2. Упорядочиваем числа — координаты центров $M[\tilde{a}(k)], k = \overline{1, n}$, по возрастанию: $M[\tilde{a}(k_1)], M[\tilde{a}(k_2)], \dots, M[\tilde{a}(k_n)]$. Тогда интервал $\tilde{a}(k_n)$, имеющий максимальное значение центра $M[\tilde{a}(k_n)]$, будет максимальным интервалом системы, а интервал $\tilde{a}(k_1)$, имеющий минимальное значение центра $M[\tilde{a}(k_1)]$, будет минимальным интервалом.

3. Усовершенствованная методика решения задачи

Вернемся к решению поставленной в п. 1 интервальной задачи условной оптимизации (4). Выделяем два случая:

1) интервальные значения $\tilde{F}(x_1, \dots, x_n)$ в (4) при различных наборах $x_1, \dots, x_n$ попарно не сравнимы;

2) интервальные значения $\tilde{F}(x_1, \dots, x_n)$ в (4) при различных наборах $x_1, \dots, x_n$ попарно сравнимы.

Интервальные значения левых и правых частей неравенств (4) всегда сравнимы — иначе эти неравенства были бы невозможны.

Рассмотрим первый случай. Здесь неполностью определенная (интервальная) задача условной оптимизации (4) на основании теорем 1 и 3 переходит в следующую эквивалентную ей пару полностью определенных (детерминированных) задач условной оптимизации:

$$\left. \begin{aligned} F_1(x_1, \dots, x_n) &= \max \\ \text{при } \Phi_{i1}(x_1, \dots, x_n) &\leq b_{i1}, i = \overline{1, m}, \end{aligned} \right\} \quad (14)$$

$$\left. \begin{aligned} F_2(x_1, \dots, x_n) &= \max \\ \text{при } \Phi_{i2}(x_1, \dots, x_n) &\leq b_{i2}, i = \overline{1, m}, \end{aligned} \right\} \quad (15)$$

где F_1 и F_2 — нижняя и верхняя граничные целевые функции интервальной целевой функции $\tilde{F} = [F_1, F_2]$ задачи (4); Φ_{i1}, Φ_{i2} — нижняя и верхняя граничные функции в интервальной функции ограничений $\tilde{\Phi}_i = [\Phi_{i1}, \Phi_{i2}]$ оптимизационной задачи (4); b_{i1} и b_{i2} — нижняя и верхняя границы интервальной правой части ограничений задачи (4). Таким образом, здесь неполностью определенная (интервальная) задача условной оптимизации (4) сводится в итоге к двум полностью определенным (детерминированным) задачам условной оптимизации (14), (15) типа (3), которые решаются известными и эффективными методами решения данного типа задач.

Во втором случае в неполностью определенной (интервальной) задаче условной оптимизации (4) на основании теорем 5 и 6 возможно перейти от сравнения интервальных значений функции $\tilde{F}$ (с выбором максимального значения) и интервальных значений функций $\tilde{\Phi}_i$ и параметров $\tilde{b}_i$ к сравнению обычных вещественных чисел — центров M соответствующих интервалов. В результате неполностью определенная задача (4) переходит в следующую эквивалентную ей полностью определенную (детерминированную) задачу условной оптимизации:

$$\left. \begin{aligned} M[\tilde{F}(x_1, \dots, x_n)] &= \max; \\ M[\tilde{\Phi}_i(x_1, \dots, x_n)] &\leq M(\tilde{b}_i), i = \overline{1, m}, \end{aligned} \right\} \quad (16)$$

или, после подстановки в (16) явных выражений центров M всех интервалов через их нижние и верхние границы:

$$\left. \begin{aligned} 0,5[F_1(x_1, \dots, x_n) + F_2(x_1, \dots, x_n)] &= \max; \\ 0,5[\Phi_{i1}(x_1, \dots, x_n) + \Phi_{i2}(x_1, \dots, x_n)] &\leq 0,5(b_{i1} + b_{i2}), \\ i &= \overline{1, m}. \end{aligned} \right\}$$

Учитывая факт, что при умножении функции на $k \neq 0$ экстремум будет наблюдаться при тех же значениях аргументов, мы имеем окончательный вид задачи (4) в форме эквивалентной ей полностью определенной (детерминированной) задачи условной оптимизации:

$$\begin{aligned} F_1(x_1, \dots, x_n) + F_2(x_1, \dots, x_n) &= \max; \\ \Phi_{i1}(x_1, \dots, x_n) + \Phi_{i2}(x_1, \dots, x_n) &\leq b_{i1} + b_{i2}, \\ i &= \overline{1, m}. \end{aligned} \quad (17)$$

В формуле (17) переменные $F_1, F_2, \Phi_{i1}, \Phi_{i2}, b_{i1}, b_{i2}$ имеют тот же самый смысл, что и в формулах (15), (16). Таким образом, во втором случае неполностью определенная (интервальная) задача условной оптимизации (4) сводится к одной полностью определенной (детерминированной) задаче условной оптимизации (17) типа (3), которая решается хорошо известными и эффективными методами решения данного типа задач.

Заключение

Изложенная выше усовершенствованная методика решения неполностью определенной (интервальной) задачи условной оптимизации (4) отличается тем, что вместо решения пары полностью определенных (детерминированных) задач условной оптимизации (14), (15), эквивалентной задаче (4),

теперь достаточно решать одну полностью определенную (детерминированную) задачу (17), которая также эквивалентна исходной задаче (4).

Список литературы

1. Юдин Д. Б., Гольдштейн Е. Г. Задачи и методы линейного программирования. М.: Советское радио, 1964.
2. Вентцель Е. С. Введение в исследование операций. М.: Советское радио, 1964.
3. Уайлд Д. Дж. Методы поиска экстремума. М.: Наука, 1967.
4. Корбут А. А., Финкельштейн Ю. Ю. Дискретное программирование. М.: Наука, 1969.
5. Моисеев Н. Н., Иванюков Ю. П., Столярова Е. М. Методы оптимизации. М.: Наука, 1978.
6. Левин В. И. Структурно-логические методы исследования сложных систем. М.: Наука, 1987.
7. Вошинин А. П., Сотиров Г. Р. Оптимизация в условиях неопределенности. М.: Изд-во МЭИ, 1989.
8. Левин В. И. Дискретная оптимизация в условиях интервальной неопределенности // Автоматика и телемеханика. 1992. № 7.
9. Левин В. И. Булево линейное программирование с интервальными коэффициентами // Автоматика и телемеханика. 1994. № 7.
10. Левин В. И. Интервальное дискретное программирование // Кибернетика и системный анализ. 1994. № 6.
11. Левин В. И. Нелинейная оптимизация в условиях интервальной неопределенности // Кибернетика и системный анализ. 1999. № 2.
12. Левин В. И. Антагонистические игры с интервальными параметрами // Кибернетика и системный анализ. 1999. № 3.
13. Левин В. И. Моделирование задач оптимизации в условиях интервальной неопределенности // Известия Пензенского гос. педагогического ун-та. 2011. № 26.

УДК 658.512

А. В. Саушев, канд. техн. наук, доц.,
Санкт-Петербургский государственный
университет водных коммуникаций,
e-mail: saushev@bk.ru

Метод и алгоритмы параметрического синтеза электротехнических систем по критерию запаса работоспособности

Рассматривается метод сужающихся областей и возможные алгоритмы его технической реализации, позволяющие решить задачу параметрической оптимизации для широкого класса электротехнических систем по критерию запаса работоспособности.

Ключевые слова: область работоспособности, метод сужающихся областей, логические R-функции, параметры системы, электротехнические системы

Введение

Проектирование электротехнических систем (ЭТС) на этапе параметрического синтеза сводится к решению двух основных задач — определению номинальных значений внутренних параметров системы и допустимых пределов их изменения. Внутренние параметры — это параметры элементов ЭТС, которые характеризуют состояние и свойства самой системы. При проектировании они определяют вектор X управляемых (варьируемых) параметров.

Математическая функциональная модель ЭТС представляет собой алгоритм вычисления вектора выходных параметров Y при заданных векторах внутренних параметров X и внешних параметров V .

Внешние параметры характеризуют свойства внешней по отношению к ЭТС среды и оказывают влияние на ее функционирование.

Выходные параметры характеризуют свойства ЭТС, интересующие потребителя. Они представляют собой параметры-функционалы, т. е. функциональные зависимости фазовых переменных и параметры, являющиеся граничными значениями

диапазонов внешних переменных, в которых сохраняется работоспособность ЭТС. К выходным параметрам на стадии параметрического синтеза относятся показатели назначения, параметрической надежности и экономичности [1, 2].

Показателем параметрической надежности при ограниченных статистических данных о законах распределения внутренних параметров ЭТС во времени является запас работоспособности [1, 2]. Область работоспособности $G = D_X \cap M_Y$ определяется условиями работоспособности ЭТС, которые в общем случае имеют вид:

$$Y_{j\min} \leq Y_j = F_j(\mathbf{X}) \leq Y_{j\max}, j = \overline{1, m}; \\ X_{i\min} \leq X_i \leq X_{i\max}, i = \overline{1, n}, \quad (1)$$

где $Y_{j\max}, Y_{j\min}$ — соответственно максимально и минимально допустимые значения j -го выходного параметра Y_j ; F — оператор, устанавливающий связь между параметрами $\mathbf{X}$ и $\mathbf{Y}$; M_Y, D_X — допусковые области, определяемые соответственно первым и вторым неравенствами системы (1).

Постановка и метод решения задачи

В качестве целевой функции при оптимизации ЭТС предлагается выбрать минимальный запас работоспособности, который необходимо максимизировать. Для некоторых ЭТС возможна оптимизация по одному из показателей назначения, например по времени переходного процесса или по интегральному показателю качества. Необходимым условием при этом является обеспечение требуемого значения запаса работоспособности. Методологическое обоснование выбора предлагаемой целевой функции приводится в работе [2].

В статье рассматривается поисковый метод оптимизации, названный методом сужающихся областей, быстроедействие которого не зависит от конфигурации области работоспособности, что гарантирует принципиальную возможность его использования и высокую сходимость практически для любых структур ЭТС. Суть метода сводится к следующему.

На первом этапе известными методами [2, 3] определяется граница области G , состоящая из конечного числа гиперповерхностей Φ_j , каждую из которых можно с заданной погрешностью описать уравнениями $\Phi_j(\mathbf{X}) = 0$, где $\Phi_j(\mathbf{X}) = Y_{j\max} - F_j(\mathbf{X})$ или $\Phi_j(\mathbf{X}) = F_j(\mathbf{X}) - Y_{j\min}$ — функции ограничения в системе неравенств (1). В пространстве R^n внутренних параметров вводится метрика l , являющаяся функцией координат двух любых точек этого пространства, например точек A и B . При этом

$$l = \sqrt{\sum_{i=1}^n \mu_i (X_i(A) - X_i(B))^2}, \text{ где } X_i(A), X_i(B) — \text{коор-}$$

динаты векторов точек A и B соответственно; μ_i — нормирующий множитель по i -й координате параметров $\mathbf{X}$. Если одна из точек, например точка A , является граничной точкой области G , а точка B находится внутри этой области и ее координаты характеризуют состояние ЭТС в рассматриваемый момент времени, то данная метрика будет определять запас работоспособности ρ системы и служить критерием сужения исходной области $G^{(0)}$ в целях определения координат оптимальной точки $\mathbf{X}_{\text{опт}}$.

Далее выполняется сужение области $G^{(0)}$ по критерию $l = \rho$. С этой целью изменяют критерий на величину Δl , т. е. $l^{(1)} = l^{(0)} + \Delta l$, получают аналитическое описание границы области $G^{(1)}$ и определяют требуемое множество ее граничных точек.

Получим уравнение, отображающее координаты граничных точек области $G^{(0)}$ в координаты граничных точек области $G^{(1)}$. Предположим, что область $G^{(0)}$ определена на плоскости двух параметров X_1 и X_2 . Для граничной точки $A^{(0)}(X_1^{(0)}, X_2^{(0)})$ области $G^{(0)}$ уравнение нормали к кривой $\Phi^{(0)} \in G^{(0)}$, на которой расположена точка $A^{(0)}$, имеет вид:

$$c_2(X_1^{(1)} - X_1^{(0)}) - c_1(X_2^{(1)} - X_2^{(0)}) = 0,$$

$$c_1 = \partial\Phi^{(0)}(X_1, X_2)/\partial X_1, c_2 = \partial\Phi^{(0)}(X_1, X_2)/\partial X_2.$$

Учитывая, что $(X_1^{(1)} - X_1^{(0)})^2 + (X_2^{(1)} - X_2^{(0)})^2 = \Delta l_1^2$ и решая эти уравнения, можно записать:

$$|X_1^{(1)} - X_1^{(0)}| = c_1 \Delta l / \sqrt{(c_1^2 + c_2^2)},$$

$$|X_2^{(1)} - X_2^{(0)}| = c_2 \Delta l / \sqrt{(c_1^2 + c_2^2)}.$$

Обобщая на произвольный случай и опуская знак модуля, получим

$$X_i^{(1)} = X_i^{(0)} \pm \frac{(\partial\Phi^{(0)}(\mathbf{X})/\partial X_i)\Delta l_1}{|\text{grad}\Phi^{(0)}(\mathbf{X})|},$$

$$|\text{grad}\Phi^{(0)}(\mathbf{X})| = \sqrt{\sum_{i=1}^n (\partial\Phi^{(0)}(\mathbf{X})/\partial X_i)^2}. \quad (2)$$

Рассмотрим вопрос выбора знака в формуле (2).

Теорема. При записи условий работоспособности (1) в виде двоичного предиката $Q(F_j(\mathbf{X})) = 1$ знак в формуле (2) всегда положительный.

Доказательство. Условие $Q(F_j(\mathbf{X})) = 1$ означает, что условия работоспособности (1) записаны в виде $\Phi_j(\mathbf{X}) \geq 0, j = \overline{1, 2m}$. Рассмотрим произвольную граничную точку R^0 , принадлежащую гиперповерхности $\Phi_j^{(0)}$, и определим знак в формуле (2):

$X_i^{(1)} = X_i^{(0)} + \text{sign} \Delta X_i$. Приращение (увеличение или уменьшение) i -й координаты точки R^0 должно увеличивать значение функции $\Phi_j(\mathbf{X})$, а следовательно, и функции $\Phi_j^{(0)}(\mathbf{X})$, поскольку лишь в этом случае точка $R^{(1)}$, принадлежащая гиперповерхности $\Phi_j^{(1)} \in G^{(1)}$ и являющаяся отображением точки R^0 , будет находиться внутри области работоспособности. Рассмотрим производную

$$\partial \Phi_j^{(0)}(\mathbf{X}) / \partial X_i = [\Phi_j^{(0)}(X_i + \Delta X_i, X_g) - \Phi_j^{(0)}(\mathbf{X})] / \Delta X_i; \\ g = 1, 2, \dots, n; g \neq i.$$

Поскольку значение функции $\Phi_j^{(0)}(X_i + \Delta X_i, X_g)$ должно быть положительным, то и числитель в последней формуле должен быть больше нуля. Таким образом, $\text{sign} \Delta X_i$ полностью определяет знак в формуле (2) и в случае сужения исходной области $G^{(0)}$ в направлении оптимума этот знак должен быть положительным. Теорема доказана.

Поскольку во многих случаях гиперповерхности $\Phi_j^{(0)}$, составляющие границу области G , аналитически описываются в результате использования П-отображений полиномиальными моделями [4], которые для ЭТС, как правило, имеют порядок не выше второй степени, получим следующие расчетные формулы. Для полинома первой степени

$$\Phi^{(0)}(\mathbf{X}) = b_0 + \sum_{i=1}^n b_i X_i + \sum_{i=2}^n \sum_{j=1}^{i-1} b_{ij} X_i X_j; \\ \partial \Phi^{(0)}(\mathbf{X}) / \partial X_i = b_i + \sum_{j=1}^n \sum_{i=2j=1}^{i-1} b_{ij} X_j; \\ X_i^{(1)} = X_i^{(0)} + \\ + \left(b_i + \sum_{j=1}^n \sum_{i=2j=1}^{i-1} b_{ij} X_j \right) \Delta l_1 / \sqrt{\sum_{i=1}^n \left(b_i + \sum_{j=1}^n \sum_{i=2j=1}^{i-1} b_{ij} X_j \right)^2}.$$

Для полинома второй степени

$$\Phi^{(0)}(\mathbf{X}) = b_0 + \sum_{i=1}^n b_i X_i + \sum_{i=1}^n b_{ii} X_i^2 + \sum_{i=2}^n \sum_{j=1}^{i-1} b_{ij} X_i X_j; \\ \partial \Phi^{(0)}(\mathbf{X}) / \partial X_i = b_i + 2b_{ii} X_i + \sum_{j=1}^n \sum_{i=2j=1}^{i-1} b_{ij} X_j; \\ X_i^{(1)} = X_i^{(0)} + \left(b_i + 2b_{ii} X_i + \right. \\ \left. + \sum_{j=1}^n \sum_{i=2j=1}^{i-1} b_{ij} X_j \right) \Delta l_1 / \sqrt{\sum_{i=1}^n \left(b_i + 2b_{ii} X_i + \sum_{j=1}^n \sum_{i=2j=1}^{i-1} b_{ij} X_j \right)^2}.$$

В тех случаях, когда вид модели $\Phi^{(0)}(\mathbf{X})$ заранее неизвестен, первоначально следует использовать линейное приближение. При неадекватности модели следует перейти к построению полинома второй степени [4].

Получим уравнения, описывающие область $G^{(1)}$. С этой целью для каждой гиперповерхности определим координаты N_j граничных точек, равных числу значимых коэффициентов этой гиперповерхности. Например, для полинома первой степени $N_j = n + 1 + n(n - 1)/2$, для полинома второй степени $N_j = 2n + 1 + n(n - 1)/2$. Подставляя эти координаты в систему уравнений

$$\phi_j^{(1)} = \Phi_j^{(1)}(\mathbf{X}) = b_0^1 + \sum_{i=1}^n b_i^1 X_{ip} + \sum_{i=1}^n b_{ii}^1 X_{ip}^2 + \\ + \sum_{i=2}^n \sum_{v=1}^{i-1} b_{iv}^1 X_{ip} X_{vp} = 0, \rho = \overline{1, N_j},$$

и решая ее относительно неизвестных коэффициентов $b_0^1, b_i^1, b_{ii}^1, b_{iv}^1$, получим искомые уравнения, описывающие границу области $G^{(1)}$.

На основе использования логических R -функций [5] и полученных уравнений $\phi_j^{(1)}$ записывается одно уравнение, которое с заданной методической погрешностью аналитически описывает допустимую область $G^{(1)}$ на первом шаге поиска:

$$\begin{cases} G^{(1)} = G_{2(m+n)}^{(1)} = 0,5(G_{2(m+n)-1}^{(1)} + \phi_{2(m+n)}^{(1)} - \\ - |G_{2(m+n)-1}^{(1)} - \phi_{2(m+n)}^{(1)}|); \\ \dots G_j^{(1)} = 0,5(G_{j-1}^{(1)} + \phi_j^{(1)} - |G_{j-1}^{(1)} - \phi_j^{(1)}|); \\ \dots G_1^{(1)} = \phi_1^{(1)}. \end{cases}$$

Затем аналогичным образом определяется допустимая область $G^{(2)} \in G^{(1)}$ и процесс сужения исходной области $G^{(0)}$ циклически повторяется до получения оптимального решения при $N = 1$.

Если на μ -м цикле оптимизации $G^{(\mu)} = \emptyset$, где $\emptyset$ — пустое множество, то на $(\mu + 1)$ -м цикле необходимо уменьшить значение критерия на величину Δl . Для вычисления значения Δl можно, например, воспользоваться формулой $\Delta l = 0,5(\Delta l^{(\mu-1)} - \Delta l^{(\mu)})$, $(\Delta l^{(\mu-1)} > \Delta l^{(\mu)})$. Если и в этом случае $G^{(\mu)} = \emptyset$, то значение Δl необходимо снова уменьшить вдвое. Если $G^{(\mu)} \neq \emptyset$, то процесс оптимизации продолжается аналогично рассмотренному выше случаю.

Момент прекращения процесса поиска определяется при таких значениях внутренних параметров, при которых область $G^{(\mu)}$ в соответствии с заданной погрешностью вырождается в точку. Значения параметров $X_{\text{опт}}$ в этой точке определяет оптимальное (максимальное) значение критерия l , которому соответствует условие $N = 1$.

Для дальнейшего изложения метода и описывающих его алгоритмов необходимо уточнить исходные данные и рассмотреть различные варианты, которые могут иметь место при оптимизации конкретных ЭТС.

1. Допустимая погрешность определения координат оптимальной точки задается независимо по каждому внутреннему параметру. Она определяется технической возможностью и целесообразностью, а также чувствительностью выходных переменных ЭТС к изменению данного параметра и не зависит от конфигурации области G .

В общем случае $\varepsilon_i \leq \varepsilon_{i\text{доп}} = (X_{i\text{max}} - X_{i\text{min}})/h_i$, где h_i — число отрезков (квантов), на которые разбивается диапазон изменения параметра X_i . Если для всех внутренних параметров $h_i = h$, то число квантов определяется исходя из обеспечения требования, при котором $\varepsilon \leq \min(\varepsilon_{i\text{доп}})$.

2. Значение шага поиска l определяется формой области G и, как следствие, используемым алгоритмом поиска, уточняется на каждом шаге поиска и выбирается из условия $0,25h\varepsilon \geq l \geq 0,5\varepsilon$, $\varepsilon = \min(\varepsilon_{i\text{доп}})$.

3. Условие принадлежности исследуемой точки пространства R^n на k -м шаге поиска области $G^{(k)}$ и условие анализа граничных точек области $G^{(k)}$ на их принадлежность оптимальному решению формируются на основе предикатного уравнения с помощью функции $Q(Y) = Q(F_j^k(X))$.

В общем случае функция $Q(Y)$ определяется конъюнкцией $2(n + m)$ уравнений ϕ_j^k . Поскольку, как правило, $n \gg m$, то основные временные затраты обусловлены анализом принадлежности исследуемой точки допускowej области D_X на k -м шаге поиска (области D_X^k). Эти затраты времени можно существенно снизить. Сформулируем следующую теорему.

Теорема. При аналитическом описании на основе R -отображения допусковых областей $G^{(k)}$ на k -м, $k = 1, 2, \dots$, шаге поиска оптимального решения нет необходимости в анализе на принадлежность

этим областям всех граней области D_X^k , достаточно выбрать одну (любую) грань.

Доказательство теоремы выходит за рамки настоящей статьи.

4. Процесс поиска координат оптимальной точки в целях снижения временных затрат (исключение "холостых ходов" поиска, более точное описание гиперповерхностей, составляющих допускową область M_Y , исключение из рассмотрения гиперповерхностей, не принадлежащих области G) следует вести по области $Q \in D_X$, которая представляет собой описывающий область G гиперпараллелепипед (брус). Для этого в алгоритм поиска необходимо ввести процедуру выделения "крайних" граничных точек X'_{min} и X'_{max} , принадлежащих области G . По мере сужения исходной области G в процессе поиска необходимо сужать и область поиска Q . Критерием сужения является выбранный шаг поиска.

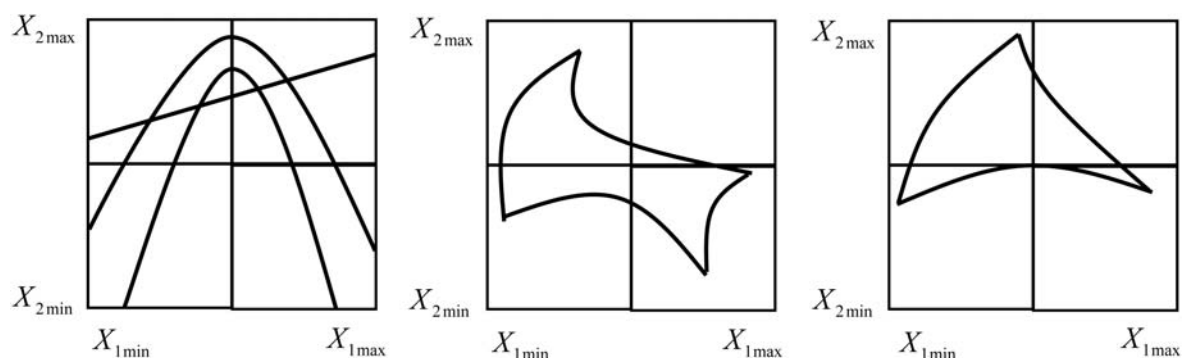
Следует отметить, что при большой размерности пространства параметров даже незначительное сужение области поиска Q приводит к заметному сокращению временных затрат, поскольку основной объем области работоспособности сосредоточен на ее периферии.

5. Априорные данные о конфигурации и задании области G .

Можно выделить следующие варианты, которые имеют место при практической реализации метода для оптимизации различных ЭТС:

- область работоспособности является односвязной (выпуклой или произвольной формы) или не односвязной (самый общий случай);
- имеется информация о границе области работоспособности в виде совокупности граничных точек и аналитических уравнений аппроксимирующих ее гиперповерхностей или такая информация отсутствует;
- область работоспособности задана в виде совокупности линейных или нелинейных гиперповерхностей.

Рисунок иллюстрирует качественно различные формы областей работоспособности на примере двух внутренних параметров X_1 и X_2 .



Возможные варианты областей работоспособности ЭТС

Алгоритмы решения задачи

Рассмотрим наиболее значимые фрагменты алгоритмического обеспечения метода сужающихся областей для случая произвольной формы области работоспособности и отсутствия информации о ее границе.

Для проверки условия $G^1 \neq \emptyset$ в области, ограниченной брусом $D^1 \in D_X$, где длина каждого ребра D^1 определяется как $[(X_{i\min} + \Delta l_1); (X_{i\max} - \Delta l_1)]$, $i = 1, n$, или в относительных единицах, $[-0,5; 0,5]$, осуществляется зондирование пространства параметров X в целях определения точки, принадлежащей области G^1 .

Зондирование выполняется методом статистических испытаний на основе использования ЛПТ-последовательностей, которые позволяют генерировать точки, равномерно распределенные в допускowej области D_X произвольной размерности R^n [6]. В качестве алгоритма формирования последовательности точек используется рекуррентная зависимость, основанная на использовании кода Грея. Пусть $\Gamma(k)$ — код Грея, соответствующий номеру генерируемой точки k . По определению $J(k) = k \lfloor k/2 \rfloor$, где $\lfloor Z \rfloor$ — целая часть Z . Два соседних кода $J(k)$ и $J(k-1)$ всегда различаются в одном и только в одном разряде $e = e(i)$, номер которого можно вычислить по формуле $e = 1 + \log_2[J(k) \cdot J(k-1)]$.

Процесс поиска заканчивается в том случае, если в процессе зондирования области $D^{(1)}$ будет определена точка, принадлежащая области $G^{(1)}$, или в случае, если число испытаний достигнет предельного значения N_1 , определяемого заданной погрешностью поиска $\alpha_{\text{зад}}$. При этом $N_1 = \log(1-p)/\log(1-\alpha_{\text{зад}}^n)$, где p — заданная доверительная вероятность. Обычно $p = 0,95$. Значение N_1 можно вычислить приближенно по формуле $N_1 = 1/(\alpha_{\text{зад}}^n)$. Аналогичным образом формируются уравнения, описывающие допускowe области G^μ , $\mu = 2, 3, \dots$. При этом значение Δl_μ , определяющее степень сужения области работоспособности, корректируется в зависимости от предыдущего результата в соответствии с алгоритмом половинного деления или другим алгоритмом одномерного поиска [1].

Зондирование пространства внутренних параметров на μ -м шаге сужения исходной области $G^{(0)}$ ограничивается брусом D^μ , при этом $X_{i\min}^\mu = X_{i\min} +$

$$+ \sum_{s=1}^{\mu} \Delta l_s, \quad X_{i\max}^\mu = X_{i\max} - \sum_{s=1}^{\mu} \Delta l_s.$$

Отношение объемов брусков $D^{(1)}$ и $D^{(\mu)}$, определяющее степень уменьшения области поиска оптимальной точки на μ -м шаге, вычисляется в относительных единицах по

$$\text{формуле } V(D^{(1)})/V(D^{(\mu)}) = (1,8)^n \left/ \left(2 - 2 \sum_{s=1}^{\mu} \Delta l_s \right)^n \right.$$

Процесс окончания поиска определяется из условия $V(D^{(\mu)}) \leq V_{\text{зад}}$ или, если шаг изменения всех параметров одинаков и они заданы в относительных единицах, из условия $\Delta l_\mu \leq \alpha_{\text{зад}}$. При этом любая точка, принадлежащая области $V_{\text{зад}}$, может быть принята в качестве оптимальной.

Рассмотренный алгоритм гарантирует получение оптимального решения при любой конфигурации области работоспособности.

При большой размерности R^n пространства параметров X целесообразна аппроксимация каждой гиперповерхности $\Phi_j^{(0)} \in G^{(0)}$ классом гиперсфер [7]. Число граничных точек, которые должны быть определены для целей аппроксимации, в этом случае по сравнению с основным алгоритмом сокращается как минимум в $h = (n+1+0,5n(n-1))/(n+1)$ раз. Например, при $n = 10$ $h \approx 5,0$, а при $n = 50$ $h \approx 25$. Основное достоинство такой аппроксимации заключается в том, что очень просто определяются уравнения подобластей G^k при сужении исходной области для поиска оптимального решения. Анализ тестовых примеров показал, что при $n > 10$ достоинства такого подхода являются решающими при его практическом использовании. При необходимости уточнения полученного решения это может быть выполнено введением в алгоритм дополнительной процедуры, реализующей разработанный алгоритм адаптивного симплексного поиска или метод спиральной развертки [8].

При большой размерности R^n пространства параметров в рассмотренный алгоритм следует ввести специальную процедуру поиска, позволяющую значительно повысить его быстродействие и обеспечить практическую реализацию алгоритма на стандартных персональных компьютерах. Данная процедура предполагает априорное разбиение области D_X на три области G_1, G_2, G_3 и псевдослучайный поиск точек, принадлежащих области G , при ее последовательном сужении в строго определенном порядке — сначала в области G_1 , затем в области G_2 и, наконец, в области G_3 . Область $G_1 \subset D_X$ подобна области D_X и имеет с ней общий центр. Введем в рассмотрение область G_X , которая подобна областям D_X и G_1 , имеет с этими областями общий центр и расположена между ними. Области G_2 и G_3 можно определить следующим образом: $G_2 = G_X - G_1$, $G_3 = D_X - G_X$.

Основная идея такой схемы поиска следует из сущности метода сужающихся областей и состоит в том, что в первую очередь анализируются точки, находящиеся в удалении от границы области D_X . Множество этих точек, ограниченное допускowym брусом G_1 , подлежит анализу при любом результате поиска. Вместе с тем множество точек, расположенных в непосредственной близости к границам области D_X , в процессе поиска в результате сужения области G и, как следствие, в результате сужения области D_X , может быть исключено из рассмотрения.

Поскольку с ростом размерности R^n пространства внутренних параметров основной объем области D_X примыкает к ее граням, то его исключение из рассмотрения при поиске приводит к существенному сокращению временных затрат.

Вначале поиск ведется в области G_1 , ограниченной пределами

$$X_i \in [X_{i\min}^1; X_{i\max}^1], X_{i\min}^1 = 0,25(X_{i\max} - X_{i\min}), \\ X_{i\max}^1 = 0,75(X_{i\max} - X_{i\min}), i = 1, n.$$

При задании параметров X в относительных единицах получим $X_i^1 = [0,5; -0,5]$. Предельные значения параметров, определяющие границу области G_1 , устанавливаются из условия гарантированного получения глобального оптимума в случае неодносвязной области G . Выбор пределов изменения параметров для областей G_2 и G_3 проводится в зависимости от полученного на первом этапе поиска значения запаса работоспособности, который представляется в относительных единицах, обозначается λ и принадлежит интервалу $[0; 1]$. Эти пределы выбирают из условия равновероятного расположения области G в области D_X :

$$G_2 = \langle X_i \in [-b; -0,5] \wedge [0,5; b], i = 1, n \rangle,$$

$$G_3 = \langle X_i \in [-c; -0,75] \wedge [0,75; c], i = 1, n \rangle.$$

Если $\lambda \leq 0,25$, то $b = 0,75$. Если $0,25 < \lambda < 0,5$, то $b = 1 - \lambda$. Если $\lambda \geq 0,5$, то поиск прекращается и точка, соответствующая этому запасу работоспособности, принимается в качестве оптимальной. При необходимости полученное значение оптимума уточняется. Во всех случаях $c = 1 - \lambda$.

На втором этапе поиск осуществляется в области G_2 . Если при анализе точек окажется, что $\lambda \geq 0,5$, поиск прекращается, если после завершения анализа точек окажется, что $\lambda < 0,25$, то переходят к третьему этапу — анализу точек в области G_3 .

Координаты точек в области G_1 формируются случайным образом по следующему правилу: $X_i = (\delta - 0,5)$, $i = \overline{1, n}$. Здесь $\delta \in [0; 1]$ — псевдослучайная величина, генерируемая по рассмотренному выше алгоритму.

В областях G_2 и G_3 правило формирования координат точек имеет вид: $X_i = \delta \text{sign}(\delta - 0,5) + \xi(\delta - 0,5)$, $i = \overline{1, n}$. При поиске в области G_2 $\delta = 0,5$, $\xi = 2b - 1$. При поиске в области G_3 $\delta = 0,75$, $\xi = 2c - 1,5 = 0,5 - 2\lambda$.

Критерием окончания поиска точек в областях G_1 , G_2 и G_3 , кроме значения запаса работоспособ-

ности λ , является также общее число "пробных точек" N_1, N_2, N_3 соответственно, которые определяются заданной погрешностью поиска $\alpha_{\text{зад}}$ и объемами этих областей. При этом

$$N_1 = \frac{\log(1-p)}{\log(1-\alpha_{\text{зад}}^n)};$$

$$N_2 = \log(1-p)[\log(2^n b^n - \alpha_{\text{зад}}^n)^{-1} - \log(1 - \alpha_{\text{зад}}^n)^{-1}];$$

$$N_3 = \log(1-p)[\log(2^n - \alpha_{\text{зад}}^n)^{-1} - \log(2^n b^n - \alpha_{\text{зад}}^n)^{-1}],$$

где p — заданная доверительная вероятность. Обычно $p = 0,95$.

Значения N_1, N_2, N_3 можно приближенно вычислить по формулам

$$N_1 = 1/(\alpha_{\text{зад}})^n; N_2 = (1 - (2b)^n)/(\alpha_{\text{зад}})^n; \\ N_3 = (1 - b^n)(2/\alpha_{\text{зад}})^n.$$

Заключение

Рассмотренный метод сужающихся областей позволяет решить задачу параметрической оптимизации для широкого класса электротехнических систем по критерию запаса работоспособности. При этом обеспечивается выполнение всех требований, предъявляемых к показателям назначения системы, и гарантируется получение единственного решения. Практическая апробация метода показала его эффективность и работоспособность.

Список литературы

1. Норенков И. П. Основы автоматизированного проектирования: М.: Изд-во МГТУ им. Н. Э. Баумана, 2002. 336 с.
2. Саушев А. В. Методы управления состоянием электротехнических систем. СПб.: СПГУВК, 2004. 126 с.
3. Саушев А. В. Метод построения границы области работоспособности электротехнических объектов // Электричество. 1990. № 4. С. 14—19.
4. Саушев А. В. Планирование эксперимента в электромеханике. СПб.: СПГУВК, 2008. 216 с.
5. Рвачев В. Л. Геометрические приложения алгебры логики. Киев: Техника, 1967. 212 с.
6. Соболев И. М., Статников Р. В. Выбор оптимальных параметров в задачах со многими критериями. М.: Дрофа, 2006. 176 с.
7. Саушев А. В. Метод аппроксимации области работоспособности // Математическое и информационное обеспечение автоматизированных систем: Сб. научных трудов. Вып. 12. СПб.: СПГУВК, 2004. С. 58—69.
8. Саушев А. В. Сеточный метод построения областей работоспособности технических объектов на основе алгоритма симплексного поиска // Журнал Университета водных коммуникаций. 2010. Вып. 1 (V). С. 58—69.

Е. М. Борчик, аспирант,
 А. И. Якимов, канд. техн. наук, доц.,
 А. И. Степанов, соискатель,
 В. В. Башаримов, соискатель,
 ГУВПО "Белорусско-Российский университет",
 г. Могилев, Беларусь
 E-mail: ykm@tut.bu

Построение плотностей распределения Пирсона для многомодальных выборок с применением кластеризации

Рассматривается задача построения статистических моделей распределений многомодальных выборок данных. Предложена процедура разделения исходной многомодальной выборки методами кластерного анализа на несколько однородных выборок-кластеров с последующим построением на каждой из них своей функции плотности обобщенного распределения Пирсона. Для проверки соответствия построенных статистических моделей распределений применяются статистические критерии Пирсона, Колмогорова—Смирнова, Мизеса.

Ключевые слова: обобщенное распределение Пирсона, статистические критерии согласия, многомодальные выборки, методы кластерного анализа

Введение

Пусть в ходе имитационных экспериментов получена выборка, являющаяся в общем случае многомодальной:

$$X = \{x_i \mid x_i \in R, i = 1, \dots, m\}. \quad (1)$$

Необходимо построить статистическую модель распределения выборочных данных (кривую), наилучшим образом описывающую данную выборку на исследуемом интервале $[a, b] \mid a, b \in R$.

Поставленную задачу можно решить с использованием существующих известных законов распределений [1], рядов специального вида [2], семейств универсальных статистических моделей распределений [3]. Каждый из подходов имеет свои достоинства и недостатки. Например, особенностью семейств универсальных моделей распределений является возможность аппроксимации лишь одномодальных, U -образных и J -образных распределений [4].

Многомодальность распределения указывает на неоднородность исследуемой выборки X . В этом случае предлагается разделение методами кластерного анализа [5] исходной выборки на несколько однородных выборок (кластеров) с последующим построением на каждой из них своей функции плотности распределения.

Построение функции плотности распределения семейства Пирсона, соответствующей эмпирическому распределению выборки X , реализовано в программном модуле BelSim2#.random [6, 7]. Процедура разделения исходной выборки X на кластеры реализована в программно-технологическом комплексе имитации сложных систем (ПТКИ) BelSim2 [5, 8].

1. Дифференциальное уравнение Пирсона

Дифференциальное уравнение (ДУ) Пирсона [3] $df/dx = (x - a)/(b_0 + b_1x + b_2x^2) \cdot f[a, b_0, b_1, b_2 \in R]$ (2) является линейным однородным дифференциальным уравнением первого порядка. Его общий интеграл имеет следующий вид:

$$f = Ce^{\psi(x)}, \quad \psi(x) = \int (x - a)/(b_0 + b_1x + b_2x^2) dx, \\ C = \text{const}, \quad (3)$$

и представляет собой функцию плотности распределения случайной величины X .

Решения ДУ (2) для различных значений корней квадратного уравнения

$$b_0 + b_1x + b_2x^2 = 0 \quad (4)$$

образуют семейство кривых Пирсона, включающее семь основных типов распределений [3] и три частных случая распределений: нормальное, экспоненциальное и равномерное.

Пусть дискриминант квадратного уравнения (4) имеет вид [3]:

$$D = b_1^2(1 - 1/\aleph), \quad \aleph = b_1^2/(4b_0b_2), \quad (5)$$

где $\aleph$ — критерий классификации типов распределений, введенный Пирсоном.

Тогда с учетом оценки коэффициентов a, b_0, b_1, b_2 ДУ (2) классическим методом моментов [8] критерий классификации $\aleph$ принимает вид

$$\aleph = \beta_1(\beta_2 + 3)^2/(4(2\beta_2 - 3\beta_1 - 6)(4\beta_2 - 3\beta_1)), \quad (6)$$

где β_1, β_2 — коэффициенты асимметрии и эксцесса, введенные Пирсоном [3],

$$\beta_1 = \mu_3^2/\mu_2^3, \quad \beta_2 = \mu_4/\mu_2^2, \quad (7)$$

при этом μ_2, μ_3, μ_4 — центральные моменты выборки X (1) 2-го, 3-го и 4-го порядков соответственно,

$$v_1 = \frac{1}{m} \sum_{i=1}^m x_i, \quad \mu_k = \frac{1}{m} \sum_{i=1}^m (x_i - v_1)^k, \\ k = 2, 3, 4, m = |X|, \quad (8)$$

где v_1 — начальный момент 1-го порядка.

Имеет место следующая классификация типов распределений Пирсона в зависимости от значений критерия $\aleph$ вида (6) и показателей асимметрии и эксцесса β_1, β_2 вида (7). Значение критерия $\aleph < 0$ соответствует типу I; $\aleph = 0$ — типам II, VII;

$0 < \aleph < 1$ — определяет тип IV; $\aleph = 1$ — тип V; $\aleph > 1$ — тип VI; $\aleph = \pm\infty$ — тип III. Для типа II выполняются следующие условия: $\beta_1 = 0$, $1 < \beta_2 < 3$; для типа VII $\beta_1 = 0$, $\beta_2 > 3$. При $(\beta_1, \beta_2) = (0; 1,8)$ имеет место равномерное распределение; при $(\beta_1, \beta_2) = (0, 3)$ — нормальное; при $(\beta_1, \beta_2) = (4, 9)$ — экспоненциальное.

Распределения семейства Пирсона обладают следующими свойствами:

1) единственность моды, т. е. $df/dx = 0$ в некоторой точке $x = a$;

2) гладкое соприкосновение с осью x на концах интервала, на котором сосредоточено распределение, т. е. $df/dx = 0$ при $f = 0$;

3) существование не более двух точек перегиба, определяемых соотношением $x^2 = b_0/b_2$ [4].

Таким образом, распределения семейства Пирсона унимодальны. Если распределение имеет две точки перегиба, то они отстоят от моды на одинаковом расстоянии. Одна из точек может быть расположена вне зоны сосредоточения распределения, т. е. распределения семейства (2) могут иметь также и J - и U -образную формы [4].

2. Построение функции плотности семейства Пирсона

Для работы модуля BelSim2#.random рассчитывают необходимые точечные оценки выборки X , в частности, начальный момент первого порядка v_1 (8), центральные моменты $\mu_0, \dots, \mu_4$ (8), коэффициенты β_1, β_2 (7).

Построение кривой $f(x)$ в соответствии с классификацией Пирсона выполняется в следующей последовательности.

Шаг А1. Классификация типа кривой $f(x)$.

Проводится в соответствии со значениями коэффициентов β_1, β_2 (7) и критерия $\aleph$ (6) классификации типа распределения Пирсона [3, 7].

Шаг А2. Оценка значений параметров $f(x)$.

Проводится в соответствии с классическим методом моментов для функции плотности распределения $f(x)$ заданного типа (см. шаг А1). В частности, определяется параметр сдвига кривой l :

$$l = \text{Mean} - \text{Mode},$$

где $\text{Mean} = v_1$ — математическое ожидание выборки, Mode — мода выборки.

В соответствии с методикой, предложенной Пирсоном [3, 4], для построенной кривой $f(x)$ определяется нормирующий множитель N , расчет которого проводится, исходя из условия равенства единице интеграла от $N \cdot f(x)$ на интервале $[a, b]$, в пределах которого выполняется построение:

$$N \int_a^b f(x) dx = 1.$$

Шаг А3. Проверка соответствия $f(x)$ выборке X .

Для проверки гипотезы о распределении выборки X по закону распределения с плотностью $f(x)$ выбраны критерии согласия: Пирсона (χ^2), Колмогорова—Смирнова (λ), Мизеса (ω^2) [1].

Первыми применяются критерии χ^2 и λ . Если логические значения результатов их работы эквивалентны, то на этом этап статистической проверки гипотезы заканчивается. Иначе — дополнительно применяется критерий ω^2 , результат работы которого принимается в качестве заключения о проверке гипотезы.

Сохраняемые результаты работы критериев χ^2 , λ , ω^2 :

1) наблюдаемые значения критериев:

$$\Delta_k, k = \chi^2, \lambda, \omega^2;$$

2) критические значения критериев:

$$\Delta_k^{\text{кр}}, k = \chi^2, \lambda, \omega^2;$$

3) отношение наблюдаемых значений критериев к критическим:

$$dL_k = \Delta_k / \Delta_k^{\text{кр}}, k = \chi^2, \lambda, \omega^2; \quad (9)$$

4) логические результаты работы критериев:

$$bL_k = \begin{cases} \text{True} & | dL_k < 1; \\ \text{False} & | dL_k \geq 1, \end{cases} k = \chi^2, \lambda, \omega^2. \quad (10)$$

3. Определение статистической модели распределения выборочных данных $f^*(x)$, наилучшим образом описывающей выборку X

В случае отклонения статистическими критериями гипотезы о принадлежности выборки X закону распределения с плотностью $f(x)$ или по специальному запросу пользователя выполняется построение с последующей проверкой комплексом статистических критериев χ^2 , λ , ω^2 всех основных типов кривых семейства и частных случаев распределений: равномерного, нормального и экспоненциального, за исключением функции $f(x)$, проверенной на предыдущем этапе. Наилучшая статистическая модель выбирается на множестве тех кривых $\{f_i(x)\}$, которые не были отклонены комплексом критериев.

Шаг В1. В том случае, если не для всех функций $f_i(x)$, $i = 1, \dots, |f_i(x)|$, применено одинаковое число критериев, для определения $f^*(x)$ проводится дополнительный расчет критерия ω^2 .

Шаг В2. Функция $f^*(x) = f_{i_0}(x) \in \{f_i(x)\}$ описывает выборку X наилучшим образом на интервале $[a, b]$ тогда и только тогда, когда

$$\phi(X, f_{i_0}(x)) = \min\{\phi(X, f_i(x)) | i = 1, \dots, |f_i(x)|\}, \quad (11)$$

где $\phi(X, f_i(x))$ — евклидова метрика от отношений dL_k вида (9):

$$\phi(X, f_i(x)) = \sqrt{\sum_k (dL_k(X, f_i(x)))^2}, k = \chi^2, \lambda, \omega^2. \quad (12)$$

Если все построенные кривые отклонены статистическими критериями, то нет возможности построения кривой плотности распределения семейства Пирсона, описывающей выборку X на исследуемом интервале.

4. Кластерный анализ данных

Кластеризация данных является одним из этапов построения аналитического решения поставленной задачи. Часто легче выделить группы схожих объектов, изучить их особенности и построить для каждой группы отдельную модель, чем создать одну общую модель для всех данных.

Для кластеризации выборки выбраны методы Tree Clustering, K-Means, Fuzzy Relation Clustering (FRC) [5]. В результате разбиения множества X на кластеры каждый из методов ставит в соответствие элементам $x_i \in X, i = 1, \dots, m$, кластеры $K_j, j = 1, \dots, |K_j|$. Для проверки и уточнения результатов разделения множества X на кластеры предлагается использование нескольких методов кластеризации с последующим обобщением результатов.

Шаг С1. Кластеризация выборки X . Число кластеров $|K_j|$, на которые делится выборка, определяется в зависимости от числа областей отдельных "пиков" в полигоне частот исследуемой выборки X .

Шаг С1.1. Проводится разделение выборки X на кластеры $K_j, j = 1, \dots, |K_j|, X = K_1 \cup \dots \cup K_{|K_j|}$, методами Tree Clustering и K-Means.

Шаг С1.2. Если результаты работы методов кластеризации, полученные на Шаге С1.1, не совпадают, применяют третий метод — FRC.

Шаг С1.3. Проводится обобщение результатов кластеризации выборки несколькими методами в соответствии с *Определением 1*.

Определение 1. Элемент $x_i \in X$ является элементом кластера $K_j, j = 1, \dots, |K_j|$, тогда и только тогда, когда он отнесен к данному кластеру по крайней мере L^* методами из L выбранных методов кластерного анализа, причем $L/2 < L^* < L, L \geq 3$.

Шаг С1.4. На основании обобщенного результата кластеризации несколькими методами выделяются кластеры K_j элементов $x_i \in X$.

Шаг С2. В том случае, если выявлены выбросы, они исключаются.

Шаг С3. Для каждого из кластеров $K_j, j = 1, \dots, |K_j|$, выполняется построение функции плотности распределения $f_j^*(x)$, соответствующей эмпирическому распределению элементов данного кластера.

В случае, если для кластера $K_{j_0}, j_0 \in 1, \dots, |K_j|$, все построенные кривые отклонены статистическими критериями, для K_{j_0} нет возможности построения кривой $f_{j_0}^*(x)$. Рекомендуется разделение K_{j_0} на несколько кластеров меньшего размера (см. шаг С1) с последующим построением на каждой из них своей функции плотности распределения.

5. Обобщение результатов кластеризации данных

Для обобщения результатов кластеризации множества X несколькими методами вводятся необходимые определения и утверждения.

Утверждение 1. Результат кластеризации множества X вида (1) для каждого из L методов кластеризации $M_l, l = 1, \dots, L$, может быть представлен в виде матрицы вероятностей принадлежности элементов $x_i \in X$ определенным кластерам [4]:

$$P_l = \|p_{lij}\|, p_{lij} \in \{0, 1\}, i = 1, \dots, |X|, j = 1, \dots, |K_j|, (13)$$

где i — номер элемента из множества X ; j — номер кластера K_j ; $|K_j|$ — число кластеров, построенных методом M_l ; p_{lij} — вероятность принадлежности i -го элемента $x_i \in X$ кластеру K_j ; l — номер примененного метода кластеризации $M_l, l = 1, \dots, L$.

Для каждого элемента $x_i \in X$ в (13) на основе классического определения вероятности вычисляют вероятности $p_{lij} \in \{0, 1\}$ попадания в определенные кластеры $K_j, j = 1, \dots, |K_j|$.

Утверждение 2. Пусть $P_1, \dots, P_L$ — матрицы вида (13) вероятностей принадлежности элементов $x_i \in X, i = 1, \dots, |X|$, определенным кластерам $K_j, j = 1, \dots, |K_j|$, согласно методам кластеризации $M_1, \dots, M_L$ соответственно. Тогда значения элементов $p_{ij} \in [0, 1]$ обобщенной (в смысле *Определения 1*) матрицы P могут быть найдены суммированием коэффициентов $P_v, v = L^*, \dots, L$, производящей функции [5]

$$\varphi(z) = \prod_{l=1}^L (q_{lij} + p_{lij} \cdot z) = \sum_{v=0}^L P_{v,L} \cdot z^v,$$

где p_{lij} — элементы матриц $P_l, l = 1, \dots, 3$; $p_{lij} \in [0, 1]$; $q_{lij} = 1 - p_{lij}, l = 1, \dots, L, i = 1, \dots, |X|, |X| = m, j = 1, \dots, |K_j|$, соответственно,

$$p_{ij} = \sum_{v=L^*}^L P_{v,L}. (14)$$

Определение 2. Элемент $x_i \in X, i = 1, \dots, |X|$, принадлежит кластеру $K_{j_0}, j_0 \in 1, \dots, |K_j|$, тогда и только тогда, когда вероятность принадлежности объекта кластеру в i -й строке обобщенной матрицы P максимальна [5]:

$$(x_i \in K_{j_0}) \Leftrightarrow (p_{ij_0} = \max\{p_{ij} | j = 1, \dots, |K_j|\}). (15)$$

Замечание 1. Для каждого из методов кластеризации свойственна собственная система нумерации кластеров K_j . Номера $j \in 1, \dots, |K_j|$ присваиваются кластерам K_j в соответствии с порядком их построения, который зависит от особенностей алгоритмов методов кластеризации [9].

Замечание 2. Для возможности обобщения результатов кластеризации необходимо проведение упорядочения столбцов обобщаемых матриц P_l вида (13) размерности $|X| \times |K_j|$ таким образом, что

для любого $j_0 \in 1, \dots, |K_j|$ кластеры $K_{j_0}^{(l)}$, построенные методами $M_l, l = 1, \dots, L$, будут соответствовать друг другу, т. е. иметь максимально возможное число общих элементов $x_i \in X$.

Предлагается следующая процедура подготовки интерпретированных результатов кластеризации P_1, P_2, P_3 к последующему этапу обобщения.

Шаг D1. Матрица P_1 определяется в качестве эталонной матрицы.

Шаг D2. Если $P_1 \neq P_2$, или $P_1 \neq P_3$, проводится упорядочение столбцов матриц P_2, P_3 , соответственно, по отношению к P_1 .

Определение 3. Обозначим через $P_\xi(I_t)$ матрицы, образованные из матрицы P_ξ перестановками I_t ее столбцов $j \in 1, \dots, |K_j|$:

$$P_\xi(I_0) = P_\xi, P_\xi(I_1), \dots, P_\xi(I_t), t = 0, \dots, |K_j| - 1,$$

где t — номер перестановки I_t , определяющей порядок следования столбцов матрицы P_ξ : $I_0 = (1, 2, \dots, |K_j|)$, $I_1 = (2, 1, \dots, |K_j|), \dots, I_t$.

Определение 4. Для матричных пар $(P_\xi(I_t), P_1)$, $t = 0, \dots, |K_j| - 1$, определяется метрика [9] $\rho: (P_\xi(I_t) \rightarrow R_+ \cup \{0\})$ вида

$$\rho(P_\xi(I_t), P_1) = \sum_{i=1}^{|X|} \sum_{j=1}^{|K_j|} \min(p_{\xi ij}(I_t), p_{1ij}),$$

$$t = 0, \dots, |K_j| - 1, \quad (16)$$

где $p_{\xi ij}(I_t)$ — элементы матрицы $P_\xi(I_t)$ со столбцами j , следующими в порядке, соответствующем проведенной перестановке I_t столбцов P_ξ .

Замечание 3. Метрика $\rho(P_\xi(I_t), P_1) \geq 0$ позволяет определить меру совпадения элементов соответствующих матриц. Значение ρ прямо пропорционально числу общих элементов $x_i \in X$ в кластерах матричной пары.

Утверждение 3. Перестановка I^* столбцов матрицы P_ξ по отношению к матрице P_1 оптимальна тогда и только тогда, когда выполняется условие [9]

$$\rho(P_\xi(I^*), P_1) = \max\{\rho(P_\xi(I_t), P_1) | t = 0, \dots, |K_j| - 1\}, \quad (17)$$

где ρ — метрика вида (16).

Возвращаясь к прежним обозначениям, результат упорядочивания будет

$$P_\xi = P_\xi(I^*). \quad (18)$$

Замечание 4. При наличии в обобщенной в соответствии с Утверждением 2 матрице P строк i_0 с нулевыми вероятностями принадлежности элемента x_{i_0} всем кластерам $K_j, j = 1, \dots, |K_j|$, вероятности принадлежности данного элемента кластерам присваиваются в соответствии со значениями p_{1i_0j} матрицы P_1 , поскольку на шаге D1 процедуры подготовки интерпретированных результатов кластеризации к обобщению P_1 определена как эталонная.

6. Экспериментальная часть

Пусть $X = \{x_i | x_i \in R, i = 1, \dots, 84\}$ — исследуемая выборка, временной ряд и гистограмма которой представлены на рис. 1, 2, где ось абсцисс i рис. 1 — ось номера элементов $x_i \in X$, ось ординат ni рис. 2 — ось частот попадания элементов x_i в частичные интервалы гистограммы.

Необходимо построить статистическую модель распределения выборочных данных, наилучшим образом описывающую выборку X на интервале [971; 1177,25].

Для решения поставленной задачи применяется программный модуль BelSim2#.random, в списке параметров которого установлены флаги обязательной проверки всех функций семейства Пирсона и трех частных случаев: равномерного ($R(x)$); нормального ($N(x)$); экспоненциального ($E(x)$).

В результате применения модуля BelSim2#.random рассчитаны точечные оценки выборки X , в частности: $\min = 971,000$, $\max = 1177,250$, $v_1 = 1069,358$ (8), $\mu_2 = 6276,194$ (8), $\mu_3 = 94791,723$ (8), $\mu_4 = 5,052E7$ (8), $\beta_1 = 0,036$ (7), $\beta_2 = 1,282$ (7), $\aleph = -0,009$ (6). Для выборочных данных построены функции плотностей распределений $f_2(x)$, $f_3(x)$, $f_5(x)$, $f_7(x)$, $R(x)$, $E(x)$, $N(x)$. С заданной точностью $\varepsilon = 0,1$ критерий классификации типов кривых Пирсона $\aleph = 0$, поэтому $f_2(x)$ — функция плотности, построенная в соответствии с классификацией Пирсона.

В результате проверки комплексом статистических критериев χ^2 , λ , ω^2 все гипотезы о принадлеж-

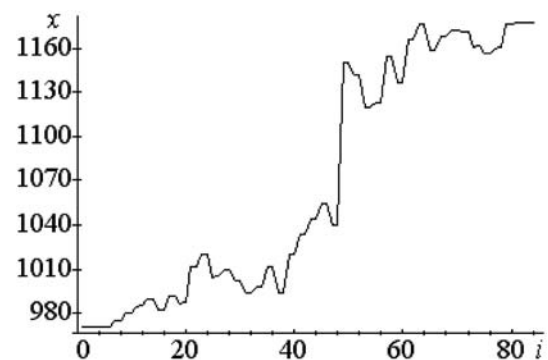


Рис. 1. Временной ряд выборки X

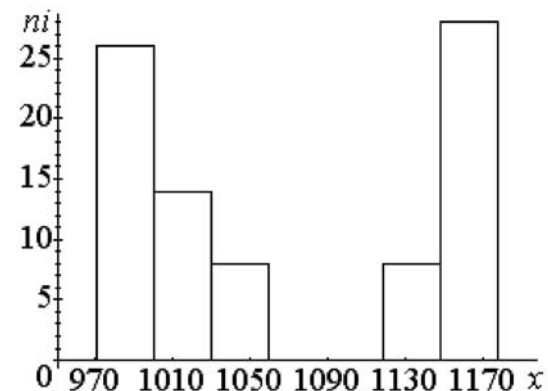


Рис. 2. Гистограмма выборки X

ности выборки X законам распределения с построенными плотностями отклонены, поэтому нет возможности построения на исследуемом интервале кривой плотности распределения семейства Пирсона, описывающей выборку X . Выборка X многомодальна.

Выполняется разделение X на кластеры с применением методов кластеризации Tree Clustering, K-Means, Fuzzy Relation Clustering (FRC). Число кластеров $|K_j| = 2$ определено в соответствии с числом областей отдельных "пиков" в гистограмме исследуемой выборки X .

Вначале проводится кластеризация выборки X методами K-Means и Tree Clustering. Пусть матрицы P_1, P_2 вида (13) — интерпретированные результаты кластеризации выборки X методами K-Means и Tree Clustering соответственно, тогда транспонированные матрицы P_1^T, P_2^T имеют вид:

$$P_1^T = \begin{pmatrix} 1_{1 \times 48} & 0_{1 \times 36} \\ 0_{1 \times 48} & 1_{1 \times 36} \end{pmatrix}, P_2^T = \begin{pmatrix} 0_{1 \times 48} & 1_{1 \times 36} \\ 1_{1 \times 48} & 0_{1 \times 36} \end{pmatrix}.$$

Поскольку $P_1^T \neq P_2^T \Leftrightarrow P_1 \neq P_2$, необходимо упорядочение столбцов матрицы P_2 по отношению к матрице P_1 , которое проводится в соответствии с Определениями 3, 4 и Утверждением 3.

Рассматривается группа перестановок столбцов матрицы P_2 , соответственно, строк матрицы P_2^T : $I_0 = (1, 2), I_1 = (2, 1)$.

Матрицы, соответствующие перестановкам I_0, I_1 :

$$P_2^T(I_0) = \begin{pmatrix} 0_{1 \times 48} & 1_{1 \times 36} \\ 1_{1 \times 48} & 0_{1 \times 36} \end{pmatrix}; P_2^T(I_1) = \begin{pmatrix} 1_{1 \times 48} & 0_{1 \times 36} \\ 0_{1 \times 48} & 1_{1 \times 36} \end{pmatrix}.$$

Оптимальной перестановкой (Утверждение 3) строк P_2^T , соответственно столбцов матрицы P_2 является $I^* = I_1$, поскольку для матричной пары $(P_1^T, P_2^T(I_1))$ значение метрики ρ вида (16) максимально:

$$\rho(P_1^T, P_2^T(I_1)) = 84; \rho(P_1^T, P_2^T(I_0)) = 0;$$

$$P_2^T(I^*) = P_2^T(I_1).$$

Таким образом, $P_2^T \equiv P_1^T$, соответственно, $P_2 \equiv P_1$. Результаты работы методов K-Means и Tree Clustering эквивалентны, поэтому нет необходимости в кластеризации третьим методом — Fuzzy Relation Clustering.

Выделение элементов кластеров K_1, K_2 выполняется на основании значений элементов матрицы $P = P_1 = P_2$ в соответствии с Определением 2:

Кластер 1: $K_1 = \{x_1, \dots, x_{48}\};$

Кластер 2: $K_2 = \{x_{49}, \dots, x_{84}\}.$

Построение модели распределения, соответствующей кластеру K_1 . Ниже приведены результаты работы BelSim2#.random для K_1 . Определены точечные оценки выборки, в частности: $\min = 971,000$, $\max = 1055,250$, $v_1 = 1001,958$, $\mu_2 = 556,655$, $\mu_3 = 7986,657$, $\mu_4 = 765267,516$, $\beta_1 = 0,370$, $\beta_2 = 2,470$, $\aleph = -0,145$, $|K_1| = 48$.

Построены функции плотностей распределений семейства Пирсона и частные случаи распределений. С заданной точностью $\varepsilon = 0,1$ критерий $\aleph = -0,1 < 0$ (6) классификации типов распределений Пирсона соответствует функция плотности $f_1(x)$. Результат работы программы BelSim2#.random:

$$f1(x) = 0,014 \cdot (1 + (x - 1001,958)/33,390)^{0,342} \cdot (1 - (x - 1001,958)/68,127)^{0,699};$$

$$f2(x) = 0,017 \cdot (1 - (x - 1001,958)^2/5184,714)^{3,157};$$

$$f3(x) = 0,019 \cdot \exp(-0,139 \cdot (x - 1001,958)) \cdot (1 + (x - 1001,958)/70,421)^{9,817};$$

$$f5(x) = 6,480E46 \cdot (x - 913,307)^{(-20,361)} \cdot \exp(-1805,092/(x - 913,307));$$

$$f7(x) = 0,019 \cdot \exp(-(x - 1001,958)^2/1113,309);$$

$$R(x) = 0,012;$$

$$E(x) = 9,980E - 4 \cdot \exp(-1 \cdot 9,980E - 4 \cdot x);$$

$$N(x) = 1/(23,594 \cdot (2 \cdot \pi)^{(1/2)}) \cdot \exp(-1 \cdot (x - 1001,958)^2/1113,309).$$

В результате проверки критериями $\chi^2, \lambda, \omega^2$ гипотезы о принадлежности выборки X законам распределения с построенными плотностями $f_1(x), f_3(x), f_5(x), R(x), E(x)$ отклонены, для отклонения гипотез о распределении выборок по законам с распределениями $f_2(x), f_7(x), N(x)$ нет оснований. Так как для функции $f_7(x)$ значение dL , рассчитанное по формуле (14), минимально, для кластера K_1 справедливо соотношение $f_1^*(x) = f_7(x)$.

Построение модели распределения, соответствующей кластеру K_2 . Ниже приведены результаты работы BelSim2#.random для K_2 . Определены точечные оценки выборки: $\min = 1120,000$, $\max = 1177,25$, $v_1 = 1158,069$, $\mu_2 = 300,290$, $\mu_3 = -4461,342$, $\mu_4 = 245791,459$, $\beta_1 = 0,735$, $\beta_2 = 2,725$, $\aleph = -0,252$, $|K_2| = 36$.

Построены функции плотностей распределений семейства Пирсона и частные случаи распределений. С заданной точностью $\varepsilon = 0,1$ критерий $\aleph = -0,3 < 0$ (6) классификации типов распределений Пирсона соответствует функция плотности $f_1(x)$. Результат работы программы BelSim2#.random:

$$f1(x) = \text{NaN} \cdot (1 + (x - 1158,07)/131,79)^{0,30} \cdot (1 - (x - 1158,07)/-62,88)^{(-0,15)};$$

$$f2(x) = 0,026 \cdot (1 - (x - 1158,069)^2/5968,841)^{8,438};$$

$$f3(x) = 0,027 \cdot \exp(0,135 \cdot (x - 1158,069)) \cdot (1 + (x - 1158,069)/-32,996)^{4,442};$$

$$f5(x) = 1,462E45 \cdot (x - 1092,199)^{(-20,704)} \cdot \exp(-1363,775/(x - 1092,199));$$

$$f7(x) = 0,027 \cdot \exp(-(x - 1158,069)^2/600,581);$$

$$R(x) = 0,017;$$

$$E(x) = 8,635E - 4 \cdot \exp(-1 \cdot 8,635E - 4 \cdot x);$$

$$N(x) = 1/(17,329 \cdot (2 \cdot \pi)^{(1/2)}) \cdot \exp(-1 \cdot (x - 1158,069)^2/600,581).$$

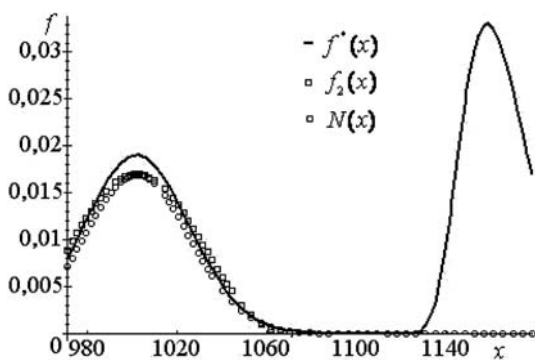


Рис. 3. Статистическая модель распределения выборочных данных

Для $f_1(x)$ коэффициент NaN обозначает, что функция не построена (множитель $C = \text{const}$ (3) не определен). В результате проверки комплексом статистических критериев χ^2 , λ , ω^2 все гипотезы о принадлежности выборки X законам распределения с построенными плотностями, за исключением $f_5(x)$, отклонены, поэтому для кластера K_2 справедливо $f_2^*(x) = f_5(x)$.

7. Обсуждение результатов

Статистическая модель распределения выборочных данных, наилучшим образом описывающая выборку X на интервале $[a, b]$, имеет вид (рис. 3):

$$f^*(x) = \begin{cases} 0,019 e^{-\frac{(x-1001,958)^2}{1113,309}}, & x \in [971,0; 1055,25]; \\ 1,462 E45 (x - 1092,199)^{-20,704} e^{\frac{-1363,775}{(x-1092,199)}}, & x \in [1120,0; 1177,25]. \end{cases}$$

На рис. 3 представлены функции плотностей, не отклоненные статистическими критериями: $f_2(x)$, $N(x)$, соответствующие кластеру K_1 , и модель распределения $f^*(x)$ выборки X . Модель $f^*(x)$ включает функции $f_7(x)$ для $x_i \in K_1$ и $f_5(x)$ для $x_i \in K_2$.

Заключение

С учетом особенностей семейств универсальных моделей распределений, т. е. возможности аппроксимации лишь одномерных и U-образных распределений, предложена процедура разделения исходной многомодальной выборки методами кластерного анализа на несколько однородных выборок-кластеров с последующим построением на каждой из них своей функции плотности обобщенного распределения Пирсона.

В случае многомодальности выборки разработана процедура разделения выборочных данных на кластеры с использованием нескольких методов кластерного анализа.

В результате разбиения выборки X на кластеры каждый из методов кластеризации ставит в соответствие элементам $x_i \in X$, $i = 1, \dots, m$, соответствующие

им кластеры K_j , $j = 1, \dots, |K_j|$. При этом для каждого из методов кластеризаций свойственна собственная уникальная система нумерации кластеров K_j .

Показано, что результат кластеризации множества многомерных данных для каждого из методов кластерного анализа можно представить в виде матрицы вероятностей принадлежности элементов $x_i \in X$ определенным кластерам K_j , $j = 1, \dots, |K_j|$.

Для обобщения результатов кластеризации предложена процедура упорядочения столбцов обобщаемых матриц P_i вида (13) таким образом, что соответствующие друг другу столбцы матриц (кластеры) будут иметь максимально возможное число общих элементов $x_i \in X$.

Анализ обобщенного результата кластеризации, получаемого посредством применения теорем сложения/умножения вероятностей, позволяет выявить кластеры элементов с эквивалентным поведением.

Для проверки гипотез о распределении выборки X либо элементов кластеров K_j , $j = 1, \dots, |K_j|$, в соответствии с законами распределений по построенным функциям плотностей предложено применение комплекса критериев согласия Пирсона (χ^2), Колмогорова—Смирнова (λ), Мизеса (ω^2).

Предложен критерий определения модели распределения выборочных данных $f^*(x)$, наилучшим образом описывающей выборку на исследуемом интервале в соответствии с результатами проверки комплексом статистических критериев χ^2 , λ , ω^2 .

Экспериментальные исследования проведены с использованием методов кластерного анализа K-Means, Tree Clustering, Fuzzy Relation Clustering и ПТКИ BelSim [7], пакета статистической обработки данных STATISTICA, табличного процессора MS Excel.

Список литературы

1. **Большев Л. Н., Смирнов Н. В.** Таблицы математической статистики. М.: Наука, 1983. 416 с.
2. **Вадзинский Р. Н.** Справочник по вероятностным распределениям. СПб.: Наука, 2001. 295 с.
3. **Elderton W. P.** Frequency curves and correlation. Camb., 1906. 172 с.
4. **Кендалл М., Стюарт А.** Теория распределений: пер. с англ. М.: Наука, 1966. 588 с.
5. **Якимов А. И., Борчик Е. М., Башаримов В. В.** О совместном использовании методов кластерного анализа многомерных данных // Доклады БГУИР. 2011. № 5 (59). С. 95—102.
6. **Программный модуль расчета кривой плотности распределения случайной составляющей в последовательности данных BelSim2#random**: свидетельство о регистрации компьютерной программы № 306 / Е. А. Якимов, А. А. Ковалевич, Е. М. Борчик, В. В. Башаримов. — Минск: НЦИС, 2011. — Заявка № С20110024. — Дата подачи: 04.04.2011.
7. **Якимов А. И., Борчик Е. М., Башаримов В. В.** Методика построения кривой семейства плотностей обобщенного распределения Пирсона для исследуемой выборки // Системный анализ и информационные технологии: материалы междунар. науч.-техн. конф. SAIT 2011 (23—28 мая 2011 г., Киев, Украина). — Киев: НТУУ "КПИ", 2011. С. 347.
8. **Якимов А. И.** Технология имитационного моделирования систем управления промышленных предприятий. Могилев: Белорус.-Рос. ун-т, 2010. 304 с.
9. **Якимов А. И., Борчик Е. М., Башаримов В. В.** Итерационная процедура обобщения результатов кластеризации многомерных данных несколькими методами // Математическое и имитационное моделирование систем. МОДС 2011: тез. докл. Шестой междунар. науч.-практич. конф., 27—30 июня 2011 г. — Чернигов: ИПММС НАН Украины, 2011. С. 419—423.

УДК 004.93'1.021

Н. Л. Щеголева¹, канд. техн. наук, доц.,
e-mail: stil_hope@mail.ru,

Г. А. Кухарев^{1, 2}, д-р техн. наук, проф.,
П. Форчманский², канд. техн. наук, доц., зав. каф.

¹ Санкт-Петербургский государственный
электротехнический университет "ЛЭТИ"
им. В. И. Ульянова (Ленина)

² Западно-Поморский технологический
университет, Щецин, Польша

Простой алгоритм распознавания изображений лиц с проблемами освещения

Обсуждается задача распознавания изображений лиц с проблемами освещения — бликами, тенями и очень низким уровнем яркости. Предлагается алгоритм решения этой задачи, основанный на методе 2DDCT в совокупности с процедурами уменьшения градиента яркости исходных изображений, элиминированием пространственных низкочастотных компонент спектра и объединением спектральных признаков по условиям средней яркости. Эксперименты выполнены на базе Yale B и Yale B+.

Ключевые слова: распознавание лиц, изображения с проблемами освещения, двумерное косинус-преобразование, гамма-коррекция, логарифмирование, фузия спектров изображений

Введение

Одной из актуальных проблем в практике распознавания изображений с лицами остается нестабильность яркости изображений на входе системы FaReS (Face Recognition System).

В этих условиях в области лица и фона вокруг него появляются изменения двух типов. Первые можно отнести к локальным ярко выраженным теням, а вторые — к глобальным теням. Локальные тени изменяют форму отдельных частей лиц (носа, рта и глаз) и искажают границы области лица, а также границы волосы/лоб и шея/подбородок. Глобальные тени существенно снижают различимость области лиц на общем фоне и/или полностью закрывают их.

Отмеченные виды нестабильности представления области лиц приводят к ухудшению результативности работы FaReS. Именно поэтому интерес к проблеме распознавания изображений лиц с про-

блемами освещения не ослабевает последние несколько лет [1—11]. Анализируя эти работы, можно отметить, что методы распознавания изображений лиц с проблемами освещения основаны на нескольких подходах, среди которых можно выделить следующие:

1) предобработка исходных изображений, включающая процедуры выравнивания яркости (эквализация); уменьшение градиента изменений яркости (гамма-коррекция, логарифмирование), инвариантное (относительно яркости) представление изображений на основе локальных бинарных эталонов (LBP — Local Binary Patterns) и логарифмических моделей общей вариации (LTV — Logarithmic Total Variation);

2) представление изображений лиц с проблемами освещения (ИЛПО) в собственных базисах и в форме соответствующих моделей, основанных на "собственных лицах";

3) представление ИЛПО в форме спектральных признаков на основе вейвлет- и косинус-преобразований с исключением низкочастотных (НЧ) компонент из этого представления;

4) дополнение базы FaReS новыми эталонами, имеющими практически все искажения, связанные с проблемами освещения изображений лиц.

Тестирование методов распознавания ИЛПО осуществляется обычно в рамках базы Yale. Полная база Yale включает первоначальную базу Yale B и ее расширенную часть Yale B+ [11]. При этом правильной необходимо считать постановку задачи распознавания изображений лиц с проблемами освещения только в том случае, когда база FaReS не дополняется эталонами с глобальными и/или локальными тенями [4, 6, 9] в противовес постановке, определенной в работах [1—3, 5, 7, 8].

Следует также отметить, что собственные базисы строятся в основном на методах PCA (principal component analysis) и LDA (linear discriminant analysis) [1—3], а в последнее время и на основе CCA (canonical correlation analysis [6]), в то время как дискретное косинус-преобразование DCT (discrete cosinus transform) используется как инструмент предварительной обработки ИЛПО.

Так, например, в работе [1] показано, что для решения задачи распознавания на основе PCA и LDA ИЛПО должны быть преобразованы в спектральные признаки на основе 2DDCT (двумерного косинус-преобразования). На этом же этапе удаляются НЧ компоненты спектра, соответствующие



Рис. 1. Схема вычислений, представленная в работе [5]

"теневым" составляющим. При этом в работе [1] предложены следующие схемы вычислений:

$$\begin{aligned} \text{Log}(X) &\rightarrow 2\text{DDCT} \rightarrow C \rightarrow C - C_{\text{low}} \rightarrow 1\text{DPCA} \text{ и} \\ \text{Log}(X) &\rightarrow 2\text{DDCT} \rightarrow C \rightarrow C - C_{\text{low}} \rightarrow 1\text{DLDA}, \quad (1) \end{aligned}$$

где $\text{Log}(X)$ — логарифмирование пикселей исходного ИЛПО; C — спектр в базисе двумерного DCT (2DDCT); $C - C_{\text{low}}$ — коррекция спектра за счет удаления НЧ составляющих C_{low} .

В работе [2] предложена схема вычислений, в которой перед выполнением PCA спектральное представление ИЛПО возвращается в исходное — яркостное признаковое пространство. Как утверждают авторы работы [2], этот шаг повышает результативность распознавания ИЛПО. Схема вычислений при этом выглядит следующим образом:

$$\begin{aligned} \text{Log}(X) &\rightarrow 2\text{DDCT} \rightarrow C \rightarrow (C - C_{\text{low}}) \rightarrow \\ &\rightarrow 2\text{DDCT}^{-1} \rightarrow 1\text{D PCA}. \quad (2) \end{aligned}$$

Аналогичные подходы представлены в работе [5], однако вычисления, связанные с реализацией PCA, оптимизируются за счет использования матриц Грамма. Полная схема вычислений при этом представлена на рис. 1.

Цель статьи — представление простого алгоритма распознавания изображений лиц с проблемами освещения — бликами, тенями и очень низким уровнем яркости.

Предлагаемый алгоритм основан на следующих подходах:

- использование только оригинальных изображений базы Yale B и Yale B+ и при условии, что база этало-

нов FaReS не содержит изображений с локальными и/или глобальными тенями;

- простая предобработка исходных изображений в целях уменьшения градиента изменений яркости (гамма-коррекция и логарифмирование);
- использование метода 2DDCT как единственного инструмента трансформации исходных данных (ИЛПО) в спектральное пространство признаков;
- использование механизма совмещения (фузии) спектральных признаков изображений на этапе их классификации в случаях низкого уровня их яркости.

1. Характеристика изображений, использованных в экспериментах

В экспериментах использовано 2452 из 2470 изображений баз Yale B и Yale B+, определяющих центральную часть лиц 38 людей (18 изображений базы нельзя прочесть с файлов, размещенных на сайте [11]). Все изображения имеют размер 192×168 пикселей, представлены в шкале GRAY (256 уровней яркости) и сгруппированы в шесть составов: Subset0...Subset5.

Изображения состава Subset0 показаны на рис. 2 в порядке увеличения средней яркости: от 71 в минимуме — левое верхнее изображение, до 172 в максимуме — правое нижнее изображение. И как видно, они имеют разную яркость, но не имеют бликов, теней и очень низкого уровня яркости.

Примеры изображений составов Subset3...Subset5 для одного и того же лица показаны на рис. 3, где цифрами 3...5 определена принадлежность изображений лиц к соответствующему составу данных.

Составы Subset1...Subset5 получены при моделировании пространственного перемещения источника света, поэтому представляют различные варианты теней — блики (Subset1 и Subset2), локальные тени (Subset3) и боковые тени (Subset4 и Subset5), а также глобальные тени (Subset5), закрывающие

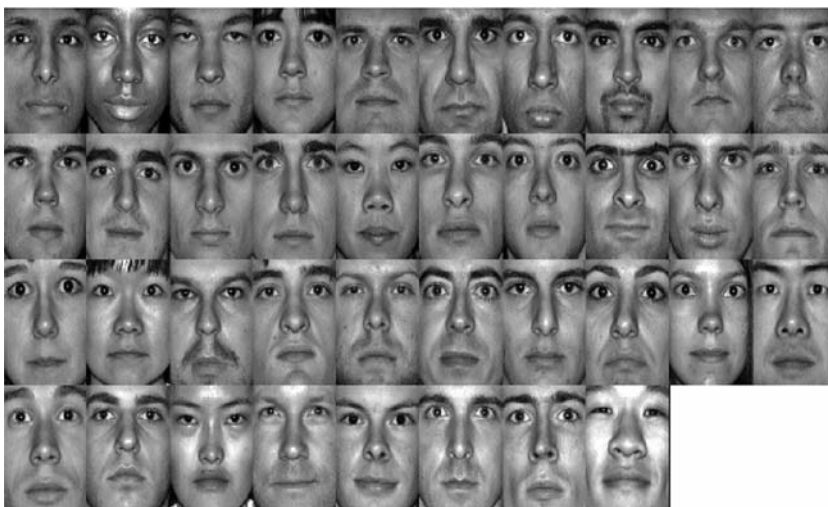


Рис. 2. Изображения лиц состава Subset0 базы Yale B и Yale B+



Рис. 3. Примеры изображений из составов Subset3...Subset5

практически всю область лица. Наибольшие трудности при распознавании вызывают изображения из составов Subset3...Subset5.

2. Предобработка изображений лиц

Ясно, что если не выравнять яркости тестовых изображений, то результативность их распознавания будет очень низкой. Как следует из работ [1–10], методы улучшения яркости изображений, показанных на рис. 2, могут быть основаны на использовании процедур гамма-коррекции и логарифмировании каждого пикселя этих изображений. Однако сами эти процедуры, с одной стороны, требуют учета характеристик исходных изображений, например, средней и локальной яркости, границ теней, контраста [10], а с другой стороны, не являются суперфинишными операциями, поскольку частично

сами искажают формы и характеристики этих изображений.

На рис. 4 приведены: оригинальное изображение, изображение на входе FaReS в условиях освещения лица "спереди и сверху" и результат применения процедур обработки — гамма-коррекции и логарифмирования значений его яркости.

Наблюдаемые при этом искажения (неудаленные тени и новые яркие области, потеря контраста, шумы) отчетливо видны на результатах обработки — особенно в сравнении с изображением-оригиналом. Но на полученных изображениях уже легко выделяются границы областей лиц и их антропометрические параметры. При этом также видно, что гамма-коррекция и процедура логарифмирования исходного изображения

"вытягивают" область лица из тени и улучшают конечный результат.

Пусть I — изображение размера $M \times N$. Тогда процедура гамма-коррекции изображения в целях повышения его яркости реализуется следующим образом:

$$\tilde{i}(m, n) = i(m, n)^\gamma, \quad \forall m = 1, 2, \dots, M \text{ и } n = 1, 2, \dots, N, \quad (3)$$

где $\tilde{i}(m, n)$ — пиксель откорректированного изображения; γ — коэффициент степенного преобразования, причем здесь $\gamma \ll 1$.

Логарифмирование выполняется в два этапа. На первом этапе все пиксели, имеющие нулевые значения яркости заменяются на значения "1" так, что

$$i(m, n) = i(m, n) + 1, \quad \forall i(m, n) \equiv 0. \quad (4a)$$

На втором этапе реализуется собственно логарифмирование:

$$\tilde{\tilde{i}}(m, n) = \log(i(m, n)), \quad \forall m = 1, 2, \dots, M \text{ и } n = 1, 2, \dots, N, \quad (4b)$$

где $\tilde{\tilde{i}}(m, n)$ — пиксель после логарифмирования его значения.

Использование DCT основывается на том, что базисные функции косинус-преобразования являются наилучшими (в смысле минимума потерь) для аппроксимации собственных функций, вычисленных для генеральной совокупности (больших наборов) цифровых изображений. Учитывая этот факт, можно полностью исключить предобработку исходных изображений в рамках PCA/KLT в FaReS, заменив ее более простым косинус-преобразованием.



Рис. 4. Пример коррекции яркости изображений базы Yale B

При этом косинус-преобразование достаточно точно представляет исходные изображения с лицами малым числом признаков — спектральных компонент DCT. Это основано на том, что энергетически значимые спектральные компоненты DCT концентрируются в левом верхнем углу матрицы — результата DCT. При этом для относительно точной реконструкции области лица достаточно не более 20 пространственных спектральных компонент, выбранных из верхнего левого угла спектра, если исходные изображения не имеют проблем с освещением [12].

Практическая реализация двумерного DCT (2DDCT). Определим 2DDCT в следующей матричной форме [12]:

$$Y = F_1 I F_2, \quad (5)$$

где I — исходное изображение размера $M \times N$; Y — результат преобразования; F_1 и F_2 — матрицы проекции размером $d_1 \times M$ и $N \times d_2$, причем

$$F_1 = [f(d_1, m)], \quad f(d_1, m) = \begin{cases} 1/\sqrt{M}, & \text{if } d_1 = 0, \\ \sqrt{\frac{2}{M}} \cos \frac{\pi(2m+1)d_1}{2M}, & \text{if } d_1 = 1, \dots, D_1 - 1, \end{cases} \quad \forall m = 0, 1, \dots, M - 1.$$

$$F_2 = [f(n, d_2)], \quad f(n, d_2) = \begin{cases} 1/\sqrt{N}, & \text{if } d_2 = 0, \\ \sqrt{\frac{2}{N}} \cos \frac{\pi(2n+1)d_2}{2N}, & \text{if } d_2 = 1, \dots, D_2, \end{cases} \quad \forall n = 0, 1, \dots, N - 1.$$

Реализация 2DDCT в матричной форме (5) представлена на рис. 5 (см. третью сторону обложки). Заметим, что в матрице-результате Y выделены две области: в левом верхнем углу и в нижнем правом углу. Вектор DCT-признаков формируется из всех компонент верхнего левого угла матрицы-результата C (иногда за исключением компоненты $C(1, 1)$). При этом вектор DCT-признаков будет содержать $d(d+1)/2$ компонент.

Отметим одну важную особенность спектра (5). Двумерное DCT позволяет получить пространственно-спектральные признаки, инвариантные относительно зеркально-симметричного поворота головы влево и/или вправо на исходном изображении. Этот факт подтверждает результат, представленный на рис. 6. Здесь показаны два изображения из базы Yale B, имеющие зеркальную симметрию источника освещения. На рис. 6 Spectrum from Y соответствует обычным (со знаком) спектрам 2DDCT для этих двух изображений. И здесь видно, что значения их не совпадают. Spectrum from $\text{abs}(Y)$ соответствует абсолютному значению спектров 2DDCT

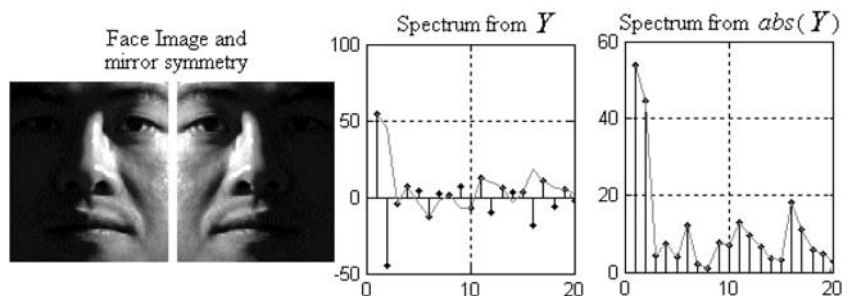


Рис. 6. Спектры 2DDCT для зеркально-симметричных изображений

для этих же изображений. Как показывает график, значения их совпадают полностью. Таким образом, абсолютное значение спектра 2DDCT является инвариантом по отношению к зеркально-симметричным изображениям лиц. Заметим, что в практике распознавания лиц отмеченный факт инвариантности может также обеспечить лучшее распознавание лиц в условиях их поворотов относительно оси Y даже при условии "не совсем точной" зеркальной симметрии источника света.

3. Эксперименты

Представленные в статье результаты выполнены в рамках двух групп экспериментов. В **первой группе экспериментов** база эталонов FaReS включает только 38 изображений — весь состав Subset0. В этом случае база содержит по одному эталону на класс, а составы Subset1—Subset5 определяют 2414 тестовых изображений, принадлежащих всем 38 классам.

Во **второй группе экспериментов** база FaReS включает 228 изображений состава Subset1 (по 6 изображений на класс), а составы Subset2—Subset5 определяют 2151 тестовое изображение также всех 38 классов. При этом ни в одном выполненном эксперименте база FaReS не дополнялась никакими другими изображениями.

Предлагаемый алгоритм обработки и распознавания ИЛПО включает **следующие пять этапов**, что отображено схематически на рис. 7.

1. Формирование двух матриц проекции F_1 и F_2 размером $d_1 \times M$ и $N \times d_2$ соответственно (здесь $d_1 \ll M$ и $d_2 \ll N$; $d_1 \neq d_2$ в общем случае).

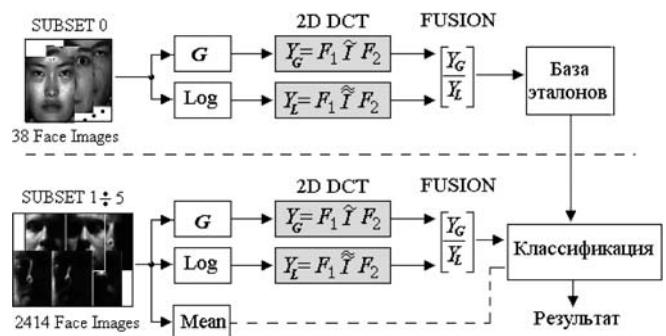


Рис. 7. Структура системы для распознавания ИЛПО

2. Реализация 2DDCT для базы эталонов с предварительной их обработкой по (3) и (4) и записью результата в базу обработанных эталонов. Заметим, что именно здесь реализуется проекция исходных данных — изображений размером $M \times N$ — в новое пространство признаков, в результате чего получаем их представление спектральными матрицами Y_G, Y_L размером $d_1 \times d_2 \ll MN$ так, что:

$$\begin{aligned} Y_G &= F_1 \tilde{I} F_2, \\ Y_L &= F_1 \tilde{\tilde{I}} F_2, \end{aligned} \quad (6)$$

где $\tilde{I}$ — изображение (эталон) после этапа предобработки с помощью гамма-коррекции; $\tilde{\tilde{I}}$ — после логарифмирования.

3. Предобработка каждого тестового изображения по варианту (3) или по вариантам (3) и (4). Выбор варианта предобработки связан со средней яркостью J тестового изображения. При этом если значение J меньше некоторого заданного порога P , то выполняются два вида предобработки, в противном случае — только один, вариант (3).

4. Проекция результата предобработки с шага 3 в новое пространство признаков на основе 2DDCT. Результат проекции — получение спектральных матриц (одной или двух) размером $d_1 \times d_2 \ll MN$.

Окончательный результат проекции с этапов 2 и 4 представляется в форме вектора, считанного с верхнего левого угла спектральной матрицы. Число элементов вектора при этом составляет $d \times (d + 1)/2$, где $d = \min(d_1, d_2)$.

5. Классификация результата, полученного на этапе 4 при его сравнении с признаками 38 классов из базы обработанных эталонов (на этапе 2). Классификация выполнена по критерию минимума расстояний в метрике L1 (KMP/L1). При этом тестовый образ будет принадлежать тому классу из 38, расстояние до которого будет наименьшим, а результат оценивается по $\text{rank} = 1$.

В процессе классификации для каждого тестового изображения вычисляется среднее значение яркости J . Если $J < P$, то результаты проекции этого изображения с этапа 4 объединяются (этап фузии) в один общий вектор и далее выполняется их классификация относительно аналогичных составов признаков, определенных для всех 38 эталонов.

Значение P было выбрано равным 50, как соответствующее случаям теней на половине или большей части области лица. Далее, в ходе решения вариационной задачи, было подтверждено, что значение $P = 50$ обеспечивает наилучший результат при распознавании тестовых изображений с подобными тенями.

Модель первой группы экспериментов имеет следующий вид:

$$\text{YaleB}(38/1/2414)\{\text{G\&Log}/2\text{DDCT}; 192 \times 168 \rightarrow d \times d\}[\Phi + \text{KMP}/\text{L1}/\text{rank} = 1]. \quad (7)$$

Модель второй группы экспериментов имеет следующий вид:

$$\text{YaleB}(228/1/2151)\{\text{G\&Log}/2\text{DDCT}; 192 \times 168 \rightarrow d \times d\}[\Phi + \text{KMP}/\text{L1}/\text{rank} = 1]. \quad (8)$$

Здесь приняты следующие обозначения:

В первой группе экспериментов: Yale(38/1/2414) — использована база изображений, состоящая из 38 эталонов (по одному на класс), а число тестовых образов составило 2414.

Во второй группе экспериментов: Yale(228/1/2151) — использовано 228 эталонов (6 на каждый класс) и использовано 2151 тестовое изображение;

— G&Log/2DDCT — реализована гамма-коррекция/логарифмирование эталонных данных и двумерное косинус-преобразование (то есть преобразования выполнены только на этапе проекции эталонных и тестовых данных);

— $192 \times 168 \rightarrow d \times d$ — представлены размеры исходных изображений и размерность матриц-результатов проекции после редукции размерности пространства признаков;

— $\Phi + \text{KMP}/\text{L1}/\text{rank} = 1$ — использована процедура объединения признаков (фузия) и классификатор по минимуму расстояния с метрикой L1, а результат оценивается по первому месту ($\text{rank} = 1$).

Динамика процесса и результат классификации тестовых изображений в рамках модели (7) показан на рис. 8. Здесь 135 изображений из 2414 были классифицированы неверно, поэтому результат распознавания (RR) составил более 94 %. Вертикальные сплошные линии определяют границы составов данных.

В табл. 1 (число эталонов равно 38 — по одному на класс) приведены результаты классификации тестовых изображений внутри каждого состава. Начальная компонента определяет номер, с которого

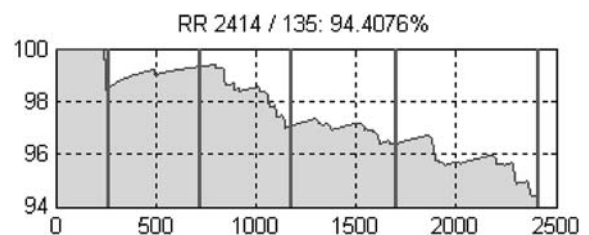


Рис. 8. Результат классификации всех 2414 тестовых изображений

Таблица 1

Параметры классификации	Subset					
	1	2	3	4	5	Все
Значение d	20	30	49	45	49	—
Начальная компонента	1	1	1	1	14	—
Число тестовых изображений	263	456	455	526	714	2414
Не распознано изображений	4	1	30	27	73	135
Результат распознавания, %	98,5	99,8	93,4	94,9	89,8	94,4

начинается отсчет элементов из левого верхнего угла спектральной матрицы. Для параметра НК = 14 это соответствует "обнулению" первых 13 (низкочастотных) компонент спектра. При этом исключается влияние глобальных теней (на исходном изображении) на результат распознавания. Значения параметров d и НК получены при моделировании и исследовании алгоритма классификации в рамках решения соответствующей вариационной задачи.

Процесс и результат классификации (в динамике) тестовых изображений в рамках модели (8) показаны на рис. 9.

В табл. 2 приведены результаты классификации тестовых изображений внутри каждого состава Subset2...Subset5.

Для эталонов использован состав Subset1. При этом поскольку классы 16–18 содержат по шесть изображений (а остальные по семь изображений), то для эталонов использовано только по шесть изображений из каждого класса (всего — 228). В этом эксперименте только 29 изображений из 2151 не были классифицированы. Окончательный результат классификации тестовых изображений составил при этом 98,7 %.

Результаты, показанные на рис. 9 и в табл. 2 (число эталонов равно 228 — по шесть на класс), соответствуют варианту контролируемой классификации тестовых образов и не представляют ошибок распознавания типа FRR. На рис. 10 представ-

лен вариант оценки границ для этих ошибок. Видно, что 74 % тестовых образов классифицировано без ошибок (расстояние $\leq 0,94$), а при значении расстояния большего, чем 0,94, возможно возникновение ошибок типа FRR. Именно эта граница соответствует классификации наиболее сложных изображений состава Subset 5.

Заключение

Показано, что метод 2DDCT в совокупности с процедурами уменьшения градиента яркости исходных изображений, объединением (фузией) признаков по условиям текущей средней яркости, а также исключением низкочастотных компонент спектра позволяет более эффективно распознавать изображения лиц с проблемами освещения — бликами, тенями и очень низким уровнем яркости.

Приведены точные модели реализованных компьютерных экспериментов, структура соответствующей FaReS, реализуемый ею алгоритм и результаты его тестирования на базе Yale B и Yale B+. Полученные при этом результаты выше, чем показанные в работах [1–10], а предложенный алгоритм представляется более простым в реализации. При этом процедуры 2DDCT (вместо, например, 2DPCA или 2DLDA) обеспечивают возможность работы с динамическими базами изображений лиц, что особенно важно в биометрических системах реального времени.

Список литературы

1. Chen W., Er M. J., Wu S. PCA and LDA in DCT domain // Pattern Recognition Letters. 2005. Vol. 26. P. 2474–2482.
2. Chen W., Er M. J., Wu S. Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain // IEEE Trans. Syst. Man Cybern. B Cybern. 2006. Vol. 36, N 2. P. 458–466.
3. Tan X., Triggs B. Preprocessing and Feature Sets for Robust Face Recognition // IEEE Conference on Computer Vision and Pattern Recognition. CVPR'07. 2007. P. 1–8.
4. Xiel X., Zheng W.-S., Lai J. et al. Face Illumination Normalization on Large and Small Scale Features // IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2008. Anchorage, AK, 23–28 June 2008. P. 1–8.
5. Abbas A., Khalil M. I., AbdelHay S., Fahmy H. M. A. // Illumination invariant face recognition in logarithm discrete cosine transform domain // IEEE ICIP'2009. P. 4157–4160.
6. Shao M., Wang Y. Joint Features for Face Recognition under Variable Illuminations // Fifth International Conference on Image and Graphics. ICIG'09, 20–23 Sept. 2009. P. 922–927.
7. Liao H. F., Isa D. New Illumination Compensation Method for Face Recognition // International Journal of Computer and Network Security. March 2010. Vol. 2, N 3. P. 308–321.
8. Hu H., Shan S., Qing L. et al. Lighting Aware Preprocessing for Face Recognition across Varying Illumination // Proc. of the 11th European conference on Computer vision. ECCV10. 2010. Part II. P. 308–321.
9. Tripti G., Nehra V., Vishwakarma V. P. Comparative Analysis of various Illumination Normalization Techniques for Face Recognition // International Journal of Computer Applications. August 2011. Vol. 28, N 9. P. 1–7.
10. Ву Зуи Линь, Кухарев Г. А. Экспресс-метод выравнивания яркости цветных изображений с лицами для систем видеонаблюдения // Известия СПбГЭТУ "ЛЭТИ". 2011. № 1. С. 24–29.
11. Base Yale B. URL: <http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html>
12. Кухарев Г. А., Щеголева Н. Л. Системы распознавания человека по изображению лица. СПб.: Изд-во СПбГЭТУ (ЛЭТИ), 2006. 176 с.

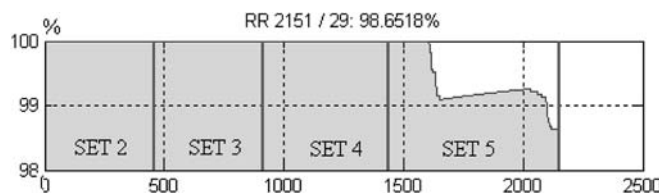


Рис. 9. Динамика классификации 2151 тестового изображения

Таблица 2

Параметры классификации	Subset				
	2	3	4	5	Все
Значение d	30	49	54	48	—
Начальная компонента	1	1	1	14	—
Число тестовых изображений	456	455	526	714	2151
Не распознано изображений	0	0	0	29	29
Результат распознавания, %	100	100	100	96,0	98,7

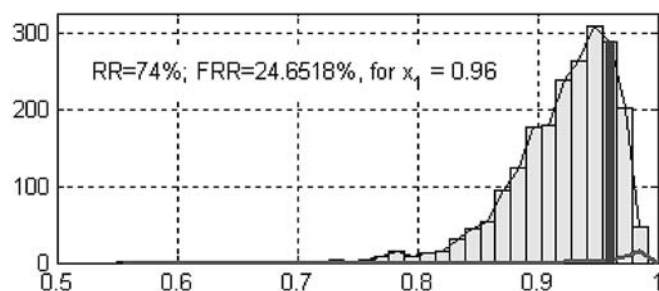


Рис. 10. К оценке ошибок FRR

А. Б. Шабунин¹, руководитель научно-технического комплекса проектирования и разработки информационных систем, e-mail: a_shabunin@hotmail.com,
Н. А. Кузнецов, д-р техн. наук, советник, e-mail: kuznetsov@cplire.ru,
П. О. Скобелев^{3, 4}, д-р техн. наук, вед. науч. сотр., ген. директор, e-mail: petr.skobelev@gmail.com,
И. О. Бабанин⁵, вед. разработчик, e-mail: babanin@gmail.com,
С. С. Кожевников⁵, директор аналитического центра, e-mail: Kozhevnikovss@gmail.com,
Е. В. Симонова⁵, канд. техн. наук, вед. аналитик, e-mail: Simonova.Elena.V@gmail.com,
М. Е. Степанов⁵, разработчик, e-mail: Simonova.Elena.V@gmail.com,
А. В. Царев⁵, ген. директор, e-mail: at@anarun.net

¹ ОАО НИИАС

² Институт радиотехники и электроники им. В. А. Котельникова РАН

³ Институт проблем управления сложными системами РАН

⁴ ООО "НПК "Мультиагентные технологии"

⁵ ООО "НПК "Разумные решения"

Разработка онтологии для мультиагентной системы управления ресурсами ОАО "РЖД"

Описываются принципы реализации онтологии для мультиагентной системы управления ресурсами ОАО "РЖД". Рассматривается структура онтологии базовых ресурсов, функции конструктора онтологий и редактора правил агентов, а также дается пример реализации сцены с разрешением конфликта для планирования расписания движения поездов.

Ключевые слова: мультиагентная система, представление знаний, онтология, сцена, принятие решения, реальное время

Введение

Система управления производственными процессами ОАО "РЖД" представляет собой сверхсложную систему, характеризующуюся большим разнообразием и числом взаимодействующих участников, обнаруживающих конфликтные интересы в

проблемных ситуациях, вызванных непредвиденными событиями. Для решения проблем управления ресурсами РЖД в реальном времени предлагается мультиагентная система (МАС), реализующая адаптивные методы и средства распределения ресурсов, планирования, согласования, мониторинга и контроля исполнения планов в реальном времени, что позволяет существенно повысить эффективность использования ресурсов [1].

В предлагаемой МАС, создаваемой на основе концепции сетей потребностей и возможностей (ПВ-сетей) [2], каждому участнику процесса движения (поезду, локомотиву, вагону, машинисту, участку пути и др.) может быть поставлен в соответствие программный агент, действующий от его лица и по его поручению для динамического планирования потребностей и возможностей с учетом имеющихся целей, предпочтений и ограничений [3].

Ресурсы ОАО "РЖД" весьма разнообразны, и внедрение любой системы управления требует учета индивидуальных особенностей используемых ресурсов и технологических процессов. Для этого в МАС предлагается для описания знаний предметной области, необходимых агентам в процессе принятия решений, использовать онтологический подход [4], развивающий концепцию ПВ-сетей, что позволит отделить предметные знания от программного кода системы и выполнять необходимые настройки специалистам ОАО "РЖД" без привлечения программистов.

Использование онтологий в процессе принятия решений

Онтология — это формализованные концептуальные знания о предметной области (РЖД), представленные в форме, допускающей компьютерную обработку и использование, например, в правилах принятия решений по планированию ресурсов. Для использования в МАС управления ресурсами требуются специализированные онтологии и средства поддержки их построения, позволяющие учитывать специфику предметной области и особенности решаемой задачи согласованного управления ресурсами при обнаружении и разрешении конфликтов, поиске баланса интересов всех участников процесса согласованного планирования [4].

В предлагаемой МАС онтология РЖД представляется в виде семантической сети, образуемой классами понятий и отношений, играющих роль связей и флагов, которые могут анализировать агенты. Фрагмент общей онтологии ОАО "РЖД" в виде семантической сети, вершинами которой являются базовые понятия, а стрелки выражают отношения (связи) между ними (часть—целое, род—вид, бронь и т. д.), представлен на рис. 1.

Примеры классов понятий и отношений:

- объекты: "Груз", "Поезд", "Вагон", "Бригада";
- свойства: Цистерна "Может перевозить" нефть;
- процессы: Груз "доставляется" потребителю, Поезд "формируется" из вагонов;
- отношения: Груз "находится" на поезде, Груз "забронировал" вагон;
- атрибуты поезда: "Плановая дата отбытия", "Станция назначения", "Скорость движения".

Онтология позволяет отделить знания об объектах и их связях от программного кода системы и пополнять описания предметной области РЖД без перепрограммирования системы в каждом случае. Например, пусть необходимо ввести класс ресурсов — локомотивы, для которых требуются бригады машинистов, новые виды запчастей, новые процессы обслуживания и т. д. В онтологию будет введен новый класс потребности и возможности — "локомотив", для которого через специальное отношение "нуждается-в" будут указаны необходимые ресурсы. Для нового класса "локомотив" будет определен процесс "обслуживание" (периодичность, кто выполняет и т. д.). При этом агент потребности локомотива имеет возможность прочитать из онтологии, каким поездам он требуется, и запланировать свою поставку к заданному сроку.

Рассмотренные примеры определяют требования к конструктору онтологий, поддерживающему механизмы работы сетей потребностей и возможностей, введенных ранее в работе [2].

Структура онтологии предметной области ОАО "РЖД"

Для решения задач адаптивного планирования в МАС управления ресурсами ОАО "РЖД" была разработана структура специализированной онтологии предметной области. Модель сущностей онтологии в части описания инфраструктуры и планирования расписания движения поездов представлена на рис. 2.

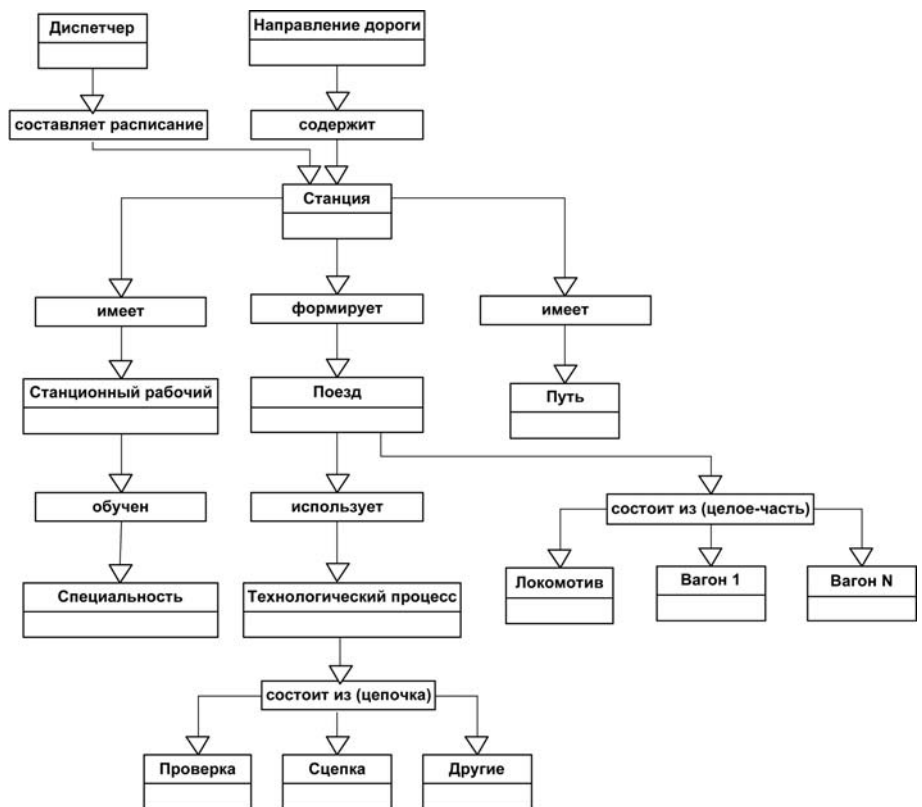


Рис. 1. Фрагмент онтологии РЖД

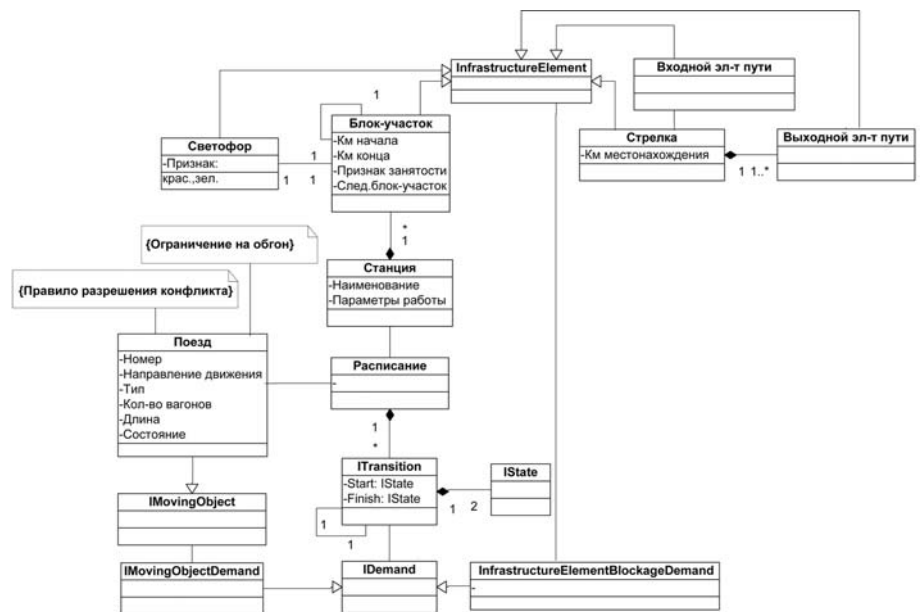


Рис. 2. Фрагмент специализированной онтологии РЖД адаптивного планирования

Кратко рассмотрим отдельные концепты предлагаемой онтологии и их использование в системе.

Элемент инфраструктуры — концепт *InfrastructureElement* описывает часть дорожной инфраструктуры. Атрибуты: входной/выходной путь, временные отношения (наследник концепта *ITransition*). Наследники концепта *InfrastructureElement* —

концепты Блок-участок, Стрелка, Светофор, Переезд.

Станция — концепт *IIEGroup* (коллекция блок-участков) описывает пункт, имеющий путевое развитие для приема, отправления, обгона поездов. Атрибуты: наименование станции, параметры работы.

Движущийся объект — концепт *IMovingObject* описывает движущийся объект, который может занимать блок-участок. Наследники концепта *IMovingObject* — концепты Вагон, Локомотив, Поезд.

Состояние движущегося объекта в определенный момент времени — концепт *IState*. Пример — прибытие поезда на блок-участок.

Временной слот — концепт *ITransition* описывает начальный и конечный моменты времени пребывания движущегося объекта в некотором состоянии. Пример — интервал занятия поездам блок-участка. Концепт осуществляет связку с предыдущим и последующим *ITransition*. Состоит из двух *IState*, описывающих начальный и конечный моменты пребывания объекта в некотором состоянии. Атрибуты: время начала, время окончания, состояние.

Расписание представляет собой множество пар экземпляров концепта *ITransition*, соответствующих экземплярам концепта *IState*.

Требование планирования какого-либо действия — концепт *IDemand*.

Требование планирования расписания движущегося объекта — концепт *IMovingObjectDemand*, наследник концепта *IDemand*. Атрибуты: блок-участок, ссылка на расписание.

Требование к планированию окон работ с подвижными или жестко заданными сроками — концепт *InfrastructureElementBlockageDemand*, наследник концепта *IDemand*.

В онтологии для концептов, отношений и процессов можно задавать правила и ограничения, на основании которых агенты будут принимать решения. Например, существует правило, согласно которому запрещен обгон высокоскоростным поездом (ВСП) грузового поезда, перевозящего щебень в открытой платформе, если оба поезда движутся по блок-участкам, принадлежащим одному перегону. На рис. 3 показано, каким образом данное правило (ограничение на обгон) вводится в онтологию.

В момент планирования обгона любым из поездов будет установлен набор отношений, показывающих условия запрета обгона, который вызовет либо задержку грузового поезда со щебнем на предыдущей станции, либо снижение скорости ВСП, следующего за грузовым поездом. После прохождения данного участка соответствующие отношения будут сняты как флаги синхронизации, регулирующие прохождение участков пути.

С использованием онтологии можно специфицировать конкретные факты и строить модели описания ситуаций (сцен) для работы агентов. Сцена описывает конкретные экземпляры понятий и отношений в заданный момент времени как набор фактов, т. е. является моментальной "фотографией" ситуации в реальном мире.

3. Пример сцены для планирования расписания поездов

Рассмотрим сцену, в которой должно быть запланировано движение нескольких поездов по одному блок-участку. При этом возникает конфликт, если скорость второго поезда больше, чем первого. Возможны следующие правила разрешения конфликта:

- 1) увеличение скорости движения поезда до максимально возможной;
- 2) уменьшение скорости движения поезда вплоть до останова;
- 3) увеличение продолжительности стоянки поезда.

Рассмотрим более подробно использование правила № 3, которое в онтологии формулируется следующим образом. Если значение атрибута "приоритет" (определяемого типом поезда) у одного поезда выше, чем у второго поезда, то для второго поезда необходимо увеличить значение атрибута "Finish" концепта "ITransition", задающего интервал времени пребывания на станции в расписании поезда.

Пусть в сцене, созданной на основе онтологии, показанной на рис. 2, участвуют 4 поезда: ВСП "Сапсан" (№ 161) и 3 пас-

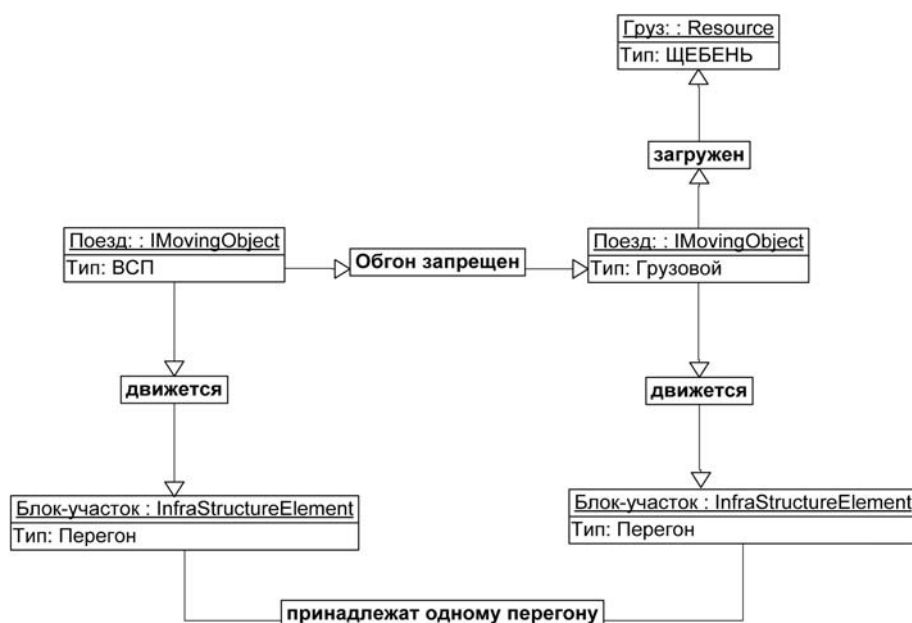


Рис. 3. Представление в онтологии правила принятия решения

сажирских поезда (№ 55, № 101, № 15). ВСП "Сапсан" имеет наивысший приоритет, приоритеты пассажирских поездов одинаковы. Требуется запланировать движение поездов через станции "Клин" — "Подсолнечная" — "Поварово" — "Крюково" в соответствии с указанным в сцене расписанием. При этом график движения ВСП "Сапсан" в районе станции "Подсолнечная" должен пересечь графики движения пассажирских поездов. После запуска сценария планирования (рис. 4), реализуемого на базе проблемно-ориентированного языка в платформе "Вектор" (разработка ООО "ПрограмПарк" [3]), конфликт будет разрешен агентами согласно правилу № 3 (рис. 5). На рис. 5 приведены графики движения поездов с указанными выше номерами.

Заключение

В настоящее время в ОАО "РЖД" ведется работа по созданию интеллектуальных систем управления пассажирским и грузовым движением, позволяющих повысить качество железнодорожных услуг, сократить расходы и сроки перевозок, снизить риски.

Предлагаемая мультиагентная система адаптивного планирования ресурсов направлена на создание интеллектуальных систем управления движением РЖД, что является одним из актуальных и значимых направлений развития автоматизированных систем управления железнодорожным транспортом будущего.

Авторы выражают благодарность Российскому Фонду Фундаментальных Исследований и Министерству науки и образования РФ, поддержавших разработку по созданию интеллектуальных систем управления железнодорожным транспортом в реальном времени по гранту РФФИ № 11-07-13119-офи-м-2011-РЖД и государственному контракту Минобрнауки РФ № 07.514.11.4094.

```
#161 поезд. Сапсан
obj = UserObject.create do ; is :K_Задачадвижения; into "Задачидвижения"; end
obj.имя = "поезд 161. расписание"; obj.исполнитель = поезд.получить 161;

ts = User::UserObject.get_by_attr_val('имя', 'поезд 161. расписание', 'задачидвижения');
sbt = ts.создать_подзадачу(поезд 161. расписание. Клин);
sbt.ресурс = User::UserObject.get_by_attr_val('имя', 'Клин', 'пассажирскаястанция');
sbt.время_начала_план = Time.parse('Tue Nov 22 13:00:00 +0300 2011');
sbt.время_окончания_план = Time.parse('Tue Nov 22 13:00:00 +0300 2011');

ts = User::UserObject.get_by_attr_val('имя', 'поезд 161. расписание', 'задачидвижения');
sbt = ts.создать_подзадачу(поезд 161. расписание. Подсолнечная);
sbt.ресурс = User::UserObject.get_by_attr_val('имя', 'Подсолнечная', 'пассажирскаястанция');
sbt.время_начала_план = Time.parse('Tue Nov 22 13:30:00 +0300 2011');
sbt.время_окончания_план = Time.parse('Tue Nov 22 13:32:00 +0300 2011');

ts = User::UserObject.get_by_attr_val('имя', 'поезд 161. расписание', 'задачидвижения');
sbt = ts.создать_подзадачу(поезд 161. расписание. Поварово I);
sbt.ресурс = User::UserObject.get_by_attr_val('имя', 'Поварово I', 'пассажирскаястанция');
sbt.время_начала_план = Time.parse('Tue Nov 22 14:30:00 +0300 2011');
sbt.время_окончания_план = Time.parse('Tue Nov 22 14:40:00 +0300 2011');

ts = User::UserObject.get_by_attr_val('имя', 'поезд 161. расписание', 'задачидвижения');
sbt = ts.создать_подзадачу(поезд 161. расписание. Крюково);
sbt.ресурс = User::UserObject.get_by_attr_val('имя', 'Крюково', 'пассажирскаястанция');
sbt.время_начала_план = Time.parse('Tue Nov 22 16:00:00 +0300 2011');
sbt.время_окончания_план = Time.parse('Tue Nov 22 16:05:00 +0300 2011');

obj.подзадачи.each do UserObject.get(x).переместить_в "подзадачидвижения"; end
#Запуск задач на исполнение
#1. Инициализация и конфигурирование (инфраструктура)
#2. Загрузка задач Plugin.call ServiceMAP.info[:name], :process_changed, [101, 55, 15, 161]
# Ожидание завершения расчета расписания
while true break if Plugin.call(User::ServiceMAP.info[:name], :is_valid) == true end;
```

Рис. 4. Фрагмент сценария планирования

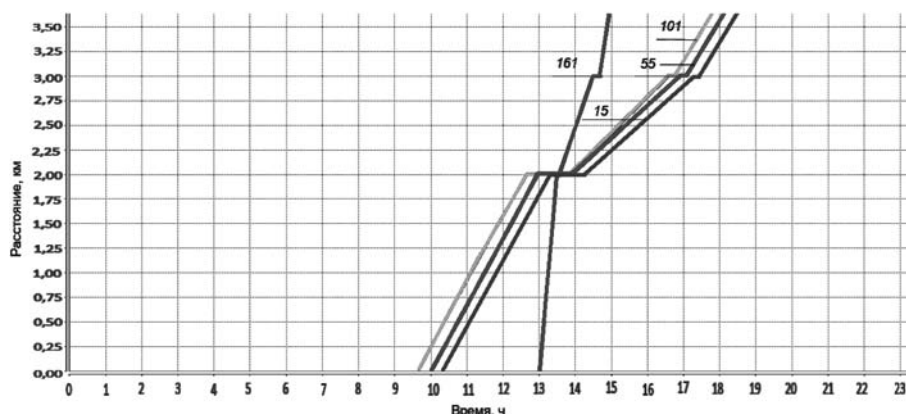


Рис. 5. Разрешение конфликта в расписании

Список литературы

1. Скобелев П. О. Мультиагентные технологии в промышленных применениях: к 20-летию основания Самарской научной школы мультиагентных систем // Мехатроника, автоматизация, управление. 2010. № 12. С. 33—46.
2. Виттих В. А., Скобелев П. О. Мультиагентные модели взаимодействия для построения сетей потребностей и возможностей в открытых системах // Автоматика и Телемеханика. 2003. № 1. С. 177—185.
3. Шабунин А. Б., Чехов А. В., Скобелев П. О., Кузнецов Н. А., Симонова Е. В., Бабанин И. О., Кожевников С. С., Степанов М. Е., Царев А. В., Сазуров С. В., Курбатов Е. В., Дмитриев Д. В. Сетевые подходы к созданию распределенных систем управления ресурсами ОАО "РЖД" на основе мультиагентных технологий // Тр. XIV Междунар. конф. "Проблемы управления и моделирования в сложных системах". Самара, 22—25 июня 2012 г. Самара: СНЦ РАН. 2012. С. 724—734.
4. Скобелев П. О. Онтология деятельности для ситуационного управления предприятиями в реальном времени // Онтология проектирования. 2012. № 1 (3). С. 6—38.

В. В. Нечаев, д-р физ.-мат. наук (ВМАКК),
канд. техн. наук, проф., зав. каф.,
e-mail: nechayev@mirea.ru,
Московский государственный
технический университет радиотехники,
электроники и автоматики

Раскрытие неопределенности системной задачи, представленной концептуальной моделью

Рассматривается возможность раскрытия неопределенности конкретных пользовательских задач (КПЗ), представленных на основе интерпретации концептуальной модели системной задачи (КМСЗ). Информационная неопределенность КПЗ рассматривается в контексте анализа КМСЗ. Для раскрытия информационной неопределенности КПЗ предлагается использовать рекурсивный механизм порождения концептуальных моделей вспомогательных задач (КМВЗ), посредством которого КМСЗ трансформируется в концептуальную модель задачной системы (КМЗС). Для оценки степени информационной неопределенности КПЗ вводится основанный на КМСЗ комплексный показатель — ПЦНД, включающий частные показатели полноты, целостности, необходимости и достаточности. Приводятся обобщенная и полная структурно-логические схемы, отражающие процессы порождения комплекса КМВЗ для случая максимальной неопределенности КМСЗ.

Ключевые слова: задача, задачная система, вспомогательная задача, системная задача, модель, концептуальная модель, концептуальный уровень (слой), среда существования задачи, решатель задач, цель задачи, условия задачи, метод, эксперт, алгоритм, программа, результат, адекватность, релевантность, качество, эффективность, полнота, целостность, необходимость, достаточность, определенность, корректность, ранжирование, рекурсия, рекурсивный механизм

Введение

Предлагаемая вниманию читателя работа является развитием статьи [1], в которой изложены вопросы создания концептуальной модели системной задачи, а также сама модель. Разработанная модель ориентирована на формирование концептуального уровня (слоя) представления задач, включая концептуальную схему, в рамках архитектуры интеллектуального решателя системных задач (ИРСЗ). Подобный подход, основанный на технологии концептуального слоя — страты представления информации, широко используется в базах данных и знаний. Ядром подобных технологий служат соответ-

ствующие концептуальные модели и схемы. Введение концептуального слоя в архитектуру решателя системных задач дает возможность сделать относительно независимыми, но адекватно отображаемыми как пользовательские (внешние), так и внутренние, реализуемые в среде ИРСЗ, представления задач. Предлагаемая архитектура позволяет существенно расширить спектр пользовательских задач, решаемых в информационной и программно-аппаратной среде ИРСЗ. Решение полностью определенных пользовательских задач на основе технологии концептуального слоя (ТКС) не вызывает существенных затруднений. Однако решение некорректных, не полностью определенных, трудно формализуемых системных задач часто связано со значительными сложностями. Для раскрытия неопределенности таких задач используются различные методы, в основном эвристические. В настоящей работе для решения задач подобного рода предлагается метод, основанный на использовании рекурсивного механизма. В рамках концептуальной модели системной задачи (КМСЗ) рекурсивный механизм дает возможность генерировать концептуальные модели подзадач исходной системной задачи, каждая из которых выполняет функцию вспомогательной задачи. Принимая во внимание функциональное назначение порождаемых подзадач, в дальнейшем изложении будем их определять как концептуальные модели вспомогательных задач (КМВЗ). Описанию и реализации предлагаемого метода и посвящена настоящая работа.

Анализ неопределенности концептуальной модели системной задачи

Концептуальная модель системной задачи, согласно [1], представляется семиотической системой, состоящей из четырех кортежей. Каждый компонент в кортежах семиотической системы определяется как многосортное множество:

$$\left. \begin{aligned} P_{\Sigma} &= \langle \Sigma_P, Z_P, C_P, I_P \rangle; \\ C_P &= \langle M_P, A_P, P_P, \Gamma_P \rangle; \\ I_P &= \langle D_P, K_P \rangle; \\ \Gamma_P &= \langle Q_{WP}, \varepsilon_{fP} \rangle. \end{aligned} \right\} \quad (1)$$

В рамках концептуальной модели (1) приняты следующие обозначения: P_{Σ} — системная задача; Σ_P — среда (предметная область) возникновения и существования системной задачи; Z_P — цель, на достижение которой направлено решение задачи; C_P — условия, при которых осуществляется решение системной задачи; I_P — информационное обеспечение (данные — D_P и знания — K_P), отражающее содержательное описание задачи и область определения ее исходных данных; M_P — метод решения системной задачи; A_P — алгоритм реализации ме-

тогда; P_p — программа реализации алгоритма; G_p — комплексный двухкритериальный показатель адекватности результата решения системной задачи, включающий показатель качества — Q_{wp} и показатель эффективности — ε_{fp} .

Для случая рутинной — полностью определенной задачи, т. е. когда на концептуальном уровне все элементы кортежей (1) определены, а для получения конечного результата R_p ее решения необходимо и достаточно выполнить над исходными данными D_p совокупность процедур и операций (команд), определяемых программой P_p , постановка рутинной задачи P_Σ будет иметь следующий вид:

$$\left. \begin{array}{l} \text{Известно:} \\ P_\Sigma = \langle \Sigma_p, Z_p, C_p, I_p \rangle; \\ C_p = \langle M_p, A_p, P_p, G_p \rangle. \\ \text{Требуется определить: } R_p = ? \end{array} \right\} \quad (2)$$

В реальных условиях теоретической и практической деятельности рутинные задачи P_Σ , т. е. такие, для которых концептуальная модель (1) полностью определена, в большинстве случаев являются исключением, а не правилом. Наиболее часто встречаются частично определенные, неполные и некорректные задачи. Для выявления факторов, характеризующих неопределенность КМСЗ, проводится ее анализ.

КМСЗ, представленная в форме системы кортежей (1), на концептуальном уровне анализа и, в частности, в структурном аспекте удовлетворяет условиям полноты, целостности, а также необходимости и достаточности. Отметим, что условия полноты и целостности определяются через соответствующие атрибуты и их значения, задаваемые для представления КМСЗ на уровне концептуального слоя ИРСЗ. Необходимость и достаточность — это условия, накладываемые на представление конкретной пользовательской задачи в форме КМСЗ.

Полнота КМСЗ предполагает, что в рамках концептуальной модели известны (заданы или определены) все компоненты системы кортежей (1) по всем аспектам, атрибутам и их значениям — характеристикам и параметрам, определяемым уровнем абстракции, т. е. стратой, на которой рассматривается КМСЗ.

Целостность КМСЗ определяется объектной (морфологической или синтаксической), а также семантической и логической связностью компонентов кортежей системы (1), т. е. объектной, семантической и логической структурами модели задачи.

Необходимость и достаточность КМСЗ рассматриваются в контексте пользовательской задачи, т. е. в соответствии с результатами интерпретации КМСЗ на основе знаний и данных, определяемых конкретной пользовательской задачей. Семантическая, синтаксическая, качественная и количественная интерпретации КМСЗ дают возможность представить пользовательскую задачу в стандартной по отношению к ИРСЗ форме и включить все сведения, требуе-

мые для достижения полной определенности пользовательской задачи и достаточные для ее решения.

Замечание. Полный комплекс перечисленных выше условий в дальнейшем будем обозначать аббревиатурой ПЦНД.

Таким образом, как только возникает потребность практического использования КМСЗ для решения конкретных пользовательских задач P_Σ , так сразу же возникает и необходимость отображения пользовательской задачи P_Σ в структуру концептуальной модели системной задачи P_Σ , а также ее комплексной интерпретации на семантическом, синтаксическом (структурном), качественном и количественном уровнях. Отметим, что интерпретации подлежат не только все компоненты и элементы всех кортежей системы (1), но и отношения между этими компонентами и элементами. У специалистов интерпретирование КМСЗ в общем случае не вызывает затруднений. Однако при работе с конкретными задачами, в силу возможной информационной неопределенности таких задач (см. [1]), связанной с неполнотой и некорректностью описаний, недостатком или отсутствием конкретных данных для их решения, возникают проблемы доопределения данных и знаний — информационного обеспечения этих задач. Доопределение задач реализуется прежде всего на уровне компонентов кортежей системы (1) в случаях их неполного или некорректного, например иррелевантного, описания. Необходимо принимать во внимание и такой фактор, как когерентность — степень влияния (значимости) каждого компонента кортежа системы (1) на другие компоненты этих же кортежей, а как следствие, и на уровень неопределенности задачи в целом. Фактор значимости (назовем его весовым фактором) компонентов в структуре КМСЗ проиллюстрируем следующим примером. Возьмем для рассмотрения из моделей (1) и (2) четыре компонента: метод M_p , алгоритм A_p , программу P_p и результат R_p . Определим связность этих компонентов посредством соответствующих отношений. В структуре задачи P_Σ между выделенными компонентами имеет место отношение следования от метода к результату:

$$\begin{aligned} & \text{Метод } (M_p) \rightarrow \text{Алгоритм } (A_p) \rightarrow \\ & \rightarrow \text{Программа } (P_p) \rightarrow \text{Результат } (R_p). \end{aligned}$$

Очевидно, что результат R_p невозможно получить, не имея программы P_p или плана решения задачи; программу P_p невозможно написать, не зная в явном или неявном виде алгоритма A_p ; алгоритм невозможно разработать, не зная метода M_p решения задачи. Таким образом, в приведенной последовательности отношение следования усилено условиями отношения строгого порядка. При заданных условиях в последовательности:

$$M_p \rightarrow A_p \rightarrow P_p \rightarrow R_p$$

в соответствии с отношением строгого порядка каждый предшествующий компонент в определенном

смысле доминирует и включает в себя последующий. Иными словами, в структуре КМСЗ имеет место иерархическая организация выделенных компонентов модели (1). Отметим, что при анализе влияния (действия) комплексного показателя адекватности Γ_P между компонентами кортежей имеет место более сложная система отношений. Для рассматриваемых выше четырех компонентов системной задачи P_Σ показатель адекватности Γ_P устанавливает следующую систему отношений, отражающих соответствия или ограничения для бинарных сочетаний:

- метод M_P находится в отношении релевантности и логической адекватности к задаче P_Σ ;
- алгоритм A_P находится в отношении логической адекватности к методу M_P решения задачи P_Σ ;
- программа P_P находится в отношении синтаксической корректности, логической адекватности и эквивалентности к алгоритму A_P реализации метода M_P ;
- результат R_P решения задачи P_Σ находится в отношении адекватности (или точности) к цели Z_P решаемой задачи P_Σ при условии положительной верификации программы P_P .

Проведенный анализ неопределенности КМСЗ дает возможность рассмотреть вопрос многоуровневого (стратифицированного) представления концептуальной модели системной задачи P_Σ , сформировать шкалу неопределенности этой задачи P_Σ и по введенной шкале ранжировать компоненты кортежей системы (1).

Ранжирование компонентов КМСЗ по шкале неопределенности

Многоуровневое представление КМСЗ P_Σ с учетом результатов анализа осуществляется на основе метода стратификации. Для формирования уровней (страт, слоев) организации задачи P_Σ необходим критерий или основание стратификации. В качестве такого критерия в рассматриваемом случае целесообразно выбрать показатель неопределенности системной задачи P_Σ . В соответствии с указанным критерием введем качественную шкалу неопределенности. На шкале неопределенности ранжируем компоненты концептуальной модели, представленной системой кортежей (1), по степени возрастания неопределенности. При этом будет иметь место следующая последовательность страт — уровней (St) неопределенности системной задачи:

- конечного результата решения задачи R_P ;
- исходной информации I_P , формируемой в рамках области определения (данных D_P и знаний K_P);
- показателя адекватности Γ_P ;
- программы P_P реализации алгоритма A_P ;
- алгоритма A_P реализации метода M_P ;
- метода M_P решения задачи P_Σ ;
- цели Z_P системной задачи P_Σ ;
- среды Σ_P существования задачи P_Σ .

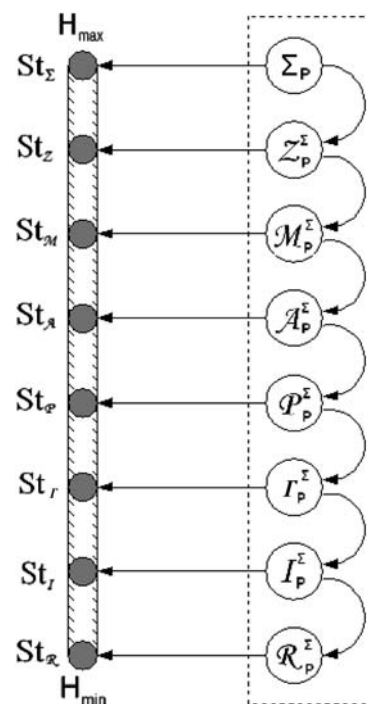


Рис. 1. Ранжированное представление компонентов концептуальной модели системной задачи на шкале неопределенности

Графическая интерпретация шкалы неопределенности имеет вид, представленный на рис. 1.

В последовательности страт (рис. 1) уровень неопределенности H задачи P_Σ возрастает при движении по шкале снизу вверх, т. е. от $H_{\min}$ до $H_{\max}$. При этом каждый выделенный на шкале уровень — страта St разбивает ее на две части. Верхняя часть шкалы включает все известные (определенные) компоненты, а нижняя — компоненты, которые необходимо определить или доопределить, т. е. неизвестные.

Замечание. В приведенном перечне выделенных страт, формирующих шкалу неопределенности, нижние шесть уровней характеризуют неопределенность непосредственно системной задачи P_Σ , а два верхних — цель Z_P и среда Σ_P — отражают этапы конкретизации проблемы Π_Σ , порождающей задачу P_Σ , т. е. этапы перехода от проблемы Π_Σ к системной задаче P_Σ .

С учетом стратифицированной шкалы неопределенности концептуальной модели системной задачи P_Σ ее многоуровневая иерархическая схема принимает вид, представленный на рис. 2. Следует отметить, что на этой схеме стрелки, связывающие блоки справа сверху вниз, отражают логическую последовательность раскрытия неопределенности задачи. Ранжировав страты на шкале неопределенности и стратифицировав системную задачу P_Σ , мы исходим из предположения, что задача P_Σ может иметь одно, два или более неизвестных. Следовательно, возникает проблема определения этих неизвестных и снятия соответствующей неопределенности.



Рис. 2. Схема стратифицированного многоуровневого представления концептуальной модели системной задачи

Для раскрытия информационной неопределенности исходной целостной концептуальной модели системной задачи P_Σ и сведения ее к полностью определенной (рутинной) используются различные, в основном эвристические, методы [2–6]. В настоящей работе разрешение проблемы снятия неопределенности осуществляется на основе преобразования КМСЗ в концептуальную модель задачной системы (КМЗС). Компонентами КМЗС выступают вспомогательные задачи — подзадачи КМСЗ. Для такого преобразования автором предлагается метод рекурсивного порождения концептуальных моделей вспомогательных задач (КМВЗ). Сущность метода состоит в том, что для каждого неизвестного компонента концептуальной модели системной задачи P_Σ оформляется по рекурсии концептуальная модель вспомогательной задачи (КМВЗ), аналогичная КМСЗ, задаваемой кортежем (1). Таким образом:

раскрытие информационной неопределенности исходной системной задачи, представленной концептуальной моделью, осуществляется посредством порождения на основе рекурсивного механизма вспомогательных задач, соответствующих информационно неопределенным компонентам КМСЗ, и решения этих задач для получения необходимой информации, раскрывающей содержание недоопределенных компонентов [5–7].

Рекурсивные процедуры формирования КМВЗ

Исходя из приведенного выше описания сущности метода раскрытия неопределенности концептуальной модели задачи на основе механизма рекур-

сии, можно сделать заключение о необходимости преобразования системной задачи P_Σ , представленной многоуровневой схемой рис. 2, в задачную систему P_S со структурой, представленной совокупностью задач P_S^* , каждая из которых является компонентом задачи P_Σ . Такое преобразование выполняется в тех случаях, когда соответствующие компоненты системной задачи P_Σ переходят в категорию "неизвестное". Процесс рекурсивного задачного оформления неизвестных компонентов, согласно шкале неопределенности, протекает снизу вверх поэтапно. При этом для каждого нового неизвестного на основе рекурсивной процедуры, по аналогии с (1), посредством переобозначения индексов и соответствующей интерпретации для каждого конкретного случая оформляется концептуальная модель системной задачи P_S^* соответствующего уровня неопределенности. Концептуальные модели порождаемых вспомогательных задач P_S^* оформляются по аналогии и для всех последующих нижних компонентов исходной КМСЗ, следующих за последним верхним неизвестным. Если в качестве неизвестного выступает, например, метод M_P , то все компоненты системной задачи P_Σ , находящиеся на шкале неопределенности ниже метода M_P , также неизвестны или недоопределены.

Следовательно, системная задача P_Σ имеет максимальную неопределенность. Согласно рекурсивной процедуре, для всех существующих неизвестных компонентов КМСЗ допускается оформление соответствующих вспомогательных концептуальных моделей задач P_S^* . Формальную процедуру такого преобразования представим гомоморфным отображением вида

$$P_S^* = \text{Morf } H_\Sigma^* : \{P_\Sigma \rightarrow P_S^*\}, \quad (3)$$

где запись $\text{Morf } H$ — оператор гомоморфного отображения; символ "*" — обобщенный индекс вспомогательной задачи, принимающий конкретные значения соответствующих вспомогательных задач.

Концептуальные модели вспомогательных задач P_S^* , составляющие задачную систему P_S , формируются исходя из стратифицированного представления концептуальной модели системной задачи P_Σ (см. рис. 2). Каждая вновь порождаемая концептуальная модель задачи P_S^* имеет однозначно фиксированное имя, которое совпадает с именем соответствующего неизвестного. Полная номенклатура таких задач ранжируется в соответствии со шкалой неопределенности (см. рис. 1) в порядке возрастания уровня неопределенности. Она задается следующей совокупностью формальных записей гомоморфных отображений, отражающих процессы формирования КМВЗ:

- задача P_S^R уровня конечного результата R_P :

$$P_S^R = \text{Morf } H_\Sigma^R : \{P_\Sigma \rightarrow R_P\}; \quad (3a)$$

- задача P_S^I уровня исходной информации I_P :

$$P_S^I = \text{Morf } H_\Sigma^I : \{P_\Sigma \rightarrow P_S^I\}; \quad (3б)$$

- задача P_S^T уровня адекватности Γ_P :

$$P_S^T = \text{Morf } H_\Sigma^T : \{P_\Sigma \rightarrow P_S^T\}; \quad (3в)$$

- задача P_S^P уровня программы P_P :

$$P_S^P = \text{Morf } H_\Sigma^P : \{P_\Sigma \rightarrow P_S^P\}; \quad (3г)$$

- задача P_S^A уровня алгоритма A_P :

$$P_S^A = \text{Morf } H_\Sigma^A : \{P_\Sigma \rightarrow P_S^A\}; \quad (3д)$$

- задача P_S^M уровня метода M_P :

$$P_S^M = \text{Morf } H_\Sigma^M : \{P_\Sigma \rightarrow P_S^M\}. \quad (3е)$$

Обобщенная структурно-логическая схема формирования КМВЗ

Гомоморфные отображения, представленные выражениями (3а)—(3е), отражают только факт порождения вспомогательных задач P_S^* на основе исходной системной задачи P_Σ , иначе — рекурсивность процедуры формирования КМВЗ. Конкретизация этих выражений в настоящей работе осуществляется на структурно-логическом уровне. Как механизм, так и процесс рекурсивного формирования КМВЗ в общем случае не зависят от конкретного компонента КМСЗ, для раскрытия неопределенности которого генерируется КМВЗ. Следовательно, возникает возможность создания обобщенной модели этого процесса для некоторой абстрактной вспомогательной задачи КМВЗ- q . Символ " q " в обозначении вспомогательных задач задает номер или имя анализируемого компонента КМСЗ. Модель процесса формирования КМВЗ- q целесообразно представить в форме структурно-логической схемы (органиграммы). Процесс рекурсивного порождения КМВЗ- q начинается с анализа компонентного состава КМСЗ, определяемого системой (1). Для каждого компонента КМСЗ в соответствии с порядком, задаваемым шкалой неопределенности (см. рис. 1), проводится анализ информационного наполнения компонентов. Для обозначения конкретной КМВЗ символ " q " заменяется соответствующим этой задаче символом, задающим элемент кортежа (1). Например, для задачи P_M метода M_P будет иметь место обозначение КМВЗ — M_P . В результате информационного анализа компонентов КМСЗ возможны различные исходы:

- компонент КМСЗ полностью определен;
- компонент КМСЗ недоопределен;
- компонент КМСЗ в существующих условиях определить невозможно. Как следствие, возможны три варианта действий. Рассмотрим эти варианты.

Полностью определенная задача. Если в процессе анализа q -го компонента КМСЗ получен результат, полностью удовлетворяющий условиям информационной полноты, целостности, необходимости и достаточности (ПЦНД), т. е. если q -й компонент КМСЗ полностью определен, то осуществляется переход к $(q + 1)$ -му компоненту. В тех случаях, когда в результате анализа условия ПЦНД выполняются для всех элементов кортежей (1), имеет место полностью определенная задача. Для получения конечного результата такая задача передается для решения в интеллектуальный решатель системных задач (ИРСЗ).

Недоопределенная задача. В тех случаях, когда условия ПЦНД не выполняются, т. е. q -й компонент КМСЗ определен не полностью, включается рекурсивный механизм его доопределения. При этом формируется вспомогательная задача КМВЗ- q , через которую и доопределяется q -й компонент КМСЗ. Процесс рекурсивного оформления вспомогательной задачи КМВЗ- q включает перезапись системы кортежей (1), индексное переобозначение элементов, а также интерпретирование всех элементов сформированной системы кортежей КМВЗ- q на семантическом, синтаксическом, качественном и количественном уровнях. Результаты процесса формирования КМВЗ- q анализируются и оцениваются по тем же четырем условиям ПЦНД, но адаптированным к q -й вспомогательной задаче. Если по результатам анализа будет сделано заключение о некорректности сформированной КМВЗ- q , то актуализируется алгоритм анализа причин некорректности, а после их выявления запускается процесс переформирования КМВЗ- q . Если же по результатам анализа корректности сформированной вспомогательной задачи получено положительное заключение, то задача фиксируется и передается в решатель системных задач. Решение сформированной вспомогательной задачи направлено на определение q -го компонента КМСЗ и реализуется в среде проблемно-ориентированного ИРСЗ или универсального, например ЭВМ. Процесс решения протекает в соответствии с алгоритмом A_P . Результат решения КМВЗ- q должен совпадать с информационным наполнением q -го компонента КМСЗ. Степень совпадения (релевантности и адекватности) определяется соответствующими комплексными условиями ПЦНД, а также частным показателем адекватности, принадлежащим комплексному показателю Γ_P . В случае выполнения заданных условий процесс определения q -го компонента КМСЗ завершается и осуществляется переход к определению следующего, не полностью определенного $(q + 1)$ -го компонента.

При отрицательном результате решения КМВЗ- q может быть использован итеративный механизм доопределения q -го компонента КМСЗ. В тех случаях, когда и итеративный процесс доопределения

не дает положительного результата, возможны: либо углубление рекурсии, т. е. формирование новой КМВЗ второго или третьего порядка и реализация ее решения, либо отказ от решения КМВЗ- q и, как следствие, признание невозможности в заданных условиях решения исходной системной задачи P_Σ . Структурно-логическая схема модели описанного выше процесса раскрытия неопределенности q -го компонента КМСЗ представлена на рис. 3.

Комментарий. При анализе КМСЗ на уровне q -го компонента возможны следующие исходы:

- q -й компонент известен или задан пользователем вместе с описанием;
- q -й компонент задан, а описание его применения имеется в базе знаний методов решения системных задач (МРСЗ);
- q -й компонент известен приближенно, и необходимо его доопределение;
- q -й компонент может быть определен по аналогии с соответствующим доопределением (корректировкой);
- q -й компонент неизвестен — необходимо его нахождение (поиск, разработка).

Замечание. Для сложных, комплексных (составных) системных задач допустимо определение q -го компонента на основе их декомпозиции на подзадачи, каждая из которых имеет известный метод решения.

Преобразование КМСЗ в концептуальную модель задачной системы

При использовании предложенного метода формирования вспомогательных задач для всех компонентов кортежа (1) концептуальная модель системной задачи P_Σ преобразуется в концептуальную модель задачной системы P_S . Формирование на основе рекурсивного механизма концептуальной модели задачной системы для случая максимальной неопределенности системной задачи P_Σ как совокупности концептуальных моделей вспомогательных задач — подзадач $\{P_S^*\}$ задачи P_Σ и составляет содержание настоящего раздела.

Замечание. Используемые выше понятия "системная задача" и "задачная система" — не игра слов, как может показаться на первый взгляд. Они имеют различную форму представления. В концептуальной схеме КМСЗ каждый ее компонент выступает в качестве супервершины графа по отношению к концептуальной схеме КМСЗ, в которой каждый компонент — это вспомогательная задача, представляемая концептуальной схемой КМВЗ. Однако,

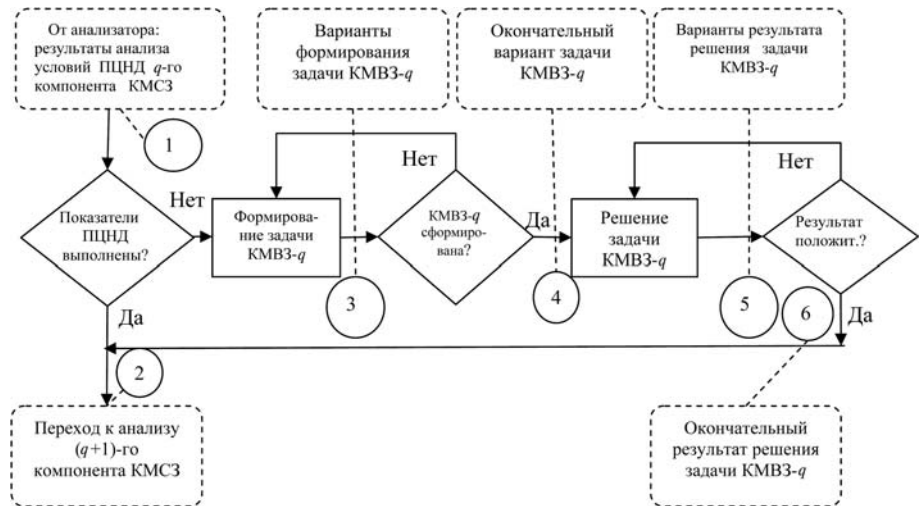


Рис. 3. Структурно-логическая схема процесса раскрытия неопределенности q -го компонента КМСЗ

с точки зрения целевого назначения, а следовательно, и конечного результата R_P системная задача P_Σ и задачная система P_S эквивалентны, т. е. $P_\Sigma \sim P_S$.

Учитывая изложенное, а также принимая во внимание содержательное определение понятия "системная задача", дадим формулировку определения понятию "задачная система".

Определение. Задачная система P_S — это совокупность $\{P_S^*\}$ взаимосвязанных и взаимодействующих вспомогательных задач $P_S^* \subset P_S$, каждая из которых рекурсивно порождается соответствующим неизвестным компонентом системной задачи P_Σ , реализует функцию этого компонента и выступает в качестве подзадачи $P_S^* \subset P_S$ исходной задачной системы P_S . Результаты решений R_P^* подзадач $P_S^* \subset P_S$ направлены на раскрытие неопределенности исходной целостной задачи P_Σ и, как следствие, ее сведение к рутинной.

Комментарий. Структура модели задачной системы P_S на концептуальном уровне представления изоморфна структуре порождающей ее концептуальной модели системной задачи P_Σ . Однако в первом случае в качестве носителя структуры выступают подзадачи $P_S^* \subset P_S$, а во втором — компоненты задачи P_Σ .

Таким образом, исходная системная задача P_Σ трансформируется в производную задачную систему P_S . Структурно-логическая схема (органиграмма), отражающая процессы формирования полного комплекса вспомогательных задач и, как следствие, преобразования концептуальной модели системной задачи P_Σ в концептуальную модель задачной системы P_S , на основе которой осуществляется раскрытие неопределенности исходной КМСЗ, представлена на рис. 4.

Комментарий. В тех случаях, когда в системе кортежей (1) нечетко определены цель Z_P , на дости-

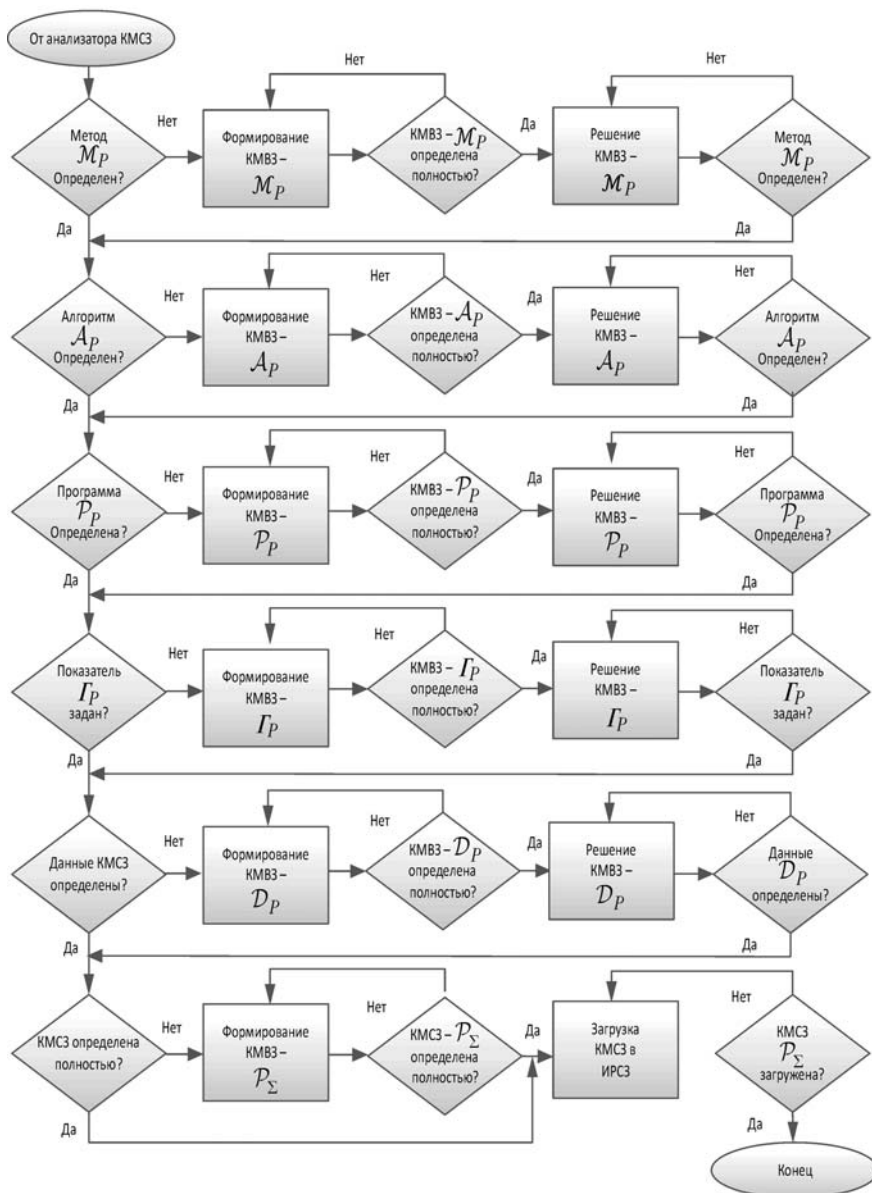


Рис. 4. Структурно-логическая схема процессов формирования полного комплекса вспомогательных задач и раскрытия неопределенности КМСЗ на основе рекурсивного механизма

жение которой направлено решение задачи, и/или среда Σ_P существования системной задачи P_Σ , т. е. когда имеет место проблема P_Σ и возникает необходимость раскрытия неопределенности на уровне проблемы P_Σ , то по аналогичной схеме могут порождаться задачи для компонент, определяющих цель Z_P и среду существования задачи Σ_P .

Заключение

В результате анализа концептуальной модели системной задачи P_Σ проведено ранжирование ее компонентов по шкале неопределенности. Осуществлена стратификация компонентов КМСЗ и сформирована иерархическая структурная модель. Для

раскрытия неопределенности системной задачи P_Σ предложен метод порождения концептуальных моделей вспомогательных задач, в основу которого положен рекурсивный механизм. Данный метод дает возможность на основе рекурсии, в определенном смысле, структурировать и формализовать процесс раскрытия неопределенности пользовательских задач. Системная организация задачи, с одной стороны, и ее структуризация на концептуальном уровне, с другой, предоставляют возможность целенаправленно привлекать к решению такого рода задач экспертов не только по предметным областям, в рамках которых возникают задачи, но и по каждой из вспомогательных задач (КМВЗ) в отдельности. Иными словами, задачная технология на основе КМСЗ и порождаемых КМВЗ дает возможность целенаправленно формировать проблемно-ориентированные коллективные решатели (ПОКР) сложных системных задач. Практическая реализация таких ПОКР может иметь различные формы: коллективы специалистов, универсальные и проблемно-ориентированные инструментальные средства, смешанные человеко-машинные комплексы, т. е. появляется возможность в определенной степени автоматизировать процессы формирования, постановки и решения системных задач.

Список литературы

1. Нечаев В. В. Концептуальное модельное представление задачи как системы // Информационные технологии. 2009. № 9. С. 26—32.
2. Спиридонов В. Ф. Психология мышления: Решение задач и проблем: учеб. пособие. М.: Генезис, 2006. 319 с.
3. Клир Дж. Системология. Автоматизация решения системных задач: Пер. с англ. М.: Радио и связь, 1990. 544 с.
4. Нечаев В. В. Метамоделное представление рекурсивного решателя творческих задач // Междунар. научн. конф. "Технологии и системы сбора, обработки и представления информации". Тезисы докл. Рязань: Русское слово, 1993.
5. Нечаев В. В. Интеллектуальные системы моделирования как методология и инструмент решения творческих задач // Междунар. конф. по проблемам моделирования в бионике "Биомод — 92". СПб.: Информсервис, 1992. С. 359—360.
6. Нечаев В. В. Концептуальное метамоделное представление творческих задач // Методы анализа и оптимизации сложных систем. М.: Изд. ИФТП, 1993. С. 118—124.
7. Нечаев В. В. Творческая задача и интеллектуальный решатель // Модели и системы представления знаний. Межвуз. сб. научн. тр. М.: МИРЭА, 1990. С. 53—68.

М. Ф. Волобуев, канд. техн. наук, нач. лаб.,
В. А. Демчук, нач. фак-та,
М. А. Замыслов, канд. техн. наук, ст. науч. сотр.,
А. М. Мальцев, канд. техн. наук,
С. Б. Михайленко, канд. техн. наук, ст. науч. сотр.,
 Военный авиационный
 инженерный университет, г. Воронеж,
 e-mail: mihzam@rambler.ru

Управление резервированной с помощью мажоритарных элементов системой с применением методов нечеткой логики

Предложен подход к построению алгоритмов автоматизированного распознавания отказов резервированной с помощью мажоритарных элементов системы и управления ее реконфигурацией, направленной на восстановление работоспособности, основанных на представлении выходных параметров элементов резервированной системы в лингвистической форме и реализации стратегии управления с использованием нечеткой логики. Результаты работы алгоритмов представляются в виде вектора предпочтительности альтернативных решений, формируемых с помощью процедур нечеткого вывода, реализованных в программной среде FuzzyTECH.

Ключевые слова: управление резервированной системой, мажоритарные элементы, нечеткое множество, нечеткая логика, функция принадлежности, фаззификация, нечеткий вывод, дефаззификация, нечеткое моделирование

Системы автоматического управления летательными аппаратами (ЛА), как пилотируемыми, так и беспилотными, состоящие из радиотехнических и нерadiотехнических средств различного функционального назначения (автопилотов, средств радиоуправления, инерциальной навигации и др.), составляют основу современных пилотажно-навигационных комплексов (ПНК) ЛА. Одним из наиболее перспективных путей повышения эффективности функционирования и надежности составных частей таких ПНК является резервирование их наиболее важных элементов. При этом для эффективного управления работой таких резервированных систем (РС) требуются как информация об отказах их элементов, получаемая в реальном масштабе времени, так и соответствующее изменение структуры систем (реконфигурация), направленное на сохранение их работоспособности. Поэтому управление конфигурацией резервированных систем, направленное на восстановление их работоспособности при отказах элементов, является одной из важных задач подсистем управления, контроля и восстановления работоспособности в системах автоматического управления и ПНК любых типов летательных аппаратов.

1. Особенности построения устройства управления резервированной с помощью мажоритарных элементов системой

Обнаружение отказов и формирование команд управления резервированной системой, направленных на восстановление ее работоспособности, возможны на основе применения элементов как аналоговой, так и цифровой техники, а также и при их совместном использовании.

В бортовых системах автоматического управления ЛА достаточно часто применяют аналого-цифровые устройства управления резервированными с помощью мажоритарных элементов системами, основанные на принципе мажоритарной логики (принципе выбора по большинству), например, в автоматизированной бортовой системе управления АБСУ-145М [2]. Этот принцип широко используют для обнаружения отказов в системах резервирования, состоящих из рабочего элемента и нескольких элементов сравнения, играющих роль резервных элементов, путем выборки из множества. В этом случае состояние рабочей системы сравнивается с состоянием ряда таких же систем. Признаком безотказности рабочей системы является совпадение ее состояния с состоянием большинства систем [1]. Считается, что состояния рабочей и i -й систем сравнения одинаковы, если разность их выходных параметров меньше ε :

$$|P - P_i| = |\dot{P} - \dot{P}_i| = |Z_i| < \varepsilon,$$

где ε — произвольная положительная величина (абсолютное значение порога сравнения).

Для конкретности и простоты изложения ниже будем рассматривать резервированную с помощью мажоритарных элементов систему минимальной кратности, в которой еще может быть реализовано мажоритарное резервирование. Система состоит из одного рабочего и двух резервных элементов (элементов сравнения). Устройство управления такой резервированной системой содержит [1] рабочий элемент (РЭ), два элемента сравнения (ЭС), три вычитающих устройства (ВУ), три компаратора (К), три электронных ключа (ЭК) и блок управления (БУ), причем входы РЭ и двух ЭС соединены вместе и являются входом системы, а выходы ЭК также соединены вместе и являются выходом системы. Вычитающие устройства, компараторы, блок управления и электронные ключи представляют собой мажоритарный элемент. Вычитающие устройства и компараторы сравнивают выходные параметры различных пар РЭ—ЭС и определяют те пары, выходные параметры которых одинаковы (разность выходных параметров не выше заданного значения ε). Блок управления и электронные ключи выбирают из этих пар один из элементов и подключают его сигнал к выходу системы. При отказе одного элемента системы (РЭ или ЭС) сигналы на выходе двух связанных с ним компараторов будут отсутствовать,

а в блок управления поступит сигнал с компаратора, не связанного с отказавшим элементом. В соответствии с этим сигналом БУ сформирует сигнал управления на один из электронных ключей для подключения одного из исправных элементов к нагрузке. Отказ системы наступает при выходе из строя каких-либо двух элементов (РЭ или ЭС). Структурная схема устройства управления такой резервированной системой приведена на рис. 1. Такое устройство управления резервированной системой обеспечивает сигнал на выходе системы при совпадении сигналов как минимум двух из трех резервируемых элементов.

Основным недостатком приведенного устройства управления резервированной системой является невозможность подключения к нагрузке исправного элемента в случае отказа двух других элементов, что не позволяет в полной мере реализовать принцип резервирования замещением и снижает надежность системы. Полного отказа резервированной системы можно было бы избежать при наличии на борту ЛА автоматического устройства, которое распознавало бы возникновение отказов резервированной системы и в реальном масштабе времени проводило бы реконфигурацию системы, т. е. переход с отказавшего элемента на другой, работоспособный. Поэтому актуальной является задача совершенствования автоматизированных процедур распознавания фактов возникновения отказов резервированной системы, т. е. такого изменения основных выходных параметров (ОВП) системы, которое может привести к потере ее работоспособности.

Указанную задачу будем решать с использованием методического подхода, предложенного в работах [3–5, 11], с применением формализованных в бортовом процессоре методик и средств, позволяющих на количественном уровне объективных параметров контролировать работоспособность резервированной системы, не допуская опасности возникновения ее отказа вследствие выхода из строя

каких-либо элементов. При этом в первую очередь необходимо решить задачу автоматизированного распознавания фактов возникновения отказов системы (ОС). Для идентификации ОС требуется представление ситуации в виде множества количественных параметров, позволяющих адекватно ее оценить, т. е. отнести к заранее описанному классу ситуаций, при котором тактика устранения ОС известна. При решении этой задачи необходимо также учитывать возможный диапазон изменения выходных параметров резервируемых элементов при различных режимах их работы, что должно обеспечить достаточно высокую вероятность обнаружения отказа. Кроме того, на эффективность работы резервированной системы оказывает негативное влияние ряд неопределенных факторов, к которым относятся:

- влияние случайных возмущений (вибрации и тряски, флуктуации параметров внешней среды — температуры, влажности, давления, нестабильности питающих напряжений и параметров аналоговых и цифровых элементов и др.);
- влияние систематических и случайных ошибок измерения значений выходных параметров элементов на формирование команд управления работой резервированной системы;
- неполная достоверная информация о работоспособности отдельных элементов системы управления резервированной системой, базирующаяся на сигналах, контролирующих этот процесс на выходе информационных датчиков.

В этих условиях для построения системы автоматизированного распознавания отказов резервированной с помощью мажоритарных элементов системы и управления ее реконфигурацией с учетом нечеткости информации целесообразно использовать подход, основанный на методах теории нечетких множеств и нечеткой логики. Будем считать, что в качестве управляющих характеристик используются отклонения основных выходных параметров резервируемых элементов от номинальных значений, характерных для соответствующих режимов работы элементов, поддерживаемых их системами стабилизации (демпферами, автоматами устойчивости и др.). Использование этих параметров обусловлено тем, что они оказывают существенное влияние на устойчивость функционирования резервированной системы при отказах как одного, так и двух любых элементов при различных режимах их работы.

2. Построение алгоритмов нечеткого управления устройством автоматизированного распознавания отказов в резервированной с помощью мажоритарных элементов системе и восстановления ее работоспособности

Ввиду ограниченного объема статьи рассмотрим только один из возможных режимов работы резервированной системы, например нагруженный резерв при номинальных условиях работы всех элементов в процессе установившегося движения носителя.

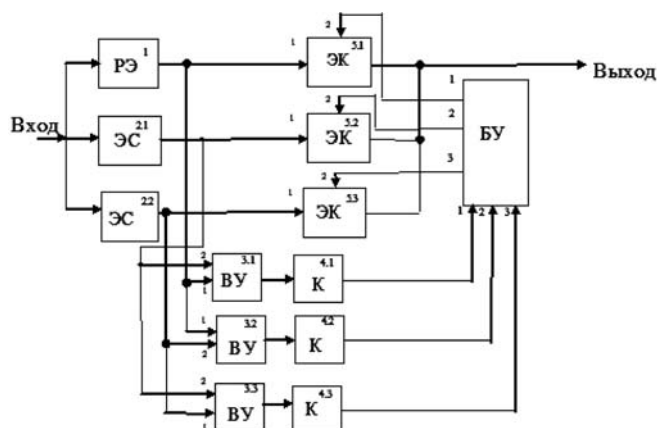


Рис. 1. Устройство управления резервированной с помощью мажоритарных элементов системой: РЭ — рабочий элемент; ЭК — электронный ключ; ЭС — элемент сравнения (резервный элемент); БУ — блок управления; ВУ — вычитающее устройство; К — компаратор (элемент сравнения)

Предположим, что в множество ОВП элементов РС включены значения фактических выходных параметров элементов (P_i) различных элементов резервированной системы. Поэтому в качестве управляющих характеристик будем использовать отклонения этих параметров от значений, задаваемых соответствующими порогами по фактическим значениям $P(t) - \Delta$, где Δ — допустимый предел отклонения выходного параметра P элемента (РЭ или ЭС) от математического ожидания m_p , при котором считается, что элемент работает безотказно.

Для перехода от числовых значений "четких" отклонений этих характеристик к их "нечеткой" интерпретации выполним операцию фаззификации [3]. Для этого возможные отклонения управляющих характеристик от нормы будем рассматривать как лингвистические переменные (ЛП), представляющие собой упорядоченную совокупность [7]

$$\langle A, T = [\langle \beta, \gamma, M = \langle \mu_i(x)/x \rangle], X \rangle, \quad (1)$$

где A — имя ЛП; T — терм-множество (ТМ) (множество значений) ЛП; β — имя нечеткой переменной; γ — базовая шкала нечеткой переменной; M — функция принадлежности (ФП) нечеткой переменной; X — базовая шкала лингвистической переменной; μ_i — значение ФП.

Функция принадлежности (1) задает степень соответствия текущего отклонения характеристик к каждому элементу терм-мноества ЛП и представляет собой непрерывную функцию, принимающую множество значений в диапазоне от нуля до единицы (0 — текущее значение полностью не соответствует данному терму; 1 — полностью соответствует).

Будем представлять ФП терма в виде нечетких чисел LR -типа, задаваемых двумя упорядоченными четверками [5]:

$$LR = (\beta, c, \alpha, \gamma) \text{ и } LR1 = (\alpha, \gamma, \beta, c).$$

Для четверки LR β, c — точки, в которых значение ФП равно единице; $\alpha(\gamma)$ — координаты ближайшей слева (справа) к $x = \beta(c)$ точки базовой шкалы, в которой ФП принимает нулевое значение. Четверка $LR1$ представляет собой перевернутое изображение LR . Поэтому для $LR1$ α и γ — точки, в которых ФП равна единице, а интервал $[\beta, c]$ соответствует нулевым значениям ФП. Вычисление значений ФП терма, представленной в виде нечеткого числа LR -типа, заданного упорядоченной четверкой, в точке $x = \xi$ выполняется по формуле [5]

$$\mu(\xi) = \begin{cases} 0, & \xi \leq \alpha, \\ (\xi - \alpha)/(\beta - \alpha), & \alpha \leq \xi \leq \beta, \\ 1, & \beta \leq \xi \leq c, \\ (\gamma - \xi)/(\gamma - c), & c \leq \xi \leq \gamma, \\ 0, & \xi \geq \gamma, \end{cases}$$

а вычисление значений ФП терма в виде нечеткого числа LR -типа, заданного упорядоченной четверкой $LR1$, в точке $x = \xi$ — по формуле

$$\mu1(\xi) = \begin{cases} 1, & \xi \leq \alpha, \\ (\beta - \xi)/(\beta - \alpha), & \alpha \leq \xi \leq \beta, \\ 0, & \beta \leq \xi \leq c, \\ (\xi - c)/(\gamma - c), & c \leq \xi \leq \gamma, \\ 1, & \xi \geq \gamma. \end{cases}$$

При этом функции принадлежности $\mu(\xi)$ и $\mu1(\xi)$ порождают нормальные выпуклые нечеткие множества с носителем — интервалом (α, γ) , границами $(\alpha, \beta)U(c, \gamma)$, ядром $[\beta, c]$.

Применительно к рассматриваемой задаче будем использовать более сложную ФП, состоящую из совокупности перекрывающихся LR чисел [5], рассмотренных выше. При этом каждая из лингвистических переменных может принимать одно из следующих значений множества:

$$L = \{NB, Z, PB\}. \quad (2)$$

В выражении (2) первая буква в обозначении указывает знак (сторону отклонения) числовой переменной (N — отрицательное, P — положительное), вторая буква говорит об абсолютном значении переменной (B (big) — большое), а символ Z соответствует попаданию переменной в допустимый предел Δ отклонения выходного параметра элемента от математического ожидания m_x , при котором считается, что элемент работает безотказно. Значения допустимых пределов изменения параметров элементов вверх и вниз должны выбираться соответственно равными $U_b = (m_x + 3\sigma_x)$ и $U_n = (m_x - 3\sigma_x)$, где σ_x — среднее квадратическое отклонение (СКО) выходного сигнала элемента, соответственно.

С использованием (2) могут быть определены функции принадлежности для каждой из лингвистических переменных. Вид такой функции принадлежности качественных оценок отклонений значений характеристик ОВП от нормы показан на рис. 2. Вычисление значений ФП, показанной на рис. 2, проводится по формулам, приведенным выше.

Это обусловлено тем, что при принятом виде функции принадлежности (см. рис. 2) каждому числовому значению отклонения характеристики ОВП соответствует (с различной степенью принадлежности) два нечетких значения. Так, например, отклонение выходного параметра РЭ на -400 мВ может являться отрицательным малым (NS) отклонением, близким к нормальному, со степенью принадлежности 0,8 и отрицательным средним (NM) со степенью 0,2.

Введем также качественные оценки и для принимаемых решений на выходе системы. Для этого

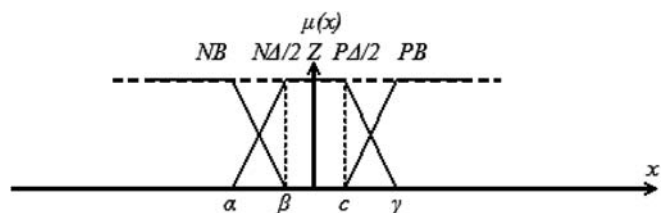


Рис. 2. Вид функции принадлежности качественных оценок отклонений значений характеристик ОВП от нормы для отдельной ЛП

сформируем множество решений, которые должны приниматься в процессе функционирования резервированной с помощью мажоритарных элементов системы. К такому множеству, соответствующему рассмотренным выше особенностям построения устройства управления резервированной с помощью мажоритарных элементов системой, относятся следующие восемь принимаемых решений: I — продолжение работы резервированной системы в исходном режиме (все элементы исправны, к нагрузке подключен РЭ); II — изменение конфигурации резервированной системы путем подключения к нагрузке ЭС1 при отказе РЭ; III — изменение конфигурации резервированной системы путем подключения к нагрузке РЭ при отказе ЭС1; IV — изменение конфигурации резервированной системы путем подключения к нагрузке РЭ при отказе ЭС2; V — изменение конфигурации резервированной системы при выходе из строя мажоритарного элемента, вызванного отказом РЭ и ЭС1, путем подключения к нагрузке ЭС2; VI — изменение конфигурации резервированной системы при выходе из строя мажоритарного элемента, вызванного отказом РЭ и ЭС2, путем подключения к нагрузке ЭС1; VII — изменение конфигурации резервированной системы при выходе из строя мажоритарного элемента, вызванного отказом ЭС1 и ЭС2, путем подключения к нагрузке РЭ; VIII — выход резервированной системы из строя вследствие отказа всех элементов (РЭ, ЭС1 и ЭС2).

Для того чтобы различать такие понятия в степени уверенности решений, как "малая уверенность" (S), "достаточная уверенность" (M) и "сильная уверенность" (B), представим для каждого из восьми принимаемых решений перечисленные понятия в виде трех термов, показанных на рис. 3.

При этом каждая из ЛП (I, II, III, IV, V, VI, VII, VIII) может принимать одно из значений множества $D = \{S, M, B\}$.

С учетом введенных выше понятий и терм-множеств под нечетким управлением будем понимать процедуры принятия решений, представленные в виде некоторой совокупности правил, основанных на логико-формальном представлении знаний о функционировании системы (в лингвистической форме) и преобразовании их с применением методов нечеткой логики. Эти правила описывают принятие решений по управлению резервированной системой и формируются квалифицированными экспер-

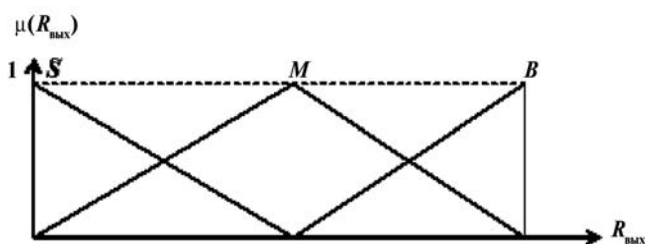


Рис. 3. Вид функций принадлежности качественных оценок уверенности в выбранных решениях по управлению резервированной системой

тами в области логического резервирования систем. Например, подобным правилом, соответствующим принимаемому решению I, и записанным на естественном языке для рассматриваемой резервированной системы, может быть следующее высказывание:

"Если отклонение фактического значения выходного параметра РЭ соответствует незначительным отклонениям в положительную сторону, близким к норме, отклонение выходного параметра ЭС1 — незначительное и отрицательного знака, а отклонение выходного параметра ЭС2 — незначительное, близкое к норме, и положительного знака, то резервированная система должна работать в прежнем режиме (к нагрузке остается подключенным РЭ).

С использованием логико-формальных методов и введенных выше терм-множеств входных и выходных лингвистических переменных это правило в соответствии с упрощенным алгоритмом нечеткого вывода [5] записывается в следующем виде:

$$((P_{RE} = Z) \wedge (P_{ES1} = Z) \wedge (P_{ES2} = Z)) \mapsto \mapsto ((I = B) \wedge (II = S) \wedge (III = S) \wedge (IV = S) \wedge (V = S) \wedge (VI = S) \wedge (VII = S) \wedge (VIII = S)), \quad (3)$$

где $\wedge$ — знак логической операции "и"; $\mapsto$ — знак логической операции "секвенция", обозначающий, что высказывание, стоящее в правой части выражения (13), "истинно" тогда и только тогда, когда "истинно" высказывание в его левой части.

Аналогичным образом с использованием терм-множеств входных и выходных лингвистических переменных и упрощенного алгоритма нечеткого вывода [5] записываются соответствующие логико-формальные правила (правила нечеткого вывода) для принятия решений II—VIII:

$$((P_{RE} = PB \vee NB) \wedge (P_{ES1} = Z) \wedge (P_{ES2} = Z)) \mapsto \mapsto ((I = S) \wedge (II = B) \wedge (III = S) \wedge (IV = S) \wedge (V = S) \wedge (VI = M) \wedge (VII = S) \wedge (VIII = S)); \quad (4)$$

$$((P_{RE} = Z) \wedge (P_{ES1} = PB \vee NB) \wedge (P_{ES2} = Z)) \mapsto \mapsto ((I = S) \wedge (II = S) \wedge (III = B) \wedge (IV = S) \wedge (V = S) \wedge (VI = S) \wedge (VII = M) \wedge (VIII = S)); \quad (5)$$

$$((P_{RE} = Z) \wedge (P_{ES1} = Z) \wedge (P_{ES2} = PB \vee NB)) \mapsto \mapsto ((I = S) \wedge (II = S) \wedge (III = S) \wedge (IV = B) \wedge (V = S) \wedge (VI = S) \wedge (VII = M) \wedge (VIII = S)); \quad (6)$$

$$((P_{RE} = PB \vee NB) \wedge (P_{ES1} = PB \vee NB) \wedge (P_{ES2} = Z)) \mapsto \mapsto ((I = S) \wedge (II = M) \wedge (III = M) \wedge (IV = S) \wedge (V = B) \wedge (VI = S) \wedge (VII = S) \wedge (VIII = S)); \quad (7)$$

$$((P_{RE} = PB \vee NB) \wedge (P_{ES1} = Z) \wedge (P_{ES2} = PB \vee NB)) \mapsto \mapsto ((I = S) \wedge (II = M) \wedge (III = S) \wedge (IV = M) \wedge (V = S) \wedge (VI = B) \wedge (VII = S) \wedge (VIII = S)); \quad (8)$$

$$((P_{RE} = Z) \wedge (P_{ES1} = PB \vee NB) \wedge (P_{ES2} = PB \vee NB)) \mapsto \mapsto ((I = S) \wedge (II = S) \wedge (III = M) \wedge (IV = M) \wedge (V = S) \wedge (VI = S) \wedge (VII = B) \wedge (VIII = S)); \quad (9)$$

$$((P_{RE} = PB \vee NB) \wedge (P_{ES1} = PB \vee NB) \wedge (P_{ES2} = PB \vee NB)) \mapsto \mapsto ((I = S) \wedge (II = S) \wedge (III = S) \wedge (IV = S) \wedge (V = S) \wedge (VI = M) \wedge (VII = S) \wedge (VIII = B)). \quad (10)$$

В общем случае ситуация, описываемая выражениями (3–10), требует рассмотрения для каждого выражения 2^3 правил, что обусловлено, как отмечалось выше, видом принятой функции принадлежности входных ЛП. Для нечетких подмножеств в (3)–(10) степень выполнения условия "Если" будет определять степень применения данного правила. Поэтому степень уверенности в выборе того или иного решения будет определяться весом, соответствующим весу функции принадлежности пересечения нечетких подмножеств NB , Z , PB в условии "Если". Как известно [8], функция принадлежности пересечения нечетких подмножеств A и B определяется следующим образом:

$$A, B \in E, \forall x \in E: \mu_{A \cap B} = \min(\mu_A(x), \mu_B(x)). \quad (11)$$

Допустим, что функции принадлежности приняли следующие значения:

$$\mu_Z(P_{RE}) = 0,8, \mu_Z(P_{ES1}) = 0,9, \mu_{PB}(P_{ES2}) = 0,6,$$

тогда в соответствии с (11) имеем

$$\begin{aligned} \beta_S(I) &= \beta_S(II) = \beta_S(III) = \beta_B(IV) = \beta_S(V) = \\ &= \beta_S(VI) = \beta_M(VII) = \beta_S(VIII) = \mu_{Z \cap Z \cap PZB}(x) = \\ &= \min(\mu_Z(P_{RE}), \mu_Z(P_{ES1}), \mu_{PB}(P_{ES2})) = 0,6, \end{aligned}$$

где $\mu_A(x)$ — функция принадлежности лингвистической переменной x нечеткому множеству A ; $\beta_D(y)$ — весовой коэффициент, характеризующий степень принадлежности лингвистической переменной y (I, II, ..., V, VI, VII, VIII) нечеткому множеству D .

Действуя в соответствии с описанным выше алгоритмом, мы получим для каждого возможного решения восемь усеченных термов. Окончательное решение задачи находим путем их объединения и дефаззификации, т. е. переходом от нечеткого множества, полученного в результате объединения, к единственному четкому значению степени (иногда трактуемой как вероятность) выполнения определенного решения на выходе системы. Для дефаззификации будем использовать метод определения взвешенного среднего значения [5], позволяющий вычислять выходное значение в реальном масштабе времени, который в матричном виде записывается следующим образом [11]:

$$R_{\text{вых}} = \beta F^T \Delta \beta^{-1}, \quad (12)$$

$$\text{где } \Delta \beta = \sum_{i=1}^n \beta_i, F^* = \{\mu_S(R_{\text{вых}}), \mu_M(R_{\text{вых}}), \mu_B(R_{\text{вых}})\},$$

$$\beta = \begin{bmatrix} \beta_S(I) & \beta_M(I) & \beta_B(I) \\ \beta_S(II) & \beta_M(II) & \beta_B(II) \\ \beta_S(III) & \beta_M(III) & \beta_B(III) \\ \beta_S(IV) & \beta_M(IV) & \beta_B(IV) \\ \beta_S(V) & \beta_M(V) & \beta_B(V) \end{bmatrix},$$

F^* — вектор центральных значений нечетких подмножеств выходной лингвистической переменной (0, 0,5, 1); β_i — веса, учитывающие уровень выполнения условия "Если" i -го правила (уровни активности соответствующих правил); n — число правил вывода, соответственно.

Проиллюстрируем работоспособность алгоритма на следующем примере.

Допустим, что значения основных выходных параметров элементов резервированной системы, в исходном состоянии, будут следующими: $P_{RE} = 1000$ мВ, $P_{ES1} = 1100$ мВ, $P_{ES2} = 900$ мВ, и гипотетическая резервированная система настроена на эти значения. Пусть в некоторый момент времени параметры изменились следующим образом: P_{RE} возросло до 1450 мВ, P_{ES1} увеличилось до 1800 мВ, P_{ES2} уменьшилось до 300 мВ.

Используя эти отклонения от первоначальных значений выходных параметров элементов резервированной системы, принятый вид ФП (см. рис. 1) и аналитические выражения для расчета значений ФП $\mu(\xi)$ в точке $x = \xi$, определим значения входных и выходных ФП лингвистических переменных. Порядок фаззификации, т. е. определения входных ФП ЛП с использованием принятого вида ФП, показан на рис. 4. В результате выполнения этой процедуры фаззификации получим следующие значения ФП лингвистических переменных P_{RE} , P_{ES1} и P_{ES2} : $\mu_Z(P_{RE}) = 0,88$; $\mu_{PB}(P_{RE}) = 0,12$; $\mu_Z(P_{ES1}) = 0,25$; $\mu_{PB}(P_{ES1}) = 0,75$; $\mu_Z(P_{ES2}) = 0,5$; $\mu_{NB}(P_{ES2}) = 0,5$.

В рассматриваемом случае активными становятся восемь правил, в результате выполнения которых степени достоверности решений по управлению резервированной системой получают весовые коэффициенты, приведенные в таблице.

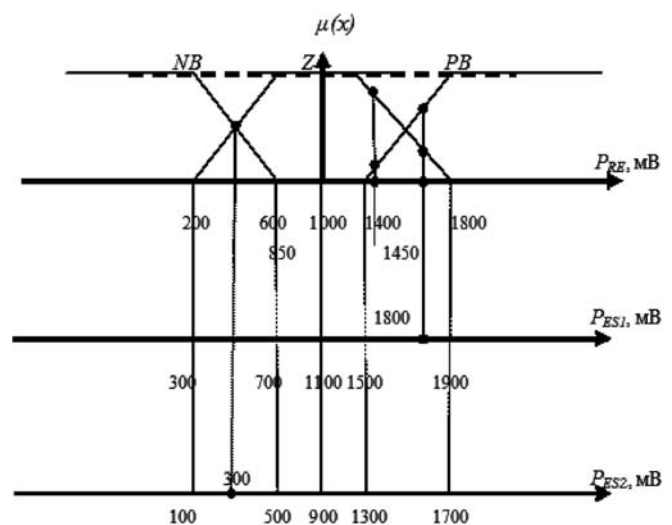


Рис. 4. Определение значений функций принадлежности (фаззификация) для лингвистических переменных


$$\mathbf{\beta}^T = \begin{bmatrix} 1,73 & 1,62 & 0,74 & 0,61 & 1,86 & 1,62 & 0,73 & 1,86 \\ 0 & 0,24 & 0,62 & 1,12 & 0 & 0,24 & 0,75 & 0 \\ 0,25 & 0,12 & 0,62 & 0,25 & 0,12 & 0,12 & 0,5 & 0,12 \end{bmatrix}.$$

$\mathbf{R}_{\text{ВЫХ}} = [0,126 \ 0,121 \ 0,47 \ 0,409 \ 0,061 \ 0,121 \ 0,442 \ 0,061]^T$,
который после нормировки имеет следующий вид:

$$\mathbf{R}_{\text{BbIX}}^{\text{H}} = [0,07 \ 0,067 \ 0,259 \ 0,226 \ 0,033 \ 0,067 \ 0,244 \ 0,033]^{\text{T}}.$$

$$\mathbf{R}_{\text{BbIX}}^* = (R_{\text{BbIX}}^{\text{III}}, R_{\text{BbIX}}^{\text{VII}}, R_{\text{BbIX}}^{\text{IV}}). \quad (13)$$

Структурная схема нечеткого контроллера [10], формирующего вектор упорядоченных по предпочтительности решений по управлению резервированной системой, изображена на рис. 5, а на рис. 6 приведена структурная схема резервированной системы, управляемой нечетким контроллером. Опи-

58

сание нечеткой модели управления резервированной системой в среде FuzzyTECH, разработанной с использованием нотации языка нечеткого управления FTL (Fuzzy Tech Language) [5], приводится ниже.

3. Нечеткая модель управления резервированной с помощью мажоритарных элементов систем, построенная с использованием программы нечеткого моделирования FuzzeTECH 6.00 Professional Demo

Содержательная постановка задачи и один из алгоритмов нечеткого моделирования подробно рассмотрены в разд. 2. Ниже приводится краткое описание нечеткой модели управления резервированной системой, разработанной в среде FuzzeTECH, и результаты ее тестирования (отладки). Все элементы модели обозначены на английском языке, так как интерфейс программы FuzzeTECH не поддерживает русский язык. В качестве входных лингвистических переменных в данной нечеткой модели используются лингвистические переменные "Outputwu" (выходной сигнал рабочего элемента), "Outputru1" (выходной сигнал первого резервного элемента) и "Outputru2" (выходной сигнал второго резервного элемента). В качестве выходных лингвистических переменных данной нечеткой модели используются лингвистические переменные "Position1", "Position2", ..., "Position8" (положение коммутатора элементов резервированной системы).

В качестве терм-множества лингвистических переменных "Outputwu", "Outputru1" и "Outputru2" используются множества T_1 , T_2 и T_3 соответственно, принимающие значения {"neg_big", "zero", "pos_big"}. Графики функций принадлежности термов из $T_1...T_3$ показаны в соответствующих окнах редакторов лингвистических переменных "Outputwu", "Outputru1", "Outputru2" рабочего интерфейса программы FuzzyTECH, изображенного на рис. 7 (см. третью сторону обложки). В качестве терм-множеств выходных лингвистических переменных "Position1", "Position2", ..., "Position8" используются множества T_4 , T_5 , ..., T_{11} соответственно, принимающие значения {"small_confidence", "middle_confidence", "big_confidence"}. Графики функций принадлежности термов из $T_4...T_{11}$ показаны в соответствующих окнах редакторов лингвистических переменных "Position1", "Position2", ..., "Position8" на рис. 7.

База правил в модели построена таким же образом, как в разд. 2, но их количество и форма записи в среде FuzzyTECH приведены в соответствие с лексикой языка FTL. Поскольку в модели во всех правилах в качестве логической связи для подусловий применяется только нечеткая конъюнкция (операция "И"), то в качестве метода агрегирования используется операция min-конъюнкции [5]. Так как в качестве схемы нечеткого вывода используется метод Мамдани, то методом активации является MIN, который рассчитывается по формуле

$$\mu(y) = \min\{c_j, \mu(y)\},$$

где c_j — значение степени истинности подзаключений для j -го правила, входящего в рассматриваемую базу правил нечеткого вывода; $\mu(y)$ — функция принадлежности терма, который является значением некоторой выходной переменной ω_j , заданной на универсуме Y .

Для аккумуляции заключений правил используется метод max-дизъюнкции, который также применяется в случае схемы нечеткого вывода методом Мамдани, а в качестве метода дефазификации — метод центра тяжести для одноэлементных множеств [5]:

$$y = \frac{\sum_{i=1}^n x_i \mu(x_i)}{\sum_{i=1}^n \mu(x_i)},$$

где n — число одноэлементных нечетких множеств, каждое из которых характеризует единственное значение рассматриваемой выходной лингвистической переменной; x_i — переменная, соответствующая выходной лингвистической переменной ω ; $\mu(x_i)$ — функция принадлежности нечеткого множества, соответствующего выходной переменной ω после этапа аккумуляции.

Рассмотрим пример выполнения нечеткого вывода для модели, разработанной в среде FuzzyTECH, при тех же исходных данных, которые использовались в примере разд. 2. Для этого выполним анализ данной системы нечеткого вывода при решении задачи управления резервированной системой. С этой целью используем интерактивный режим отладки (Interactive Debug Mode), окно которого показано на рис. 7. В этом окне приведены значения входных данных, соответствующих примеру разд. 2, и выходных данных нечеткой модели, представляющих собой степени уверенности (взвешенное среднее) $R_{\text{вых}}$ для различных решений:

$$R_{\text{вых}} = [0,3333 \ 0,2499 \ 0,6429 \ 0,3333 \ 0,1999 \ 0,0999 \ 0,4001 \ 0,1999]^T.$$

С учетом полученных значений $R_{\text{вых}}$ ранжированный ряд состоятельных решений имеет следующий вид:

$$R_{\text{вых}}^* = (R_{\text{вых}}^{\text{III}}, R_{\text{вых}}^{\text{VII}}, (R_{\text{вых}}^{\text{IV}}, R_{\text{вых}}^{\text{I}})). \quad (14)$$

Сравнение выражений (13) и (14) позволяет сделать следующие выводы.

1. Решения, принимаемые по первому и второму элементам ранжированных рядов, полученных на различных нечетких моделях, совпадают.

2. В выражении (14) третий по предпочтительности элемент включает в себя два эквивалентных решения $R_{\text{вых}}^{\text{IV}}$ и $R_{\text{вых}}^{\text{I}}$, а в выражении (13) — только решение $R_{\text{вых}}^{\text{IV}}$, однако к ошибкам это не приводит, так как в результате выполнения этих решений

($R_{\text{вых}}^{\text{IV}}$ или $R_{\text{вых}}^{\text{I}}$) к выходу системы будет подключаться один и тот же рабочий элемент (РЭ).

3. Различия в численных значениях элементов векторов $\mathbf{R}_{\text{вых}}$, полученных в примере разд. 2 и на модели, реализованной в среде FuzzyTECH, обусловлены применением различных алгоритмов нечеткого вывода. В модели разд. 2 используется упрощенный алгоритм нечеткого вывода [5], а в программе FuzzyTECH — алгоритм Мамдани.

Для общего анализа рассматриваемой нечеткой модели войдем в меню анализатора правил ("Analyzer") и откроем графическое окно просмотра трехмерной поверхности нечеткого вывода (окно "3D Plot — 1" на рис. 7). Данная поверхность характеризует зависимость значений выходных переменных от значений отдельных входных переменных нечеткой модели. Отсутствие разрывов на поверхности нечеткого вывода свидетельствует об устойчивом характере данной нечеткой модели, что является одним из убедительных аргументов в пользу применения этой модели в практике принятия решений по управлению резервированной системой. Зависимости, описываемые трехмерными поверхностями нечеткого вывода, могут служить основой для программирования контроллера или аппаратной реализации нечеткого алгоритма управления в форме соответствующей таблицы решений. При этом рассматриваемая программа, написанная на языке FTL, может использоваться в некоторых внешних программах как процедура, формальными параметрами которой являются входные переменные блока правил, а возвращаемыми значениями — дефаззинфицированные значения выходных переменных. Порядок организации вызова таких процедур во внешних программах рассмотрен в работе [5].

Решения по управлению резервированной системой с алгоритмом нечеткого управления (решения по реконфигурации системы) должны выполняться в реальном масштабе времени в автоматическом режиме и в последовательности, задаваемой векторами (13) или (14). Причем в первую очередь должны реализоваться решения, имеющие больший вес (ранг), и выполняться соответствующая этим решениям реконфигурация системы. При этом взвешенное среднее различных решений (13), (14), т. е. выходные коэффициенты уверенности, полученные с применением методов нечеткой логики, позволяют снизить до минимума риск выбора окончательного решения.

Сравним надежность резервированных систем, структурные схемы которых приведены на рис. 1 и 6. Первая из этих систем функционирует на основе мажоритарной логики, а во второй системе реализован алгоритм нечеткого управления, построенный на основе методов нечеткой логики. Как видно из изложенного выше, отказ системы с мажоритарным резервированием происходит при выходе из строя любых двух элементов, а резервированной системы с нечетким управлением — только при от-

казе всех элементов. Поэтому расчеты вероятности безотказной работы этих систем проводили по следующим формулам [1], полученным в предположении, что все элементы равнонадежны: для РС с мажоритарной логикой

$$P_6 = 3p^2 - 2p^3, \quad (15)$$

для РС с нечетким управлением

$$P_6 = 1 - (1 - p)^3, \quad (16)$$

где p — вероятность безотказной работы элемента.

Анализ результатов расчетов показывает, что применение алгоритмов нечеткого управления резервированной системой позволяет в 1,73...1,14 раза повысить вероятность безотказной работы резервированной системы по сравнению с типовым алгоритмом мажоритарного резервирования при значениях вероятностей безотказной работы элементов, равных соответственно 0,5...0,8, за счет обнаружения и подключения к нагрузке оставшегося работоспособным элемента мажоритарного органа. Предложенный алгоритм может быть использован не только для восстановления работоспособности резервированной с помощью мажоритарных элементов системы, но и для выявления всех отказавших элементов. При этом следует ожидать сокращения времени восстановления работоспособности отказавшего элемента за счет снижения временных затрат на поиск неисправностей.

Список литературы

1. Шевцов Г. А., Шерemet Е. М. Логическое резервирование. — Львов: Изд-во Львовского университета, 1973. С. 43, 44.
2. Головченко В. В., Ерошин С. А., Миллер Ю. В. Автоматизированная бортовая система управления АБСУ-145М 3-я серия: учеб. пособие. Иркутск: Изд. ИВВАИУ, 1997.
3. Лебедев Г. Н., Алисултанов Ш. М. Нечеткое управление геолого-разведочным полетом ДПЛА в сложных погодных условиях // Информационные технологии. 2000. № 8. С. 40—42.
4. Замыслов М. А., Михайленко С. Б., Замыслов Е. М., Замыслов А. М. Применение методов теории нечетких множеств и отношений для решения задачи выбора потребителем наиболее предпочтительных поставщиков по совокупности показателей // Информационные технологии. 2002. № 5.
5. Леоненков А. В. Нечеткое моделирование в среде MATLAB и FuzzyTECH. — СПб.: БХВ-Петербург, 2005. 736 с.
6. Мистров Л. Н., Замыслов М. А., Михайленко С. Б. Эквивалентность методов выбора альтернативных вариантов группового ранжирования на основе аппарата теории нечетких множеств // Техника машиностроения. 2010. Вып. 4 (76).
7. Розен В. В. Цель—оптимальность—решение. — М.: Радио и связь, 1982. 168 с.
8. Кофман А. Введение в теорию нечетких множеств. М.: Радио и связь, 1982.
9. Заде Л. А. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Наука, 1982.
10. Procyk T. J., Mamdani E. H. A Linguistic Self-Organizing Process Controller // Automatica. 1979. Vol. 15. P. 15—30.
11. Волобуев М. Ф., Демчук В. А., Замыслов М. А., Михайленко С. Б. Применение методов нечеткой логики для автоматизированного распознавания начальных стадий возникновения летных ситуаций, опасных для пилотирования летательного аппарата // Материалы XII Международной научно-технической конференции "Кибернетика и высокие технологии XXI века". Воронеж, 2011. Т. 2. С. 531—539.

УДК 519.873:519.218.4:004.056.53

А. И. Перегуда, д-р техн. наук, проф.,
Обнинский институт атомной энергетики —
филиал федерального государственного
автономного образовательного учреждения
высшего профессионального образования

"Национальный исследовательский
ядерный университет "МИФИ",
e-mail: Pereguda@iate.Obninck.ru,

Д. А. Тимашов, канд. техн. наук, разработчик,
ЗАО "Ай-Текс",
e-mail: i18n@yandex.ru

Математическая модель надежности информационных систем с эшелонированной защитой от несанкционированного доступа

Рассмотрена математическая модель надежности информационной системы систем защиты информации, состоящей из объекта защиты информации и двух систем безопасности. Процесс функционирования изучаемой системы защиты информации описывается как наложение альтернирующих процессов восстановления. Получены верхняя и нижняя оценки для математического ожидания времени до осуществления несанкционированного доступа к информации.

Ключевые слова: надежность, система безопасности, случайные величины, наработка до отказа, случайный процесс, математическое ожидание времени, функция распределения

Введение

При построении высоконадежных систем защиты информации нередко используется эшелонированная защита от угроз [1, 2]. Для обеспечения защиты информации от несанкционированного доступа требуется обеспечить определенный уровень надежности таких систем, включая надежность их аппаратных средств и программного обеспечения. Математические модели надежности систем защиты информации рассматривались в различных работах, например в [3]. При анализе надежности таких систем недостаточно просто учитывать структуру системы безопасности, необходимо также принять во внимание резерв времени, имеющийся у систем безопасности, который возникает за счет того, что на преодоление последовательных уровней

защиты злоумышленнику потребуется некоторое время. Здесь мы получим соотношения, позволяющие оценивать, насколько данный запас времени повышает надежность системы защиты информации.

Описание модели

Пусть информационная система состоит из объекта защиты и двух систем безопасности, каждая из которых парирует одну из разновидностей атак на объект защиты. Системы безопасности не являются абсолютно надежными и могут отказывать. Предполагается, что отказы систем безопасности являются независимыми. Обнаружение отказа системы безопасности происходит только во время периодического контроля ее состояния. После обнаружения отказа системы безопасности выполняется ее полное восстановление к исходному состоянию. Если первая система безопасности была не способна немедленно париовать атаку, то несанкционированный доступ к информации может быть осуществлен только спустя некоторое время и лишь при условии, что за это время ни одна из систем безопасности не сработала.

Обозначим χ_1 случайное время до первой атаки на объект защиты, обнаруживаемой первой системой безопасности. Случайное время отражения первой атаки на объект защиты, обнаруженной первой системой безопасности, обозначим γ_1 . Тогда случайное время до второй атаки на объект защиты после отражения первой атаки, обнаруживаемой первой системой безопасности объекта защиты, обозначим χ_2 , после отражения второй атаки — χ_3 и т. д. Будем считать, что все χ_i , $i = 1, 2, \dots$, как обычно, независимы и одинаково распределены с функцией распределения $F_\chi(t) = P(\chi \leq t)$. Случайное время отражения i -й атаки на объект защиты, обнаруженной первой системой безопасности, будем обозначать γ_i , $i = 1, 2, \dots$, причем γ_i — независимые одинаково распределенные случайные величины с функцией распределения

$$F_\gamma(t) = P(\gamma \leq t).$$

Обозначим δ_i случайное время между i -й атакой на объект защиты, которая не была обнаружена первой системой безопасности, и осуществлением несанкционированного доступа. Очевидно, что δ_i , $i = 1, 2, \dots$, — независимые одинаково распределенные случайные величины с функцией распределения $F_\delta(t) = P\{\delta \leq t\}$. Таким образом, первая система безопасности все еще может предотвратить несанкционированный доступ в интервале времени $[\chi_i, \chi_i + \delta_i]$.

Поскольку процесс функционирования первой системы безопасности состоит из циклов "атака — отражение атаки", следовательно, моменты завершения отражения атаки на объект защиты, обнаруженной первой системой безопасности, являются моментами регенерации информационной системы.

Если же первая система безопасности так и не сработала в указанном интервале времени, то вторая система безопасности все еще может предотвратить несанкционированный доступ в момент $\chi_i + \delta_i$. Продолжительности отражения атаки второй системой безопасности после первой атаки обозначим α_i , после второй атаки — α_2 и т. д. Считаем, что α_i , $i = 1, 2, \dots$, — независимые, одинаково распределенные случайные величины с функцией распределения $F_{\alpha}(t) = P\{\alpha \leq t\}$.

Случайное время до i -й атаки на объект защиты, обнаруживаемой второй системой безопасности, обозначим φ_i , $i = 1, 2, \dots$. Пусть φ_i , $i = 1, 2, \dots$, — независимые одинаково распределенные случайные величины с функцией распределения $F_{\varphi}(t) = P\{\varphi \leq t\}$. Продолжительность отражения i -й атаки на объект защиты, которая была обнаружена второй системой безопасности, обозначим ψ_i , $i = 1, 2, \dots$. Считаем, что ψ_i , $i = 1, 2, \dots$, — независимые, одинаково распределенные случайные величины с функцией распределения $F_{\psi}(t) = P\{\psi \leq t\}$. Предполагаем, что моменты окончания отражения таких атак являются моментами регенерации процесса функционирования информационной системы. Кроме того, пусть математические ожидания χ_i , γ_i , δ_i , α_i , φ_i и ψ_i , $i = 1, 2, \dots$, существуют и конечны.

Поскольку процесс отражения атак состоит из циклов "атака — отражение атаки", следовательно, моменты завершения отражения атаки на объект защиты, обнаруженной как первой системой безопасности, так и второй системой безопасности, являются моментами регенерации информационной системы.

Рассмотрим теперь процессы функционирования систем безопасности. Пусть $\xi_i^{(1)}$, $i = 1, 2, \dots$, — наработка до i -го отказа первой системы безопасности. Предполагаем, что $\xi_i^{(1)}$ — независимые одинаково распределенные случайные величины с функцией распределения $F_{\xi_i^{(1)}}(t) = P\{\xi_i^{(1)} \leq t\}$. Продолжительность восстановления первой системы безопасности после i -го отказа обозначим $\eta_i^{(1)}$, $i = 1, 2, \dots$. Здесь $\eta_i^{(1)}$ — независимые, одинаково распределенные случайные величины с функцией распределения $F_{\eta_i^{(1)}}(t) = P\{\eta_i^{(1)} \leq t\}$. Обнаружение отказов систем безопасности происходит только во время периодического контроля состояния. Так, исправность первой системы безопасности проверяется с периодом $T^{(1)}$, а длительность периодического контроля требует времени, равного $\theta^{(1)}$.

Если первая система безопасности (СБ) исправно функционировала случайное время $\xi^{(1)}$, то за это время было выполнено $\left\lceil \frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}} \right\rceil$ периодов контрольной профилактики, и за эти периоды первая СБ находилась в рабочем состоянии $T^{(1)} \left\lceil \frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}} \right\rceil$ единиц времени. Между последней перед отказом контрольной профилактикой, которая произошла в момент $(T^{(1)} + \theta^{(1)}) \left\lceil \frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}} \right\rceil$, и отказом в момент времени $\xi^{(1)}$ остается еще время $\xi^{(1)} - (T^{(1)} + \theta^{(1)}) \times \left\lceil \frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}} \right\rceil$, в течение которого первая СБ все еще исправна. На последнем цикле контрольной профилактики система безопасности находилась в нерабочем состоянии в течение времени $(T^{(1)} + \theta^{(1)}) - \left\lceil \frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}} \right\rceil (T^{(1)} + \theta^{(1)})$, так как отказ ее произошел в момент времени $\xi^{(1)}$, а был обнаружен по окончании цикла контрольной профилактики. Отказ первой системы безопасности будет обнаружен в момент времени $(T^{(1)} + \theta^{(1)}) \left(\left\lceil \frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}} \right\rceil + 1 \right)$, и после проведения ремонтно-восстановительных работ продолжительностью $\eta^{(1)}$ СБ начнет опять исправно функционировать.

Полагаем $\xi_i^{(2)}$, $i = 1, 2, \dots$, — наработка до i -го отказа второй системы безопасности, где $\xi_i^{(2)}$, $i = 1, 2, \dots$, — это независимые, одинаково распределенные случайные величины с функцией распределения $F_{\xi_i^{(2)}}(t) = P\{\xi_i^{(2)} \leq t\}$. Продолжительность восстановления второй системы безопасности после i -го отказа обозначим $\eta_i^{(2)}$, $i = 1, 2, \dots$, где $\eta_i^{(2)}$, $i = 1, 2, \dots$, — независимые, одинаково распределенные случайные величины с функцией распределения $F_{\eta_i^{(2)}}(t) = P\{\eta_i^{(2)} \leq t\}$.

Период контроля состояния второй системы безопасности обозначим $T^{(2)}$, а длительность ее периодического контроля — $\theta^{(2)}$. Отказ второй системы безопасности будет обнаружен в момент времени $(T^{(2)} + \theta^{(2)}) \left(\left\lceil \frac{\xi^{(2)}}{T^{(2)} + \theta^{(2)}} \right\rceil + 1 \right)$. После проведения ремонтно-восстановительных работ с продолжительностью, равной $\eta^{(2)}$, вторая СБ будет опять исправно функционировать. Таким образом, процессы функ-

ционирования первой и второй систем безопасности с контрольными профилактиками есть альтернирующий процесс восстановления. Здесь мы предполагаем, что во время контроля состояния системы безопасности не выполняют своих функций.

Кроме того, пусть математические ожидания $\xi_i^{(1)}$, $\eta_i^{(1)}$, $\xi_i^{(2)}$ и $\eta_i^{(2)}$, $i = 1, 2, \dots$, существуют и конечны, а функции распределения этих случайных величин не являются арифметическими.

Процессы функционирования первой и второй систем безопасности представляются как чередование интервалов "работа — восстановление", которые состоят из независимых, одинаково распределенных случайных величин $\xi_i^{(1)}$, $\eta_i^{(1)}$, $i \geq 1$, и $\xi_i^{(2)}$, $\eta_i^{(2)}$, $i \geq 1$, образующих два альтернирующих процесса восстановления ($\xi_i^{(1)}$, $\eta_i^{(1)}$), $i \geq 1$, и ($\xi_i^{(2)}$, $\eta_i^{(2)}$), $i \geq 1$.

Введем некоторые обозначения и пояснения к процессу проведения контрольных профилактик, которые будут в дальнейшем учтены при составлении математической модели. Через $[x]$ и $\{x\}$ будем обозначать целую и дробную части числа x ,

$$x^+ = \max(x, 0) = \begin{cases} x, & \text{если } x \geq 0; \\ 0, & \text{если } x < 0, \end{cases}$$

$$x \wedge a = \min(x, a) = \begin{cases} x, & \text{если } x \leq a; \\ a, & \text{если } x > a; \end{cases}$$

$$J_{x < a} = \begin{cases} 1, & \text{если } x < a; \\ 0, & \text{если } x \geq a \end{cases} \text{ — индикатор события } x < a.$$

Случайное время до осуществления несанкционированного доступа обозначим ω . Нашей задачей является построение математической модели надежности систем защиты информации с эшелонированной системой безопасности и получение оценки среднего времени $M\omega$ (M — математическое ожидание) до осуществления несанкционированного доступа к информации.

Основные результаты

Поскольку процесс функционирования системы защиты информации описывается как наложение альтернирующих процессов восстановления, то случайное время до осуществления несанкционированного доступа записывается в виде следующей суммы:

$$\omega = \sum_{i=1}^{v-1} \sigma_i + \sigma'_v, \quad (1)$$

где длительность периода регенерации процесса функционирования информационной системы, на котором не произошло несанкционированного доступа, равна

$$\sigma_i = \chi_i \wedge \varphi_i + ((\beta_i + \gamma_i)J_{B_i} + (\delta_i + \alpha_i)J_{\bar{B}_i})J_{\chi_i \leq \varphi_i} + \psi_i J_{\varphi_i < \chi_i},$$

и длительность периода регенерации процесса функционирования информационной системы, на котором произошел несанкционированный доступ, равна

$$\sigma'_i = \chi_i \wedge \varphi_i + \delta_i J_{\chi_i \leq \varphi_i}.$$

В соотношении σ_i используются следующие обозначения: β_i — длительность интервала времени между осуществлением атаки на объект защиты, которая должна быть парирована первой системой безопасности, и срабатыванием этой системы безопасности при условии, что она смогла парировать атаку; B_i — событие, заключающееся в том, что на i -м цикле регенерации процесса функционирования информационной системы при осуществлении атаки, соответствующей первой системе безопасности, эта система безопасности парировала данную атаку

в интервале времени $\left[\sum_{j=1}^{i-1} \sigma_j + \chi_i, \sum_{j=1}^{i-1} \sigma_j + \chi_i + \delta_i \right)$;

$\bar{B}_i$ обозначается противоположное событие — первая система безопасности не парировала данную атаку на объект защиты в указанном интервале времени. Для рассматриваемой модели очевидно, что $0 \leq \beta_i < \delta_i$.

Для функции распределения времени до осуществления несанкционированного доступа, очевидно, справедливо следующее соотношение:

$$F_\omega(t) = P(\omega \leq t) = P\left(\sum_{i=1}^{v-1} \sigma_i + \sigma'_v \leq t\right).$$

Для дальнейших преобразований функции $F_\omega(t)$ воспользуемся методом условных распределений и выразим вероятность сложного события через условные вероятности этого события при соответствующих условиях. Если условия несовместны и образуют полную группу событий, то безусловное преобразование функции распределения $F_\omega(t)$ вычисляется по формуле полной вероятности

$$\tilde{F}_\omega(s) = M e^{-s\omega} = \sum_{n=1}^{\infty} M(e^{-s\omega} | v = n) P(v = n),$$

$$\text{где } \tilde{F}_\omega(s) = \int_0^{\infty} e^{-st} dF_\omega(t) = E[e^{-s\omega}].$$

Поскольку все χ_i и γ_i — независимые случайные величины, то с учетом того, что $\tilde{F}_\sigma(s) = M e^{-s\sigma}$ и $\tilde{F}_{\sigma'}(s) = M e^{-s\sigma'}$ и рассматриваемый процесс регенерирующий, то $P(v = n) = q(1 - q)^{n-1}$, где q — вероятность осуществления несанкционированного доступа на периоде регенерации процесса функ-

ционирования информационной системы. Нетрудно заметить, что

$$M(e^{-s\omega}|v = n) = \\ = M\left(e^{-s\left(\sum_{i=1}^{v-1} \sigma_i + \sigma'_v\right)}\right)|v = n = (\tilde{F}_\sigma(s))^{n-1} \tilde{F}_{\sigma'}(s).$$

Следовательно, преобразование Лапласа—Стилеса функции $F_\omega(t)$ можно переписать в виде

$$\tilde{F}_\omega(s) = \sum_{n=1}^{\infty} (\tilde{F}_\sigma(s))^{n-1} \tilde{F}_{\sigma'}(s) q(1-q)^{n-1} = \\ = \frac{q \tilde{F}_{\sigma'}(s)}{1 - (1-q) \tilde{F}_\sigma(s)}.$$

Выражение для математического ожидания времени до осуществления несанкционированного доступа можно непосредственно вычислить из соотношения (1) или воспользовавшись известным

соотношением $M\omega = \frac{d\tilde{F}_\omega(s)}{ds}\Big|_{s=0}$. Получаем

$$M\omega = M\sigma' + \frac{1-q}{q} M\sigma. \quad (2)$$

Для нахождения математического ожидания наработки комплекса до первой аварии необходимо вычислить отдельно математические ожидания $M\sigma'$ и $M\sigma$. При вычислении математического ожидания $M\sigma'$ будем учитывать, что рассматриваемые случайные величины взаимно независимы. Тогда математическое ожидание $M\sigma'$ запишем следующим образом:

$$M\sigma' = M((\chi \wedge \varphi) + \delta J_{\chi \leq \varphi}) = M(\chi \wedge \varphi) + M\delta P(\chi \leq \varphi) = \\ = \int_0^\infty (1 - F_\chi(t))(1 - F_\varphi(t))dt + \int_0^\infty (1 - F_\delta(t))dt \int_0^\infty F_\chi(t) dF_\varphi(t).$$

По аналогии с $M\sigma'$ вычислим математические ожидания σ :

$$M\sigma = M((\chi \wedge \varphi) + ((\beta + \gamma)J_B + (\delta + \alpha)J_{\bar{B}})J_{\chi \leq \varphi} + \\ + \psi J_{\chi > \varphi}) = M(\chi \wedge \varphi) + M(((\beta + \gamma)J_B + \\ + (\delta + \alpha)J_{\bar{B}})J_{\chi \leq \varphi}) + M(\psi J_{\chi > \varphi}) = \\ = \int_0^\infty (1 - F_\chi(t))(1 - F_\varphi(t))dt + P(\chi \leq \varphi) \times \\ \times ((M\beta + M\gamma)P(B) + (M\delta + M\alpha)P(\bar{B})) + M\psi P(\varphi < \chi).$$

Подставляя вычисленные математические ожидания $M\sigma'$ и $M\sigma$ в формулу (2), получим

$$M\omega = M(\chi \wedge \varphi) + \delta J_{\chi \leq \varphi} + \frac{1-q}{q} M((\chi \wedge \varphi) + \\ + ((\beta + \gamma)J_B + (\delta + \alpha)J_{\bar{B}})J_{\chi \leq \varphi} + \psi J_{\chi > \varphi}).$$

Отметим, что записать функции распределения для случайных величин β_i , от которых зависят величины σ_i , не представляется возможным, но можно

получить верхнюю и нижнюю оценки для математического ожидания времени до осуществления несанкционированного доступа к информации, воспользовавшись отношениями порядка на множестве функций распределения [4]. Данная оценка записывается в следующем виде:

$$M\sigma' + \frac{1-q}{q} (M(\chi \wedge \varphi) + (M\gamma P(B) + (M\delta + M\alpha) \times \\ \times P(\bar{B}))P(\chi \leq \varphi)) + M\psi P(\varphi < \chi) \leq M\omega \leq \\ \leq M\sigma' + \frac{1-q}{q} (M(\chi \wedge \varphi) + ((M\delta + M\gamma)P(B) + \\ + (M\delta + M\alpha)P(\bar{B}))P(\chi \leq \varphi)) + M\psi P(\varphi < \chi). \quad (3)$$

Для того чтобы вычислить значения верхней и нижней оценок, необходимо сначала оценить вероятность q осуществления несанкционированного доступа на периоде регенерации процесса функционирования информационной системы. Чтобы получить оценку этой вероятности, рассмотрим подробнее процессы функционирования систем безопасности.

Рассмотрим отдельный цикл процесса функционирования информационной системы и вычислим $P(\bar{B})$, введя в рассмотрение две вспомогательные случайные величины U_n и V_n , определяемые соотношениями

$$U_n = \sum_{i=1}^n \xi_i^{(1)} + \sum_{i=1}^{n-1} \left((T^{(1)} + \theta^{(1)}) - \right. \\ \left. - \left\{ \frac{\xi_i^{(1)}}{T^{(1)} + \theta^{(1)}} \right\} (T^{(1)} + \theta^{(1)}) \right) + \sum_{i=1}^{n-1} \eta_i^{(1)}; \\ V_n = \sum_{i=1}^n \xi_i^{(1)} + \sum_{i=1}^n \left((T^{(1)} + \theta^{(1)}) - \right. \\ \left. - \left\{ \frac{\xi_i^{(1)}}{T^{(1)} + \theta^{(1)}} \right\} (T^{(1)} + \theta^{(1)}) \right) + \sum_{i=1}^n \eta_i^{(1)},$$

где U_n — момент n -го отказа первой системы безопасности, а V_n — момент окончания восстановления первой системы безопасности после n -го отказа.

Поскольку процесс функционирования системы безопасности является альтернирующим процессом восстановления, мы можем записать

$$U_n = \sum_{i=1}^n \xi_i^{(1)} + \sum_{i=1}^{n-1} \left((T^{(1)} + \theta^{(1)}) - \right. \\ \left. - \left\{ \frac{\xi_i^{(1)}}{T^{(1)} + \theta^{(1)}} \right\} (T^{(1)} + \theta^{(1)}) \right) + \sum_{i=1}^{n-1} \eta_i^{(1)}; \\ V_n = \sum_{i=1}^n \xi_i^{(1)} + \sum_{i=1}^n \left((T^{(1)} + \theta^{(1)}) - \right. \\ \left. - \left\{ \frac{\xi_i^{(1)}}{T^{(1)} + \theta^{(1)}} \right\} (T^{(1)} + \theta^{(1)}) \right) + \sum_{i=1}^n \eta_i^{(1)}.$$

Тогда несанкционированный доступ на рассматриваемом цикле регенерации процесса функционирования информационной системы осуществляется, если

$$U_n \leq \chi < V_n - \delta, \delta \leq V_n - U_n$$

или если

$$V_{n-1} + T^{(1)} \leq \chi < V_{n-1} + T^{(1)} + \theta^{(1)} - \delta;$$

$$V_{n-1} + (T^{(1)} + \theta^{(1)}) + T^{(1)} \leq \chi < V_{n-1} + 2(T^{(1)} + \theta^{(1)}) - \delta;$$

...

$$V_{n-1} + \left(\left\lfloor \frac{\xi_n^{(1)}}{T^{(1)} + \theta^{(1)}} \right\rfloor - 1 \right) (T^{(1)} + \theta^{(1)}) + T^{(1)} \leq$$

$$\leq \chi < V_{n-1} + \left\lfloor \frac{\xi_n^{(1)}}{T^{(1)} + \theta^{(1)}} \right\rfloor (T^{(1)} + \theta^{(1)}) - \delta;$$

$$\delta < \theta^{(1)}.$$

Для дальнейшего изложения необходимо ввести неотрицательные случайные величины

$$\varepsilon_n = (T^{(1)} + \theta^{(1)}) - \left\lfloor \frac{\xi_n^{(1)}}{T^{(1)} + \theta^{(1)}} \right\rfloor (T^{(1)} + \theta^{(1)});$$

$$\Delta_n = \eta_n^{(1)} + \varepsilon_n - \delta, \zeta = \theta^{(1)} - \delta,$$

где ε_n — продолжительность времени от момента возникновения отказа первой системы безопасности до обнаружения отказа и начала ее восстановления.

Принимая во внимание условия осуществления несанкционированного доступа к информации, запишем выражение для вероятности $P(\bar{B})$:

$$P(\bar{B}) = \sum_{n=1}^{\infty} \int_0^{\infty} M \left(J_{U_n \leq x < V_n - \delta} J_{\Delta_n > 0} + \right.$$

$$\left. + \sum_{i=1}^{\left\lfloor \frac{\xi_n^{(1)}}{T^{(1)} + \theta^{(1)}} \right\rfloor} J_{V_{n-1} + (i-1)(T^{(1)} + \theta^{(1)}) \leq x < V_{n-1} + i(T^{(1)} + \theta^{(1)}) - \delta} J_{\zeta > 0} \right) dF_{\chi}(x).$$

Отсюда следует, что

$$P(\bar{B}) = \sum_{n=1}^{\infty} \int_0^{\infty} P(U_n \wedge (U_n + \Delta_n) \leq x) dF_{\chi}(x) -$$

$$- \sum_{n=1}^{\infty} \int_0^{\infty} P(U_n + \Delta_n \leq x) dF_{\chi}(x) +$$

$$+ \sum_{n=1}^{\infty} \int_0^{\infty} M \left(\sum_{i=1}^{\left\lfloor \frac{\xi_n^{(1)}}{T^{(1)} + \theta^{(1)}} \right\rfloor} P((V_{n-1} + i(T^{(1)} + \theta^{(1)}) - \theta^{(1)}) \wedge \right.$$

$$\left. \wedge (V_{n-1} + i(T^{(1)} + \theta^{(1)}) - \theta^{(1)} + \zeta) \leq x) \right) dF_{\chi}(x) -$$

$$- \sum_{n=1}^{\infty} \int_0^{\infty} M \left(\sum_{i=1}^{\left\lfloor \frac{\xi_n^{(1)}}{T^{(1)} + \theta^{(1)}} \right\rfloor} P(V_{n-1} + i(T^{(1)} + \theta^{(1)}) - \right.$$

$$\left. - \theta^{(1)} + \xi \leq x) \right) dF_{\chi}(x) = q_1 + q_2.$$

Заметим, что

$$q_1 = \sum_{n=1}^{\infty} \int_0^{\infty} \int_0^{\infty} ((F_{\xi^{(1)}} * (F_{\xi^{(1)}} * F_{\eta^{(1)}} * F_{\varepsilon})^{*(n-1)})(x) -$$

$$- (F_{\xi^{(1)}} * (F_{\xi^{(1)}} * F_{\eta^{(1)}} * F_{\varepsilon})^{*(n-1)})(x - y)) dF_{\Delta}(y) dF_{\chi}(x),$$

где $F_{\xi^{(1)}} * F_{\eta^{(1)}}(t) = \int_0^t F_{\xi^{(1)}}(t - z) dF_{\eta^{(1)}}(z)$ — свертка

функций распределения $F_{\xi^{(1)}}(t)$ и $F_{\eta^{(1)}}(t)$, $F^{*(n)}(t) = F * F^{*(n-1)}(t)$ — n -кратная свертка функций $F(t)$. Вероятность q_1 можно записать в виде

$$q_1 = \int_0^{\infty} \int_0^{\infty} (H_0(x) - H_0(x - y)) dF_{\Delta}(y) dF_{\chi}(x),$$

где $H_0(x) = \sum_{n=1}^{\infty} F_{\xi^{(1)}} * (F_{\xi^{(1)}} * F_{\eta^{(1)}} * F_{\varepsilon})^{*(n-1)}(x)$ — функция 0-восстановления процесса функционирования первой системы безопасности.

Второе слагаемое преобразуется аналогично:

$$q_2 = \sum_{n=1}^{\infty} \int_0^{\infty} \int_0^{\infty} M \left(\sum_{i=1}^{\left\lfloor \frac{\xi_n^{(1)}}{T^{(1)} + \theta^{(1)}} \right\rfloor} (P(V_{n-1} + i(T^{(1)} + \theta^{(1)}) - \right.$$

$$- \theta^{(1)} \leq x) - P(V_{n-1} + i(T^{(1)} + \theta^{(1)}) -$$

$$- \theta^{(1)} \leq x - y)) \right) dF_{\zeta}(y) dF_{\chi}(x).$$

Полученное выражение можно записать с использованием функций 0-восстановления:

$$q_2 = \int_0^{\infty} \int_0^{\infty} M \left(\sum_{i=1}^{\left\lfloor \frac{\xi_n^{(1)}}{T^{(1)} + \theta^{(1)}} \right\rfloor} (H_{0i}(x) - H_{0i}(x - y)) \right) dF_{\zeta}(y) dF_{\chi}(x),$$

где $H_{0i}(x) = \sum_{n=1}^{\infty} F_{2i, n}(x)$, а $F_{2i, n}(x) = P(V_{n-1} + i(T^{(1)} + \theta^{(1)}) - \theta^{(1)} \leq x)$.

Дальше упростить полученные соотношения не представляется возможным, но можно получить асимптотические оценки, воспользовавшись предельными теоремами теории восстановления [6, 5]:

$$q_1 \approx \frac{1}{M\eta^{(1)} + (T^{(1)} + \theta^{(1)}) + (T^{(1)} + \theta^{(1)})M\left[\frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}}\right]} \int_0^\infty y dF_\Delta(y) =$$

$$= \frac{M\left(\eta^{(1)} + (T^{(1)} + \theta^{(1)}) - \left\{\frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}}\right\}(T^{(1)} + \theta^{(1)}) - \delta\right)^+}{M\eta^{(1)} + (T^{(1)} + \theta^{(1)}) + (T^{(1)} + \theta^{(1)})M\left[\frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}}\right]}; \quad (4)$$

$$q_2 \approx \frac{1}{M\eta^{(1)} + (T^{(1)} + \theta^{(1)}) + (T^{(1)} + \theta^{(1)})M\left[\frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}}\right]} \times$$

$$\times M\left[\frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}}\right] \int_0^\infty y dF_\zeta(y) =$$

$$= \frac{1}{M\eta^{(1)} + (T^{(1)} + \theta^{(1)}) + (T^{(1)} + \theta^{(1)})M\left[\frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}}\right]} \times$$

$$\times M\left[\frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}}\right] M(\theta^{(1)} - \delta)^+. \quad (5)$$

Суммируя (5) и (4), получим вероятность того, что первая система безопасности не парировала атаку на объект защиты в интервале времени $[\chi, \chi + \delta)$:

$$P(\bar{B}) \approx \frac{M\left(\eta^{(1)} + (T^{(1)} + \theta^{(1)}) - \left\{\frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}}\right\}(T^{(1)} + \theta^{(1)}) - \delta\right)^+}{M\eta^{(1)} + (T^{(1)} + \theta^{(1)}) + (T^{(1)} + \theta^{(1)})M\left[\frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}}\right]} +$$

$$+ \frac{M\left[\frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}}\right] M(\theta^{(1)} - \delta)^+}{M\eta^{(1)} + (T^{(1)} + \theta^{(1)}) + (T^{(1)} + \theta^{(1)})M\left[\frac{\xi^{(1)}}{T^{(1)} + \theta^{(1)}}\right]}.$$

Записать в явном виде $\int_0^\infty y dF_\Delta(y)$ и $\int_0^\infty y dF_\zeta(y)$ достаточно сложно даже для относительно простого случая экспоненциальных распределений случайных величин, но можно воспользоваться методом Монте-Карло для получения требуемых численных оценок. Отметим также, что

$$P(B) = 1 - P(\bar{B}).$$

Вероятность q осуществления несанкционированного доступа на периоде регенерации процесса функционирования информационной системы в соответствии с формулой полной вероятности запишем в виде

$$q = P(\text{авария на цикле регенерации} |_{\chi \leq \varphi}) P(\chi \leq \varphi) +$$

$$+ P(\text{авария на цикле регенерации} |_{\chi > \varphi}) P(\chi > \varphi) =$$

$$= q^{(1,2)} P(\chi \leq \varphi) + q^{(2)} P(\chi > \varphi) =$$

$$= q^{(1,2)} \int_0^\infty F_\chi(t) dF_\varphi(t) + q^{(2)} \left(1 - \int_0^\infty F_\chi(t) dF_\varphi(t)\right),$$

где $q^{(1,2)}$ — вероятность того, что на цикле регенерации первая, а за ней вторая системы безопасности не смогут парировать атаку, а $q^{(2)}$ — вероятность того, что на цикле регенерации вторая система безопасности не сможет парировать соответствующий ей тип атаки. Принимая во внимание, что системы безопасности независимы друг от друга, можем записать

$$q^{(1,2)} = P(\bar{B}) q^{(2)}.$$

И, наконец, оценку для вероятности $q^{(2)}$ можно получить, используя тот же подход, который был применен выше при вычислении $P(\bar{B})$. Опустим промежуточные вычисления и приведем сразу конечный результат:

$$q^{(2)} \approx 1 - \frac{M\xi^{(2)} - \theta^{(2)} M\left[\frac{\xi^{(2)}}{T^{(2)} + \theta^{(2)}}\right]}{M\eta^{(2)} + (T^{(2)} + \theta^{(2)}) + (T^{(2)} + \theta^{(2)}) M\left[\frac{\xi^{(2)}}{T^{(2)} + \theta^{(2)}}\right]}.$$

Отметим, что $q^{(2)}$ — коэффициент неготовности второй системы безопасности, принимающий минимальное значение при проведении контрольной профилактики с оптимальным периодом $T_{\text{опт}}^{(2)}$ [7]. Оптимальный период профилактики определяется по формуле

$$T_{\text{опт}}^{(2)} = \sqrt{2\theta^{(2)}(M\xi^{(2)} + M\eta^{(2)})}.$$

Таким образом, нам удалось получить верхнюю и нижнюю асимптотические оценки (3) для среднего времени до осуществления несанкционированного доступа.

Предложенная математическая модель надежности систем защиты информации с эшелонированной системой безопасности с восстанавливаемыми элементами позволяет учитывать временную избыточность, когда одна система безопасности "страхует" другую. Предложенная двухсторонняя оценка для математического ожидания до осуществления несанкционированного доступа к информации позволяет получить достаточно узкий интервал для значений этого показателя надежности системы, достаточно просто вычисляется и учитывает большое число различных параметров процесса функционирования системы защиты информации. Полученные соотношения справедливы без каких-либо предположений относительно функций распределения случайных величин.

Список литературы

1. Щеглов А. Ю. Защита компьютерной информации от несанкционированного доступа. СПб.: Наука и техника, 2004. 384 с.
2. Смит Р. Ф. Эшелонированная оборона. // Windows IT Pro. 2006. № 04.
3. Перегуда А. И., Тимашов Д. А. Математическая модель надежности систем защиты информации // Информационные технологии. 2009. № 8. С. 10—17.
4. Штойян Д. Качественные свойства и оценки стохастических моделей: Пер. с нем. М.: Мир, 1979. 268 с.
5. Rausand M. & Høland A. System Reliability Theory: Model, Statistical Methods and Applications. 2nd ed. NY: John Wiley & Sons. 2004.
6. Байхельт Ф., Франкен П. Надежность и техническое обслуживание. Математический подход: Пер. с нем. М.: Радио и связь, 1988. 392 с.
7. Островский Е. И., Перегуда А. И. О периодическом контроле восстанавливаемых систем // Автоматика и телемеханика. 1984. № 11. С. 39—43.

Указатель статей, опубликованных в журнале "Информационные технологии" в 2012 г.

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Алекперов Р. К.** Создание распределенных вычислительных сред на основе технологии вычислительных облаков. № 2.
- Барский А. Б.** Алгоритмические, архитектурные и структурные методы организации управляющих процессов в виртуальном пространстве средств Grid-системы. № 5.
- Барский А. Б.** Двух- и трехкомпьютерная кластеризация Grid-ресурсов для оптимизированной сборки результатов распределенных вычислений по SPMD-технологии. № 8.
- Бобков С. Г.** Архитектура микропроцессоров суперЭВМ эксафлопсного диапазона. № 12.
- Богатырев В. А.** Отказоустойчивые кластеры дублированных вычислительных комплексов. № 1.
- Васенин В. А., Шундеев А. С.** Эволюция технологии Грид. № 1.
- Викторов Ю. О., Готманов А. Н.** Проблемы качества обслуживания при проектировании коммуникационных фабрик в системах на кристалле. № 11.
- Докучаев А. Н.** Особенности диспетчеризации сверхлегких задач в мультипроцессорных вычислительных системах реального времени. № 2.
- Егоров Г. А., Шяудкулис В. И., Финотти М.** Проблемы построения современных архивных хранилищ данных. № 12.
- Захаров В. М., Шалагин С. В.** Вычисление нелинейных полиномиальных функций на многопроцессорной системе с программируемой архитектурой. № 5.
- Колосков В. А., Колоскова Г. П., Динь Туан Лонг.** Управляемая клеточная непрерывная среда самореконфигурации многопроцессорных систем. № 9.
- Крючков А. О., Крищенко В. А.** Отслеживание жизненного цикла IP-пакетов. № 5.
- Минухин С. В., Знахур С. В.** Оптимизация энергопотребления вычислительных ресурсов двухуровневого Grid на основе балансировки их загрузки. № 4.
- Немолочнов О. Ф., Зыков А. Г., Поляков В. И., Меженин А. В.** Неравенства-отношения и выбор альтернативных решений управления вычислительными процессами. № 5.
- Саак А. Э.** Диспетчеризация в Grid-системах на основе однородной квадратичной типизации массивов заявок пользователей. № 4.
- Саак А. Э.** Сравнительный анализ полиномиальных алгоритмов диспетчеризации в Grid-системах. № 9.
- Салибеян С. М., Панфилов П. Б.** Объектно-атрибутная архитектура — новый подход к созданию объектных систем. № 2.
- Червяков Н. И., Авербух В. М., Бабенко М. Г., Лавриненко И. Н., Ляхов П. А.** Эффективный метод приближенного вычисления позиционной характеристики модулярного представления числа. № 11.

СЕТИ И СИСТЕМЫ СВЯЗИ

- Вознюк М. А., Дворников С. В., Винокуров М. Е., Петросян А. П., Романенко П. Г.** Работа линий радиосвязи с ППРЧ в условиях преднамеренных помех. № 10.
- Дворников С. В., Казаков Е. В., Устинов А. А., Чихонадских А. П., Андриянов С. В.** Обоснование модели секвентного сигнала для систем связи. № 9.
- Карутин С. Н.** Метод автономного эфемеридно-временного обеспечения навигационных космических аппаратов системы ГЛОНАСС. № 6.

Мошевикин А. П., Галов А. С., Волков А. С. Точность расчета локации в беспроводных сетях датчиков стандарта napo-LOC (IEEE 802.15.4a). № 9.

Тарасов В. Н., Мезенцева Е. М. Организация защиты компьютерных сетей. Метод многомодульной фильтрации спама на Web-сайтах. № 6.

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Александров А. Е., Востриков А. А., Шалыминов И. В.** Программная система "Прогноз" для анализа безопасности энергетических систем. № 11.
- Андреев Д. В.** Универсальный логический модуль для обработки многозначных данных. № 7.
- Борчик Е. М., Якимов А. И., Степанов А. И., Башаримов В. В.** Построение плотностей распределения Пирсона для многомодальных выборок с применением кластеризации. № 12.
- Владимирский Э. И., Исмаилов Б. И.** Синхронизация в управлении хаотическими системами. № 1.
- Воловиков В. В., Сарафанов А. В.** Моделирование теплового сопротивления при вынужденном конвективном теплообмене в каналах конструкций радиоэлектронной аппаратуры. № 2.
- Горохов А. В.** Алгоритмизация поиска источников хаотических колебаний в крупномасштабных энергосистемах. № 11.
- Елтаренко Е. А.** Конъюнктивные операторы агрегирования в многокритериальных задачах. № 8.
- Зак Ю. А.** Адаптивные методы глобального случайного поиска в решении одного класса многоэкстремальных задач в условиях ограничений. № 1.
- Зак Ю. А.** Распределение множества заданий и определение оптимальных очередностей их выполнения на параллельных машинах методами динамического программирования. № 8.
- Иванова К. Ф.** Знаковый подход к оценке решения интервальных линейных систем. № 9.
- Казаков П. В.** Оценка эффективности генетических алгоритмов многокритериальной оптимизации. Части 1, 2. № 8, 9.
- Кухаренко Б. Г., Пономарев Д. И.** Аппроксимация смесью Гауссовых распределений в модели переключающегося фильтра Калмана для идентификации режимов колебаний временных рядов. № 7.
- Левин В. И.** Методы оптимизации систем в условиях интервальной неопределенности параметров. № 4.
- Левин В. И.** Упрощенная методика оптимизации систем в условиях интервальной неопределенности. № 12.
- Львович Я. Е., Каширина И. Л., Тузилов А. А.** Генетический алгоритм решения многокритериальной задачи повышения надежности резервирования. № 6.
- Меркушева А. В., Малыгина Г. Ф., Малыхин В. М.** Структуры, методы и алгоритмы адаптивной фильтрации нестационарных сигналов. № 2.
- Оцоков Ш. А.** Метод проверки необходимости округления при организации высокоточных вычислений в модулярной арифметике. № 7.
- Панин И. Г., Благовещенский В. В.** Численное решение уравнения эволюции дислокационного источника Франка — Рида. № 6.
- Сандуляну Л. Н., Стрижов В. В.** Выбор признаков в авторегрессионных задачах прогнозирования. № 7.
- Саушев А. В.** Метод и алгоритмы параметрического синтеза электротехнических систем по критерию запаса работоспособности. № 12.
- Скрибцов П. В., Долгополов А. В.** Применение GPU для решения задач математического моделирования в области гидродинамики и гидрологии. № 5.
- Струченков В. И.** Новые алгоритмы оптимизации в задачах обеспечения надежности сложных систем. № 11.
- Струченков В. И.** Устаревшие стереотипы и новые алгоритмы решения прикладных задач дискретной оптимизации. № 5.

- Трудоношин В. А., Федорук В. Г.** Сравнение математических моделей шарнирного соединения. № 8.
- Ураков А. Р., Тимеряев Т. В.** Многоуровневый алгоритм разбиения графов по критерию средней длины. № 4.
- Чепурко В. А., Чепурко С. В.** Непараметрическая оценка коэффициента деградации геометрических процессов. № 7.
- Юсупова Н. И., Валеева А. Ф., Файзрахманов Р. И.** Вероятностный алгоритм муравьиной колонии для решения задач раскроя промышленных материалов на заготовки различных геометрических форм. № 5.

КОДИРОВАНИЕ И ОБРАБОТКА СИГНАЛОВ

- Агиевич С. Н.** Обработка сигналов методами сплайн-алгебраического гармонического анализа. № 5.
- Агиевич С. Н.** Оценка параметров сигналов в базисах функций сплайн-характеров. № 6.
- Агиевич С. Н.** Теоретические основы сплайн-алгебраического гармонического анализа сигналов систем радиосвязи. № 8.
- Браиловский И. В., Соломеш Н. А.** Моделирование качества для видеокодирования. № 1.
- Васильев В. В., Симак Л. А.** Полиномиальная аппроксимация сигналов как операционное исчисление: применение к моделированию динамических систем дробного порядка. № 3.
- Елесина С. И., Никифоров М. Б.** Повышение эффективности генетического алгоритма. № 3.
- Захаров А. В., Богданов М. Р., Мухамедьянов А. Р.** Аналитический метод решения задачи позиционирования. № 8.
- Кухаренко Б. Г., Пономарев Д. И.** Использование метода Прони для оценки временного масштаба при обнаружении паттернов во временных рядах. № 1.
- Лысенков И. Н., Дворников С. В., Дворников С. С., Ишин Д. М., Устинов А. А.** Пороговое декодирование самоортогональных кодов. № 5.
- Мандрикова О. В., Полозов Ю. А.** Критерии выбора вейвлет-функции в задачах аппроксимации природных временных рядов сложной структуры. № 1.

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

- Андрейчикова О. Н., Андрейчиков А. В.** Что делать с авторским правом в сети Интернет? Поиск решения методом аналитических сетей. № 8.
- Афонин С. А., Бахтин А. В.** Построение иерархии понятий на основе лексических шаблонов. № 3.
- Балакирев Н. Е., Полицина Е. В., Мозина Е. В.** Инструмент прогнозирования времени обработки в системе автоматизированного анализа текста. № 10.
- Баранов Д. В.** Логические уравнения в эквивалентных LP-структурах. № 8.
- Болховитянов А. В., Чеповский А. М.** Методика автоматического выделения структурных единиц в предложениях на русском языке. № 2.
- Борисов В. В., Сысков В. В.** Мультиагентное моделирование сложных организационно-технических систем в условиях противоборства. № 4.
- Бочков М. В., Бородащенко А. Ю.** Перспективы развития методов семантической фильтрации текстовых документов. № 4.
- Бочков М. В., Бойков П. Н.** Способ автоматического рубрицирования неструктурированной информации сети Интернет. № 2.
- Волобуев М. Ф., Демчук В. А., Замыслов М. А., Мальцев А. М., Михайленко С. Б.** Управление резервированной с помощью мажоритарных элементов системой с применением методов нечеткой логики. № 12.
- Костенко К. И.** Операции унифицированной технологии построения цифровых пространств знаний. № 3.

- Левин В. И.** Логико-автоматная теория множеств. № 10.
- Левков А. А.** Интеграция знаний и данных в больших информационных системах. № 2.
- Нечаев В. В.** Раскрытие неопределенности системной задачи, представленной концептуальной моделью. № 12.
- Норенков И. П., Уваров М. Ю.** Задачи обработки знаний на основе ролевой кластеризации онтологий. № 2.
- Норенков И. П., Уваров М. Ю.** Извлечение знаний из текстовых документов на основе концептно-ориентированной типизации запросов. № 4.
- Норенков И. П., Уваров М. Ю., Грушин М. А., Овешников А. А.** Классификация документов на основе кластеризации онтологий. № 8.
- Паламарь И. Н., Юлин С. С.** Анализ применимости современных методов статистического моделирования для распознавания и понимания речи. № 3.
- Ходашинский И. А.** Идентификация нечетких систем на базе алгоритма имитации отжига и методов, основанных на производных. № 3.
- Шабунин А. Б., Кузнецов Н. А., Скобелев П. О., Бабанин И. О., Кожевников С. С., Симонова Е. В., Степанов М. Е., Царев А. В.** Разработка онтологии для мультиагентной системы управления ресурсами ОАО "РЖД". № 12.
- Щеголева Н. Л., Кухарев Г. А., Форчманский П.** Простой алгоритм распознавания изображений лиц с проблемами освещения. № 12.

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Андрейчиков А. В., Хорычев А. А., Андрейчикова О. Н.** Экспертная система для начальных стадий проектирования инноваций. № 3.
- Бибило П. Н.** Экспертные знания в САПР цифровых устройств. № 6.
- Бобков С. Г., Задябин С. О., Крыницкий А. В., Новожилов Е. Е.** Верификация модели контроллера ведущего интерфейса USB 2.0. № 3.
- Князев Н. А., Малинаускас К. К.** Алгоритмы поиска критических путей в задаче статического временного анализа СБИС. № 11.
- Кривошеин Д. Ю., Марченко А. М.** Инкрементальный алгоритм поиска кратчайших путей в графе. № 7.
- Кузнецов Е. С., Талицкий Е. Н., Шумарин С. В.** Автоматизация проектирования виброзащиты ячеек электронной аппаратуры демпфирующими вставками. № 3.
- Талалай М. С., Трушин К. В., Венгер О. В.** Логический синтез булевой функции для проектирования интегральных схем на транзисторных шаблонах. № 6.
- Чувашева Е. С., Чувашев С. Н., Зорина И. Г.** Комплексная математическая модель для концептуального проектирования высокоскоростных летательных аппаратов. № 11.

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ

- Бражник С. А., Малафеев С. И.** Синхронизация в промышленных контроллерах с операционной системой Linux. № 7.
- Мартынов П. Н., Ковшов Е. Е.** Разработка средств автоматизации тестирования интерфейсов пользователя в человеко-машинных системах управления. № 7.

БАЗЫ ДАННЫХ

- Керимов С. Г., Керимова Е. С.** Репозитории в информационных системах и программном обеспечении. № 3.

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Бочков М. В., Лалушкин Ю. П., Шкадов А. А. Имитационное моделирование процессов управления защищенностью ресурсов компьютерной сети на основе иерархических раскрашенных временных сетей Петри. № 6.
- Васенин В. А. К созданию международной системы мониторинга и анализа информационного пространства для предотвращения и прекращения военно-политических киберконфликтов. № 9.
- Вялых А. С., Вялых С. А., Сирота А. А. Оценка уязвимости информационной системы на основе ситуационной модели динамики конфликта. № 9.
- Имамвердиев Я. Н. Метод обнаружения переделанных отпечатков пальцев на основе фрактальных характеристик. № 9.
- Каленик А. Н., Коляда А. А., Коляда Н. А., Чернявский А. Ф., Шабинская Е. В. Умножение и возведение в степень по большому модулю с использованием минимально избыточной модулярной арифметики. № 4.
- Козленко А. В. Метод оценки защищенности информации от несанкционированного доступа в компьютерных сетях в условиях комплексных угроз. № 6.
- Крупнов И. В. Анализ проблем обеспечения информационной безопасности системы электронного голосования в условиях российского информационного пространства. № 4.
- Перегуда А. И., Тимашов Д. А. Математическая модель надежности информационных систем с эшелонированной защитой от несанкционированного доступа. № 12.

ВЕБ-ТЕХНОЛОГИИ

- Алгулиев Р. М., Алыгулиев Р. М., Гянджалиев Ф. С. Извлечение социальных сетей научных исследователей в Веб с использованием информации из нескольких источников. № 5.

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Астафуров В. Г., Скороходов А. В. Нейросетевой классификатор облачности по спутниковым данным. № 7.
- Бобков В. А., Мельман С. В., Май В. П. Визуализация динамики циклонов. № 2.
- Косьяков С. В., Гадалов А. Б., Садыков А. М. Моделирование пространственных данных при решении задач дискретной оптимизации в среде ГИС. № 7.

ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ СИСТЕМЫ

- Асадов Х. Г., Гашимов Дж. Г. Общая теория энтропийной фильтрации сигналов дистанционного зондирования тектонических объектов. № 10.

КОМПЬЮТЕРНАЯ ГРАФИКА И ГЕОМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ

- Бимаков В. А., Бимаков Е. В. Принципы построения и функционирования устройств для высокоскоростной обработки n -мерных пиксельных геометрических моделей. № 6.
- Бобков В. А., Роньшин Ю. И., Машенцев В. Ю. Построение карт глубин на множественной последовательности видов. № 6.
- Малыхин В. М., Малыхина Г. Ф., Меркушева А. В. Метод регуляризации для улучшения качества восстановления изображения. № 1.

ОРГАНИЗАЦИОННО-ИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Сенотрусов И. А., Нешков П. А. Анализ и заполнение пропусков данных в базах данных информационно-аналитических систем. № 1.

НАДЕЖНОСТЬ И ДИАГНОСТИКА

- Антонов А. В., Соколов С. В., Чепурко В. А. Бутстреп-метод оценки характеристик надежности восстанавливаемых объектов по специфическим данным об отказах. № 4.

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Ворожцов Е. В. Реализация интерфейса между Фортран-программами и графической системой комплекса *Mathematica*. № 11.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Авдошин С. М., Савельева А. А. Междисциплинарный подход к изучению информационной безопасности на основе анализа конкретных ситуаций. № 5.
- Куравский Л. С., Юрьев Г. А. Применение фильтра Калмана для фильтрации артефактов при адаптивном тестировании. № 4.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕДИЦИНЕ

- Мажуга В. В., Хачумов В. М. Цифровая фильтрация и анализ электрокардиограмм. № 4.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ И УПРАВЛЕНИИ

- Архипова Я. А. Автоматическая периодизация динамики биржевых котировок на основе критерия минимума информационного рассогласования. № 6.
- Вашенко Г. В., Капулин Д. В. Возможности интеграции бизнес-процессных моделей и ERP-систем. № 1.
- Галиев Д. Р., Исавнин А. Г. Модели выбора оптимального инвестиционного портфеля с нечетко-множественными доходностями. № 8.
- Димитриенко Ю. И., Димитриенко О. Ю. Континуальное моделирование динамики кластеров экономических данных в условиях внешних кризисных воздействий. № 1.
- Капулин Д. В. Прикладное решение по подготовке информации о бизнес-процессах для платформы IC: Предприятие с использованием ERwin Process Modeler. № 4.
- Караев Р. А., Гюльмамедов Р. Г., Садыхова Н. Ю., Нагиев М. А. Индикаторы состояния и факторы развития ИКТ-сектора регионов. № 4.
- Мезенцев Ю. А., Павлов П. С. К программной реализации декомпозиционного алгоритма решения одного класса задач дискретной оптимизации с полуопределенной релаксацией. № 2.
- Рындни А. А., Ульев В. П. Проектирование гибридной системы скоринговой оценки кредитоспособности юридических лиц. № 6.
- Сологуб Р. А. Порождение регрессионных моделей поверхности волатильности биржевых опционов. № 8.
- Топка В. В. Управление стоимостью проекта с учетом показателя его надежности. № 2.

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Барбасова Т. А., Вставская Е. В., Константинов В. И. Кодирование информации методом пропуска полупериодов сетевого напряжения в системах освещения. № 2.
- Боровский А. С., Тарасов А. Д. Метод обработки экспертной информации на основе нечетких гиперграфов для проектирования систем физической защиты. № 2.
- Гриняк В. М., Девятисильный А. С. Нечеткая система экспертных оценок характера движения судна. № 8.
- Набибекова Г. Ч. Применение OLAP-технологий в системах поддержки принятия решений в сфере внешней политики. № 2.

Немтинов В. А., Пеншин Н. В., Донских Ю. А., Немтинов К. В., Егоров Е. С. Имитационное моделирование динамических процессов при управлении городским пассажирским транспортом. № 4.

ДИСКУССИОННЫЙ КЛУБ

Вяткин В. Б. Информационно-квантовые характеристики и отраженные образы конечных множеств. № 7.

ОБМЕН ОПЫТОМ

Федорова Е. В., Гетьман М. А., Савельева Е. В. Использование облачных технологий Google Apps Education Edition для создания саморазвивающейся информационной платформы вуза. № 7.

КАФЕДРЕ

САПР МГТУ им. Н. Э. БАУМАНА — 30 ЛЕТ

Артемченко Ю. Н., Волкоморов С. В., Карпенко А. П., Мартынюк В. А., Пашенко В. Н., Темерев К. А., Шарыгин А. В. Многосекционный манипулятор параллельной структуры для управления ориентацией космической обсерватории "Миллиметр". № 10.

Афиногенов Е. И., Волосатова Т. М., Сельвесюк Н. И., Чичварин Н. В. Системный подход к проектированию алгоритмов информационно-управляющей системы летательного аппарата с учетом противодействия информационным подсистемам. № 10.

Божко А. Н. Моделирование позиционных связей в механических системах. № 10.

Жук Д. М., Перфильев С. А. Системы автоматизированного проектирования в хирургии — CAS-системы. № 10.

Маничев В. Б., Жук Д. М. SADEL — Си-библиотека для решения алгебраических и дифференциальных уравнений с максимально возможной компьютерной точностью. № 10.

Норенков И. П. Научные исследования кафедры "Системы автоматизированного проектирования" МГТУ им. Н. Э. Баумана и развитие информационных технологий. № 10.

Норенков И. П., Уваров М. Ю. Семантический анализ текстовых документов. № 10.

Трудоношин В. А. Эволюция средств моделирования динамических систем. № 10.

Журнал в журнале

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

Алгазинов Э. К., Дрюченко М. А., Митрофанова Е. Ю., Сирота А. А. Математическое и программное обеспечение для создания цифровых водяных знаков с использованием искусственных нейронных сетей. № 9.

Борисов В. В., Мисник А. Е. Комбинированный нейросетевой способ моделирования для оперативного управления сложными системами. № 7.

Девятисильный А. С., Гриняк В. М. Нейросетевой алгоритм сопровождения судов в системах управления движением. № 3.

Демиденков К. А., Мельников И. И., Евсеенко И. А. Совершенствование способов определения плотности транспортного потока и его состава на базе искусственных нейронных сетей путем использования параллельных вычислений. № 11.

Добротворский А. С. Оценка перспектив использования ассоциативных инструментальных платформ квантовых компьютеров, работающих в многозначном алфавите. № 3.

Емельянова Н. А., Гафаров Ф. М., Сулейманов Я. А., Хуснутдинов Н. Р. Математическая модель эволюции нейронной сети. № 9.

Кулик С. Д., Николец Д. А. Автоматизация классификационно-диагностических почерковедческих исследований с помощью нейронных сетей. № 1.

Мышев А. В. Информационная модель нейросети в технологиях вычислительного интеллекта и формах реализации компьютеринга. № 1.

Мышев А. В. Метрологическая теория динамики взаимодействующих объектов в информационном поле нейросети. № 11.

Нгуен Виет Хунг. Сжатие изображения с использованием нейронной сети в преобразованном пространстве. № 1.

Нгуен Виет Хунг, Муравьев А. В. Применение нейросетевого алгоритма в задаче компенсации движения в видеокодировании с помощью технологии NVIDIA CUDA. № 5.

Осинов В. Ю. Метод настройки ассоциативной интеллектуальной системы на входные сигналы. № 9.

Петрушан М. В., Самарин А. И. Метод контрастирования описаний изображений лиц для системы санкционированного доступа. № 3.

Салибекян С. М., Панфилов П. Б. Реализация искусственных нейронных сетей с помощью атрибутной архитектуры вычислительной системы. № 5.

Скрибцов П. В., Казанцев П. А., Червоненкис М. А. Применение искусственных нейронных сетей для решения обратных задач в гидрологии. № 7.

Соломаха А. А., Милова К. А., Горбаченко В. И. Нейросетевая экспертная медико-технологическая система прогнозирования риска развития осложнений у больных с гнойно-деструктивными заболеваниями легких. № 3.

Степанов С. Ю., Кабак И. С. Алгоритм фрагментации больших нейронных сетей и исследование его сходимости. № 7.

Томашевич Н. С. Алгоритм построения признаковой нейронной сети для задач обработки изображений на примере задачи распознавания букв. № 11.

Энгель Е. А. Решение задач управления, принятия решений и обработки информации методом нечеткой селективной нейросети. № 5.

Указатель приложений к журналу "Информационные технологии", опубликованных в 2012 г.

Барский А. Б. История российских суперкомпьютеров специального назначения: свидетельства и размышления. № 9.

Бобков С. Г., Косарев И. М. Методы повышения производительности вычислительных систем. № 10.

Волосатова Т. М., Денисов А. В., Чичварин Н. В. Комбинированные методы защиты данных в САПР. № 5.

Галушкин А. И. Стратегия развития современных суперкомпьютеров на пути к эксафлопным вычислениям. № 2.

Карпенко А. П. Популяционные алгоритмы глобальной поисковой оптимизации. Обзор новых и малоизвестных алгоритмов. № 7.

Котенко И. В., Коновалов А. М., Шоров А. В. Исследовательское моделирование бот-сетей и механизмов защиты от них. № 1.

Кудинов Ю. И., Келина А. Ю. Методы синтеза и настройки нечетких ПИД регуляторов Мамдани. № 6.

Кухаренко Б. Г. Аккумулирующий анализ главных компонент потоков тензорных данных. № 12.

Кухаренко Б. Г. Алгоритмы выделения объектов переднего плана из фона и интерактивного редактирования изображений. № 4.

Лейпяля Т., Шилов В. В. Из истории вычислительной техники в России XIX — начала XX веков. № 8.

Меркушева А. В. Элементы прикладной теории времячастотных преобразований и обработки нестационарных сигналов в информационно-измерительных системах. № 3.

Трахтенгерц Э. А. Компьютерные методы подготовки к противодействию группе прогнозируемых катастроф. № 11.

CONTENTS

Bobkov S. G. <i>Microprocessors Architecture for Exascale Computers</i>	2
--	---

New decisions for designing high-reliability microprocessors for exascale computers are consider.

Keywords: supercomputer, supercomputer technologies, computer architecture

Egorov G. A., Chiaoudkoulis V. I., Finotti M. <i>Problems of the Modern Archive Data Storage</i>	9
---	---

This white paper analyzes problems of the architecture of the modern archiving storage and how those problems resolve in modern data storage systems. The cloud storage concept is suggested as a basis for the now types of the data storage. An example of implementation of the modern data archiving system is described.

Keywords: data storage, archiving storage, cloud storage

Levin V. I. <i>The Simplified Method of Systems Optimization in Conditions of Intervals Uncertainty</i>	19
--	----

The method of conditional optimization of functions with interval parameters proposed and justified. In this method interval problem is reduced to single optimization problem with exactly known parameters, not two, as it was before.

Keywords: system, constrained optimization, uncertainty, interval mathematics, determinization method

Saushev A. V. <i>Method and Algorithms of Parametrical Synthesis of Electrotechnical Systems by Criterion of a Stock of Working Capacity</i>	24
---	----

The method of narrowed areas and possible algorithms of its technical realization is considered, allowing to solve a problem of parametrical optimization for a wide class of electrotechnical systems by criterion of a stock of working capacity.

Keywords: working capacity area, a method of narrowed areas, logic R-functions, system parameters, electrotechnical systems

Borchyk E. M., Yakimau A. I., Stepanov A. I., Basharimov V. V. <i>Construction of the Density Distribution of Pearson for Multimodal Samples with the Use of Clustering</i>	30
--	----

The problem of building statistical models for multimodal distributions of sample data is done. The procedure of separation of the original sample of multi-modal methods of cluster analysis on the number of homogeneous samples of clusters is given. For samples of clusters constructed its density function of the generalized distribution of Pearson. To verify that the constructed statistical models used statistical distributions of the Pearson criterion, the Kolmogorov-Smirnov, Mises.

Keywords: generalized Pearson distribution, statistical goodness; multimodal sampling, methods of cluster analysis

Shchegoleva N. L., Kukharev G. A., Forcznanski P. <i>Simple Algorithm for the Recognition Face Images with Lighting Problem</i>	36
--	----

An algorithm for solving this problem based on the method of 2DDCT, together with procedures to reduce the gradient of the brightness of the source images (gamma correction, logarithm) and the union under the terms of the spectral features high brightness. Experiments were carried out based on the Yale B and Yale B +.

Keywords: face recognition, highlight and shadows in the image, two-dimensional cosine transform, gamma correction, logarithm, fusion spectra of the images

Shabunin A. B., Kuznetsov N. A., Skobelev P. O., Babanin I. O., Kozhevnikov S. S., Simonova E. V., Stepanov M. E., Tsarev A. V. <i>Multiagent Adaptive Resource Management System Ontology Development for JSC "Russian Railways"</i>	42
--	----

The paper describes the principles of ontology development for multi-agent adaptive resource management system created for JSC "Russian Railway". The structure of basic ontology resources, functions of ontology management toolset and agents rules editor are shown. As an example in this paper is used the scene implementing and solving the conflict in trains schedule.

Keywords: multi-agent system, knowledge representation, adaptive planning, ontology, decision-making, real time

The study considers the possibility of evaluation of indeterminate form of specific user problems (SUP), represented by the interpretation of a system problem conceptual model (SPCM). The informational indetermination of the SUP is considered in the context of SPCM analysis. A recursive mechanism for creation of auxiliary problems conceptual models (APCM) is suggested to evaluate the informational indetermination of SUP. With the help of this mechanism the SPCM is transformed into a conceptual model of the problem system (CMPS). To evaluate the degree of informational indetermination of the SUP a complex indicator (ICINS) is introduced, which is based on the SPCM and includes the partial indicators of completeness, integrity, necessity, and sufficiency. Generalized and complete structural-logical schemes are provided, that reflect the processes of APCM complex creation for a case of SPCM maximum indetermination.

Keywords: problem, problem system, model, conceptual model, conceptual stratum, problem solver, goal of problem, method, Algorithm, adequacy, relevance, houl, requirement, intellectual problem solver, recursion, recursive mechanism

The offered approach to building algorithm automatic recognition refusal reserved by means of voting system element and management her(its) rearranging, directed on reconstruction of capacity to work, founded on presentation output parameter element of the reserved system in linguistical form and realization to strategies of management with use the fuzzy logic. The results of the work algorithm introduce in the manner of vector preferable alternative decisions, formed by means of procedures of the fuzzy conclusion, marketed in programme ambience fuzzy TECH.

Keywords: management of redundant system, majority elements, fuzzy set, the fuzzy logic, accessory function, fuzzification, an fuzzy conclusion, defuzzification, fuzzy modelling

In this paper propose mathematical model of reliability of information system and security systems of information which consist of security object of information and two security systems. Process of functioning of security systems of information describe as superposition of alternation process of renewal. Computed upper and lower bounds for mean time before realization of illegal access.

Keywords: reliability, security system, random values, failure time, stochastic process, mean time, distribution function

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е. В. Конова*.

Корректор *Е.В. Комиссарова*.

Сдано в набор 04.10.2012. Подписано в печать 20.11.2012. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT1212. Цена договорная.

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

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Авансесд солюшнз". Отпечатано в ООО "Авансесд солюшнз".

105120, г. Москва, ул. Нижняя Сыромятническая, д. 5/7, стр. 2, офис 2.