

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ
УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

ISSN 0135-1710

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И ПРИБОРЫ АВТОМАТИКИ

**Всеукраинский межведомственный
научно-технический сборник**

Основан в 1965 г.

Выпуск 145

Харьков
2008

В сборнике представлены результаты исследований, касающихся компьютерной инженерии, управления, технической диагностики, автоматизации проектирования, оптимизированного использования компьютерных сетей и создания интеллектуальных экспертных систем. Предложены новые подходы, алгоритмы и их программная реализация в области автоматического управления сложными системами, оригинальные информационные технологии в науке, образовании, медицине.

Для преподавателей университетов, научных работников, специалистов, аспирантов.

У збірнику наведено результати досліджень, що стосуються комп'ютерної інженерії, управління, технічної діагностики, автоматизації проектування, оптимізованого використання комп'ютерних мереж і створення інтелектуальних експертних систем. Запропоновано нові підходи, алгоритми та їх програмна реалізація в області автоматичного управління складними системами, оригінальні інформаційні технології в науці, освіті, медицині.

Для викладачів університетів, науковців, фахівців, аспірантів.

Редакционная коллегия:

В.В. Семенец, д-р техн. наук, проф. (гл. ред.), М.Ф. Бондаренко, д-р техн. наук, проф., И.Д. Горбенко, д-р техн. наук, проф., Е.П. Пуятин, д-р техн. наук, проф., В.П. Тарасенко, д-р техн. наук, проф., Г.И. Загарий, д-р техн. наук, проф., А.Штефан, доктор-инженер, Г.Ф. Кривуля, д-р техн. наук, проф., О.Г. Руденко, д-р техн. наук, проф., Н.В. Алипов, д-р техн. наук, проф., Е.В. Бодянский, д-р техн. наук, проф., Э.Г. Петров, д-р техн. наук, проф., В.Ф. Шостак, д-р техн. наук, проф., В.М. Левыкин, д-р техн. наук, проф., В.И. Хаханов, д-р техн. наук, проф. (отв. ред.).

Свидетельство о государственной регистрации
печатного средства массовой информации

КВ № 12073-944ПР от 07.12.2006 г.

Адрес редакционной коллегии: Украина, 61166, Харьков, просп. Ленина, 14, Харьковский национальный университет радиоэлектроники, комн. 321, тел. 70-21-326

© Харківський національний університет
радіоелектроніки, 2008

СОДЕРЖАНИЕ

ЕВЛАНОВ М.В., СЛИПЧЕНКО Е.В., НИКИТЮК В.А. ПОДХОД К ПРОЕКТИРОВАНИЮ ХРАНИЛИЩ ДАННЫХ В ИНФОРМАЦИОННЫХ СИСТЕМАХ.....	4
БАБИЧ А.В., КУДИНА М.В., ЕМЕЛЬЯНОВ И.В. ИССЛЕДОВАНИЕ МЕТОДОВ РЕШЕНИЯ ЗАДАЧ ИНЖИНИРИНГА ТРАФИКА В СЕТЯХ СЛЕДУЮЩЕГО ПОКОЛЕНИЯ.....	8
ЗАЙЧЕНКО С.А., ХАХАНОВ В.И. ФОРМАЛЬНАЯ СЕМАНТИКА СЛОЖНЫХ ОПЕРАТОРОВ ЛИНЕЙНОЙ ТЕМПОРАЛЬНОЙ ЛОГИКИ.....	14
ОЛЙНИК О.І. ЧИСЕЛЬНЕ МОДЕЛЮВАННЯ ДИФУЗІЙНИХ ПРОЦЕСІВ У СИСТЕМАХ З МАСИВАМИ ЗАГЛИБЛЕНИХ МІКРОЕЛЕКТРОДІВ У ФОРМІ ДИСКУ.....	29
ЛИТВИНОВА Е.И. ТЕХНОЛОГИИ ВСТРОЕННОГО РЕМОНТА КОМПОНЕНТОВ SYSTEM-IN-PACKAGE.....	40
ТЕВЯШЕВ А.Д., ЗОЛОТАРЕВ Д.А. ОБ ОДНОМ МЕТОДЕ РЕШЕНИЯ ЗАДАЧИ ОПТИМИЗАЦИИ ПЛАНОВЫХ РЕЖИМОВ ТРАНСПОРТА И РАСПРЕДЕЛЕНИЯ ПРИРОДНОГО ГАЗА В ГОРОДСКИХ ГАЗОРАСПРЕДЕЛИТЕЛЬНЫХ СЕТЯХ.....	48
ЧУБ И.А., НОВОЖИЛОВА М.В. МОДИФИКАЦИЯ ТОЧНОГО МЕТОДА РЕШЕНИЯ ЗАДАЧИ РАЗМЕЩЕНИЯ ПРЯМОУГОЛЬНЫХ ОБЪЕКТОВ.....	57
ТАЯНОВ С.А., ТАЯНОВ В.А. МЕТОДИКА КЛАСТЕРИЗАЦІЇ ЗОБРАЖЕНЬ ДЛЯ ЇХ КОМПРЕСІЇ НА ОСНОВІ КОМПОНЕНТНОГО АНАЛІЗУ.....	63
ШАХОВСЬКА Н.Б., УГРИН Д.І. ТЕХНОЛОГІЯ ETL В ІНТЕГРАЦІЇ ДАНИХ ТУРИСТИЧНОГО БІЗНЕСУ.....	68
ДОЛГОВА Н.Г., НОВОЖИЛОВА М.В., СИНЕЛЬНИКОВА О.И. МЕТОД ОЦЕНКИ АЛЬТЕРНАТИВНЫХ ВАРИАНТОВ ФУНКЦИОНАЛЬНОГО ЗОНИРОВАНИЯ ТЕРРИТОРИИ ГОРОДА.....	73
ЕВГРАФОВ В.Н. ПРОИЗВОДИТЕЛЬНОСТЬ БУФЕРНЫХ АСИНХРОННЫХ МНОГОСТУПЕНЧАТЫХ СЕТЕЙ С ПРОИЗВОЛЬНЫМ ЧИСЛОМ ПРИОРИТЕТНЫХ МОДУЛЕЙ ПАМЯТИ.....	80
КАКУРИН Н.Я., КОВАЛЕНКО С.Н., ЛОПУХИН Ю.В., МАКАРЕНКО А.Н. СПОСОБ ПОВЫШЕНИЯ БЫСТРОДЕЙСТВИЯ ПРЕОБРАЗОВАТЕЛЕЙ КОДОВ НА СЧЕТЧИКАХ.....	86
ЗАЯЦЬ В.М. ПЕРСПЕКТИВИ ЗАСТОСУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ТА ІДЕНТИФІКАЦІЇ КОРИСТУВАЧІВ КОМП'ЮТЕРА, ОПИСАНОЇ НА ОСНОВІ ДИСКРЕТНОЇ МОДЕЛІ.....	96
ЛЫСЕНКО Э.В., ПОНОМАРЕНКО В.П., ПИСКЛАКОВА В.П. СИСТЕМОЛОГИЧЕСКИЙ АНАЛИЗ ПРОБЛЕМЫ ПРИНЯТИЯ РЕШЕНИЙ В УСЛОВИЯХ МНОГОКРИТЕРИАЛЬНОСТИ И НЕОПРЕДЕЛЕННОСТИ.....	104
ВИШНЯК М.Ю., ДОВГАНЬ С.С. РАСШИРЕНИЕ ФУНКЦИЙ АСУ: УПРАВЛЕНИЕ ЗНАНИЯМИ.....	109
ОМРИ КАРИМ ПОСТРОЕНИЕ АЛГОРИТМА РАСПОЗНАВАНИЯ ОТПЕЧАТКОВ ПАЛЬЦЕВ ДЛЯ СИСТЕМЫ КОНТРОЛЯ ДОСТУПА.....	116
КРИВУЛЯ Г.Ф., РЯБЕНЬКИЙ В.М., ПЕТРЕНКО Л.П. МЕТОДИКА ФОРМИРОВАНИЯ МАТЕМАТИЧЕСКОЙ МОДЕЛИ СУММАТОРА В ФОРМАТЕ ТРОИЧНОЙ СИСТЕМЫ СЧИСЛЕНИЯ.....	121
КУЗЬМИН А. Я., ГОЛОВИЙ (ГУСАРЬ) Н.В., ДАЮБ Я. РЕАЛИЗАЦИЯ МОДЕЛИ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В ОБЛАСТИ СЕРВИСНОГО ОБСЛУЖИВАНИЯ БАНКОМАТОВ.....	134
ВЕРЕЩАК И.А. МЕТОД НИЗКОУРОВНЕВОЙ ОБРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ.....	139
РЕФЕРАТИ	143

ПОДХОД К ПРОЕКТИРОВАНИЮ ХРАНИЛИЩ ДАННЫХ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

Анализируются основные проблемы, возникающие в процессе разработки хранилищ данных. Выделяются главные подходы к моделированию хранилищ данных. Рассматривается подход, позволяющий автоматизировать процесс проектирования схемы хранилища данных. Предлагается метод синтеза схемы хранилища данных типа «звезда» исходя из особенностей представления схемы источника данных.

1. Введение

По мере развития информационных систем, которые эксплуатируются на современных предприятиях, технологии и средства хранения данных претерпевают существенные изменения. Одним из этапов этого изменения является создание и эксплуатация активных хранилищ данных (ХД). Основной особенностью традиционных ХД является организация периодического выполнения пакетных работ, которые направлены на извлечение из операционных данных неких больших выборок исторических данных, их очистку, преобразование и загрузку в ХД. Активное ХД требует постоянного обнаружения и доставки в него данных в реальном времени или в режиме времени, близком к реальному, из баз данных информационных систем, функционирующих на предприятии [1].

Факторами, стимулирующими развитие активных ХД, являются растущие потребности клиентов, в частности, к скорости предоставления услуг, а также ужесточение нормативных требований к поддержке исторической информации о бизнес-процессах, которые осуществляются на предприятии. Внедрение и грамотная эксплуатация активных ХД может сыграть ключевую роль в реализации всей стратегии выполнения бизнес-процессов предприятия в целом. Расширяя возможности применения ХД до уровня сотрудников, занимающихся операционной деятельностью, поставщиков и клиентов, предприятие получает возможность принятия научно обоснованных решений путем увеличения объема хранимых данных. Поскольку информация является ресурсом, ценность которого зависит от времени, ее ценность для бизнеса прямо пропорциональна ее своевременности и достоверности. Поэтому проблема разработки методов и технологий проектирования активных ХД на основе моделей эксплуатируемых баз данных является актуальной.

2. Анализ современных достижений в области разработки хранилищ данных

Основными принципами проектирования ХД в настоящее время являются простота (с точки зрения конечных пользователей) и производительность (оправдывающая ожидания конечных пользователей). Дополнительными аспектами, учет которых желателен при проектировании, являются накладные расходы (на само проектирование, на повседневное администрирование и т.д.), стоимость (персонала, программных и аппаратных средств), риски неполучения требуемых результатов, риски чрезмерной централизации.

Эти принципы и аспекты положены в основу современных подходов к моделированию хранилищ данных. Указанные подходы можно разделить на две основные группы:

- подходы, основанные на традиционном ER-моделировании;
- подходы, основанные на многомерном моделировании данных.

Использование подходов первой группы позволяет наиболее полно описать предметную область предприятия и формализовать процесс хранения исторической информации. Подходы второй группы позволяют достаточно быстро построить модель для анализа различных показателей, однако обладают рядом недостатков, таких как принятие компромиссных решений в ходе выбора способа хранения исторических данных и т.п. [2, 3].

3. Выделение нерешенной части проблемы и постановка задачи исследования

В то же время использование существующих подходов к разработке ХД порождает целый ряд проблем. Среди этих проблем как основные можно указать следующие [2, 3]:

- в ХД оказываются недоступными данные, необходимые для принятия решений;
- недостаток партнерских отношений между конечными пользователями и специалистами в области информационных технологий, возникающий в процессе проектирования ХД;
- отсутствие явной познавательной и концептуальной модели конечных пользователей, определяющих их точки зрения на разрабатываемое ХД;
- запаздывание экспорта в ХД данных, требуемых для принятия решений;
- недостаточная подробность данных, хранимых в ХД;
- неудобные форматы данных, определенные в процессе разработки ХД;
- слишком медленная доставка данных из ХД конечным пользователям;
- низкое качество данных, сберегаемых в ХД;
- чрезмерная затрата сил и времени на создание модели данных разрабатываемого ХД.

Одним из основных симптомов возникновения ошибок в процессе создания и эксплуатации активных ХД является появление тысяч сущностей, которые никогда не наполняются реальными данными. Этот симптом является следствием распространенного мнения, будто усилия, затрачиваемые на разработку модели данных, только задерживают работу по созданию активного ХД. Согласно этому мнению предпочтительней разрабатывать модель активного ХД на основе моделей реальных источников данных, заранее рассчитывая на то, что ошибки и несоответствие данных будут выявляться непосредственно в ходе выполнения процедур физической реализации и эксплуатации активного ХД.

Таким образом, решение задачи автоматизации моделирования активных ХД является актуальным и позволит добиться как сокращения времени разработки активных ХД, так и повышения качества создаваемого ХД за счет отказа от создания избыточных сущностей на ранних этапах разработки активного ХД.

4. Изложение материала исследования

Рассматриваемый в этой работе подход к решению задачи автоматизации моделирования активных ХД предполагает использование в качестве исходных данных моделей структур источников данных. В общем случае такими источниками являются нормализованные схемы баз данных информационных систем, которые эксплуатируются на предприятии. В результате модели источников данных не имеют ярко выраженного центра (рис. 1), что затрудняет их преобразование к представлению многомерного ХД.

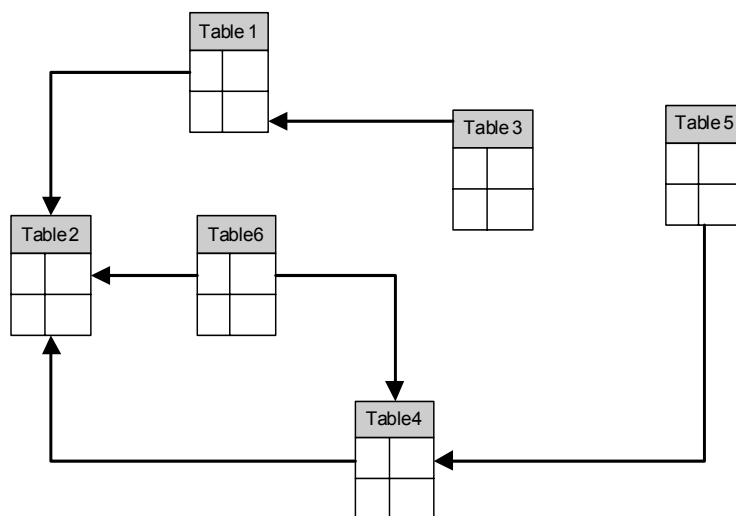


Рис. 1. Пример схемы нормализованной структуры данных

Результатом решения задачи следует признать модель активного ХД, построение которой основывается на использовании принципов многомерного моделирования данных. При этом в качестве основной схемы ХД предлагается выбрать схему типа «звезда» (рис. 2).

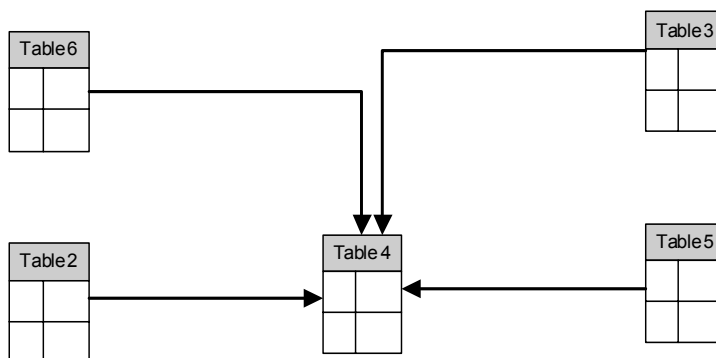


Рис. 2. Пример ненормализованной многомерной структуры хранилища данных типа «звезда»

Такой выбор объясняется следующими причинами. Во-первых, схема типа «звезда» обеспечивает реализацию многомерных ХД (в том числе и активных ХД) в большинстве современных корпоративных систем управления базами данных. Во-вторых, эта схема обеспечивает высокое быстродействие разрабатываемого ХД. В-третьих, схема типа «звезда» частично совпадает со способом восприятия и использования данных конечными пользователями ХД, что позволяет разработчикам и конечным пользователям без труда управлять метаданными. В-четвертых, указанную схему можно модифицировать и развивать по мере роста потребности в решении задач анализа, прогноза и управления на конкретном предприятии.

Для получения структуры данных типа «звезда» (см. рис. 2) предлагается метод, состоящий из следующих этапов.

Этап 1. ER-диаграмма структуры данных источника представляется в виде ориентированного графа, вершины которого описывают сущности исходной ER-диаграммы, а дуги описывают связи, выделенные между этими сущностями.

Этап 2. Каждой вершине сопоставляется кортеж из двух элементов. Первый элемент этого кортежа описывает количество дуг, входящих в вершину. Второй элемент описывает количество дуг, исходящих из вершины. Пример результата выполнения этапов 1 и 2 для ER-диаграммы, показанной на рис. 1, приведен на рис. 3.

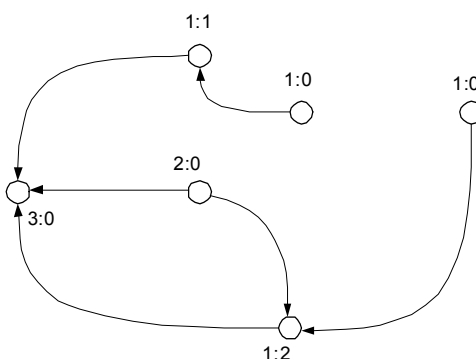


Рис. 3. Представление схемы данных источника в виде графа

Этап 3. Выбирается вершина графа, у которой разница между количеством входящих дуг и количеством исходящих дуг максимальна. Данная вершина определяет сущность, которая будет использоваться как основа таблицы фактов разрабатываемого ХД.

Этап 4. Приведение графа к виду, показанному на рис. 4, путем выполнения над исходными сущностями стандартного алгоритма соединения (JOIN).

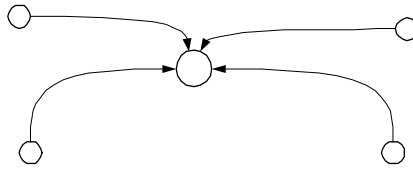


Рис. 4. Представление графа, определяющего многомерную структуру хранилища данных типа «звезда»

5. Выводы

Предложенный метод проектирования ХД позволяет упростить и формализовать процедуру построения схем данных ХД типа «звезда». Реализация этого метода позволит создать принципиально новые информационные системы управления крупными и средними предприятиями и организациями, которые будут основаны на оперативном мониторинге экономических процессов предприятия и управлении этими процессами в режиме времени, близком к реальному.

В то же время для детализации и реализации рассмотренного метода в рамках специальных информационных технологий необходимы дальнейшие исследования в области интеграции разнородных БД, модели данных которых могут быть заранее неизвестны. Решение такой задачи традиционными методами чрезвычайно затруднено, поэтому рекомендуется рассмотреть возможность решения данной проблемы с использованием математического аппарата динамической теории информации.

Список литературы: 1. *Черняк Л.* На пути к предприятию, управляемому в реальном времени // Открытые системы. 2002. № 12. С. 43-47. 2. *Кузнецов С.Д.* Базы данных: языки и модели. Учебник. М.: ООО «Бином-Пресс», 2008. 720 с. 3. *Крѣнке Д.* Теория и практика построения баз данных. 9-е изд. СПб.: Питер, 2005. 859 с.

Поступила в редколлегию 14.11.2008

Евланов Максим Викторович, канд. техн. наук, доцент кафедры ИУС ХНУРЭ. Научные интересы: проблемы эволюционного проектирования информационных управляющих систем. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-451.

Слипченко Елена Викторовна, канд. техн. наук, ст.н.с., в.н.с. ПНИЛ АСУ ХНУРЭ. Научные интересы: информационные технологии моделирования и управления бизнес-процессами предприятий. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-890.

Никитюк Владимир Анатольевич, студент гр. ИУСТм-08-1 ХНУРЭ. Научные интересы: применение технологии Windows Workflow Foundation для моделирования и отображения данных в информационных системах управляющих систем. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-451.

ИССЛЕДОВАНИЕ МЕТОДОВ РЕШЕНИЯ ЗАДАЧ ИНЖИНИРИНГА ТРАФИКА В СЕТЯХ СЛЕДУЮЩЕГО ПОКОЛЕНИЯ

Рассматриваются процедуры инжиниринга трафика в сетях следующего поколения (ССП) и проблемы обеспечения их эффективности. Анализируется модель инжиниринга трафика на базе технологии MPLS и расширений протоколов маршрутизации по алгоритму состояния связей в целях определения ее узких мест, для устранения которых предлагается использование системы AntNet как дополнительной структуры данных. Формулируются цели и задачи дальнейших исследований для практической реализации расширения технологии MPLS на базе системы AntNet.

Введение

Одной из особенностей сети следующего поколения (ССП) является одновременная циркуляция в ней множества разнотипных потоков, каждый из которых требует безусловного соблюдения одних параметров передачи и допускает некоторые уступки по другим. Например, в случае возникновения перегрузок в ССП для одного потока может быть урезана полоса пропускания, для другого – увеличено время доставки, для третьего – опущены требования к целостности передаваемых данных. Как следствие, требования к эффективности использования сетевого транспорта в ССП возрастают и, в частности, включают в себя необходимость решения задач выбора многокритериального оптимального пути и сокращения числа узких мест. Так как в качестве одной из составляющих транспортной платформы ССП используются традиционные IP-сети, не включающие в себя механизмы оптимизации производительности, вопросы повышения эффективности передачи информации остаются по-прежнему актуальными. Более того, традиционные IP-сети ориентируются на коммуникационную модель «по мере возможности», когда в процессе маршрутизации для выбора кратчайшего пути используются простые параметры, такие как число промежуточных узлов или задержка, с помощью которых невозможно выполнить сбалансированное распределение трафика в реальном режиме времени, что приводит к возникновению таких проблем, как несбалансированность нагрузки, осцилляция маршрутов или «проблема рыбы».

Таким образом, обеспечение эффективного управления трафиком в ССП в настоящее время является актуальной задачей, которая может быть выбрана в качестве базового направления дальнейших исследований в области развития технологий ССП.

Проблемы традиционной маршрутизации IP

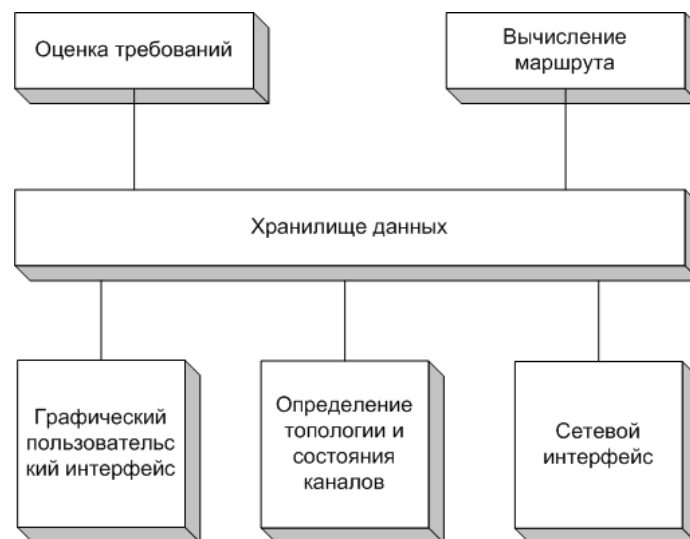
По своей природе маршрутизация ориентирована на получателя, следовательно, при передаче по протоколу IP все пакеты, адресаты которых имеют один и тот же префикс, будут отправляться на следующий общий для них промежуточный узел. Маршрутизация в соответствии с адресатом не всегда дает возможность задействовать все разнообразные соединения, существующие в сети. Распределение трафика обычно предсказать очень сложно, и зачастую оно несбалансированно. В силу этого ресурсы на многих участках сети могут использоваться недостаточно эффективно. Например, при наличии нескольких маршрутов между узлами выбранный кратчайший путь может оказаться перегруженным, что приведет к возрастанию задержек передачи данных, тогда как альтернативный путь не задействуется вообще.

Вторая особенность маршрутизации IP состоит в том, что маршруты рассчитываются с учетом локальной оптимизации. Поскольку каждый узел выбирает путь со своей «точки зрения», в рамках всей сети такой выбор может оказаться неоптимальным. Кратчайший путь может оказаться самым напряженным, поскольку множество узлов в сети могут выбрать его независимо друг от друга. В данном случае длинный путь может оказаться

более эффективным. Чтобы оптимизировать использование ресурсов в масштабах всей сети, решение о маршрутизации должно приниматься с учетом назначения всей сети и общего представления о ней. В исключительных случаях маршрутизация по кратчайшему пути может привести к осцилляции маршрутов. Протоколы маршрутизации работают с системами, где имеют место значительные задержки (так называемые системы с обратной задержанной связью). Допустим, некоторый узел, оценив метрику нескольких маршрутов, принял решение перейти на новый маршрут. Если он это сделает, окружающие маршрутизаторы узнают об этом с заметной задержкой и соответствующим образом изменят свои оценки метрик фрагментов пути. Когда (снова с задержкой) исходный маршрутизатор получит информацию по этим оценкам, может случиться, что старый маршрут окажется предпочтительнее нового, и рассматриваемый процесс, теоретически, может продолжаться до бесконечности.

Система инжиниринга трафика

В настоящее время для решения задачи обеспечения эффективного управления трафиком в ССП задействуется технология инжиниринга трафика (Traffic Engineering). Под инжинирингом трафика понимают методы и механизмы сбалансированной загрузки всех ресурсов сети, а также быстрого восстановления маршрутов после сбоя за счет рационального выбора пути прохождения трафика через сеть. Система инжиниринга трафика (рисунок) состоит из следующих основных компонентов: выявление топологии/состояния, вычисление маршрута, оценка требований трафика, графический пользовательский интерфейс, сетевой интерфейс и модуль хранилища данных.



Структурная схема системы инжиниринга трафика

Модуль хранилища данных системы инжиниринга трафика состоит из центральной базы данных и постоянной памяти для всех разделяемых объектов данных, таких как топология сети, информация о состоянии канала, требования трафика, маршруты и правила. Другие модули могут хранить информацию, получать к ней доступ и обмениваться ею через базу данных. Кроме того, каждый объект имеет соответствующий конечный автомат, где каждый переход связан с модулем-отправителем, входным событием, модулем-получателем и выходным событием.

Эффективной системе инжиниринга трафика необходима точная информация о требованиях трафика, которая может быть получена либо из соглашений об уровне сервиса (Service Level Agreement, SLA), либо из статистического анализа параметров трафика. После определения системой инжиниринга трафика оптимальных маршрутов необходимо сконфигурировать элементы сети так, чтобы эти требования выполнялись.

Двухстороннее взаимодействие системы инжиниринга трафика и сетевых элементов может выполняться либо через Web-интерфейсы, либо посредством использования протокола SNMP или службы COPS.

Требования и задачи инжиниринга трафика

Согласно [1], управляющие действия системы инжиниринга трафика должны включать:

- модификацию параметров управления трафиком,
- модификацию параметров, связанных с маршрутизацией,
- модификацию атрибутов и констант, связанных с ресурсами.

Уровень человеческого вмешательства в процесс управления трафиком должен быть минимизирован. Это может быть реализовано путем автоматизации операций (распределенных и масштабируемых), описанных выше.

В модели “оптимальных усилий” для Интернет-сервиса ключевая задача управления трафиком включает в себя: минимизацию потерь пакетов и задержек, оптимизацию пропускной способности и согласование наилучшего уровня услуг. Статистически заданные характеристики трафика (разброс времени доставки пакетов, вероятность потери и максимальное время доставки) становятся важными в дифференцированных услугах Интернет. Одним из подходов решения таких проблем является оптимизация использования всех имеющихся ресурсов сети. В частности, желательно гарантировать, чтобы поднаборы сетевых ресурсов не были перегружены, в то время как аналогичные ресурсы на альтернативных маршрутах недогружены. Полоса пропускания является критическим ресурсом современных сетей. Следовательно, центральной функцией управления трафиком является эффективное управление пропускной способностью, в том числе и минимизация перегрузок. Здесь речь идет не о кратковременных перегрузках, а о долгосрочных, влияющих на поведение сети в целом. Перегрузка обычно проявляется двояко:

- когда сетевых ресурсов недостаточно или они не соответствуют существующей загрузке;

- когда потоки трафика неэффективно распределены по имеющимся ресурсам.

Первый тип проблем перегрузки может быть решен путем:

- расширения ресурса,
- применения классических средств управления перегрузкой,
- сочетания этих подходов.

Классическое управление перегрузкой нацелено на регулирование уровня потребности путем снижения его до имеющегося в распоряжении уровня ресурсов. Классическое управление перегрузкой включает в себя: ограничение потока, управление шириной окна для потока, управление очередями в маршрутизаторе, диспетчеризацию и т.д.

Второй тип проблем перегрузки, связанный с неэффективным размещением ресурсов, может быть решен посредством управления трафиком.

Вообще, перегрузка, связанная с неэффективным размещением ресурсов, может быть уменьшена с помощью политики балансировки нагрузки в различных фрагментах сети. Задачей таких стратегий является минимизация максимальной перегрузки или, напротив, минимизация максимума использования ресурса. Когда перегрузка минимизирована путем оптимального размещения ресурсов, потери пакетов и задержка доставки падают, а совокупная пропускная способность возрастает. Таким образом, восприятие конечным пользователем качества сетевого обслуживания становится лучше. Одним из вариантов практической реализации процедур инжиниринга трафика является протокол MPLS, широко используемый для передачи данных в сетях следующего поколения. Далее будет рассмотрена модель балансировки нагрузки в MPLS.

MPLS и инжиниринг трафика

Модель балансировки нагрузки представляет собой наведенный MPLS-граф, аналогичный виртуальной топологии в модели наложений. Модель наложений – полная сеть логических соединений между всеми пограничными узлами. Логические соединения – заданные явным образом точные маршруты, реализуемые путем резервирования пропускной способности и организованные в физической сети таким образом, чтобы распределение трафика по всем магистралям в сети было сбалансированным.

Наведенный MPLS-граф состоит из набора LSR [2], которые представляют собой вершины графа, и набора LSP [2], которые предоставляют логические соединения «точка-точка» между указанными LSR и, следовательно, служат в качестве дуг наведенного графа. Формируется иерархический наведенный MPLS-граф, базирующийся на концепции стеков меток [2], и в рамках данной модели базовые проблемы управления полосой пропускания в MPLS определяются эффективностью совмещения наведенного MPLS-графа с физической топологией сети.

Пусть $G = (V, E, c)$ является графом, отражающим физическую топологию сети. Здесь V – набор узлов сети и E – набор каналов; т.е. для v и w из V объект $(v, w) \in E$, если v и w являются непосредственно связанными в рамках G . Параметр “ c ” представляет собой набор емкостей и других ограничений, сопряженных с E и V . Далее G будет рассматриваться как “основа” сетевой топологии.

Пусть $H = (U, F, d)$ является ориентированным наведенным MPLS-графом, где U является поднабором V , представляющим набор LSR в сети, или более точно набор LSR, которые являются конечными точками по крайней мере одного LSP. Здесь F представляет собой набор LSP, так что для x и y из U объект $(x, y) \in F$. Параметр “ d ” представляет собой набор требований и ограничений, ассоциированных с F . Таким образом, базовая проблема инжиниринга трафика сводится к установлению соответствия между наведенным MPLS-графом (H) и базовой топологией сети (G). Отсюда видно, что H зависит от переходных характеристик G , а эффективность наведенного MPLS-графа будет тем выше, чем больше управляемых логических соединений сформировано между пограничными узлами.

Несмотря на свою простоту и эффективность, наложенный подход может привести к определенным ограничениям масштабируемости (проблема « N в квадрате»). Чтобы сформировать все возможные логические соединения между N пограничными узлами, каждый из них должен установить LSP с $(N-1)$ других узлов, так что в итоге количество LSP будет равно $N \times (N - 1)$. В крупной сетевой магистрали, где существует множество граничных узлов, подобный подход не будет масштабироваться приемлемым образом.

Следовательно, задача поиска механизма, позволяющего сочетать алгоритмы выбора оптимального маршрута для множества LSR, что позволит ликвидировать указанное узкое место в наведенном MPLS-графе, с процедурами инжиниринга трафика, реализованными в протоколе MPLS, является достаточно актуальной.

Теория муравьев и инжиниринг трафика

Для решения задачи резервирования пропускной способности технология MPLS использует расширения протоколов маршрутизации, работающих на основе алгоритма состояния связей (OSPF, IS-IS). Данные расширения содержат новые типы объявлений о номинальной и незарезервированной пропускной способности каждой связи, а также параметры потоков данных, для которых необходимо определить оптимальные пути с учетом требований инжиниринга трафика [3]. Тем не менее, проблемы, связанные с динамической загрузкой каналов связи, описанные в разделе «Проблемы традиционной маршрутизации IP», остаются актуальными ввиду сложности параллельной обработки и оптимизации выбора путей для всех потоков данных, возрастающей прямо пропорционально размерам транспортной платформы компьютерной сети. Таким образом, имеет место необходимость разработки параллельного алгоритма обработки маршрутной информации с условием динамической адаптации к изменяющимся характеристикам транспортной платформы сети.

Здесь для решения указанной задачи предлагается использование аппарата роевого интеллекта, хорошо зарекомендовавшего себя в областях, требующих решения задач нахождения кратчайших маршрутов к источнику, а также динамического перераспределения и оптимизации этих маршрутов.

Система маршрутизации на базе мобильных агентов AntNet [4] является модификацией системы муравьиных колоний (Ant Colony Systems, ACS). ACS – это метод оптимизации, где группа искусственных муравьев (агентов) обходит граф, представляющий собой объект оптимизации. Агенты собирают и сохраняют информацию о каждом узле графа в парал-

тельном режиме, что позволяет перейти от локальной оптимизации к общей и сделать адекватный выбор наилучшего решения при заданных условиях. Данная концепция используется и в системе AntNet [5].

Создаются два набора мобильных агентов – «обычные муравьи» (forward ants) и «возвращающиеся муравьи» (backward ants). Эти агенты не связаны друг с другом напрямую и могут передавать свой опыт только через две информационные структуры, расположенные на каждом узле, а именно через:

– массив M_k , представляющий собой статистическую модель распределения трафика для каждого пункта назначения d , такой, какой эта модель видится из узла k ;

– таблицу маршрутизации, содержащую вероятностное значение P_{dn} для каждой пары узлов (d, n) . Это значение характеризует оптимальность выбора (по времени прохождения маршрута) следующего узла n при условии, что пункт назначения – это узел d .

Сначала в каждом узле сети s создается «обычный муравей» $F_{s \rightarrow d}$ со случайно выбранным узлом назначения d . Каждый агент выбирает следующий пункт своего назначения, используя информацию в таблице маршрутизации. Каждый следующий узел выбирается по случайной схеме с вероятностью P_{dn} , зависящей от оптимальности маршрута к соседнему, еще не посещенному узлу n . Если агент уже прошел через все соседние узлы, то следующий определяется случайным образом. Если выбранный маршрут на данный момент недоступен, агент ожидает доступа к линии связи в очереди с другими пакетами данных. Идентификатор каждого посещенного узла k и время, затраченное на перемещение к этому узлу $T_{s \rightarrow k}$, заносится в память агента. Если происходит циклическое перемещение, т. е. если агент попадает на уже посещенный им узел, все данные об узлах внутри цикла обнуляются.

Когда «обычный муравей» $F_{s \rightarrow d}$ достигает узла назначения d , он создает «возвращающегося муравья» $B_{d \rightarrow s}$, передает ему свою память и погибает. «Возвращающийся муравей» проделывает тот же самый путь, но в обратном направлении. Прибыв на узел k с предыдущего узла f , агент обновляет массив M_k значениями времени $T_{k \rightarrow i}$, затраченного на путешествие с узла k на каждый посещенный узел i по пути $k \rightarrow d$. Он также вносит изменения в таблицу маршрутизации, повышая вероятность P_{if} , относящуюся к каждой паре узлов i и f , в соответствии с $P_{if} \leftarrow P_{if} + (1 - r') \cdot (1 - P_{if})$, где r' – оценочный параметр, показывающий, насколько хорошо (мало) время обхода маршрута T , которое наблюдалось в среднем вплоть до искомого узла, и понижая для нормализации вероятности P_{in} остальных соседних узлов n в соответствии с $P_{in} \leftarrow P_{in} - (1 - r')P_{in}; \forall j \in N_k, f \neq n$. Время $T_{k \rightarrow i}$, затраченное на прохождение маршрута агентом $F_{s \rightarrow d}$, используется для определения коэффициента повышения.

Таким образом, в любой момент времени параллельно решаются две задачи: нахождение оптимальных маршрутов по вероятностной таблице маршрутизации, а также динамическое перераспределение и оптимизация этих маршрутов за счет постоянного обновления информации на каждом узле при каждом посещении им агента. Более того, если какой-либо участок пути оказывается перегруженным, то прокладывается обходной маршрут, пусть и более длинный. Когда же трафик уменьшается, все агенты вновь начинают проходить по оптимальному пути.

Постановка задачи и дальнейшие исследования

В свете сказанного выше возникает необходимость в практической реализации расширения технологии MPLS на базе системы AntNet как альтернативы расширениям протоколов маршрутизации, работающих на основе алгоритма состояния связей, чему и будут посвящены дальнейшие исследования. Для этого необходимо решить следующие задачи:

– разработать модель инжиниринга трафика путем сочетания структур данных MPLS и AntNet;

– выполнить имитационное моделирование в целях анализа эффективности предложенной модели в сравнении с используемой в настоящее время парой «MPLS-расширение алгоритма состояния связей»;

- выполнить программную реализацию предложенного расширения MPLS на базе AntNet;
- выполнить натурное моделирование в целях определения значений показателей эффективности и производительности предложенной модели.

Выводы

Рассмотрены основные задачи инжиниринга трафика и методы их решения, указаны недостатки и узкие места рассмотренных методов. Так, для технологии MPLS ее узкими местами являются экспоненциально возрастающая сложность масштабируемости маршрутов («проблема N²»), а также использование для решения ряда задач инжиниринга трафика, в частности, резервирования пропускной способности, расширений протоколов маршрутизации классической IP-сети, в результате чего остаются актуальными проблемы, связанные с отсутствием параллельного алгоритма обработки маршрутной информации с условием динамической адаптации к изменяющимся характеристикам транспортной платформы канала связи.

В целях устранения указанных проблем авторами предложено использование системы AntNet, базирующейся на аппарате роевого интеллекта, как альтернативы расширениям протоколов маршрутизации IP-сетей. Алгоритм маршрутизации AntNet отвечает требованиям параллельности, динамической адаптации и, как следствие, обеспечивает сбалансированную загрузку каналов связи.

Сформулированы цели и задачи исследований, решение которых сводится к практической реализации расширения технологии MPLS на базе системы AntNet и анализу эффективности предложенного расширения.

Список литературы: 1. *Awduche D.* Requirements for Traffic Engineering Over MPLS, RFC-2702 - <http://www.rfc-editor.org/rfc/rfc2702.txt>. 2. *Олейн В.* Структура и реализация современной технологии MPLS. Руководство Cisco. Киев: Диалектика-Вильямс, 2004. 480 с. 3. *Гулевич Д.С.* Сети связи следующего поколения - www.intuit.ru/forum2site/course/ndnets. 4. *Gianni Di Caro, Marco Dorigo.* AntNet: Distributed Stigmergetic Control for Communications Networks // Journal of Artificial Intelligence Research. 1998. №9. P. 317-365. 5. *Острогляд Е.* Методики дрессировки “цифровых муравьев” - <http://ko-online.com.ua/node/10475>.

Поступила в редколлегию 09.12.2008

Бабич Анна Витальевна, канд. техн. наук, доцент кафедры АПВТ ХНУРЭ. Научные интересы: сетевые технологии, технологии дистанционного образования. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.

Кудина Марина Владимировна, аспирантка кафедры АПВТ ХНУРЭ. Научные интересы: сетевые технологии, компьютерная вирусология. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.

Емельянов Игорь Валерьевич, аспирант кафедры АПВТ ХНУРЭ. Научные интересы: сетевые технологии, диагностика и оптимизация компьютерных систем и сетей. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.

ФОРМАЛЬНАЯ СЕМАНТИКА СЛОЖНЫХ ОПЕРАТОРОВ ЛИНЕЙНОЙ ТЕМПОРАЛЬНОЙ ЛОГИКИ

Рассматривается проблема нечеткости интерпретации сложных операторов линейной темпоральной логики в применении к анализу цифровых систем в рамках динамических методов верификации. Вводятся формальные определения и спецификации ключевых вычислительных процедур рабочего цикла верификации, ориентированные на максимальное быстродействие анализа.

1. Введение

Любая система на кристалле является параллельной. В общем случае под параллельными понимаются программно-аппаратные системы с несколькими параллельными процессами, которые непрерывно обмениваются сообщениями с внешней средой. Большинство цифровых аппаратных систем имеют конечное число состояний, определяющееся набором достижимых значений всех сигналов. В отличие от аппаратных систем, большинство программных систем, где применяются динамические структуры данных, имеют бесконечное число состояний, что значительно усложняет процессы их верификации [1,2].

Задача верификации параллельных систем состоит в доказательстве наличия или отсутствия в их работе определенных последовательностей состояний, называемых ограничениями. Если число состояний конечно, представляется возможным формальное доказательство удовлетворения системой заданных ограничений.

Очевидно, анализ работы параллельных систем, также как и анализ ограничений, предполагает формальные рассуждения об изменении состояния системы с течением времени. Формальное утверждение о том, что система всегда или в выбранном интервале времени удовлетворяет заданному логико-временному ограничению последовательностей состояний, называется темпоральным утверждением или темпоральной *ассерцией*. Необходимы формальные механизмы для моделирования параллельных систем и механизмы описания ассерций.

Темпоральная логика позволяет задать порядок событий во времени без внесения понятия времени в явном виде. Применение темпоральной логики для описания ограничений параллельных систем показало свою эффективность при специфицировании требований и верификации цифровых систем.

С точки зрения цели специфицирования различают два принципиальных вида темпоральных ограничений, которые можно задавать при помощи утверждений темпоральной логики [2]:

– *инвариантные* ограничения (safety) – заданная последовательность состояний никогда не случится (например, ресурс никогда не будет использован двумя процессами одновременно);

– *достижимые* ограничения (liveness) – желаемая последовательность состояний будет достигнута (например, пришедший запрос когда-нибудь обязательно будет обработан).

Помимо цели, темпоральные ограничения могут дополнительно характеризоваться *требованиями допустимости* (fairness constraints). Допустимость является требованием к множеству разрешенных в системе вычислительных путей. Характеристика допустимости является важным фактором для путей бесконечной длины, возникающих при циклических вычислительных процедурах. Различают три вида требования допустимости:

– *безусловная* допустимость (unconditional fairness) – для всех переходов независимо от их вхождения в состав выбранного пути каждый переход выполняется бесконечно часто;

– *слабая условная* допустимость (weak fairness) – некоторые из переходов, включенных в состав выбранного пути, будут выполнены бесконечно часто;

– *сильная условная* допустимость (strong fairness) – все переходы, включенные в состав выбранного пути, будут выполнены бесконечно часто.

Существует два основных подхода к решению задачи верификации цифровых систем на основе темпоральных ассерций: формальные методы [3] (математическое доказательство удовлетворения системой инвариантных и достижимых ограничений без тестовых воздействий) и динамические методы [4] (проверка выполнения системой заданных ограничений во время имитационного моделирования, или симуляции, под воздействием тестовых стимулов). Формальные методы гарантируют 100% точность результатов верификации, однако их применение сопряжено со значительными вычислительными затратами в связи с экспоненциальной сложностью. На практике, несмотря на отсутствие 100% гарантии результатов, более распространено применение динамических методов, более приспособленных к проверке работоспособности системы в основных операционных режимах.

Рассуждения о бесконечных вычислительных путях в рамках процесса динамической верификации системы имеют условный характер. Разумеется, число шагов верификации в системе с тестовыми воздействиями определяется длиной теста, и не может идти речи о фактической бесконечности вычислительного пути. В этом контексте ограничения допустимости следует интерпретировать относительно момента завершения тестирования.

Типичная математическая база анализа ассерций во время симуляции, состоящая в трансформации операторов LTL-логики к детерминированным конечным автоматам, лишь частично способна удовлетворить основные требования к верификации на основе ассерций [5]. Основной проблемой классической модели является экспоненциальный рост количества состояний в зависимости от структурной сложности реализуемой темпоральной формулы. Кроме того, накладывается существенное ограничение на языки описания ассерций в виде простого подмножества, поскольку классическая модель неспособна осуществить проверку ряда темпоральных конструкций с приемлемой для практического применения вычислительной сложностью. Поддержка режима глобально-во времени практически невозможна на основе модели конечных автоматов.

Очевидно, для обеспечения существенно лучшей производительности анализа ассерций и поддержки темпоральных операторов, определяемых для бесконечных вычислительных путей, необходима принципиально иная модель. Такими свойствами обладает ранее предложенная модель динамических регистровых очередей (DRTLQ – Dynamic Register-Transfer-Level Queues) [6]. Модель ориентирована на высокопроизводительный анализ элементов линейной темпоральной логики и эффективность обнаружения и локализации нарушений [7]. Ключевые понятия, вводимые в модели (последовательностные функции, темпоральные свойства), представляют собой обобщенную надстройку над общеизвестными терминами LTL и языков описания ассерций [8].

Цель данной работы – формальное определение семантики сложных операторов линейной темпоральной логики в контексте их реализации в модели DRTLQ, осуществляющей высокопроизводительный анализ ассерций в рамках динамических методов верификации цифровых систем, а также создание формальной спецификации вычислительных процедур для наиболее сложных элементов модели DRTLQ.

К основным *задачам* исследования относятся:

– уточнение ранее предложенной модели процесса анализа темпоральных формул в рамках динамических методов верификации (под управлением HDL-симулятора);

– формальное определение семантики сложных операторов LTL-логики последовательностного и темпорального уровня, призванное обеспечить строгую однозначность интерпретации формул при реализации системы верификации на основе ассерций;

– создание формальной спецификации ключевых вычислительных процедур цикла работы последовательностных функций и темпоральных свойств в различных режимах интерпретации времени, ориентированных на достижение максимального быстродействия верификации.

2. Цикл работы ассерционного процесса в модели DRTLQ

Наиболее общим понятием модели DRTLQ является ассерционный процесс – совокупность элементов булевого, темпорального и верификационного уровня семантики ассерций, реализующая верификацию логически связанной группы темпоральных утверждений. Самым близким к ассерционному процессу языковым термином является единица верификации (verification unit) из языка PSL. Ассерционным процессом AP в модели DRTLQ называется целостная совокупность множества верификационных переменных B , списка текущих потоков активации Ω , очереди последовательностных функций F и темпоральных свойств P , одного ассерционного монитора M , как правило, относящаяся к определенному функциональному блоку системы и реализующая верификацию логически связанной группы темпоральных утверждений в процессе моделирования:

$$AP = \{B, \Omega, F, P, M\}. \quad (1)$$

В моменты изменения верификационные переменные создают события:

$$e = \{e_{\leftarrow}, e_{\rightarrow}, e_{\downarrow}, \rho, t_b, t_a\}, \quad (2)$$

где $e_{\leftarrow}, e_{\rightarrow}, e_{\downarrow}$ – связи события с другими событиями; ρ – характеристический вектор события; t_b и t_a – время создания и активации события соответственно, измеряемое в количестве тактовых событий с момента начала моделирования.

Событие называется активированным, если в ходе обработки оно достигло элемента ассерционного процесса, создающего поток активации. Поток (или кольцо) активации $\omega_k \in \Omega$ в модели DRTLQ – динамическое множество всех обрабатываемых событий $e_k^i, i \geq 0, k \geq 0$, соответствующих одному вычислительному пути π , который начинается с тактового цикла с порядковым номером k .

Событие может быть связано с другими событиями в двух различных видах цепочек. Правая цепочка r_{\rightarrow} , формируемая двунаправленными событийными связями $e_{\rightarrow}, e_{\downarrow}$, соответствует событиям, связанным одним кольцом активации. Основная цель связывания событий в правые цепочки состоит в возможности каскадных операций над всем потоком (например, уничтожение при завершении потока). Левая однонаправленная цепочка r_{\leftarrow} , формируемая событийными связями e_{\leftarrow} , соответствует событиям различных колец активации, которые одновременно транспортируются через выбранный элемент модели. Правая цепочка неупорядочена, она формируется свободно, здесь порядок не играет никакой роли, а левая цепочка упорядочена по времени создания события, что принципиально важно для конвейерной обработки нескольких потоков активации.

Характеристический вектор события ρ – короткое двоичное слово, транспортируемое в составе события, индивидуальные биты которого моделируют семантические флаги-модификаторы, оказывающие влияние на обработку события. Важнейшие флаги-модификаторы приведены в табл. 1.

Таблица 1

Обозначение	Описание
ρ^{VAL}	Текущее значение события
ρ^{LDEAD}	Событие должно быть отделено от левой цепочки
ρ^{RDEAD}	Событие должно быть отделено от правой цепочки
ρ^{ACTIVE}	Событие активировано
ρ^{RTOP}	Событие является вершиной потока активации
ρ^{ABORT}	Событие исходит от элемента прерывания анализа

Транспортирование событий осуществляется путем перегруппировки левых цепочек. Левые цепочки ассоциируются с элементами модели более высокого уровня, формирующими пути транспортирования событий. Пусть $r_{\leftarrow}^1 : \{e_1^1 \cdots e_{N_1}^1\}$ представляет собой левую цепочку, из которой транспортируются события, а $r_{\leftarrow}^2 : \{e_1^2 \cdots e_{N_2}^2\}$ – левую цепочку, к которой они направляются. Суть задачи простого (а) и копирующего транспортирования (б):

$$\begin{aligned} \text{а) } r_{\leftarrow}^{\prime 2} &: r_{\leftarrow}^2 \cup r_{\leftarrow}^1, \\ \text{б) } r_{\leftarrow}^{\prime 2} &: \kappa(r_{\leftarrow}^2) \cup \kappa(r_{\leftarrow}^1) \end{aligned} \quad (3)$$

состоит в корректном расширении цепочки r_{\leftarrow}^2 событиями из цепочки r_{\leftarrow}^1 с сохранением относительного порядка событий в результирующей цепочке $r_{\leftarrow}^{\prime 2}$:

$$\begin{cases} e_{\leftarrow}(e_0^{\prime 2}) = \varepsilon; \\ \forall k, 1 \leq k < |r_{\leftarrow}^{\prime 2}| - 1, e_{\leftarrow}^{\prime 2}(e_{k-1}^{\prime 2}) = e_{\leftarrow}(e_k^{\prime 2}) \Rightarrow t_b(e_{k-1}^{\prime 2}) < t_b(e_k^{\prime 2}). \end{cases} \quad (4)$$

Слияние цепочек событий происходит начиная с правого края с одновременным итерированием в единственно допустимом направлении, при этом не теряется позиция вставки, что обеспечивает линейную сложность алгоритма относительно суммы $|r_{\leftarrow}^1| + |r_{\leftarrow}^2|$.

Вставка предполагает изменение связей e_{\leftarrow} непосредственно вносимого в цепочку события, а также события, следующего за ним. При копирующем транспортировании, оригинальные цепочки не модифицируются, в то время как простое транспортирование полностью очищает r_{\leftarrow}^1 и перезаписывает результатом $r_{\leftarrow}^{\prime 2}$ цепочку r_{\leftarrow}^2 .

Помимо основной задачи (3), осуществляется дополнительная задача элиминации лишних событий, руководствуясь двумя критериями:

1. Если $\rho^{\text{RDEAD}}(e_1^k) = 1$, т.е. вычислительный поток события уничтожен, событие должно быть отключено от левой цепочки и уничтожено. Гарантируется, что событие с установленным флагом ρ^{RDEAD} будет уничтожено при первой же попытке транспортирования, соответственно, можно утверждать, что для любого события целевой цепочки $\rho^{\text{LDEAD}}(e_2^k) = 0$.

2. Если для двух соседних событий в результирующей цепочке выполняется условие $t_b(e_{k-1}^{\prime 2}) = t_b(e_k^{\prime 2})$, а $\rho(e_{k-1}^{\prime 2}) = \rho(e_k^{\prime 2})$ (характеристические векторы идентичны), то одно из событий можно уничтожить.

Частным случаем задачи транспортирования (3) является случай, при котором $|r_{\leftarrow}^2| = 0$. Такое упрощение сводит задачу к переносу тех элементов цепочки r_{\leftarrow}^1 в результирующую цепочку $r_{\leftarrow}^{\prime 2}$, которые не попадают под действие описанных выше критериев элиминации. Этот процесс, аналогично полноценному транспортированию, характеризуется линейной сложностью, однако только относительно $|r_{\leftarrow}^1|$, и, в силу отсутствия необходимости редактирования событийных связей, имеет существенно лучшие показатели производительности.

Анализ ассерций в модели DRTLQ осуществляется по событию, доставка которого в ассерционный процесс является обязанностью родительской системы моделирования. Каждая итерация цикла работы всегда соответствует уникальному моменту времени. Ситуация, когда одному моменту времени соответствует более одной итерации анализа, исключается.

Итерации анализа разделяются на типы в зависимости от момента запуска:

– *регулярная* итерация – запускается по тактовому событию (если тактового события не определено, по событию изменения любой из множества верификационных переменных, принадлежащих ассерционному процессу);

– *итерация прерывания* – запускается при срабатывании условия прерывания (операторы *async_abort* и *sync_abort*), прерывает все активные вычислительные потоки путем обработки множеств внутренних событий элементов-очереди и элементов-репетиций;

– *финальная итерация* – запускается один раз в конце моделирования, транспортирует все цепочки событий непосредственно монитору на обработку с учетом требований условной допустимости (безусловная, слабая, сильная).

Финальная итерация и итерация прерывания предназначены только для поддержки специальных видов ассерционных языковых конструкций и сводится к итерированию подготовленных заранее отдельных элементов системы (очереди и репетиций) и обработке множеств их внутренних событий.

Регулярная итерация анализа ассерций включает в себя следующие шаги:

1. Считывание значений всех ассерционных переменных.
2. Запуск шага вычислительной процедуры каждой из последовательностных функций, в соответствии с очередью обработки:

$$\begin{cases} e_{\leftarrow}(e'_0{}^2) = \varepsilon; \\ \forall k, 1 \leq k < |r_{\leftarrow}'^2| - 1, e_{\leftarrow}'^2_{k-1} = e_{\leftarrow}(e'_k{}^2) \Rightarrow t_b(e'_{k-1}{}^2) < t_b(e'_k{}^2). \end{cases} \quad (4a)$$

3. Запуск шага обработки ассерционного монитора.

Очередь обработки последовательностных функций упорядочивает запуск всех элементов таким образом, чтобы осуществить полную обработку модели за один проход. Первыми выполняются функции-генераторы, порождающие события от ассерционных переменных. Далее выполняются все последовательностные функции, в соответствии с иерархией.

Ассерционный монитор непосредственно не контактирует с уровнем последовательностей, а обрабатывает события по запросу. Для обеспечения взаимодействия уровня последовательностей с ассерционным монитором, вершиной иерархии последовательностных функций, т.е. последним элементом в очереди на выполнение, используется служебная функция действия f_{ACTION} . Данная функция не имеет ни внутренних, ни выходных событий, поскольку ее единственной задачей является доставка поступившей на вход цепочки событий к ассерционному монитору. Функции действия применяются также для реализации режима глобального времени, однако в этом случае цепочки поступивших событий доставляются не монитору, а темпоральным свойствам.

Пусть r_{\leftarrow} – цепочка событий, транспортированных к монитору с последовательностного уровня. Независимо от типа монитора сначала проверяется статус активации каждого события: если $\rho^{\text{ACTIVE}}(e \in r_{\leftarrow}) \neq 1$, событие уничтожается без каких-либо нотификаций внешней среде. Дальнейшие шаги обработки событий на уровне монитора зависят от его типа.

Предположим, обрабатываемая ассерция моделирует инвариантное LTL-ограничение вида $\text{never}\{\dots\text{seq}\dots\}$. В таком случае монитор называется запрещающим, и принцип его работы следующий:

$$\forall e \in r_{\leftarrow}, \begin{cases} \rho^{\text{VAL}}(e) = 1 \Rightarrow \begin{cases} \text{report(failed);} \\ \forall e' \in \{e_{\uparrow}, e_{\downarrow}\} \text{ recursively } (e), \rho^{\text{RDEAD}}(e') \leftarrow 1; \end{cases} \\ \rho^{\text{VAL}}(e) = 0 \Rightarrow \begin{cases} \rho^{\text{LDEAD}}(e) \leftarrow \rho^{\text{RDEAD}}(e) \leftarrow 1; \\ e_{\uparrow}(e) = e_{\downarrow}(e) = e \Rightarrow \text{report(passed)}. \end{cases} \end{cases} \quad (5)$$

Если значение события $\rho^{\text{VAL}}(e) = 1$, это означает, что дочерняя последовательность выполнена успешно, а значит, ассерция нарушена. В таком случае весь поток активации ω , соответствующий времени активации t_a рассматриваемого события $e \in r_{\leftarrow}$, признается удовлетворенным, и дальнейший его анализ (могут существовать другие события этого потока) не имеет смысла. Все события, относящиеся к данному потоку, уничтожаются моментально, используя связи события $e_{\uparrow\rightarrow}, e_{\downarrow\rightarrow}$ по правой цепочке. Монитор сигнализирует внешней среде о неудаче анализа ассерции, вызывая абстрактную процедуру *report* с неудачным статусом.

Если же значение события $\rho^{\text{VAL}}(e) = 0$, это означает, что один из сценариев последовательности удовлетворяет свойству. В таком случае, событие просто уничтожается из всех цепочек. Если событие было последним в собственном потоке активации, монитор сигнализирует внешней среде об успехе анализа ассерции, на этот раз вызывая процедуру *report* с положительным статусом.

Работа разрешающего монитора, моделирующего инвариантное ограничение вида *always*({...seq...}), подобна запрещающему монитору. Различие состоит только в противоположности условия срабатывания: событие с отрицательным значением $\rho^{\text{VAL}}(e) = 0$ является в данном случае определяющим неудачный статус и досрочное завершение анализа потока активации. Напротив, $\rho^{\text{VAL}}(e) = 1$ считается успешным результатом, и, если такое событие последнее в потоке активации, то ассерция удовлетворяется на данном вычислительном пути.

Также существует третий тип монитора – ожидающий, соответствующий LTL-ограничениям достижимости вида *eventually!*({...seq...}) :

$$\forall e \in r_{\leftarrow}, \begin{cases} \rho^{\text{VAL}}(e) = 1 \Rightarrow \begin{cases} \text{report}(\text{passed}); \\ \text{recursively} \\ \forall e' \in \{e_{\uparrow\rightarrow}, e_{\downarrow\rightarrow}\}(e), \rho^{\text{RDEAD}}(e') \leftarrow 1; \end{cases} \\ \rho^{\text{VAL}}(e) = 0 \Rightarrow \rho^{\text{LDEAD}}(e) \leftarrow \rho^{\text{RDEAD}}(e) \leftarrow 1. \end{cases} \quad (6)$$

В случае прихода события с положительным значением $\rho^{\text{VAL}}(e) = 1$ поток активации удовлетворяется, и монитор сигнализирует об успешном статусе анализа. Отрицательный статус $\rho^{\text{VAL}}(e) = 0$ лишь уничтожает поступившее событие. Никакие действия с потоком активации не происходят, поскольку в режиме локального времени свойство *eventually!* не может дать отрицательного результата, за исключением конца моделирования.

3. Семантика последовательностных функций

Последовательностной функцией $f \in F$ в модели DRTL Q называется некоторая вычислительная процедура, характеризующаяся множествами входных, внутренних и выходных событий, осуществляющая их пошаговое преобразование на каждом тактовом цикле верификации. Результатом последовательностной функции в некоторый момент времени (номер тактового цикла с начала моделирования) является содержимое множества выходных событий в этом моменте времени.

Активирующей последовательностной функцией является такая последовательностная функция, выходные события которой, во-первых, инициируют некоторый поток активации, соответствующий текущему тактовому циклу, если таковой еще не был инициирован, во-вторых, назначают связи между событиями по правой цепочке в соответствии с принадлежностью к инициированному потоку, в-третьих, устанавливают значение флага $\rho^{\text{ACTIVE}} = 1$.

С точки зрения обработки одного события последовательные функции можно интерпретировать как конечные автоматы. Событие, поступив на вход последовательной функции, попадает в некоторое начальное состояние. Из начального состояния в зависимости от особенностей вычислительной процедуры и внешних условий событие попадает в одно из внутренних состояний. Наконец, достигнув конечного состояния, событие передается на выход функции и используется следующими элементами модели.

Детали вычислительного процесса, скрытого последовательной функцией, зависят от ее типа. Многие функции также предусматривают параметризацию. К основным последовательным функциям относятся:

– функция-генератор – последовательная функция, связанная с некоторой верификационной переменной $b \in B$, не имеющая входных и внутренних событий, мгновенно порождающая событие e в выходном множестве с текущим значением верификационной переменной:

$$f_{\text{gen}(b \in B)}(t) = [e_0 : \{e_{\leftarrow} = e_0, \begin{cases} e_{\uparrow} = \varepsilon, \rho^{\text{VAL}} = b(t) \end{cases}\}]; \quad (7)$$

– функция конъюнктивного конкатенирования – последовательная функция, имеющая два операнда, одним из которых обязательно является функция-генератор, значение выходных событий которого используется в качестве условия транспортирования цепочек событий из второго операнда:

$$f_{\text{conj}}(t, f_1, f_{\text{gen}}) : a = \rho^{\text{VAL}}(e^0 = \text{first}(f_{\text{gen}}(t))), \begin{cases} a = 1 \Rightarrow f_{\text{conj}}(t, f_1, f_{\text{gen}}) = f_1(t); \\ a = 0 \Rightarrow f_{\text{conj}}(t, f_1, f_{\text{gen}}) = \emptyset; \end{cases} \quad (8)$$

где $\text{first}(f(t))$ – первое событие, принадлежащее множеству (в случае операнда-генератора первое событие является единственным);

– функции-очереди, имеющие единственный операнд и моделирующие временные интервалы;

– функции-репетиции, реализующие циклический анализ булевого выражения или другой последовательной функции;

– функции, реализующие логические последовательные операторы, такие как OR, AND, INTERSECT, WITHIN, а также импликации $|\rightarrow$ и $|\Rightarrow$.

Наиболее часто применяемой последовательной функцией в предлагаемой модели является функция-очередь, лежащая в основе выбранного названия DRTLQ. Очереди применяются для моделирования интервалов между интересующими событиями в системе. Наглядно очередь можно представить в виде цепочки триггеров. На рис. 1 показана схема обработки последовательности событий $\{a; [*2]; b\}$ на основе очереди, начинающейся с момента, в котором сигнал $a = 1$, и в прошествии 3 тактовых циклов завершающейся событием $b = 1$.

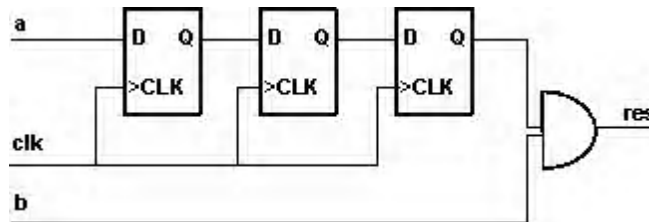


Рис. 1. Наглядная иллюстрация понятия DRTLQ-очереди

В данном случае функция-очередь $f_{\#}$ является операндом функции конъюнктивного конкатенирования f_{conj} с $f_{\text{gen}(b)}$ и имеет функцию-генератор $f_{\text{gen}(a)}$ на входе:

$$f(t, \{a; [*2]; b\}) = f_{\text{conj}}(t, f_{\#[3:3]}(t, f_{\text{gen}(a)}), f_{\text{gen}(b)}). \quad (9)$$

В общем виде, функция-очередь, имеющая идентичные конечные интервалы, функционирует путем хранения и продвижения к выходу списка цепочек событий, считанных с

входа в различные моменты моделирования, и упорядоченного по времени считывания. Цепочка событий, достигнувшая последней в очереди ячейки, является фактическим результатом функции на следующем такте обработки:

$$\begin{aligned}
& f_{\#[N:N]}(t, f_1(t)), N > 0 : \langle r_0(t), \dots, r_{N-1}(t) \rangle; \\
& r_0(t) = f_1(t); \\
& \forall i, 0 < i < N, r_i(t) = r_{i-1}(t-1); \\
& f_{\#[N:N]}(t, f_1(t)) = r_{N-1}(t-1). \tag{10}
\end{aligned}$$

Задача усложняется, если значения интервалов очереди различаются. Копии событий, время нахождения которых в очереди превысило минимальный заданный интервал, но еще не достигло максимального интервала, направляются на выход. Сами события остаются внутри очереди до момента достижения максимального интервала и продолжают транспортирование по очереди. Копии событий связывают по правой цепочке:

$$\begin{aligned}
& f_{\#[N:M]}(t, f_1(t)), N > 0, N < M : \langle r_0(t), \dots, r_{M-1}(t) \rangle; \\
& r_0(t) = f_1(t); \\
& \forall i, 0 < i < M, r_i(t) = r_{i-1}(t-1); \\
& f_{\#[N:M]}(t, f_1(t)) = r_{M-1}(t-1) \cup \bigcup_{i=N-1}^{M-2} \kappa(r_i(t-1)). \tag{11}
\end{aligned}$$

Если $M \rightarrow \infty$, суть вычислительной процедуры (11) не изменяется, однако структура внутренних цепочек событий $\langle r_0(t), \dots, r_{M-1}(t) \rangle$ преобразуется из статической формы в динамическую. Очередь с бесконечным интервалом будет генерировать и продвигать к выходу копии считанного события до тех пор, пока не будет остановлен породивший событие поток активации.

Если $N = 0$, копии считанных с входа событий моментально транспортируются к выходу, независимо от обработки события внутри очереди:

$$f_{\#[0:M]}(t, f(t)_1), M > 0 = f_{\#[1:M]}(t, f_1(t)) \cup \kappa(f_1(t)). \tag{12}$$

Функции-репетиции в модели DRTLQ разделяются на булевы и сложные, в зависимости от типа операнда (булево выражение и SERE соответственно). В целом, схема обработки булевых репетиций имеет определенное сходство с обработкой очередей. Дополнительно к сопоставлению времени регистрации в очереди, в репетициях проверяется значение входного булевого выражения. В том случае, если выражение не выполняется, все цепочки событий, в текущий момент хранящиеся в репетиции, поступают на выход со значением 0, и репетиция полностью очищается:

$$\begin{aligned}
& f_{[*N]}(t, f_{\text{gen}(b \in B)}(t)), N > 0 : \langle v(t), r_0(t), \dots, r_{N-1}(t) \rangle; \\
& v(t) = \rho^{\text{VAL}}(\text{first}(f_{\text{gen}(b)}(t))); \\
& \left[\begin{array}{l} v(t) = 1 \Rightarrow \left\{ \begin{array}{l} r_0(t) = f_{\text{gen}(b)}(t); \\ \forall i, 0 < i < N, r_i(t) = r_{i-1}(t-1); \\ f_{[*N]}(t, f_{\text{gen}(b)}(t)) = r_{N-1}(t); \end{array} \right. \\ \\ v(t) = 0 \Rightarrow \left\{ \begin{array}{l} \forall i, k, 0 \leq i < N, 0 \leq k < |r_i(t-1)|, \rho^{\text{VAL}}(e_k \in r_i(t-1)) \leftarrow 0; \\ f_{[*N]}(t, f_{\text{gen}(b)}(t)) = \bigcup_{i=0}^{N-1} r_i(t-1); \\ \forall i, 0 \leq i < N, r_i(t) = \emptyset. \end{array} \right. \end{array} \right. \tag{13}
\end{aligned}$$

Аналогично очереди (11), интервальная репетиция требует создания и продвижения к выходу копий цепочек событий, выдержанных внутри репетиции минимальное количество итераций, но не достигших максимума:

$$v(t) = 1 \Rightarrow f_{[*N:M]}(t, f_{\text{gen}(b \in B)}), N > 0, N < M = \bigcup_{i=N}^{M-1} \kappa(r_i(t-1)). \quad (14)$$

Случаи $M \rightarrow \infty$ и $N = 0$ определяются аналогичным очередям образом, разумеется, с учетом процедуры полного очищения репетиции при $v(t) = 0$.

Реализация галолирующих репетиций модифицирует базовую репетиционную функцию (13) добавлением элемента $\varphi(t)$, который должен обеспечивать буферизацию последнего успешного входного события до момента появления следующего. Соответственно, галолирующая репетиция никогда не выполняет полную очистку, как обычная репетиция:

$$f_{[\rightarrow N]}(t, f_{\text{gen}(b \in B)}(t)), N > 0 : \langle v(t), \varphi(t), r_0(t), \dots, r_{N-1}(t) \rangle, \\ v(t) = \rho^{\text{VAL}}(\text{first}(f_{\text{gen}(b)}(t))); \begin{cases} v(t) = 1 \Rightarrow \varphi(t) = f_{\text{gen}(b)}(t); \\ v(t) = 0 \Rightarrow \varphi(t) = \kappa(\varphi(t-1)); \end{cases} \\ \begin{cases} r_0(t) = \varphi(t); \\ \forall i, 0 < i < N, r_i(t) = r_{i-1}(t-1); \\ f_{[\rightarrow N]}(t, f_{\text{gen}(b)}) = r_{N-1}(t). \end{cases} \quad (15)$$

Неконсекutive репетиции расширяют реализацию галолирующих репетиций дополнительным выходным буфером, порождающим копии ранее транспортированных на выход событий до удовлетворения потока активации:

$$f_{[=N]}(t, f_{\text{gen}(b)}(t)) = \kappa(f_{[\rightarrow N]}(t-1, f_{\text{gen}(b)}(t-1))) \cup f_{[\rightarrow N]}(t, f_{\text{gen}(b)}). \quad (16)$$

Наиболее сложными являются функции, реализующие логические операторы над последовательностями. Пусть имеется SERE $s = \{\{a; b\} \parallel \{c; d\} \} \&\& \{!e[*2]\}$. На рис. 2 в графическом виде представлена DRTLQ-реализация данного SERE, а формула (17) демонстрирует то же SERE в аналитическом виде.

$$f_s(t) = f_{\supset}^1(t); f_{\supset}^1(t) = \text{join}_{2-\&\&}(t, f_{\supset}^2(t), f_{[*2:2]}(t, f_{\text{gen}(-e)}(t)));$$

$$f_{\supset}^2(t) = \text{join}_{2-\parallel} \left(\begin{array}{l} t, \\ f_{\text{conj}}(t, f_{\#[1:1]}(t, f_{\text{conj}}(t, f_c^2(t), a)), b), \\ f_{\text{conj}}(t, f_{\#[1:1]}(t, f_{\text{conj}}(t, f_c^2(t), c)), d); \end{array} \right)$$

$$f_c^2(t) = \text{split}_{2-\parallel}(t, f_c^1(t)); f_c^1(t) = \text{split}_{2-\&\&}(t, f_{\text{gen}(l)}(t)). \quad (17)$$

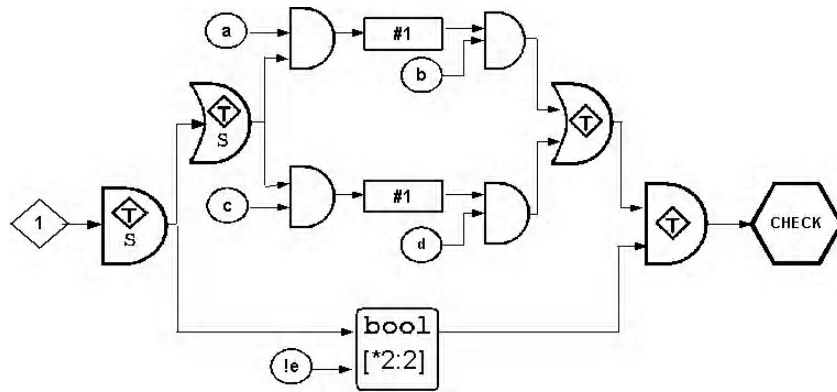


Рис. 2. DRTLQ-реализация последовательности с логическими операторами

Любой логический последовательный оператор реализуется двумя DRTLQ-функциями: разделителем f_{\leftarrow} (splitter) и соединителем f_{\rightarrow} (joiner). Функция-соединитель является вершиной пути операндов, а начало каждого из операндов исходит из функции-разделителя. Ряд операторов (WITHIN, импликации) могут иметь только два операнда, но степень разветвления логических операций (AND, OR, INTERSECT) не ограничивается. Разделитель всегда имеет единственный вход и множество выходов, соответствующее количеству операндов. Событие, считываемое разделителем с входа, расслаивается на самостоятельные копии, каждая из которых транспортируется по пути одного из операндов. Соединитель имеет единственный выход, порождая выходные цепочки событий в соответствии с реализуемой функцией.

Аналогичным образом реализуются репетиции с операндами-последовательностями: вводятся функция-разделитель и функция-соединитель, причем разделитель, определяющий направление транспортирования событий, находится выше по иерархии, чем соединитель, помещаемый на нижнем уровне. В зависимости от количества итераций, пройденных событием в рамках репетиции, событие либо направляется разделителем на выход, либо возвращается к соединителю на очередную итерацию.

4. Семантика сложных темпоральных свойств

Темпоральное свойство $p \in P$ в модели DRTLQ – совокупность логических и темпоральных отношений между булевыми выражениями, последовательными функциями и другими темпоральными свойствами, задающая пространство поведения системы. При помощи темпоральных свойств задаются утверждения о поведении системы на протяжении всего цикла моделирования. Элементы-свойства создаются только в режиме глобального времени и являются промежуточным звеном между последовательными функциями и ассерционным монитором.

При наличии в модели уровня темпоральных свойств задачи проверки критериев удовлетворения потоков активации выполняются именно этими элементами. Вспомогательные последовательные функции f_{ACTION} доставляют выходные цепочки событий на первый уровень иерархии темпоральных свойств. С ассерционным монитором взаимодействует только вершина иерархии свойств. Роль самого монитора упрощается и сводится к обработке статуса единственного за весь сеанс моделирования события.

Каждое свойство состоит из следующих структурных элементов:

$$p \in P : \{p_{\uparrow}, P_{\downarrow} \in P\} \times \{\text{INIT}, \text{ABORT}, \text{EVAL}, \text{FINAL}\}, \quad (18)$$

где P_{\uparrow} – родительское свойство; P_{\downarrow} – множество дочерних свойств; INIT – рекурсивная процедура активации вычислительного потока; ABORT – рекурсивная процедура отмены активации всех текущих вычислительных потоков; EVAL – процедура обработки события; FINAL – процедура обработки завершения моделирования.

Тело процедур и дополнительные элементы структуры различаются в зависимости от типа свойства. К основным типам свойств относятся:

- унарные свойства, в свою очередь подразделяющиеся на группы:
- буферные свойства (BUF, NOT);
- глобальные свойства (always, never, eventually!);
- бинарные свойства (\rightarrow , \leftrightarrow , $\mid\rightarrow$, $\mid\Rightarrow$, until *, before *, next *) – свойства, имеющие по два операнда-свойства;
- коммутативные свойства (& &, ||) – свойства, которые могут иметь неограниченное число равноправных операндов-свойств.

Унарные свойства в качестве операнда могут иметь как другие свойства, так и последовательную функцию.

Цикл работы ассерционного процесса в режиме глобального времени отличается от простого режима дополнительными шагами по обработке уровня темпоральных свойств:

1. На каждом шаге моделирования осуществляется активация свойств путем вызова процедуры $INIT$ на свойстве верхнего уровня. Активации обеспечивают разрешение создания вычислительных потоков на уровне последовательностей. В отличие от мониторов, всегда разрешающих активацию вычислительных потоков, свойство может блокировать часть дочерних последовательностей в случае необходимости.

2. События, транспортируемые функциями действия f_{ACTION} с последовательностного уровня, приводят к запуску процедуры $EVAL$ у свойств первого уровня. В зависимости от этой процедуры, событие может быть либо уничтожено, либо транспортировано к родительскому свойству. Если событие признается удовлетворяющим процедуру $EVAL$ свойства верхнего уровня, событие направляется к монитору, и весь цикл анализа ассерции завершается.

3. Финальная итерация предполагает вызов процедуры $FINAL$ для всех свойств p , принадлежащих модели. Если анализ ассерции не был завершен на момент остановки симуляции, результат определяется именно в этот момент.

Рассмотрим простейшее темпоральное свойство BUF , которое либо может иметь последовательностный вход, создаваемый при продвижении булевого выражения или последовательности на уровень свойств, либо дочернее свойство-операнд:

$$P_{BUF} : \left\{ p_{\uparrow}, \left[\begin{array}{l} P_{\downarrow} = \emptyset, f_{ACTION} \in F; \\ P_{\downarrow} : \{p_1 \in P\}; \end{array} \right], \Omega_{BUF} \in \Omega \right\} \times \{INIT_{BUF}, ABORT_{BUF}, EVAL_{BUF}, FINAL_{BUF}\}, \quad (19)$$

где f_{ACTION} – дочерняя последовательностная функция; p_1 – дочернее свойство-операнд; Ω_{BUF} – множество активных вычислительных потоков, порожденных свойством. Очевидно, такое свойство не имеет дочерних свойств.

Процедура $INIT_{BUF}$ состоит в расширении Ω_{BUF} новым потоком активации ω_t , соответствующим текущему моменту времени, в который добавляется искусственное событие e_t^0 . На уровне последовательностей наличие потока разрешает активацию событий, примыкающих по правой цепочке к e_t^0 .

Процедура $ABORT_{BUF}$ состоит в очищении множества потоков Ω_{BUF} вместе с уничтожением всех порожденных правых цепочек событий.

Процедура $EVAL_{BUF}$, вызываемая либо функцией действия f_{ACTION} с последовательностного уровня, либо дочерним свойством p_1 , удаляет из множества Ω_{BUF} поток ω' , для которого время активации равно времени активации пришедшего с входа события $t_A(\text{first}(\omega')) = t_A(e)$. Само событие e направляется в процедуру $EVAL$ родительского свойства.

Процедура $FINAL_{BUF}$ для всех вычислительных потоков Ω_{BUF} выбирает порожденное при активации событие e^0 и направляет к родительскому свойству: статус определяется режимом условной допустимости.

Аналогичным образом функционирует свойство NOT , имеющее идентичную структуру, а также процедуры активации и отмены активации:

$$P_{NOT} : \left\{ p_{\uparrow}, \left[\begin{array}{l} P_{\downarrow} = \emptyset, f_{ACTION} \in F; \\ P_{\downarrow} : \{p_1 \in P\}; \end{array} \right], \Omega_{NOT} \in \Omega \right\} \times \left\{ \begin{array}{l} INIT_{NOT} = INIT_{BUF}; \\ ABORT_{NOT} = ABORT_{BUF}; \\ EVAL_{NOT}; \\ FINAL_{NOT}; \end{array} \right\}. \quad (20)$$

Отличие процедур $EVAL_{NOT}$ и $FINAL_{NOT}$ от аналогов в BUF состоит в инверсии значения $\rho^{VAL}(e)$, транспортируемого к родительскому свойству события.

Рассмотрим работу глобальных темпоральных свойств на примере свойства *always* :

$$P_G : \left\{ p_{\uparrow}, \left[p_{\downarrow} = \emptyset, f_{\text{ACTION}} \in F; \right], p_{\downarrow} : \{p_1 \in P\}; \right\}, \Omega_G \in \Omega \times \left\{ \begin{array}{l} \text{INIT}_G; \\ \text{ABORT}_G = \text{ABORT}_{\text{BUF}}; \\ \text{EVAL}_G; \\ \text{FINAL}_G; \end{array} \right\}. \quad (21)$$

Отличие процедуры INIT_G от INIT_{BUF} состоит в том, что в случае наличия дочернего свойства-операнда и открытых потоков активации в нем на каждом такте анализа создается дополнительный поток активации, помещаемый в Ω_{BUF} . Такой прием обеспечивает неявное ветвление потоков активации, порождаемых глобальными темпоральными операторами, без фактического ветвления вычислений на уровне дочерних последовательностей.

Процедура EVAL_G состоит в завершении всех вычислительных потоков $\omega \in \Omega_G$, для которых выполняется соотношение $t_{\wedge}(\text{first}(\omega)) \leq t_{\wedge}(e)$, и транспортировании соответствующих потокам событий e^0 на уровень выше. Здесь ветвление свойств-операндов воссоздается, однако это не имеет такого эффекта на производительность анализа, как если бы производить все вычисления на последовательностном уровне.

Процедура FINAL_G с успешным статусом завершает все стартовавшие потоки Ω_G . Очевидно, активность свойства *always* в момент остановки симуляции свидетельствует об успешном выполнении данного свойства.

Свойство *never* отличается от свойства *always* инверсией значений $\rho^{\text{VAL}}(e)$ транспортируемых событий в процедуре EVAL , а свойство *eventually!* – инверсией значений в процедуре FINAL .

Коммутативные свойства, такие как AND, имеют следующую структуру:

$$p_{\wedge} : \left\{ p_{\uparrow}, p_{\downarrow} : \{p_1, \dots, p_N\}, \mu(t) : \langle k \in [0; N], v \in [0; 1] \rangle \right\} \times \left\{ \begin{array}{l} \text{INIT}_{\wedge} : \forall i, 0 \leq i < N, \text{INIT}(p_i); \\ \text{ABORT}_{\wedge} : \forall i, 0 \leq i < N, \text{ABORT}(p_i); \\ \text{EVAL}_{\wedge}; \\ \text{FINAL}_{\wedge}; \end{array} \right\} \quad (22)$$

где N – число операндов; p_1, \dots, p_N – свойства-операнды, а $\mu(t)$ – функция истории выполнения потока активации, каждый элемент которой соотносится с некоторым тактовым циклом и включает информацию о количестве завершенных на текущий момент операндов, а также общий статус завершения.

Работа процедур INIT_{\wedge} и ABORT_{\wedge} тривиальна и состоит из вызова аналогичных процедур на всех свойствах-операндах по очереди.

Работа процедуры EVAL_{\wedge} состоит из двух шагов:

1. Изменение статуса истории потока активации по принципу $v(\mu(t)) \leftarrow v(\mu(t)) \wedge \rho^{\text{VAL}}(e)$ (очевидно, начальное значение $v = 1$). В случае изменения статуса $v(\mu(t-1)) = 1 \rightarrow v(\mu(t)) = 0$ поток активации признается неудачным, и событие e продвигается на уровень выше.

2. Инкремент числа выполненных операндов: $k(\mu(t)) \leftarrow k(\mu(t)) + 1$. Если новое значение $k(\mu(t)) = N$, поток активации признается успешным, и событие e также продвигается на уровень выше.

Работа процедуры FINAL_{\wedge} заключается в продвижении на уровень выше событий всех потоков активации, для которых имеется хотя бы один незавершенный поток операнда: $k(\mu(t)) < N$ – а начальный статус по-прежнему не изменился: $v(\mu(t)) = 1$.

Отличие свойства OR в деталях вычислительных процедур. Во-первых, начальным значением $v(\mu(t))$ является 0. Во-вторых, условие досрочного завершения анализа потока в процедуре $EVAL_{\downarrow}$ заключается в переходе $v(\mu(t-1)) = 0 \rightarrow v(\mu(t)) = 1$, и продвигает успешный статус. В-третьих, достижение значения $k(\mu(t)) = N$, при котором $v(\mu(t)) = 0$, завершает поток с неудачным статусом.

Аналогичным способом строится обработка темпоральных свойств \rightarrow и \leftrightarrow , с той разницей, что число операндов здесь всегда равно двум.

Бинарное свойство, реализующее семейство операторов *until**, имеет более сложную структуру и вычислительные процедуры:

$$p_U : \{p_{\uparrow}, p_{\downarrow} : \{p_1, p_2\}, v(t) : \in \{0, 1, X\}, \{s_1, s_2, s'_1, s'_2\}(t) : \in \{0, 1, X\}\} \times \{INIT_U, ABORT_U = \{ABORT(p_1); ABORT(p_2)\}, EVAL_U, FINAL_U\}; \quad (23)$$

где p_1, p_2 – дочерние свойства-операнды; $v(t)$ – функция текущего статуса вычислительного потока; $s_1(t)$ и $s_2(t)$ – функции фактического состояния потоков, принадлежащих операндам, а $s'_1(t)$ и $s'_2(t)$ – функции интерпретации состояния потоков, принадлежащих операндам.

Процедура $INIT_U$ активирует свойства-операнды и кроме того создает записи для текущего момента t с начальными значениями $s_1(t) = s_2(t) = X$, а также $v(t) = X$.

Работа процедуры $EVAL_U$ состоит из следующих шагов:

1. Обновление состояния операнда, с которого пришло событие: $s_{\downarrow}^{\downarrow}(t) \leftarrow \rho^{VAL}(e)$.

2. Обновление состояния ранее не разрешенных активированных потоков, начиная со стартовавших одновременно с событием, до текущего времени:

$$\exists k, 0 \leq k < (t_{now} - t_A(e)),$$

$$s'_2(t_A(e) + k) = \begin{cases} 0 \Rightarrow \begin{cases} s'_1(t_A(e) + k) = 0 \Rightarrow v(t_A(e) + k); \\ v(t_A(e) + k) \leftarrow X; \end{cases} \\ 1 \Rightarrow \begin{cases} k > 0 \Rightarrow v(t_A(e) + k) \leftarrow s'_1(t_A(e) + k - 1); \\ k = 0 \Rightarrow v(t_A(e) + k) \leftarrow 1; \end{cases} \\ X \Rightarrow v(t_A(e) + k) \leftarrow X; \end{cases} \quad (24)$$

3. Разрешение потоков (отправка события на уровень выше), для которых $v(t)$ на предыдущем шаге претерпело изменение $X \rightarrow 0$ или $X \rightarrow 1$.

В табл. 2 приведены определения функций $s'_1(t)$ и $s'_2(t)$ на основе функций $s_1(t)$ и $s_2(t)$, изменяющиеся в зависимости от конкретного оператора.

Таблица 2

Оператор	$s'_1(t)$	$s'_2(t)$
$f_1 \text{ until } f_2$	$s_1(t)$	$s_2(t)$
$f_1 \text{ until}_- f_2$	$s_1(t)$	$s_1(t) \wedge s_2(t)$
$f_1 \text{ before } f_2$	$\neg s_2(t)$	$s_1(t) \wedge \neg s_2(t)$
$f_1 \text{ before}_- f_2$	$\neg s_2(t)$	$s_1(t) \wedge \neg s_2(t)$

Процедура $FINAL_U$ для всех потоков с $v(t) = X$ направляет событие, принадлежащее потоку, на родительский уровень со статусом, определяющимся соотношением $\neg \text{strong}(p_U) \wedge s_1(t_\infty)$ для операторов before^* и $\neg \text{strong}(p_U) \wedge s_2(t_\infty)$ – для операторов until^* .

5. Выводы

Задача функциональной верификации параллельных систем состоит в проверке логико-временных ограничений между атомарными утверждениями, ассоциированными с состояниями системы на всех возможных вычислительных путях. Утверждения об удовлетворении системой определенных ограничений, называемые ассерциями, отражают требования спецификации в виде формальной математической формулы, использующей аппарат темпоральной логики, который пришел в компьютерную инженерию из философии. Экспоненциальная сложность формальной проверки накладывает существенные ограничения на применимость метода к крупным моделям. Лучшую производительность верификации LTL-ограничений обеспечивают динамические методы, функционирующие путем конвертирования LTL к детерминированному конечному автомату. Несмотря на проблему низкого покрытия пространства состояний системы (зависящую от внешнего теста) в динамических методах, а также нелинейный рост размера проверяющих автоматов в зависимости от структурной сложности формулы, использование LTL-логики во время симуляции способно обеспечить обнаружение большинства функциональных нарушений с более приемлемой производительностью по сравнению с формальными методами.

Для обеспечения существенно лучшей производительности анализа ассерций и поддержки режима полноценной интерпретации глобальных темпоральных операторов была предложена модель динамических регистровых очередей (DRTLQ). Она ориентирована на высокопроизводительный анализ элементов линейной темпоральной логики и эффективность обнаружения и локализации нарушений. Модель состоит из четырех структурных уровней: 1) переменных верификации, обеспечивающих абстракцию выражений булевого уровня, 2) последовательностных функций, предназначенных для реализации регулярных выражений, учитывающих параметр времени, 3) темпоральных свойств и ассерционного монитора, контролирующего потоки активации при анализе формул, 4) ассерционного процесса, реализующего верификацию логически связанной группы темпоральных утверждений. В отличие от модели на основе конечных автоматов, количество элементов модели растет линейно относительно размера моделируемой LTL-формулы. Высокий уровень компактности модели достигается за счет минимальной структурной сложности представления простых и интервальных временных сдвигов, циклических ограничений, логических разветвлений – конструкций, порождающих сложные деревья вычислительных процессов, приводящих модель на основе конечных автоматов к нелинейному росту количества состояний.

Событийная модель обработки конструкций-свойств и вычислительные процедуры, контролирующие множества инициированных вычислительных путей, обеспечивают эффективную реализацию всех LTL-операторов с линейной сложностью. Производительность достигается иерархической декомпозицией событийных потоков. Теоретическая древовидность низкоуровневых вычислений, определяемая семантикой LTL, подменяется линейной нагрузкой на последовательностный уровень, а фактический учет сложных ветвлений эффективно реализуется на высоком уровне абстракции. Вычислительная сложность реализации такова, что применение на практике оказывается существенно быстрее (от 25 до 500%) по сравнению с традиционным способом реализации на основе конечных автоматов.

Ключевым научным результатом работы является формальное определение семантики сложных операторов LTL-логики последовательностного и темпорального уровня, а

также создание формальной спецификации ключевых вычислительных процедур цикла работы последовательных функций и темпоральных свойств в различных режимах интерпретации времени.

Основным *практическим результатом* исследования является разработка модели процесса реализации сложных операторов линейной темпоральной логики, характеризующаяся линейной вычислительной сложностью относительно обработки одного вычислительного пути, возможностью одновременной обработки нескольких вычислительных путей за счет конвейерной архитектуры. Предложенные алгоритмы обеспечивают системе анализа ассерций параметры быстродействия, позволяющие уверенно конкурировать с аналогичными системами на EDA-рынке.

Список литературы: 1. *Emerson E.A.* Temporal and modal logic // In J. Van Leeuwen, editor, "Handbook of Theoretical Computer Science", Elsevier Science Publishers, 1990, volume B. P. 996–1072. 2. *Clarke E.M.Jr, Grubmerg O., Peled D.A.* Model Checking // The MIT Process, 2002. 314p. 3. *Drechsler R.* Advanced Formal Verification // Kluwer Academic Publishers, 2004. 277p. 4. *Foster H., Krolnik A., Lacey D.* Assertions-based Design", Kluwer Academic Publishers, 2003. 392p. 5. *Forczek M, Hryniewicz K.* Formal properties evaluation // Proc. of Programmable Devices and Systems Workshop 2003 (PDS-2003). P.305-311. 6. *Хаханов В.И., Зайченко С.А.* Модель динамических регистровых очередей для быстродействующего анализа линейных темпоральных ограничений // АСУ и приборы автоматики. 2007. Вып. 136. С. 10-25. 7. *Хаханов В. И., Егоров А. А., Зайченко С. А., Обризан В. И., Каминская М. А.* Механизм ассерций для функциональной верификации проектируемых цифровых систем // АСУ и приборы автоматики. 2005. Вып.130.С. 85-95. 8. *Зайченко С.А., Хаханов В.И.* Эффективная функциональная верификация моделей цифровых систем на кристалле на основе ассерций глобального времени // АСУ и приборы автоматики. 2007. Вып. 137.С. 4-13.

Поступила в редколлегию 19.09.2008

Зайченко Сергей Александрович, аспирант кафедры АПВТ ХНУРЭ, начальник отдела разработки компании Aldec-Kharkov Ltd. Научные интересы: системы автоматизированного проектирования, моделирования и верификации цифровых систем на кристаллах. Увлечения: литература, музыка, футбол. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (097)-367-62-93. E-mail: Sergei.Zaychenko@aldec.com

Хаханов Владимир Иванович, декан факультета КИУ ХНУРЭ, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (057)-702-13-26. E-mail: hahanov@kture.kharkov.ua

ЧИСЕЛЬНЕ МОДЕЛЮВАННЯ ДИФУЗІЙНИХ ПРОЦЕСІВ У СИСТЕМАХ З МАСИВАМИ ЗАГЛИБЛЕНИХ МІКРОЕЛЕКТРОДІВ У ФОРМІ ДИСКУ

Розглядаються дифузійні процеси, що відбуваються у масивах заглиблених дискових мікроелектродів. Наводиться математична модель для дослідження цих процесів та пропонується використовувати квазіконформне відображення для чисельного розв'язку даної задачі. Отримані чисельні результати наводяться у вигляді часозалежних струмів, обчислених для різних кінцевих заглиблень.

1. Вступ

Масиви заглиблених мікро- та наноелектродів отримали широке застосування з розвитком методу їх виготовлення – літографії. Завдяки своїм особливим властивостям, масиви мікроелектродів (у формі дисків або смужок) використовуються для детектування та моніторингу навколишнього середовища, у фізіології для вивчення клітин та тканин тощо. Слід відзначити, що незважаючи на широке використання, теоретичних робіт щодо пояснення усіх їх властивостей ще дуже мало [1, 2]. Теорія систем з заглибленими одиночними мікроелектродами, а також підхід для чисельного моделювання відгуку системи для довільної геометрії заглиблення були запропоновані у нашій попередній роботі [3].

Метою даної роботи є дослідження залежностей відгуку системи від її геометричних параметрів, а також розробка чисельного підходу для моделювання дифузійних процесів у масивах мікроелектродів. Для досягнення цієї мети автор продовжує розвиток універсального підходу, запропонованого у [3], для дослідження поведінки масивів мікроелектродів з довільною геометрією системи на прикладі мікродисків. В основу цього чисельного підходу покладено застосування квазіконформного відображення для чисельного моделювання. Це дозволяє відображати область моделювання у одиничний квадрат, що спрощує обчислення цієї складної двовимірної проблеми у порівнянні з застосуванням реальних координат. Також застосування квазіконформного відображення вирішує проблему сингулярності навколо отвору заглиблення електродів, яка виникає через надзвичайно високі струми навкруги цієї точки і обумовлена складною геометрією системи. Ці дві переваги значно спрощують чисельний розв'язок проблеми та обумовлюють високу точність чисельних результатів.

2. Математична модель у реальних координатах

Розглянемо масив заглиблених дискових мікроелектродів, тобто систему, що складається з низки дискових мікроелектродів, які працюють одночасно. Будемо вважати спочатку, що всі електроди у масиві мають однакові характерні розміри. Тоді кожному електроду можна поставити у відповідність «комірку», в якій він знаходиться, яка характеризується такими геометричними параметрами: r_d – радіус електроду; α – кут між поверхнею електроду та твірною кінцевого (чи циліндричного) заглиблення, в якому знаходиться електрод; h – висота заглиблення; r_c – радіус одиничної комірки з електродом; β – кут між поверхнею ізолятора та площиною симетрії, що оточує комірку з одиничним електродом (рис. 1). Радіус отвору кінцевого заглиблення r_p (див. рис. 1) є залежним параметром та обчислюється з наведених вище параметрів так:

$$r_p = r_d - h \operatorname{ctg}(\alpha). \quad (1)$$

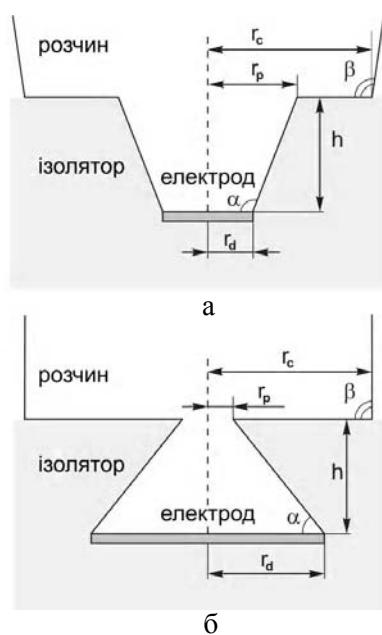


Рис. 1. Осьовий перетин комірки з заглибленим електродом з (а) $\alpha, \beta > \pi/2$ та (б) $\alpha < \pi/2, \beta = \pi/2$

Припустимо, що кожен з електродів працює в амперометричному режимі при постійному та однаковому для усіх електродів у масиві потенціалі (більш складні ситуації буде обговорено нижче), тобто на поверхні електроду відбувається реакція переносу електрона $A \pm e \rightarrow B$ у дифузійно контрольованому режимі. В такому випадку для опису поведінки всієї системи достатньо дослідити поведінку однієї комірки, тому що сумарний відгук системи буде пропорційним відгуку одиничної комірки.

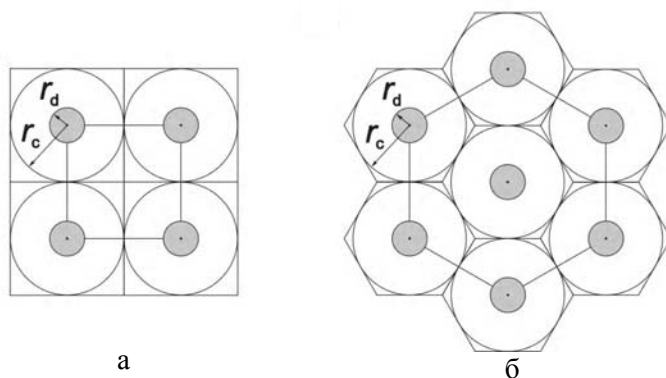


Рис. 2. Погляд зверху на квадратний (а) та гексагональний (б) масиви заглиблених електродів. Відповідними багатокутниками позначено реальну зону поширення дифузійного шару, колами позначено фактичну зону моделювання; штриховане коло – електрод

Комірка з одиничним електродом має осьову симетрію, тому математичну модель зручно формулювати у циліндричних координатах. Однак слід відзначити особливість, що виникає у цьому випадку. Розглянемо квадратний чи гексагональний масив заглиблених електродів (рис. 2) (випадок неструктурованого розподілу електродів чи електродів різного розміру буде обговорено нижче). Дифузійний шар кожного з електродів буде поширюватися в рамках відповідного багатокутника, квадрата (рис. 2,а) чи гексагону (рис. 2,б), тоді як формулювання проблеми у циліндричних координатах зменшує область, що розглядається, до кола, зображеного всередині відповідного багатокутника. Слід, однак, відзначити, що у роботах [1, 2, 4] було доведено, що вплив цих ділянок на відгук системи є малим і ним можна знехтувати, якщо розглядати кола з площею, що дорівнює площі відповідного багатокутника.

Для надання результатам більшої загальності математичну модель та результати наведемо у безрозмірному вигляді за допомогою таких безрозмірних змінних:

$$C = \frac{c}{c_0}, \quad R = \frac{r}{r_d}, \quad Z = \frac{z}{r_d}, \quad \tau = \frac{Dt}{r_d^2}, \quad (2)$$

де r, z – просторові циліндричні координати; t – час; $c = c(r, z, t)$ – концентрація генерованої на поверхні електроду речовини В; c_0 – концентрація речовини А до початку експерименту; D – коефіцієнт дифузії.

Математична модель для описаного вище процесу має вигляд:

$$\frac{\partial C}{\partial \tau} = \frac{\partial^2 C}{\partial R^2} + \frac{1}{R} \frac{\partial C}{\partial R} + \frac{\partial^2 C}{\partial Z^2}, \quad (3a)$$

початкові та граничні умови (див. рис. 1):

$$\tau = 0: \quad \forall R, Z; \quad C = 0; \quad (3б)$$

$$\tau > 0:$$

$$R = 0 \quad 0 < Z < \infty \quad \frac{\partial C}{\partial R} = 0; \quad (\text{вісь симетрії}) \quad (3в)$$

$$0 \leq R \leq 1, \quad Z = 0, \quad C = 1; \quad (\text{електрод}) \quad (3г)$$

$$\theta\left(\alpha - \frac{\pi}{2}\right) + \theta\left(\frac{\pi}{2} - \alpha\right)R_p \leq R \leq \theta\left(\alpha - \frac{\pi}{2}\right)R_p + \theta\left(\frac{\pi}{2} - \alpha\right), \quad Z = R \operatorname{tg}(\alpha), \quad \frac{\partial C}{\partial n} = 0; \quad (\text{ізолятор}) \quad (3д)$$

$$R_p < R < R_c, \quad Z = H, \quad \frac{\partial C}{\partial Z} = 0; \quad (\text{ізолятор}) \quad (3е)$$

$$R_c \leq R < \infty, \quad Z = -R \operatorname{tg}(\beta), \quad \frac{\partial C}{\partial n} = 0; \quad (\text{площина симетрії}) \quad (3е)$$

$$R^2 + Z^2 \rightarrow \infty, \quad C \rightarrow 0, \quad (\text{нескінченно видалена точка}), \quad (3ж)$$

де H, R_c, R_p – безрозмірні аналоги відповідних розмірних параметрів; n – вектор одиничної нормалі; $\theta(x)$ – функція Хевісайда.

Як було відзначено вище, для опису поведінки усього масиву мікроелектродів достатньо дослідити поведінку одного з електродів, що складають масив, у разі коли геометричні параметри усіх заглиблених електродів однакові. Якщо геометричні параметри електродів у масиві є різними, то маючи статистичний розподіл геометричних параметрів комірок одиничних електродів у масиві, необхідно обчислити відгук кожної з таких комірок (або зробити це з деяким кроком, якщо розподіл є неперервним) та просумувати всі відгуки з ваговими коефіцієнтами, що відповідають статистичному розподілу цих параметрів (див. також додаток у [5]).

3. Квазіконформне відображення

Область моделювання, що зображена на рис. 1, є досить складною для вирішення проблеми (3) у реальних координатах. Крім того, в околі точки (2) (рис. 3,а) спостерігаються великі градієнти концентрації, що призводить до накопичування похибок в цій області та розбіжності чисельного розв'язку у разі використання непристосованих обчислювальних сіток. Тому для чисельного розв'язку задачі (3) було запропоновано використовувати квазіконформне відображення [3]. У нашому випадку область моделювання є багатокутником, тому інтеграл Шварца-Кристофеля [6]:

$$X = K \int_0^\zeta \frac{(\zeta - u_2)^{1-\alpha/\pi}}{\sqrt{\zeta} (\zeta - 1)^{1-\alpha/\pi} (\zeta - u_3)^{1-\beta/\pi}} d\zeta \quad (4)$$

відображає багатокутник (що знаходиться у першому квадранті X -площини, з границею, позначеною неперервною лінією) на верхню півплощину (див. рис. 3), де $X = R + iZ$, $\zeta = u + iv$ – комплексні змінні; K – масштабувальний множник, що однозначно визначається геометрією системи; u_2 та u_3 є образами точок (2) та (3) на рис. 3,а. Зазначимо, що як відомо з

теорії конформних відображень [6], для будь-якого конформного відображення три дійсних параметри можуть бути задані довільно. Інтеграл (4) записано з такою відповідністю точок: початок координат та точка (1; 0) у ζ -площині відображаються у початок координат (центр електроду) та точку (1; 0) (край електроду) в X -площині відповідно; нескінченно видалена точка у X -площині є образом нескінченно видаленої точки у ζ -площині. Слід також відзначити, що незалежно від значень параметрів α , β , H та R_c перетворення (4) буде відображати заданий багатокутник на верхню півплощину. При зміні зазначених параметрів будуть змінюватися параметри інтегралу, тобто K , u_2 та u_3 .

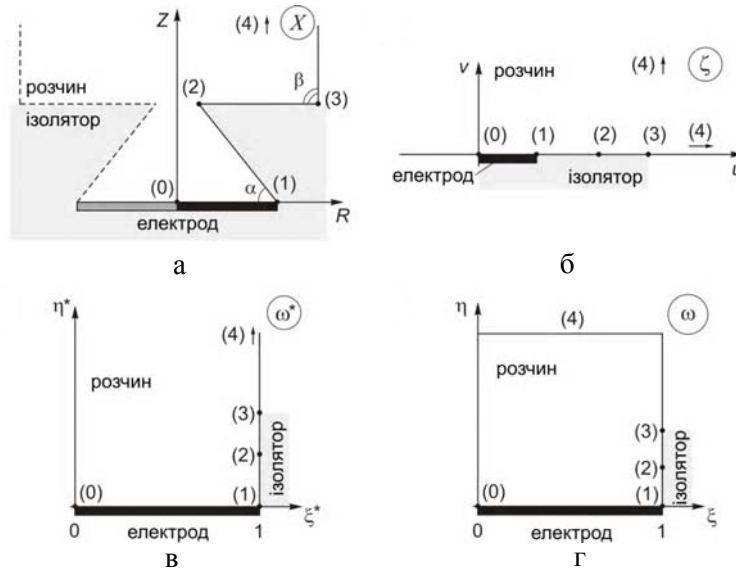


Рис. 3. Послідовність перетворень області моделювання. Область моделювання зображено: а – у реальних координатах; б – після застосування відображення (4); в – після застосування функції (5); г – після застосування додаткової функції стиснення (9)

Далі використання функції

$$\omega^* = \frac{2}{\pi} \arcsin(\sqrt{\zeta}), \quad (5)$$

де $\omega^* = \xi^* + i\eta^*$, відображає верхню півплощину ζ - площини на напівнескінченну смугу у ω -просторі з відрізком $[0; 1]$ в основі (рис. 3,в). Нагадаємо, що ми явно використовуємо симетрію області моделювання і розглядаємо тільки область у першому квадранті X -площини ($R \geq 0, Z \geq 0$) та її відповідні образи (див. рис. 3).

Зворотне відображення для функції (5) має вигляд:

$$\zeta = \sin^2\left(\frac{\pi}{2} \omega^*\right). \quad (6)$$

Підставляючи останнє рівняння в інтеграл Шварца-Кристофеля (4), після деяких спрощень отримаємо:

$$X = \pi K \times \int_0^{\omega^*} \frac{(u_2 - \sin^2(\pi\omega^*/2))^{1-\alpha/\pi}}{(1 - \sin^2(\pi\omega^*/2))^{0.5-\alpha/\pi} (\sin^2(\pi\omega^*/2) - u_3)^{1-\beta/\pi}} d\omega^*. \quad (7)$$

Для визначення параметрів перетворення (7) K , u_2 та u_3 для заданої геометрії системи необхідно розв'язати таку систему інтегральних рівнянь:

$$\left\{ \begin{array}{l} \pi K \int_0^1 \frac{(u_3 - \sin^2(\pi\omega^*/2))^{1-\alpha/\pi}}{(1 - \sin^2(\pi\omega^*/2))^{0.5-\alpha/\pi} (\sin^2(\pi\omega^*/2) - u_3)^{1-\beta/\pi}} d\omega^* = 1, \\ \pi K \int_0^{u_2} \frac{(u_3 - \sin^2(\pi\omega^*/2))^{1-\alpha/\pi}}{(1 - \sin^2(\pi\omega^*/2))^{0.5-\alpha/\pi} (\sin^2(\pi\omega^*/2) - u_3)^{1-\beta/\pi}} d\omega^* = R_p + iH, \\ \pi K \int_0^{u_3} \frac{(u_3 - \sin^2(\pi\omega^*/2))^{1-\alpha/\pi}}{(1 - \sin^2(\pi\omega^*/2))^{0.5-\alpha/\pi} (\sin^2(\pi\omega^*/2) - u_3)^{1-\beta/\pi}} d\omega^* = R_c + iH. \end{array} \right. \quad (8)$$

Систему (8) було розв'язано чисельно за допомогою спеціальних методів чисельного інтегрування сингулярних інтегралів [7, 8] та методу золотого поділу [7].

Отримана область моделювання (рис. 3,в) є набагато зручнішою для чисельного моделювання, ніж область моделювання у реальних координатах (рис. 3,а). Крім того, у трансформованому просторі вирішено проблему з сингулярністю в околиці точки (2) (рис. 3,а) [3]. Але у разі використання ω^* -простору для чисельного розв'язку виникає проблема залежності розмірів області моделювання від часу експерименту, що моделюється. Крім того, коли $\beta = \pi/2$ (тобто коли масив електродів розташовано на площині), система не має стаціонарного режиму, тому область моделювання буде збільшуватися необмежено зі збільшенням часу експерименту.

Для розв'язання цієї проблеми була застосована додаткова функція стиснення вздовж координати η^* , що відображає необмежену смугу на замкнений прямокутник. Зрозуміло, що таких функцій існує безліч і обговорення щодо ефективності застосування різних функцій наведено у роботах [3, 9]. У даній роботі було застосовано таку функцію стиснення [3, 9, 10]:

$$\eta^* = f(\eta) = \frac{\eta}{1-\eta}, \quad \eta = f^{-1}(\eta^*) = \frac{\eta^*}{1+\eta^*}, \quad (9)$$

де η змінюється на відрізьку $[0; 1]$.

Площини ω та ω^* пов'язані таким співвідношенням:

$$\omega = \xi + i\eta = \xi^* + if^{-1}(\eta^*). \quad (10)$$

Усі результати, описані нижче, отримані при обчисленні у ω -просторі. Слід зазначити також, що суперпозиція відображень (4) та (6) є конформним відображенням, тоді як застосування функції (9) робить результативне відображення квазіконформним, тобто задовольняючим більш загальній еліптичній системі, ніж система Коші-Рімана [6].

Наведені вище викладки було проведено для масиву заглиблених мікродискових електродів. Але слід зазначити, що для побудови квазіконформного відображення ніякої специфічної інформації, крім осьового перетину системи, не використовувалося. У разі масиву з заглибленими мікросмужковими електродами перетин буде мати такий саме вигляд і тому наведене конформне відображення може бути застосовано для чисельного розв'язання задач масопереносу в системах з масивами заглиблених мікросмужкових електродів [3]. Зазначимо, що у цьому випадку математична модель буде іншою, тобто у рівнянні (3) буде відсутній радіальний член з першою похідною за R .

4. Часткові випадки, коли $\alpha = \pi/2$ чи $\beta = \pi/2$

У частковому випадку, коли $\alpha = \pi/2$ (тобто у випадку прямого циліндричного заглиблення), перетворення (7) дещо спрощується й має такий вигляд:

$$X = \pi K \int_0^{\omega^*} \frac{\sqrt{u_2 - \sin^2(\pi\omega^*/2)}}{(\sin^2(\pi\omega^*/2) - u_3)^{1-\beta/\pi}} d\omega^*. \quad (11)$$

Відзначимо, що випадок, коли $\beta = \pi/2$ (тобто масив електродів розташований на площині, а не на скривленій поверхні), є найбільш розповсюдженим в експериментальній практиці [1, 2, 4]. Для цього випадку масштабувальний множник K можна визначити аналітично за допомогою такої стандартної процедури [6].

Якщо точка в площині X знаходиться на осі симетрії, то при її переміщенні паралельно осі R на площину симетрії (тобто на пряму, що виходить з точки (3) і паралельну осі Z , див. рис. 3,а) вона отримує приріст, рівний $\Delta X = R_c$. Цьому переміщенню (при $Z \rightarrow \infty$) буде відповідати перехід вздовж кола нескінченного радіуса з центром у початку координат в площині ζ (рис. 3,б). Використовуючи полярну форму запису комплексного числа $\zeta = re^{i\varphi}$ та у граничному випадку, коли $\zeta \rightarrow \infty$, інтеграл (4) можна переписати так:

$$\Delta X = K \int_{C_R} \frac{d\zeta}{\zeta} = K \int_0^\pi i\varphi d\varphi = -K i \pi. \quad (12)$$

З останнього рівняння та попереднього спостереження отримуємо вираз для константи K :

$$K = \frac{R_c}{\pi} i. \quad (13)$$

Рівняння (13) виконується у разі, коли $\beta = \pi/2$, незалежно від значень решти геометричних параметрів системи.

5. Математична модель у перетворених координатах

Рівняння дифузії у ω -площині має вигляд:

$$\begin{aligned} \frac{\partial C}{\partial \tau} = \det J^* \left[\frac{\partial^2 C}{\partial \xi^2} + \frac{1}{[f'(\eta)]^2} \frac{\partial^2 C}{\partial \eta^2} \right] + \\ + \frac{1}{R(\xi, \eta)} \left[\frac{\partial C}{\partial \xi} \frac{\partial \xi^*}{\partial R} + \frac{\partial C}{\partial \eta} \left\{ \frac{1}{f'(\eta)} \frac{\partial \eta^*}{\partial R} - \frac{f''(\eta)}{[f'(\eta)]^3} \right\} \right], \end{aligned} \quad (14)$$

де $\det J^*$ – якобіан перетворення (7):

$$\det J^* = \left(\frac{\partial \xi^*}{\partial R} \right)^2 + \left(\frac{\partial \xi^*}{\partial Z} \right)^2 = \left(\frac{\partial \eta^*}{\partial R} \right)^2 + \left(\frac{\partial \eta^*}{\partial Z} \right)^2, \quad (15a)$$

тут часткові похідні визначаються таким чином:

$$\frac{\partial \xi^*}{\partial R} = \frac{\partial \eta^*}{\partial Z} = \operatorname{Re} \left(\frac{d\omega}{dX} \right); \quad -\frac{\partial \xi^*}{\partial Z} = \frac{\partial \eta^*}{\partial R} = \operatorname{Im} \left(\frac{d\omega}{dX} \right); \quad R(\xi^*, \eta^*) = \operatorname{Re} X(\omega), \quad (15б)$$

$$\frac{d\omega}{dX} = \left[\frac{dX}{d\omega} \right]^{-1} = \frac{1}{\pi K} \times \frac{\left(1 - \sin^2 \left(\frac{\pi}{2} \omega \right) \right)^{0.5 - \alpha/\pi} \left(\sin^2 \left(\frac{\pi}{2} \omega \right) - u_3 \right)^{1 - \beta/\pi}}{\left(u_2 - \sin^2 \left(\frac{\pi}{2} \omega \right) \right)^{1 - \alpha/\pi}}. \quad (15в)$$

Початкові та граничні умови у перетвореному просторі є такими (рис. 3,г):

$$\begin{aligned} \tau = 0: \\ 0 \leq \xi \leq 1, \quad 0 \leq \eta \leq 1, \quad C = 0; \end{aligned} \quad (16a)$$

$$\begin{aligned} \tau > 0: \\ 0 \leq \xi \leq 1, \quad \eta = 0, \quad C = 1; \end{aligned} \quad (\text{електрод}) \quad (16б)$$

$$\xi = 0, \quad 0 \leq \eta < 1, \quad \frac{\partial C}{\partial \xi} = 0; \quad (\text{вісь симетрії}) \quad (16в)$$

$$\xi = 1, \quad 0 \leq \eta < 1, \quad \frac{\partial C}{\partial \xi} = 0; \quad (\text{ізолятор, площина симетрії}) \quad (16г)$$

$$0 \leq \xi \leq 1, \quad \eta = 1, \quad C = 0. \quad (\text{нескінченно видалена точка}) \quad (16д)$$

6. Струм

Струм, що протікає через одиничний електрод масиву, записується у безрозмірних координатах таким чином:

$$i(t) = 2\pi nFDc_0r_d \int_0^1 \frac{\partial C}{\partial Z} R dR. \quad (17)$$

Після застосування перетворень (7) та (9) вираз для струму у перетворених координатах переписується як:

$$i(t) = 2\pi nFDc_0r_d \int_0^1 \frac{\partial C}{\partial \eta} R(\xi, 0) d\xi. \quad (18)$$

Безрозмірний струм визначимо як відношення $i(t)$ до стаціонарного струму мікродискового електрода:

$$\psi = \frac{i(t)}{4nFDc_0r_d} = \frac{\pi}{2} \frac{1}{f'(0)} \int_0^1 \frac{\partial C}{\partial \eta} R(\xi, 0) d\xi. \quad (19)$$

Для отримання відгуку всієї системи (тобто масиву) треба помножити відповідний вираз (18) чи (19) на кількість електродів у масиві.

7. Деталі чисельного моделювання

Задача (14), (16) розв'язувалася чисельно у ω -просторі за допомогою методу змінних напрямків [11] та нерівномірної сітки за часом [12]. Усі результати було отримано на просторовій сітці $N\xi \times N\eta = 200 \times 200$ точок та з наступними параметрами сітки за часом $\Delta\tau_0 = 10^{-5}$, $\mu = 10^{-3}$, де $\Delta\tau_0$ – початковий крок за часом та μ – параметр, що контролює швидкість збільшення кроку сітки. Усі програми було написано на Borland Delphi 7 Enterprise Edition.

Задача, що досліджується, має осьову симетрію, тому рівняння дифузії у реальних та перетворених координатах має радіальний член $R^{-1}(\xi, \eta)$. Для виконання обчислень у ω -просторі інтеграли (7) повинні бути обчислені для усіх вузлів обчислювальної сітки, крім тих, що знаходяться на границі області моделювання. Тому що інтеграл (7) не може бути вираженим через елементарні функції, його значення обчислювалися чисельно. Однак слід зазначити, що потрібна додаткова увага при обчисленні цього інтеграла, тому що (i) чисельні похибки в обчислених значеннях інтеграла можуть впливати на розв'язок всієї задачі; (ii) інтеграл (7) є сингулярним (тобто значення інтеграла є необмеженими у точках 0, 1 та u_3); (iii) інтеграл повинен бути обчислений у комплексній площині. Для врахування пунктів (i)-(iii) було написано спеціальні процедури інтегрування у комплексній площині, що повертають значення інтеграла з зазначеною точністю. Ядром цих процедур є алгоритми, наведені у [7].

Для значного зменшення часу обчислень використовувалася адитивність інтеграла, тобто якщо координати вузла $X_{i,j}$ (образ $\omega_{i,j}$, де i, j є індексами вузла за координатами ξ та η відповідно) було вже обчислено, то координати сусіднього вузла, наприклад, $X_{i+1,j}$ можна знайти як

$$X_{i+1,j} = \pi K \left[\int_0^{\omega_{i,j}} \frac{(u_3 - \sin^2(\pi\omega/2))^{1-\alpha/\pi}}{(1 - \sin^2(\pi\omega/2))^{0.5-\alpha/\pi}} d\omega + \int_{\omega_{i,j}}^{\omega_{i+1,j}} \frac{(u_3 - \sin^2(\pi\omega/2))^{1-\alpha/\pi}}{(1 - \sin^2(\pi\omega/2))^{0.5-\alpha/\pi}} d\omega \right], \quad (20)$$

використовуючи, таким чином, вже обчислені значення та зменшуючи кількість операцій, що значно зменшує час обчислень.

Слід однак відзначити, що кожен частковий інтеграл обчислюється з заданою точністю, що може призвести до накопичування похибок при використанні формули (20). Тому точність чисельного інтегрування повинна бути ретельно обраною. У даній роботі використовувалося значення 10^{-7} . Воно було визнано придатним, тому що подальше покращення точності не приводило до зміни чисельних результатів.

8. Результати та їх обговорення

Розглянемо випадок, коли $\alpha = \beta = \pi/2$, тобто масив електродів з циліндричним заглибленням розташовано на площині. Очевидно, що на початку експерименту дифузійний шар, що створюється біля поверхні електрода, буде однаковим вздовж всього електрода і не залежатиме від радіальної координати. Тому струм, що спостерігається у системі, буде Котрелівським (тобто струмом, що спостерігається на планарному електроді), що у безрозмірному вигляді визначається таким рівнянням [13]:

$$\psi(\tau) = \frac{1}{4} \sqrt{\frac{\pi}{\tau}}, \quad (21)$$

що у розмірних термінах означає пропорційність струму площі електрода (πr_d^2) та зворотну пропорційність \sqrt{t} .

Далі, коли розмір дифузійного шару збільшиться до висоти заглиблення, в якому знаходиться електрод, ефекти двовимірної дифузії [2, 13] змусять відхилитися струм від планарної поведінки. Але коли розмір дифузійного шару буде дорівнювати відстані декількох ($h + r_c$), відгук системи буде знову Котрелівським (бо розподіл концентрації знову не буде залежати від радіальної координати), але пропорційним площі «комірки» одиничного електрода у масиві (πr_c^2) [1-2], тобто задовольнятиме такому безрозмірному рівнянню:

$$\psi(\tau) = \frac{R_c^2}{4} \sqrt{\frac{\pi}{\tau}}. \quad (22)$$

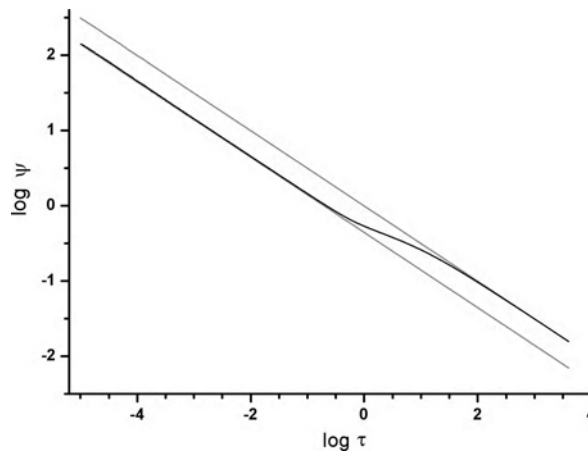


Рис. 4. Логарифмічна залежність струму від часу. Товста лінія – обчислений струм для $H = 1$;

$R_c = 1,5$; $\alpha = \beta = \pi/2$; тонкі лінії – граничні Котрелівські струми, обчислені за (21) та (22)

Підтвердження цього якісного аналізу зображено на рис. 4, де наведено обчислений струм для таких геометричних параметрів: $H = 1$; $R_c = 1,5$; $\alpha = \beta = \pi/2$, а також зображено два граничні Котрелівських струми, що обчислено за рівняннями (21) та (22). На рис. 4 також чітко виявлено перехід, обумовлений двовимірною дифузією в околі отвору заглиблення.

Для підтвердження точності отриманих чисельних результатів було досліджено чисельну збіжність обчислених струмів. Для цього було обчислено часову залежність струму (аналогічну тій, що зображено на рис. 4) у системі з фіксованою геометрією на послідовності сіток [14]. Струм, отриманий на сітці $N\xi \times N\eta = 800 \times 800$, вважався точним і відносна похибка ($\varepsilon = 100\% \times (\psi - \psi_{800}) / \psi_{800}$, де ψ_{800} – струм обчислений на сітці 800×800) чисельних розв'язків, отриманих на менш щільних сітках, обчислювалася відносно цього розв'язку. Результати цієї процедури для масиву електродів з параметрами $H = 1$, $R_c = 1,5$, $\alpha = \beta = \pi/2$ зображено на рис. 5. З цього рисунка можна бачити, що навіть сітка $N\xi \times N\eta = 100 \times 100$ є

задовільною, але для більшої впевненості для всіх наведених результатів використовувалася сітка $N\xi \times N\eta = 200 \times 200$. Похибка обчисленого струму на такій сітці не перевищує 0,35% протягом всього розглянутого часового інтервалу.

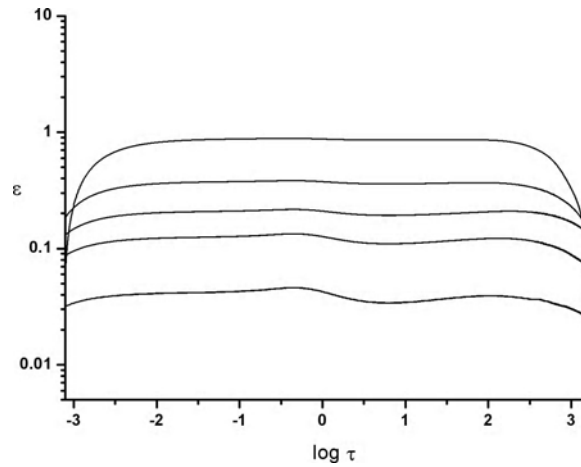


Рис. 5. Відносна похибка для струму, зображеного на рис. 4, що обчислено на послідовності сіток (зверху вниз): 100, 200, 300, 400 та 600

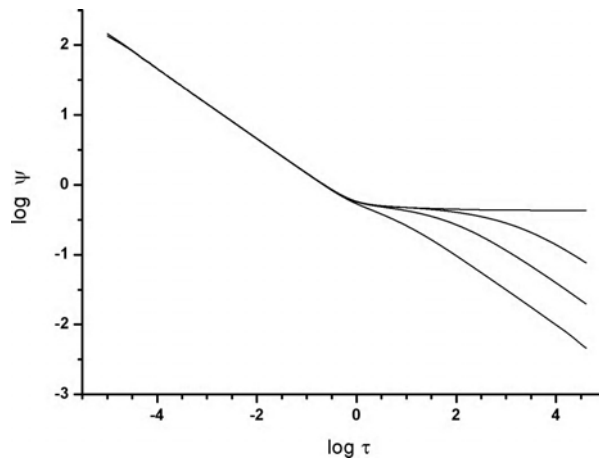


Рис. 6. Обчислений струм для $H = 1$, $\alpha = \beta = \pi/2$ та різних значень параметра R_c (знизу вгору):
 $R_c = 1,5; 3; 6$ та $R_c \rightarrow \infty$

Іншим граничним випадком у поведінці масивів електродів є тенденція відгуку системи до відгуку одиничного електрода у разі зростання відстані між електродами у масиві, тобто параметра R_c . На рис. 6 наведені струми системи, обчислені при збільшенні значення параметра R_c , а саме для $R_c = 1,5; 3$ та 6. Інші геометричні параметри були фіксовані: $H = 1$, $\alpha = \beta = \pi/2$. Також на рис. 6 наведена гранична поведінка одиничного електрода, обчисленого при тих же самих параметрах H , α та β .

Слід відзначити, що, як відомо, одиничні заглиблені мікродискові електроди мають стаціонарний режим [3]. Цей режим чітко спостерігається на рис. 6 (тобто незмінність струму при великих значеннях τ). Струми, обчислені у масивах заглиблених мікроелектродів, наближаються до струму одиничного електрода зі збільшенням R_c , але з часом Котрелівський режим набирає чинності і відгуки двох систем відхиляються один від одного. Але чим більше значення параметра R_c , тим пізніше це відбувається. Тому вказана вище тенденція спостерігається, що є додатковим доказом вірності отриманих чисельних розв'язків.

На рис. 7 та 8 наведено залежності струму за часом для випадку, коли кут $\alpha \neq \pi/2$. Розглядання таких систем є важливим з практичної точки зору, тому що вони мають значні застосування, як у разі, коли $\alpha < \pi/2$ [15-16], так і з $\alpha > \pi/2$ [17]. Отримані залежності також демонструють гнучкість застосованого підходу, тому що він дозволяє обчислювати відгуки систем з різною геометрією, змінюючи тільки параметри перетворення.

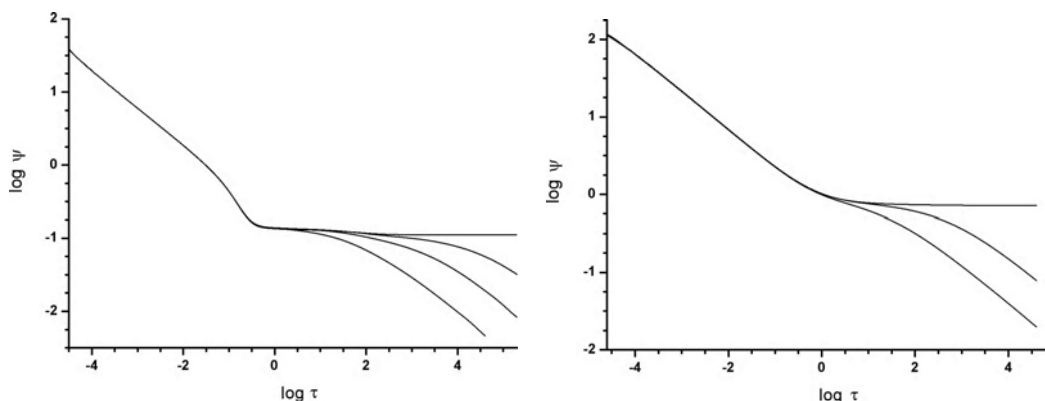


Рис. 7. Обчислений струм для $H = 0,75$, $\alpha = \pi/4$, $\beta = \pi/2$ та різних значень параметра R_c (знизу вгору): $R_c = 1,5; 3; 6$ та $R_c \rightarrow \infty$
 Рис. 8. Обчислений струм для $H = 1$, $\alpha = 3\pi/4$, $\beta = \pi/2$ та різних значень параметра R_c (знизу вгору): $R_c = 3; 6$ та $R_c \rightarrow \infty$

На рис. 7 наведено обчислені результати для масиву мікродискових електродів з $\alpha = \pi/4$. Як і у попередньому випадку, на початку експерименту незалежно від значення параметра R_c чітко спостерігається Котрелівський режим, за яким, однак, слідує досить різке падіння струму і тільки після нього спостерігається перехід до другого Котрелівського режиму. Падіння струму пояснюється зменшенням отвору пори, коли $\alpha = \pi/4$ (див. рис.1,б). Тому, коли дифузійний шар досягає отвору пори, швидкість масопереносу частинок до/від електрода уповільнюється відносно первісного значення на початку експерименту, викликаючи досить різку зміну струму. Надалі, коли у отворі пори встановлюється стаціонарний режим, спостерігається перехід від сферичної дифузії (що відповідає стаціонару в отворі пори) до планарної дифузії (що відповідає другому Котрелівському режиму), який спричиняє вгнутість кожної кривої струму, помітну в діапазоні $1 \leq \log \tau \leq 4$, залежно від значення параметра R_c .

Поведінка струму в системах з $\alpha > \pi/2$ є подібною розглянутій для систем з $\alpha = \pi/2$, за винятком того, що амплітуда струму є збільшеною та перехід між Котрелівськими режимами є коротшим. Перше пояснюється більш сприятливими умовами для масопереносу через більший отвір заглиблення. Друге – меншим контрастом двовимірної дифузії у перехідному режимі, що також обумовлено збільшенням кута в отворі пори. Згадані особливості легко помітні на рис. 8, де зображені безрозмірні струми, обчислені для системи з $\alpha = 3\pi/4$.

9. Висновки

Досліджено процеси дифузійного масопереносу в системах з масивами заглиблених дискових мікроелектродів. Математичну модель було сформульовано та чисельно розв'язано за допомогою побудованого квазіконформного відображення. Застосування квазіконформного відображення вирішує проблему сингулярності навколо отвору заглиблення мікроелектрода та спрощує область моделювання, в якій здійснюються обчислення. Крім того, заключна область моделювання є замкненою та однаковою для усіх можливих значень геометричних параметрів, які описують масив мікроелектродів, що значно спрощує та уніфікує процес чисельного розв'язку задачі. Це дозволило вперше отримати відгуки масивів дискових мікроелектродів, розташованих у конічних заглибленнях з різними кутами нахилу бокової стінки.

Отримані чисельні результати було перевірено на чисельну збіжність, а також на узгодженість з низкою граничних випадків. Проведений аналіз продемонстрував збіжність та точність отриманих результатів. Результати обчислень наведено у формі корисних залежностей для порівняння з експериментальними даними та визначення фізико-хімічних (початкова концентрація, дифузійні коефіцієнти тощо) та геометричних параметрів експериментальних систем.

Список літератури: **1.** *Amatore C., Savuiant J.-M., Tessier D.* Charge transfer at partially blocked surfaces. A model for the case of microscopic active and inactive sites // *J. Electroanal. Chem.* 1983. V. 147. P. 39-51. **2.** *Amatore C.,* Electrochemistry at Ultramicroelectrodes, in: I. Rubinstein, M. Dekker (Eds.), *Physical Electrochemistry: Principles, Methods and Applications*, New York, 1995 (Chapter 4). **3.** *Amatore C., Oleinick A., Svir I.* Construction of optimal quasi-conformal mappings for the 2D numerical simulation of diffusion at microelectrodes. Part 2. Application to recessed or protruding electrodes and their arrays // *J. Electroanal. Chem.* 2006. V. 597. P. 77-85. **4.** *Chevallier F.G., Fietkau N., Campo J. del, Mas R., Mucoz F. X., Jiang L., Jones T.G.J., Compton R.G.* Experimental cyclic voltammetry at partially blocked electrodes: Elevated cylindrical blocks: Significantly blocked and non-flat electrodes can appear to show one-dimensional diffusion // *J. Electroanal. Chem.* 2006. V.596(1). P. 25-32. **5.** *Amatore C.* Theoretical trends of diffusion-reaction into tubular nano- and mesoporous structures: A general physicochemical and physicomathematical modeling // *Chem. Eur. J.* 2008. V. 14(18). P. 5449-5464. **6.** *Лаврентьев М.А., Шабам Б.В.* Методы теории функций комплексного переменного. М.: Наука, 1973. 736 с. **7.** *Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P.* Numerical recipes in C: The art of scientific computing. 2nd ed., Cambridge University Press, Cambridge, 1992. 994 p. **8.** *Driscoll T.A., Trefethen L.N.* Schwarz-Christoffel mapping, Cambridge University Press, 2002. 132 p. **9.** *Amatore C., Oleinick A.I., Svir I.* Construction of optimal quasi-conformal mappings for the 2D-numerical simulation of diffusion at microelectrodes. Part 1: Principle of the method and its application to the inlaid disk microelectrode // *J. Electroanal. Chem.* 2006. V.597. P. 69-76. **10.** *Verbrugge M.W., Baker D.R.* Transient diffusion and migration to a disk electrode // *J. Phys. Chem.* 1992. V. 96. P. 4572-4580. **11.** *Флетчер К.* Вычислительные методы в динамике жидкости. М.: Мир. 1991. Т.1. 502 с. **12.** *Amatore C., Svir I.* A new and powerful approach for simulation of diffusion at microelectrodes based on overlapping sub-domains: application to chronoamperometry at the microdisk // *J. Electroanal. Chem.* 2003. V. 557. P. 75-90. **13.** *Bard A.J. and Faulkner L.R.* Electrochemical methods: fundamentals and applications. John Wiley & Sons: N-Y. 2002. 834 p. **14.** *Пухтмайер Р., Мортон К.* Разностные методы решения краевых задач. М.: Мир. 1972. 418 с. **15.** *Zhang B., Zhang Y., White H.S.* The nanopore electrode // *Anal. Chem.* 2004. V. 76. P. 6229-6238. **16.** *Lee S., Zhang Y., White H.S., Harrel C.C., Martin C.R.* Electrophoretic capture and detection of nanoparticles at the opening of a membrane pore using scanning electrochemical microscopy // *Anal. Chem.* 2004. V. 76. P. 6108-6115. **17.** *Aguiar F.A., Gallant A.J., Rosamond M.C., Rhodes A., Wood D., Katakya R.* Conical recessed gold microelectrode arrays produced during photolithographic methods: Characterisation and causes // *Electrochem. Commun.* 2007. V. 9(5). P. 879-885.

Надійшла до редколегії 05.12.2008

Олійник Олександр Ігорович, канд. техн. наук, старший науковий співробітник, докторант науково-дослідної лабораторії математичного та комп'ютерного моделювання ХНУРЕ. Наукові інтереси: чисельне та математичне моделювання фізико-хімічних та біологічних процесів. Адреса: Україна, 61166, Харків, пр. Леніна, 14, тел. 702-09-69.

ТЕХНОЛОГИИ ВСТРОЕННОГО РЕМОНТА КОМПОНЕНТОВ SYSTEM-IN-PACKAGE

Проводится анализ современных технологий встроенного сервисного обслуживания функциональностей цифровой системы в пакете. Рассматриваются особенности архитектуры «System-in-Package» и существующие стратегии восстановления работоспособности цифровых систем, а также метод оценки надежности восстановления их работоспособности.

1. Введение

В последние годы появились десятки работ, которые рассматривают вопросы, связанные с проблемой тестирования, диагностирования и ремонта цифровых систем на кристаллах и в пакетах (SoC, SiP) [1-24]. Особое место здесь занимает задача тестирования и ремонта компонентов цифровых систем ввиду технологической сложности восстановления работоспособности логических элементов при возникновении в них дефектов. Традиционные решения, предлагаемые в публикациях, сводятся к следующей классификации:

1. Дублирование логических элементов или областей кристалла, приводящее к удвоению аппаратной реализации функциональности. При фиксации неисправного элемента или области осуществляется переключение на исправный компонент с помощью мультиплексора [24]. Предложенные Xilinx модели FPGA применимы также при ремонте компонентов FPGA от компании Altera. Основная единица измерения при ремонте – столбец или строка.

2. Использование генетических алгоритмов для диагностирования и восстановления работоспособности на основе автономной конфигурации кристалла FPGA без использования внешних устройств управления [23]. Надежность диагностирования дефектов равна 99%, время ремонта – 36 миллисекунд вместо 660 секунд, необходимых для стандартного конфигурирования проекта.

3. Метод, не критичный к времени восстановления работоспособности FPGA, путем замены локальных CLB на избыточные запасные компоненты предложен в [20,22]. Доступный уровень объединения CLB, подлежащий замене, для критически важных приложений составляет порядка тысячи логических элементов.

Рассматривается структурно-технологический метод восстановления работоспособности логической части цифровых систем, имплементированных в кристаллы программируемой логики, основанный на существовании или внесении избыточности LUT-компонентов FPGA после выполнения процедур Place and Route. Анализируются физические дефекты кристалла, возникающие в процессе его изготовления или эксплуатации, которые проявляются как логические или временные, приводящие к неправильному функционированию цифровой схемы. Данные дефекты привязываются не только к вентилям или LUT-компонентам, но и к конкретному месту на кристалле. Идея метода сводится к исключению такого места при повторном выполнении процедур Place and Route после установления диагноза о наличии в нем дефекта. При этом возможны две технологии ремонта: 1) Запрет дефектной области путем написания управляющих скриптов для достаточно длительной процедуры Place and Route, что не всегда приемлемо для цифровых систем, работающих в реальном масштабе времени. Однако указанный подход приемлем и ориентирован на исключение дефектных областей любой кратности, имеющих на кристалле. Запрет таких областей при повторном выполнении процедуры Place and Route приводит к восстановлению работоспособности. 2) Для цифровых систем реального времени выполнение процедуры Place and Route при восстановлении работоспособности может привести к катастрофическим последствиям. Необходим технологический подход, способный восстановить функциональность цифровой системы за миллисекунды, необходимые для перепрограммирования FPGA путем подачи нового битстрима, исключаящего дефектные

области из формирования функциональности. Указанный подход может быть основан только на предварительной подготовке всех возможных битстримов, изолирующих будущие дефектные области путем их попадания в избыточную нефункциональную область кристалла. Чем больше такая резервная область, тем меньше число битстримов – проектных вариантов, привязанных к топологии, которое необходимо сгенерировать априори. Что касается кратных дефектов, не покрываемых одной резервной областью, здесь следует сегментировать цифровой проект, разбив его предварительно на непересекающиеся части, которые имеют собственные карты Place and Route. В данном случае можно ремонтировать цифровую систему, которая имеет для n распределенных на пластине дефектов n резервных сегментов. Здесь общая площадь кристалла состоит из $n+m$ одинаковых частей.

Цель исследования – анализ современного состояния технологий встроенного сервисного обслуживания функциональностей цифровой системы в пакете и определение подходов к решению наиболее актуальных задач восстановления работоспособности SiP.

Задачи: 1) анализ технологий встроенного восстановления работоспособности цифровой системы в пакете; 2) обзор моделей и методов восстановления работоспособности SiP.

2. Восстановление и ремонт SiP

FPGA широко используются для быстрого прототипирования и уменьшения стоимости сложных цифровых систем [20]. Модульная структура FPGA позволяет выполнять перепрограммирование в целях замены дефектных логических ресурсов (блоков) исправными резервными элементами. Указанное свойство дает возможность повысить отказоустойчивость приложений. Особенно это важно для систем, эксплуатируемых в сложных внешних условиях (космическое пространство, среды с повышенным уровнем радиации). Промышленные FPGA могут быть подвергнуты полному тестированию до реализации этапа программирования. Оперативный контроль FPGA и автономное тестирование кристалла возможны благодаря наличию специальных средств, применимых для FPGA. Однако использование кристаллов в критических приложениях обуславливает повышенный интерес к цифровым системам с возможностями оперативного контроля.

Систематические и неустойчивые неисправности могут быть обнаружены и локализованы путем использования различных методов тестирования. Использование определенной части ресурсов FPGA в качестве резервных позволяет заменить дефектный ресурс, в котором обнаружена систематическая неисправность. Замена происходит путем перепрограммирования кристалла при сохранении его функциональности. Схема, по которой резервные ресурсы распределяются внутри FPGA (и, следовательно, алгоритм реконfigurирования), зависит от типа кристалла. Использование процесса частичного конфигурирования позволяет существенно уменьшить среднее время восстановления и длину битового потока (bitstream), который обычно используется для реконfigurирования FPGA. Для выбора эффективной стратегии распределения резервных ресурсов важное значение имеет структура межсоединений кристалла.

В опубликованных работах [25-31] описаны различные технологии восстановления работоспособности FPGA, однако сравнительный анализ методов и метрика оценки их эффективности практически отсутствует. Поэтому далее рассматривается объект исследования, представленный в виде FPGA, его структура и особенности использования, а также существующие методы диагностирования и ремонта, использующие специфику перепрограммирования.

FPGA есть массив комплексных логических блоков (CLB – Complex Logic Block), каждый из которых представляет собой аппаратную реализацию от двух до восьми таблиц истинности на 4 входа; каждая из них соединена с триггером с помощью ресурсов трассировки, представляющих собой программируемые матрицы переключателей (PSM).

В целях обеспечения отказоустойчивости и надежности FPGA выполняется резервирование функциональной части CLB путем априорного формирования подмножества запасных блоков (spares), мощность которого зависит от структуры межсоединений кристалла.

Первый тип межсоединений ориентирован на матричную топологию CLB, когда оптимальная структура функциональности представлена соединением соседних по вертикали и горизонтали логических блоков. Если дефектный и резервный блоки не являются соседни-

ми в «манхэттенской» системе координат, то для восстановления работоспособности первого CLB необходимо выполнить достаточно сложную процедуру трассировки, используя матрицы переключателей. Описанная структура идеально подходит для обеспечения гибкости при выборе CLB при распределении резервных ресурсов. Недостатки: 1) Большое количество блоков, через которые проходит сигнал, приводит к снижению производительности системы путем увеличения длины логического пути, который зависит от числа используемых промежуточных переключательных матриц. 2) Процесс перепрограммирования межсоединений или трассировки является достаточно сложной задачей, когда замена дефектного логического блока требует значительного времени для поиска даже квазиоптимального пути.

Второй тип межсоединений представлен на рис. 1. Здесь нижний уровень иерархии связей позволяет соединить соседние блоки CLB. Группы блоков используют межсоединения высшего уровня иерархии, что позволяет уменьшить количество промежуточных переключательных матриц. Данная структура дает возможность эффективно использовать мозаичную (tile) и иерархическую стратегии распределения резервных ресурсов [18].

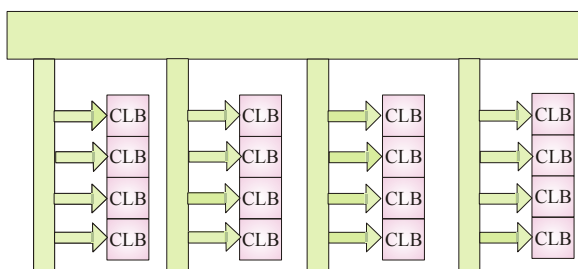


Рис. 1. Структура FPGA с иерархической топологией

Третий тип межсоединений основан на сегментации блоков CLB с помощью программируемых матриц переключателей, разбивающих логические компоненты на подмножества, как показано на рис. 2. Здесь имеются 4 сегмента по 9 логических блоков в каждом. Такая структура не требует выполнения процедуры перетрассировки, если резервный и дефектный блоки расположены на одной линии, вертикальной или горизонтальной, в пределах одного сегмента. В этом случае может быть использован «мозаичный» подход [18], позволяющий использовать резервный CLB для восстановления работоспособности соседнего по горизонтали или вертикали дефектного блока в пределах одного сегмента.

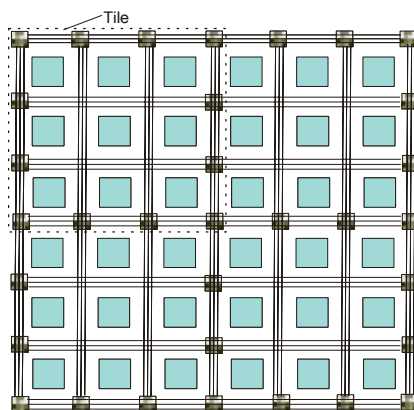


Рис. 2. Структура FPGA с частичной сегментацией связей

Интересным представляется также решение задачи, связанной с выбором стратегии восстановления, которая сводится к созданию протокола перепрограммирования FPGA. Здесь следует рассмотреть два метода частичного реконфигурирования. Первый позволяет определить ресурсы Atmel AT40K FPGA, которые должны быть перепрограммированы, используя пару управляющих регистров для выбора координаты конкретного блока в виде номера строки и столбца реконфигурируемого ресурса. После программирования новая конфигурация ресурсов может быть загружена в управляющие регистры. Данная стратегия

характеризуется незначительным временем исправления ошибки и высокой степенью детализации в процессе реконфигурирования кристалла. Вторым методом, используемым компанией Xilinx, заключается в разделении на столбцы перепрограммируемых ресурсов, что может обеспечивать степень детализации (ячеистость – granularity) не хуже, чем в предыдущем случае.

Для устранения постоянной во времени неисправности в кристалле FPGA используется модель процесса восстановления работоспособности, которая представлена на рис. 3.

Шаг 1. Обнаружение неисправностей в FPGA. До начала реконфигурирования выполняется процесс их диагностирования. Тестирование неисправностей происходит с использованием самопроверяемых схем (self-checking circuits). Приложение, имплементируемое в FPGA, разделяется на части, к каждой из которых добавляется избыточность, в результате чего функциональный модуль становится самопроверяемой схемой относительно дефектов, которые могут возникнуть внутри модуля. Степень детализации, или глубина диагностирования неисправностей, определяется количеством CLB, подозреваемых в наличии дефектов. Их поиск осуществляется в процессе непрерывного и последовательного циклического тестирования структурных компонентов FPGA, каждый из которых представляет собой совокупность блоков, объединенных в область самотестирования (self-testing areas – STARs). В каждый конкретный момент выполняется проверка только одной области, которая выводится из штатного режима путем ее замены резервом, в то время как оставшаяся часть FPGA продолжает функционировать в нормальном режиме. После завершения процедуры тестирования одной области осуществляется реконфигурирование FPGA в целях функционального отключения следующей области, подлежащей тестированию, и возврата проверенного структурного компонента в штатный режим. Описанная процедура позволяет автоматически устранять перемежающиеся (transient) неисправности с помощью конфигурационной памяти FPGA, которая сохраняет информацию о тестируемой области. Поскольку кристалл постоянно реконфигурируется для тестирования очередного компонента функциональности FPGA, процедура диагностирования имеет высокую степень детализации или глубину поиска дефектов, определяемую не более чем 10 CLB. Недостатком метода является зависимость времени диагностирования дефекта от длительности цикла тестирования функциональных компонентов FPGA, которая задается множеством самопроверяемых областей FPGA, полученным на стадии тестопригодного проектирования. Кроме того, наличие задержки между двумя событиями – возникновением и обнаружением дефекта – может приводить к неверному диагнозу о работоспособности системы в определенный момент времени, что приводит к нежелательным последствиям для критических систем реального времени.

Шаг 2. Позволяет различить перемежающиеся и систематические (константные) неисправности. Когда устройство диагностирования фиксирует факт наличия дефекта, выполняется обновление конфигурационной памяти FPGA. Затем выполняется инициализация таймера, контролирующего MTBF (Mean Time Between Failures – средняя наработка на отказ), для различения перемежающихся и систематических неисправностей. Если две ошибки обнаружены в одном месте в течение интервала времени меньшего, чем MTBF, фиксируется факт наличия систематической неисправности с последующим выполнением шагов 3 и 4.

Шаг 3. В случае обнаружения систематической неисправности, выполняется ее диагностирование со степенью детализации лучшей, чем при разделении схемы на самотестируемые модули. Для поиска дефектного CLB может быть использован метод, предложенный в [21].

Шаг 4. Ремонт дефектного модуля CLB. Механизм восстановления работоспособности непос-

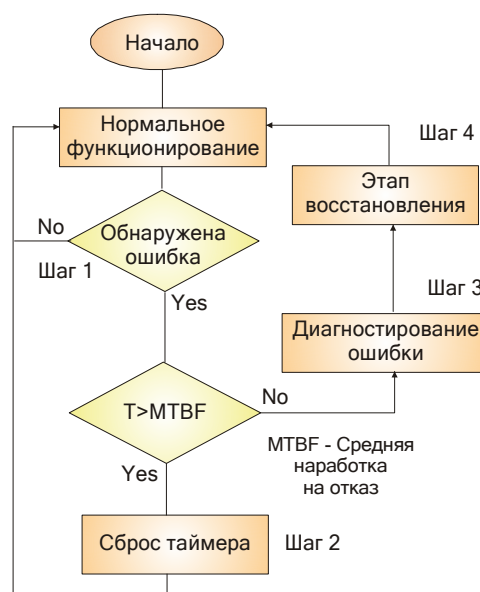


Рис. 3. Алгоритм восстановления работоспособности FPGA

редственно зависит от архитектуры FPGA. При этом используются различные технологии в зависимости от возможностей частичной или динамической реконфигурации FPGA, а также от структуры битового потока для перепрограммирования кристалла и межсоединений: 1) Иерархическая модель предполагает формирование двух уровней избыточности. На нижнем выполняется разделение FPGA на клетки (tile) – подматрицы блоков CLB, в каждой из них создаются резервные блоки. На верхнем уровне дефектные клетки могут быть заменены резервными клетками, как показано на рис. 4.

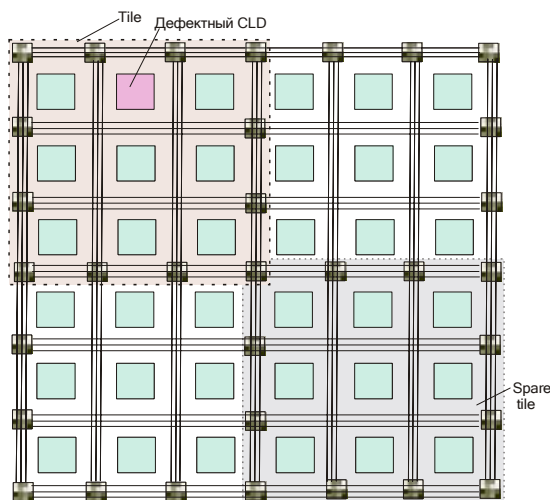


Рис. 4. Иерархическая модель восстановления работоспособности

2) Оптимальная модель предполагает, что резервные CLB могут быть использованы для восстановления работоспособности любого дефектного модуля цифровой системы на кристалле FPGA. При этом нет необходимости выполнять времязатратные процедуры перерасшировки (рис. 5).

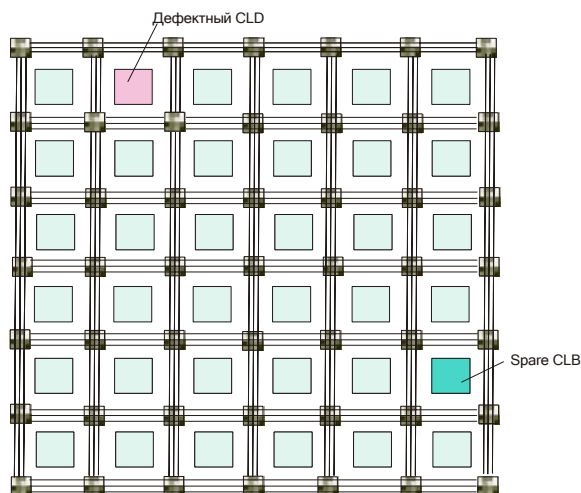


Рис. 5. Оптимальная модель восстановления работоспособности

Данный подход инвариантен к структуре FPGA, но требует существенного времени восстановления работоспособности и дополнительного объема памяти для хранения предварительно скомпилированного битового потока (bitstream), перепрограммирующего кристалл.

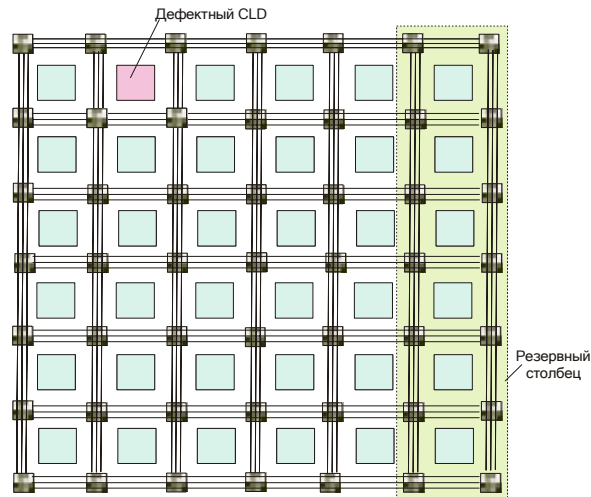
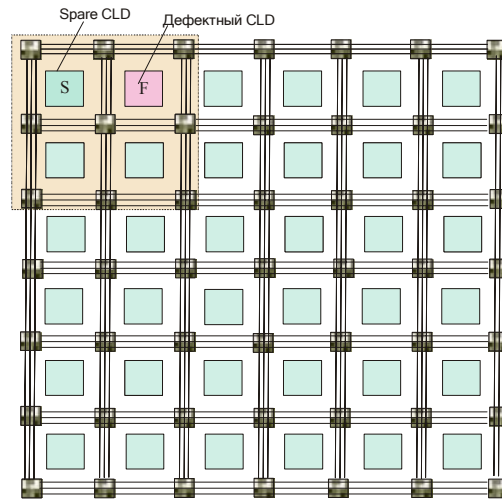


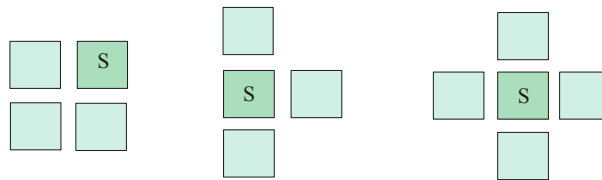
Рис. 6. Модель с избыточностью

Использование bitstream является обязательным, поскольку процедура перепрограммирования может затрагивать все ресурсы FPGA. Следовательно, должен быть реализован полный алгоритм place-and-route. Данная процедура является времязатратной и не может быть выполнена в режиме on-line. Она реализуется во время компиляции проекта с использованием методов уменьшения размерности предварительно скомпилированного битового потока. 3) Модель с избыточностью оперирует матрицей блоков CLB, разделенной на столбцы, среди которых имеется один или несколько избыточных, используемых для восстановления работоспособности цифровой системы (рис.6). Если неисправность обнаружена в функциональном столбце, он маркируется как дефектный и заменяется резервным. Модель процесса позволяет выполнять частичное реконфигурирование структуры FPGA путем разделения bitstream, как это используется в кристаллах компании Xilinx. Процедура реконфигурирования выполняется в реальном масштабе времени. Также имеется возможность упростить, а значит – ускорить ее реализацию с помощью укрупнения детализации функциональных модулей. Недостаток – при обнаружении дефектного CLB другие исправные блоки, принадлежащие столбцу, также маркируются как дефектные и выводятся из функционирования. 4) Мозаичная модель. Здесь структура FPGA делится на клетки, содержащие резервные CLB, каждый из которых может восстановить один дефектный блок в рассматриваемой клетке (рис.7).

В процессе диагностирования определяется местоположение дефектной области с глубиной детализации выше, чем размер клетки (ячейки). Это дает возможность заменить дефектный CLB резервным элементом клетки. Реконфигурация клетки обеспечивает исходную функциональность на новой карте. Межсоединения по периметру клетки с остальной частью FPGA должны быть неизменными в процессе реконфигурации. Такой подход позволяет уменьшить время восстановления работоспособности цифровой системы после обнаружения дефекта. Запасные конфигурации генерируются на стадии проектирования и хранятся в памяти. Каждая клетка формируется как набор блоков CLB и межсоединений FPGA с помощью описания интерфейса, который определяет и обеспечивает связи с другими клетками кристалла. Использование интерфейса клеток позволяет не вовлекать другие компоненты в процесс реконфигурирования, уменьшая используемую память. Модель позволяет восстанавливать не только CLB, но и локальные дефектные межсоединения. Ошибки глобальных межсоединений требуют других подходов, поскольку они пересекают периметр клеток, делая их зависимыми друг от друга. Структура клетки зависит от межсоединений FPGA (рис. 7, а). Как правило, один из нескольких CLB, принадлежащих клетке, является резервным элементом. Остальные блоки используются в штатном режиме. При обнаружении неисправности клетка реконфигурируется путем исключения из функционирования дефектного блока. На рис. 7,б показаны структуры клетки, форма которых отличается от квадратной. В кристаллах Atmel FPGA используются диагональные межсоединения, для которых модель, представленная на рис. 7,б, является более предпочтительной [22].



а



б

Рис. 7. Мозаичная модель восстановления: а – клетка с одним резервным и одним дефектным CLB; б – клетки разной формы

Иерархическая модель представляет собой наиболее общий случай восстановления работоспособности. Все остальные, описанные выше, могут рассматриваться как ее частные случаи. Оптимальная структура применима только на нижнем уровне иерархии, в то время как модель с избыточностью имеет резервные ресурсы только на верхнем уровне для замены дефектного столбца блоков. Мозаичная модель имеет избыточность на нижнем уровне иерархии, поэтому один из двух дефектных блоков, расположенных в одной клетке, является невозстанавливаемым. Для устранения данного недостатка следует использовать дополнительные резервные клетки.

При использовании двухуровневой архитектуры цифровой системы в кристалле FPGA вероятность ее исправного функционирования при наличии не более чем q дефектов, равного количеству резервных элементов, определяется выражением [20]:

$$P_{nf}(t) = \sum_{i=0}^q \binom{m}{i} P_{tile}(t)^{m-i} (1 - P_{tile}(t))^i,$$

где m – общее число клеток в FPGA или блоков CLB в клетке; q – количество резервных клеток или блоков в каждой из них.

Для сравнения четырех моделей восстановления работоспособности SiP в [20] использовались матричная (квадратная) структура FPGA (число строк и столбцов равно 100) с резервом CLB, равным 25%. 1) Оптимальная модель: количество CLB, используемых в процессе функционирования системы, равно 7500, резервных элементов – 2500. 2) Мозаичное восстановление: каждая клетка (tile) включает 4 CLB, 3 из которых используются в процессе функционирования и один является резервным, количество клеток равно 2500. 3) Модель с избыточностью: количество столбцов равно 100, каждый столбец состоит из 100 CLB, 75 из которых используется в работе и 25 – резервные. 4) Иерархическая модель: количество клеток равно 2000, каждая из них содержит 5 CLB, 4 из которых – функциональные и один – резервный. Используется 1875 рабочих клеток и 125 резервных. Общее число резервных блоков CLB равно 2500.

Для малых значений интенсивности отказов надежность восстановления с избыточностью превосходит надежность ремонта на основе мозаичной модели. При увеличении интенсивности отказов надежность первого подхода резко падает ввиду ограниченного количества резервных элементов, а надежность второго – плавно уменьшается благодаря меньшей избыточности блоков CLB для восстановления работоспособности. Таким образом, существующие структуры восстановления работоспособности можно разделить на две группы: 1) Модели, ориентированные на замену единичного элемента (оптимальная и мозаичная). 2) Модели, ориентированные на замену группы дефектных элементов (иерархическая и модель с избыточностью).

Реализация рассмотренных в работе моделей восстановления работоспособности цифровых систем основана на выполнении времязатратной процедуры Place and Route или использовании управляющих регистров для оперативного реконфигурирования структуры FPGA в реальном масштабе времени в целях исключения из процесса функционирования дефектных блоков.

Список литературы: 1. Wang F. Z., Wu S., Helian N., Parker M. A., Guo Y., Deng Y., and Khare V. R. Grid-Oriented Storage: A Single-Image, Cross-Domain, High-Bandwidth Architecture // IEEE Transactions on Computers. 2007. P.474–487. 2. Hamdioui S., Gaydadjiev G. N., van de Goor A. J. The State-of-the-art and Future Trends in Testing Embedded Memories // Records IEEE International Workshop on Memory Technology, Design, and Testing, San Jose, CA, August 2004. 2004. P. 54–59. 3. Zhong Y., Dropsho S. G., Shen X., Studer A., Ding C. Miss Rate Prediction Across Program Inputs and Cache Configurations // IEEE Transactions on Computers. 2007. P. 328–343. 4. Memory Repair Primer – A guide to understanding embedded memory Repair options and issues. Logic Vision. 2007. 5. Shoukourian S., Vardanian V., Zorian Y. SoC Yield Optimization via an Embedded-Memory Test and Repair Infrastructure // IEEE Design and Test of Computers. 2004. P. 200–207. 6. Youngs L., Paramanandam S. Mapping and Repairing Embedded-Memory Defects // IEEE Design and Test of Computers. 1997. P. 18–24. 7. Zorian Y., Shoukourian S. Embedded-Memory Test and Repair: Infrastructure IP for SoC Yield // IEEE Design and Test of Computers. 2003. P. 58–66. 8. Huang R., Chen Ch., Wu Ch. Economic Aspects of Memory Built-in Self-Repair // IEEE Design & Test. 2007. P. 164–172. 9. Choi M., Park N., Lombardi F., Kim Y. B., Piuri V. Optimal Spare Utilization in Repairable and Reliable Memory Cores // 2003 International Workshop on Memory Technology, Design and Testing (MTDT'03). 2003. P. 64–71. 10. Ohler Ph., Hellebrand S., Wunderlich H.-J. An Integrated Built-In Test and Repair Approach for Memories with 2D Redundancy // 12th IEEE European Test Symposium (ETS'07). 2007. P. 91–96. 11. Kwang-Ting (Tim) Cheng. The Need for a SiP Design and Test Infrastructure // IEEE Design and Test of Computers. May–June, 2006. P. 181. 12. Peter Rickert, William Krenik. Cell Phone Integration: SiP, SoC, and PoP // IEEE Design and Test of Computers. May–June, 2006. P. 188-195. 13. FSA SiP Market and Patent Analysis Report. FSA SiP Subcommittee // IEEE Design & Test of Computers. Vol. 24, Issue 2, March–April, 2007. P. 184-192. 14. Lee Whetsel. System-in-Package Testing Using Existing IEEE Test Standards // International Test Conference 2001 (ITC'01). 2001. P. 1167. 15. Ajay Khoche. System-in-Package is Coming to Consumer Products: Is Test Ready? // Proceedings of the International Test Conference 2001 (ITC'01). 2001. P. 1166. 16. Fontanelli A. System-in-Package Technology: Opportunities and Challenges // Quality Electronic Design, 2008. ISQED 2008, 9th International Symposium. March, 2008. P. 589–593. 17. Lim S.K. Physical design for 3D system on package // IEEE Design & Test of Computers. Vol. 22, Issue 6. Nov.-Dec., 2005. P. 532 – 539. 18. Tummala R.R., Madiseti, V.K. System on chip or system on package? // IEEE Design & Test of Computers. Volume 16, Issue 2. April-June, 1999. P. 48 – 56. 19. Appello D., Bernardi P., Grosso M., Reorda M.S. System-in-package testing: problems and solutions // IEEE Design & Test of Computers. Vol. 23, Issue 3. May-June, 2006. P. 203 – 211. 20. Pontarelli S., Ottavi M., Vankamamidi V., Salsano A., Lombardi F. Reliability Evaluation of Repairable/Reconfigurable FPGAs // 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06). October, 2006. P. 227-235. 21. Wang S.-J., Tsai T.-M. Test and diagnosis of faulty logic blocks in FPGAs // IEEE Proceedings Computers and Digital Techniques. Vol. 146, Issue 2. March, 1999. P.100 – 106. 22. Pontarelli S., Cardarilli G.C., Malvoni A., Ottavi M., Re M., Salsano A. System-on-chip oriented faulttolerant sequential systems implementation methodology // IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems. Oct., 2001. P. 455 - 460. 23. Ross R., Hall R. A FPGA Simulation Using Asexual Genetic Algorithms for Integrated Self-Repair // Adaptive Hardware and Systems, 2006. AHS 2006. First NASA/ESA Conference on Volume. Issue 15-18 June 2006. P. 301-304. 24. Subhasish Mitra, Huang W.-J., Saxena N.R., Yu S.-Y., McCluskey E.J. Reconfigurable architecture for autonomous self-repair // IEEE Design & Test of Computers. May-June, 2004. Vol. 21, Issue 3. P. 228- 240. 25. Koal T., Vierhaus H. T. Basic Architecture for Logic Self Repair // 14th IEEE International On-Line Testing Symposium. 2008. P. 177–178. 26. Kothe R., Vierhaus H.T., Coym T., Vermeiren W., Straube B. Embedded Self Repair by Transistor and Gate Level Reconfiguration // IEEE Design and Diagnostics of Electronic Circuits and Systems. 2006. P. 208–213. 27. Mange D., Sipper M., Stauffer A., Tempesti G. Toward self-repairing and

self-replicating hardware: the Embryonics approach // Evolvable Hardware, Proceedings of the Second NASA/DoD Workshop. 2000. P. 205-214. **28.** Subhasish M.; Huang W.-J., Saxena N.R., Yu S.-Y., McCluskey E.J. Reconfigurable architecture for autonomous self-repair // IEEE Design & Test of Computers.– Volume 21, Issue 3. May-June, 2004. P. 228-240. **29.** Miclea L., Szilard E., Benso A. Intelligent agents and BIST/BISR - working together in distributed systems // Proceedings of the Test Conference. 2002. P. 940-946. **30.** Rashad S. Oreifej, Carthik A. Sharma, Ronald F. DeMara. Expediting GA-Based Evolution Using Group Testing Techniques for Reconfigurable Hardware // Reconfigurable Computing and FPGA's, 2006. ReConFig 2006. IEEE International Conference. Sept., 2006. P. 1-8. **31.** Anand D., Cowan B., Farnsworth O., Jakobsen P., Oakland S., Ouellette M.R., Wheeler D.L. An on-chip self-repair calculation and fusing methodology // IEEE Design & Test of Computers. Volume 20, Issue 5. Sept.-Oct., 2003. P. 67-75.

Поступила в редколлегию 14.12.2008

Литвинова Евгения Ивановна, канд. техн. наук, доцент кафедры технологии и автоматизации производства РЭС и ЭВС ХНУРЭ. Научные интересы: алгоритмизация задач автоматизированного проектирования электронных вычислительных средств, автоматизация диагностики и встроенный ремонт компонентов цифровых систем в пакете (SiP). Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-421.

УДК 681.5:69:621:039

А.Д. ТЕВЯШЕВ, Д.А. ЗОЛОТАРЕВ

ОБ ОДНОМ МЕТОДЕ РЕШЕНИЯ ЗАДАЧИ ОПТИМИЗАЦИИ ПЛАНОВЫХ РЕЖИМОВ ТРАНСПОРТА И РАСПРЕДЕЛЕНИЯ ПРИРОДНОГО ГАЗА В ГОРОДСКИХ ГАЗОРАСПРЕДЕЛИТЕЛЬНЫХ СЕТЯХ

Рассматривается математическая постановка задачи оптимизации плановых режимов транспорта и распределения природного газа в многоуровневых газораспределительных сетях и метод ее решения. Данная задача относится к классу многокритериальных задач математического программирования с алгоритмически заданной недифференцируемой целевой функцией, алгоритмически заданными недифференцируемыми критериальными ограничениями, нелинейными ограничениями в виде равенств при двусторонней ограниченности переменных. Используются три модифицированных метода: деформируемого многогранника Нелдера-Мида; гидравлического расчета газораспределительных сетей; метод статистической линеаризации системы неявно заданных функций.

1. Введение

В работе [1] была приведена математическая постановка задачи оптимизации плановых режимов транспорта и распределения природного газа в многоуровневых газораспределительных сетях (МГРС) в виде многокритериальной задачи нелинейного стохастического программирования и получен ее детерминированный эквивалент.

В настоящее время не существует общих методов решения задач такого типа. Специфической особенностью этой задачи является ее огромная размерность, особенно для газораспределительных сетей низкого и среднего давления. Это обстоятельство исключает возможность ее непосредственного решения.

Целью данного исследования является разработка эффективного метода решения поставленной задачи, основанного на декомпозиции исходной задачи на k (где k – количество уровней газораспределительной сети) однотипных задач оптимизации, которые решаются последовательно, начиная с нижнего уровня. Координация оптимальных решений между $(k+1)$ -м и k -м уровнями осуществляется путем учета условий согласования параметров газовых потоков, определяемых математическими моделями регуляторов давления (ГРП, ПРП) и дополнительными условиями, характеризующими качество функционирования газораспределительной сети в зоне нагрузки соответствующего регулятора давления.

Для достижения поставленной цели решается задача оптимизации режима транспорта и распределения природного газа на каждом из k уровней МГРС. Это осуществляется на

основе трех модифицированных методов: деформируемого многогранника Нелдера-Мида, обобщенного на случай решения задач условной оптимизации; гидравлического расчета, обобщенного на случай расчета многоуровневых газораспределительных сетей и алгоритмически заданной нижней границей минимально-допустимого давления на каждом из ее выходов; расчета статистических свойств зависимых переменных математической модели стационарного режима транспорта и распределения природного газа в МГРС в зависимости от статистических свойств независимых переменных на основе модифицированного метода статистической линейаризации системы нелинейных неявно заданных функций.

2. Математическая постановка задачи оптимизации плановых режимов транспорта и распределения газа в МГРС

Детерминированный эквивалент стохастической задачи оптимизации плановых режимов транспорта и распределения природного газа в МГРС на интервале времени $[0, T]$ представляет собой задачу нелинейного математического программирования вида [1]:

$$J_0 = \sum_{k=1}^K \sum_{j=1}^{N^k} (\bar{P}_j^k(T) - P_j^{k-}) \rightarrow \min_{q \in L^k \in \Omega}; \quad (1)$$

$$\Omega_1: \min_{k=1, K} \min_{j \in L^k} \frac{1}{2} \Phi \left(\frac{m_{q_j^+} + 3\sigma_{q_j^+} - m_{q_j^k}}{\sigma_{q_j^k}} \right) - \frac{1}{2} \Phi \left(\frac{m_{q_j^-} - 3\sigma_{q_j^-} - m_{q_j^k}}{\sigma_{q_j^k}} \right) \geq \alpha; \quad (2)$$

$$\max_{k \in K} \max_{j \in N^k} \frac{1}{2} \Phi \left(\frac{m_{p_j^-} - 3\sigma_{p_j^-} - m_{p_j^k}}{\sigma_{p_j^k}} \right) \leq \beta, \quad (3)$$

$$\Omega_k': c_r \bar{q}_r |\bar{q}_r|^{\gamma-1} + \sum_{i \in M_1} b_{1ir} c_i \bar{q}_i |\bar{q}_i|^{\gamma-1} = 0, \quad r \in M_2^k; \quad (4)$$

$$\bar{P}_1^\alpha - \bar{P}_r^\alpha + \sum_{i \in M_1} b_{1ir} c_i \bar{q}_i |\bar{q}_i|^{\gamma-1} = 0, \quad r \in L_{22}^k; \quad (5)$$

$$q_r = \sum_{i \in M_2^k \cup L_{22}^k \cup N_{22}^k} b_{1ir} \bar{q}_i + \sum_{i \in L_{21}^k \cup N_{21}^k} b_{1ir} \bar{q}_i, \quad r \in M_1^k \cup L_1^k; \quad (6)$$

$$P_j^{(k)-} \leq \bar{P}_j^{(k)} \leq P_j^{(k)+}, \quad j \in N_2^k; \quad (7)$$

$$\Omega_k'': q_i^{(k)-} \leq q_i^{(k)} \leq q_i^{(k)+}, \quad i \in L^k, \quad (8)$$

где неравенства (2) и (3) определяют область Ω_1 , ограничения в виде равенств (4) – (6) и неравенство (7) определяют область Ω_k' , двусторонняя ограниченность переменных $\bar{q}_j^{(k)}$ (8) определяет область Ω_k'' ; $k = \overline{1, K}$ – количество уровней сети (обычно $K = 3$); $j = \overline{1, N^k}$ – количество выходов на k -м уровне сети; $\Omega = \Omega_1 \cup \Omega_2$, где Ω_1 – область, на которой выполняются неравенства (2) и (3), $\Omega_2 = \bigcup_{k=1}^K \Omega_k$, где Ω_k – область, на которой выполняются ограничения типа равенств (4) – (8) на k -м уровне, $\Omega_k = \Omega_k' \cup \Omega_k''$; M^k – множество реальных дуг графа сети: M_1^k – реальные ветви, M_2^k – фиктивные хорды; L^k – множество фиктивных дуг графа по входам сети: L_1^k – фиктивные ветви, L_2^k – фиктивные хорды; N_2^k – множество фиктивных хорд графа по выходам сети; $\bar{P}_j, j \in N_2^k$ – математическое ожидание давления на выходах сети k -го уровня; $\bar{q}_i, i \in L^k$ – математическое ожидание расхода на входах сети k -го уровня; c_i – коэффициент гидравлического сопротивления i -го участка трубопровода; b_{1ir} – элементы цикломатической матрицы для сети k -го уровня; α, β –

коэффициенты нелинейности модели транспорта газа по участку трубопровода для сетей высокого, среднего и низкого давления[9].

Критерий (1) характеризует сумму избыточных давлений во всех узлах многоуровневой газораспределительной сети, минимизация которой обеспечивает сокращение производственных потерь природного газа, минимизацию рисков возникновения аварийных ситуаций и техногенных катастроф, связанных с утечками природного газа (взрывами, пожарами и т.п.).

Критериальное ограничение (2) гарантирует, что режимная устойчивость всех регуляторов давления (ГРС, ГРП, ПРП) на всех уровнях газораспределительной сети будет больше либо равна некоторому фиксированному числу $\alpha \approx 1$. При этом минимизируется вероятность автоматического отключения групп потребителей, находящихся в зоне нагрузки соответствующего регулятора (ГРС, ГРП, ПРП), и соответствующие потери, связанные с восстановлением режимов.

Выполнение критериального ограничения (3) гарантирует, что вероятность возникновения дефицита природного газа во всех узлах газораспределительной сети будет меньше либо равна заданной величине $\beta \approx 0$.

Связь между k -м и $(k+1)$ -м уровнями осуществляется с помощью модели регулятора, которая имеет вид:

$$P_j^{(k-1)-} = f(P_j^{(k)}, q_j^{(k)+}) \quad (9)$$

и связывает между собой минимально-допустимое давление на входе регулятора $P_j^{(k-1)-}$ с максимальным расходом $q_j^{(k)+}$ и величиной стабилизируемого давления $P_j^{(k)}$ на его выходе.

3. Алгоритм решения задачи оптимизации плановых режимов транспорта и распределения газа в МГРС

Исходными данными для решения задачи (1) – (9) на интервале времени $[0, T]$ является нормативно-справочная информация о структуре и параметрах газопровода для каждого из k уровней МГРС.

Оперативные данные: плановые прогнозируемые значения математических ожиданий и дисперсий расходов $\bar{q}_j, \sigma_{q_j}^2, j \in N_2^k, k = \overline{1, K}$ всех потребителей МГРС k -го уровня; дисперсии стабилизируемых давлений на выходах всех регуляторов давлений (ГРС, ГРП, ПРП) на входах каждого из уровней МГРС $\sigma_{P_i}^2, i \in L^k, k = \overline{1, K}$.

Результатом решения задачи являются: оптимальное значение математических ожиданий $\bar{q}_j^{(k)*}, j \in L^k, k = \overline{1, K}$ плановых расходов газа на всех регуляторах давления на каждом из k уровней, при котором достигает минимума целевая функция (1) и выполняются критериальные ограничения (2) – (3), ограничения в виде равенств (4) – (6) и системы неравенств (7) – (8); дисперсии прогнозируемых расходов природного газа через каждый из регуляторов $\sigma_{q_i}^2, i \in L^k, k = \overline{1, K}$ и дисперсии давлений на входах каждого из потребителей $\sigma_{P_j}^2, j \in N_2^k, k = \overline{1, K}$; математические ожидания давлений во всех узлах газораспределительной сети каждого из k уровней и математические ожидания расходов по каждому реальному участку трубопровода на каждом из k уровней; оптимальное значение целевой функции I_0^* .

Для обеспечения выполнения системы ограничений, входящих в область Ω_k , используется гидравлический расчет сети k -го уровня с применением метода «диктующей точки».

3.1. Алгоритм расчета сети k-го уровня и определение местоположения «диктующей точки»

Известны следующие начальные данные на каждом из k уровней сети для интервала времени $[0, T]$: нормативно-справочная информация для каждого из k уровней; математические ожидания расходов на входах k-го уровня: $\bar{q}_j, j \in L_2^k$; математические ожидания расходов на выходах k-го уровня: $\bar{q}_i, i \in N_2^k$; минимально-допустимые давления на выходах k-го уровня: $P_i^-, i \in N_2^k$.

Решается задача гидравлического расчета сети k-го уровня, которая сводится к системе уравнений (10) – (11).

Для получения начального распределения давлений решаем следующую систему уравнений:

$$f_r = c_r \bar{q}_r | \bar{q}_r | + \sum_{i \in M_1} b_{1ri} c_i \bar{q}_i | \bar{q}_i | = 0, \quad r \in M_2^k, \quad (10)$$

$$\bar{q}_i = \sum_{r \in M_2^k \cup L_{22}^k \cup N_{22}^k} b_{1ri} \bar{q}_r + \sum_{r \in L_{21}^k \cup N_{21}^k} b_{1ri} \bar{q}_r, \quad i \in M_1^k \cup L_1^k. \quad (11)$$

В результате решения системы (10) – (11) получаем перепады давлений по всем дугам k-го уровня сети. Далее находится «диктующая точка» сети k-го уровня. «Диктующая точка» – это номер выхода (ГРП), выполнение ограничения на давление в которой влечет за собой выполнение этих условий на всех остальных ГРП (т.е. на всех ГРП, кроме «диктующего», устанавливается не менее чем минимально-допустимое давление).

Для определения «диктующей точки» будем использовать следующее уравнение:

$$f_r = \bar{P}_r^2 - P_1^2 + \sum_{i \in M_1} b_{1ri} c_i \bar{q}_i | \bar{q}_i | = 0, \quad r \in N_{22}^k. \quad (12)$$

Для r-го выхода ($r \in N_{22}^k$) задаем давление, равное минимально-допустимому P_r^- . Далее рассчитываем по формуле (12) математическое ожидание давления на входе \bar{P}_1 (давление в начале 1-й дуги дерева, соединенной с нулевой вершиной графа). Пропуская указанную операцию для всех $r \in N_{22}^k$ выходов, находим максимальное давление \bar{P}_1 . Соответствующий ему номер выхода и будет «диктующей точкой».

Используя найденное давление \bar{P}_1 , находим математические ожидания давлений на выходах. Для этого будем использовать ту же формулу (12), но в обратном направлении: известно \bar{P}_1 , не известны $\bar{P}_r, r \in N_{22}^k$.

Необходимым условием успешного поиска «диктующей точки» является выполнение неравенства $\bar{P}_r \geq P_r^-$ для найденных таким образом $\bar{P}_r, r \in N_{22}^k$, причем в самой «диктующей точке» должно достигаться равенство: $\bar{P}_r^* = P_r^-$.

3.2. Вычисление оценок статистических свойств зависимых переменных в зависимости от статистических свойств независимых переменных

Для выполнения ограничений на расход и давление, заданных неравенствами (7) и (8), используется метод дисперсий.

На интервале времени $[0, T]$ даны:

1. Математические ожидания давлений на входах k-го уровня сети: $\bar{P}_j, j \in L^k$.
2. Математические ожидания расходов на выходах k-го уровня сети: $\bar{q}_i, i \in N_2^k$.
3. Дисперсии давлений на входах k-го уровня сети: $\sigma_{P_j}^2, j \in L^k$.
4. Дисперсии расходов на выходах k-го уровня сети: $\sigma_{q_i}^2, i \in N_2^k$.

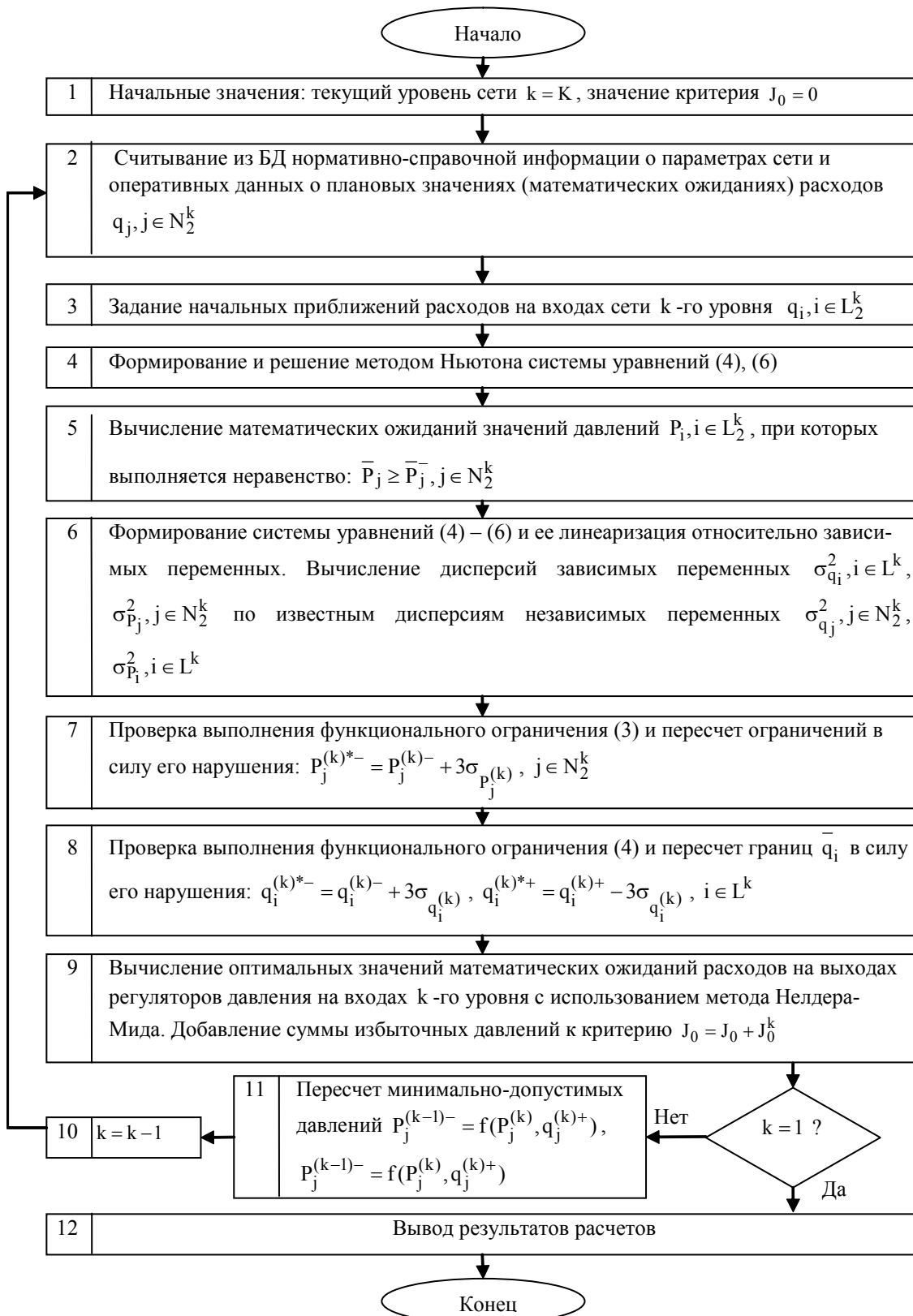


Рис. 1. Блок-схема общего алгоритма решения

Требуется найти:

1. Дисперсии расходов на входах k-го уровня сети: $\sigma_{q_j}^2, j \in L^k$.

2. Дисперсии давлений на выходах k -го уровня сети: $\sigma_{P_i}^2, i \in N_2^k$.

Представим зависимые переменные $q_j, j \in L^k$ и $P_i, i \in N_2^k$ в виде неявных функций:

$$\bar{q}_j = q_j(\bar{P}_j, j \in L^k; \bar{q}_i, i \in N_2^k), j \in L^k; \quad (13)$$

$$\bar{P}_i = P_i(\bar{P}_j, j \in L^k; \bar{q}_i, i \in N_2^k), i \in N_2^k. \quad (14)$$

Для получения явного выражения для дисперсий воспользуемся методом статистической линеаризации. Получим:

$$\sigma_{q_j}^2 = \sum_{i \in N_2^k} \left[\frac{\partial \bar{q}_j}{\partial q_i} \right]^2 \sigma_{q_i}^2 + \sum_{i \in L^k} \left[\frac{\partial \bar{q}_j}{\partial P_i} \right]^2 \sigma_{P_i}^2, j \in L^k; \quad (15)$$

$$\sigma_{P_j}^2 = \sum_{i \in N_2^k} \left[\frac{\partial \bar{P}_j}{\partial q_i} \right]^2 \sigma_{q_i}^2 + \sum_{i \in L^k} \left[\frac{\partial \bar{P}_j}{\partial P_i} \right]^2 \sigma_{P_i}^2, j \in N_2^k. \quad (16)$$

Эти выражения получены в предположении, что ковариационная матрица случайных величин $\bar{P}_j, j \in L^k$ и $\bar{q}_i, i \in N_2^k$ диагональна (т.е. случайные величины независимы).

Для вычисления дисперсий далее требуется найти частные производные зависимых переменных по независимым. Воспользуемся следующей системой уравнений:

$$f_r = c_r \bar{q}_r | \bar{q}_r | + \sum_{i \in M_1} b_{1ri} c_i \bar{q}_i | \bar{q}_i | = 0, r \in M_2^k; \quad (17)$$

$$f_r = \bar{P}_1^2 - \bar{P}_r^2 + \sum_{i \in M_1} b_{1ri} c_i \bar{q}_i | \bar{q}_i | = 0, r \in L_{22}^k; \quad (18)$$

$$f_r = -\bar{q}_r + \sum_{i \in M_2^k \cup L_{22}^k \cup N_{22}^k} b_{1ir} \bar{q}_i + \sum_{i \in L_{21}^k \cup N_{21}^k} b_{1ir} \bar{q}_i, r \in M_1^k \cup L_1^k. \quad (19)$$

Искать производные будем из решения следующих систем линейных уравнений [2]:

$$S_r = \begin{cases} \frac{\partial f_1}{\partial x_r} + \frac{\partial f_1}{\partial y_1} \cdot \frac{\partial y_1}{\partial x_r} + \dots + \frac{\partial f_1}{\partial y_m} \cdot \frac{\partial y_m}{\partial x_r} = 0, \\ \dots \\ \frac{\partial f_m}{\partial x_r} + \frac{\partial f_m}{\partial y_1} \cdot \frac{\partial y_1}{\partial x_r} + \dots + \frac{\partial f_m}{\partial y_m} \cdot \frac{\partial y_m}{\partial x_r} = 0, \end{cases} \quad (20)$$

где $f_r, r \in \overline{1, m}$ – уравнения системы (17) – (19); $x_r \in V_x = \{\bar{P}_j, j \in L^k; \bar{q}_i, i \in N_2^k\}$ – независимые переменные; $y_r \in V_y = \{q_j, j \in L^k; P_i, i \in N_2^k\}$ – зависимые переменные.

Систем такого вида будет $r = \overline{1, \text{Card}(V_x)}$. Эти системы однозначно разрешимы, так как однозначно разрешима исходная система (17) – (19). Соответственно, их определители отличны от нуля:

$$J_r = \frac{D(F_1, \dots, F_m)}{D(y_1, \dots, y_m)} \neq 0.$$

В явном виде решение каждой из систем можно записать следующим образом:

$$\frac{\partial y_1}{\partial x_r} = - \frac{D(F_1, \dots, F_m)}{D(x_r, \dots, y_m)}, \dots, \frac{\partial y_m}{\partial x_r} = - \frac{D(F_1, \dots, F_m)}{D(F_1, \dots, F_m)}.$$

3.3. Минимизация суммы избыточных давлений в узлах МГРС

Суть задачи оптимизации сводится к такой последовательности шагов:

1. При заданных расходах на входах $\bar{q}_j, j \in L^k$ и выходах $\bar{q}_i, i \in N_2^k$ k -го уровня сети находим посредством гидравлического расчета (10) – (11) перепады давлений на всех участках трубопровода.
2. Находим диктующую точку (см. пункт 3.1).
3. Основываясь на давлении в «диктующей точке», находим давления на остальных выходах k -го уровня сети.
4. Далее находим сумму разностей текущего давления и минимально-допустимого на

каждом выходе k -го уровня сети: $\sum_{i \in N_2^k} (\bar{P}_j - P_j^-)$.

Полученная сумма и будет суммой избыточных давлений для k -го уровня.

Для минимизации этой суммы по переменным $q_j, j \in L^k$ будем использовать модифицированный метод Нелдера-Мида. Его отличием от классического [3] является наличие проекции точки оптимизации на многогранник после проведения каждой из операций: отражение, растяжение, сжатие, редукция.

Для проецирования на многогранник требуется указать нижнюю и верхнюю границу для расходов $\bar{q}_j, j \in L^k$, а для подсчета суммы избыточных давлений – нижнюю границу (минимально-допустимое давление) для выходов k -го уровня сети: $P_i^-, i \in N_2^k$. Но учитывая ранее введенные критериальные ограничения, скорректируем границы изменения расходов:

$$q_j^{*-} \leq q_j \leq q_j^{*+}, j \in L^k,$$

где $q_j^{*-} = q_j^- + 3\sigma_{q_j}, j \in L^k$; $q_j^{*+} = q_j^+ - 3\sigma_{q_j}, j \in L^k$.

При этом гарантируется, что полученное решение будет фактически более устойчивым, чем расчетное.

Также скорректируем нижнюю границу давления на каждом из входов потребителей $P_i^-, i \in N_2^k$:

$$P_i^{*-} = P_i^- + 3\sigma_{P_i}, i \in N_2^k.$$

В этом случае удовлетворение потребителя в необходимом количестве природного газа будет происходить с вероятностью $1 - \beta \approx 0,9975$.

4. Пример использования рассмотренного алгоритма

Не нарушая общности, рассмотрим решение задачи (1) – (9) для сети первого уровня (сети высокого давления). На рис. 2 представлен граф расчетной схемы кольца высокого давления МГРС города. В данной сети присутствуют:

1. На входе: 4 газораспределительные станции – ГРС (на рис. 2 вершины: 1, 40, 41, 42).
2. На выходах: 22 потребителя – газораспределительных пункта (ГРП) (на рис. 2 вершины: 50 – 74).

Для каждого участка также известны: длина участка трубопровода; внутренний диаметр трубы.

Известны оперативные данные по сети.

Критерий технической устойчивости равен 0.81.

Вероятность возникновения дефицита у потребителей: 0.51.

Для получения перепадов давлений на всех участках трубопровода задаем начальное распределение расходов по входам сети, решаем систему уравнений (4), (6).

Следующим этапом решения задачи оптимизации является поиск «диктующей точки», затем – вычисление дисперсий для решения системы (4) – (6) в точке разложения. Заданы математические ожидания и дисперсии расходов на выходе сети, математические ожидания и дисперсии давлений на входах сети.

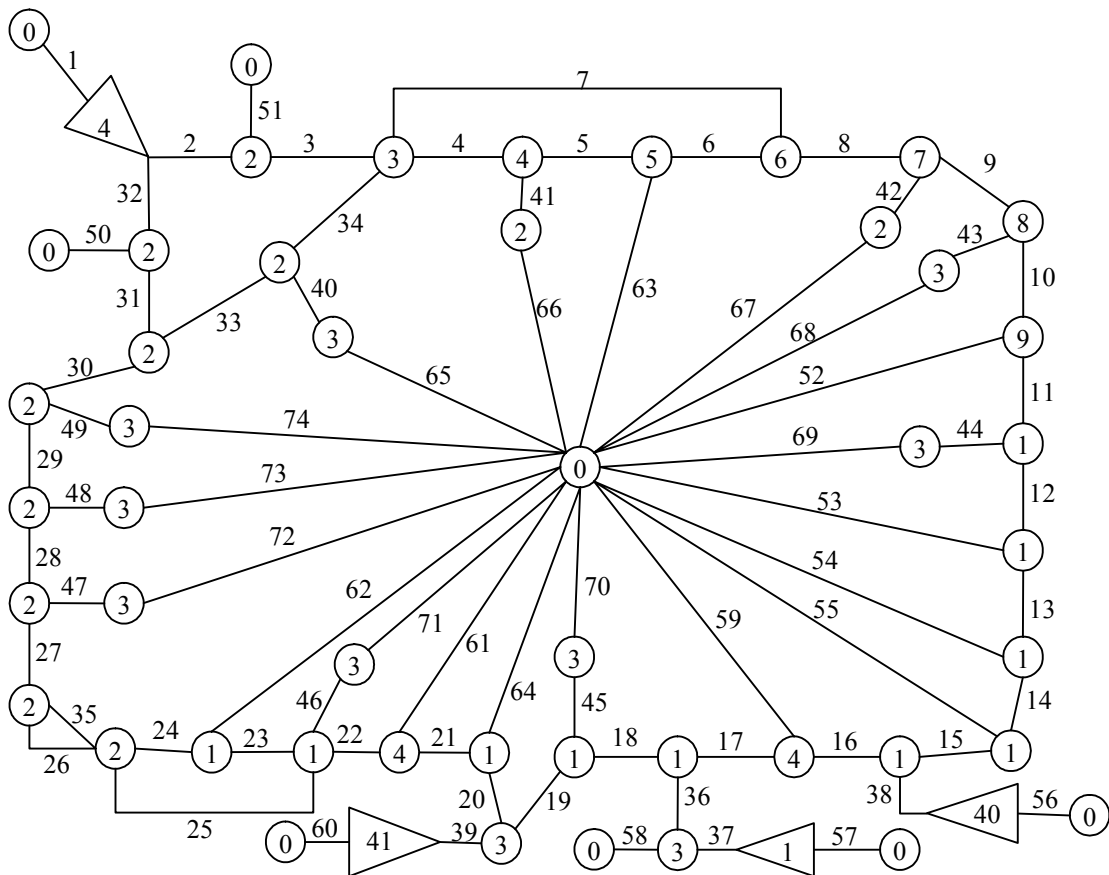


Рис. 2. Граф расчетной схемы кольца высокого давления МГРС города

Дисперсии по давлению брались из расчета, что длина диапазона изменения $(\bar{P}_j - 3\sigma_{P_j}; \bar{P}_j + 3\sigma_{P_j})$ равна $0.03 \cdot \bar{P}_j$, $j \in L$. Для расходов же этот коэффициент равнялся 50%: $0.5 \cdot \bar{q}_i$, $i \in N_2$. Таким образом, значения дисперсий вычислялись по формулам:

$$\sigma_{P_j}^2 = [(0.03 \cdot \bar{P}_j) / 6]^2; \quad \sigma_{q_i}^2 = [(0.5 \cdot \bar{q}_i) / 6]^2.$$

На основе приведенных данных составляем уравнения (17) – (19).

Используя эти уравнения, составляем системы вида (20). Решая их, получаем искомые частные производные зависимых переменных по независимым в выражениях (15) – (16).

Рассчитав дисперсии давлений на выходах и расходов на входах, переходим непосредственно к оптимизации режима газораспределительной сети.

Исходные данные для задачи оптимизации указаны выше.

В качестве начальных приближений используем точки вида $\{Q_{ГРС \text{ №}2}, Q_{ГРС \text{ №}3}, Q_{ГРС \text{ №}4}\}$. ГРС №1 не указывается ввиду того, что:

$$Q_{ГРС \text{ №}1} = \sum_{i \in N_2^k} q_i - (Q_{ГРС \text{ №}2} + Q_{ГРС \text{ №}3} + Q_{ГРС \text{ №}4})$$

и данное уравнение участвует в решаемой системе (10) – (11). Начальные приближения имеют вид:

$$\vec{q}^0 = \{36.0; 123.0; 90.0\}; \quad \vec{q}^1 = \{125.0; 104.167; 58.33\}; \quad \vec{q}^2 = \{125.0; 83.33; 0.0\}; \quad \vec{q}^3 = \{137.5; 83.33; 41.66\}.$$

С учетом рассчитанных дисперсий давлений на выходах значения P_i^- , $i \in N_2$ будут изменены. Скорректированы также значения границ изменения расходов на входах (ГРС).

С учетом введенных ограничений сумма избыточных давлений для точек приближений такова:

$$\vec{q}^0 : 22.6165 \text{ (атм.)}; \quad \vec{q}^1 : 27.2674 \text{ (атм.)}; \quad \vec{q}^2 : 29.1340 \text{ (атм.)}; \quad \vec{q}^3 : 21.9298 \text{ (атм.)}.$$

Допустимой погрешностью вычислений примем $\varepsilon = 10^{-4}$.

К заданной точности вычислений метод сошелся за 61 итерацию. Достигнутая погрешность: 0.0000865913. Точка оптимума:

$$\{q_{ГРС\ №1} = 173.472 \text{ (тыс. м}^3 \text{ / час)}; q_{ГРС\ №2} = 37.6645 \text{ (тыс. м}^3 \text{ / час)}; \\ q_{ГРС\ №3} = 32.971 \text{ (тыс. м}^3 \text{ / час)}; q_{ГРС\ №4} = 162.892 \text{ (тыс. м}^3 \text{ / час)}\} .$$

Сумма избыточных давлений в точке оптимума: $J_0 = 20.1257$ (атм.).

Полученное оптимальное решение необходимо проверить на адекватность поставленной задаче. Для этого вычислим нижнюю границу критерия технической устойчивости (2):

$$q_{ГРС\ №1} : J_{j2}^- = 1.0; q_{ГРС\ №2} : J_{j2}^- = 0.998; q_{ГРС\ №3} : J_{j2}^- = 0.998; q_{ГРС\ №4} : J_{j2}^- = 1.0 .$$

Она равна 0.998. Таким образом, мы полностью удовлетворяем критерию технической устойчивости .

Вероятность же возникновения дефицита у потребителей равняется 0.04, что является очень хорошим результатом.

Выводы

Предложен метод решения задачи оптимизации плановых режимов транспорта и распределения природного газа в газотранспортных сетях.

В настоящее время не существует общих методов решения задач такого типа ввиду их огромной размерности, особенно для газораспределительных сетей низкого и среднего давления (для $k \geq 2$), что делает работу актуальной.

Основными отличиями данного метода от существующих являются: при расчетах учитываются дисперсии расходов и давлений, а значит построенные прогнозы и доверительные интервалы, используемые при оптимизации, значительно больше отвечают реальности, чем в других аналогичных алгоритмах; в методе Нелдера-Мида добавлена операция проецирования на многогранник, что позволило оптимизируемые переменные держать в пределах реальных значений независимо от расчетных; при оптимизации используются модификации ограничений вида $(x_{\min} + 3\sigma_x; x_{\max} - 3\sigma_x)$ с учетом рассчитанных дисперсий. Использование такого подхода позволяет добиться результатов оптимизации, которые будут верны независимо от реальных значений расходов у потребителей с вероятностью, близкой к единице.

Разработанный метод является экономически целесообразным и выгодным в силу существенной минимизации вероятности отключения потребителей природного газа и связанных с ним издержек на восстановление режимов.

Использование данного метода для решения практической задачи показало его эффективность и возможность использования для задач ежедневного (суточного) планирования расхода газа в МГРС города.

Список литературы: 1. *Тевяшев А.Д., Ткаченко В.Ф., Попов А.В.* Об одном классе задач оперативного планирования режимов работы многоуровневых газораспределительных сетей // Радиоэлектроника и информатика. 2007. № 3. С.103-110. 2. *Фихтенгольц Г. М.* Курс дифференциального и интегрального исчисления. В 3 т. Т. I. М.: ФИЗМАТЛИТ, 2003. 680 с. 3. *Химмельблау Д.* Прикладное нелинейное программирование. М.: МИР, 1975. 536 с. 4. *Simone.* Equations and methods. V.5. SIMONE Research Group s.r.o. LIWACOM Informationstechnik GmbH. 2002. 157 с. 5. *ГОСТ 30319.0-96* Газ природный. Методы расчета физических свойств. Общие положения, 1996. 11 с. 6. *ГОСТ 30319.1-96* Газ природный. Методы расчета физических свойств. Определение физических свойств природного газа, его компонентов и продуктов его переработки, 1996. 25 с. 7. *ГОСТ 30319.2-96* Газ природный. Методы расчета физических свойств. Определение коэффициента сжимаемости, 1996. 36 с. 8. *ГОСТ 30319.3-96* Газ природный. Методы расчета физических свойств. Определение физических свойств по уравнению состояния, 1996. 28 с. 9. *ДБН В.2.5-20-2001.* «Газоснабжение» Госстрой Украины. Киев 2001. 68 с.

Поступила в редколлегию 17.09.2008

Тевяшев Андрей Дмитриевич, академик УНГА, д-р техн. наук, проф., зав. кафедрой прикладной математики ХНУРЭ. Научные интересы: теория стохастических моделей. Увлечения и хобби: теннис, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-14-36, e-mail: tevjashev@kture.kharkov.ua.

Золотарев Денис Алексеевич, стажер кафедры прикладной математики ХНУРЕ. Научные интересы: математическое моделирование. Увлечение и хобби: программирование. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-14-36, e-mail: Fenix_5@samsobaka.org.ua

МОДИФИКАЦИЯ ТОЧНОГО МЕТОДА РЕШЕНИЯ ЗАДАЧИ РАЗМЕЩЕНИЯ ПРЯМОУГОЛЬНЫХ ОБЪЕКТОВ

Исследуется оптимизационная задача размещения прямоугольных объектов в полосе, выделяются дополнительные свойства математической модели задачи, на основе которых предлагается модификация точного метода решения задачи, основанного на методе ветвей и границ, позволяющая улучшить оценку вычислительной сложности алгоритма

Постановка задачи. Многие практические оптимизационные задачи управления ресурсами, задачи размещения и упаковки [1-2], в том числе размещения источников физических полей, сводятся к задаче размещения конечного набора прямоугольных геометрических объектов в прямоугольной области. Данная задача оптимизационного геометрического проектирования [1], несмотря на наличие множества подходов к ее решению, в основном приближенных, представляет теоретический и практический интерес, о чем свидетельствует постоянно растущее число публикаций [3-4] как в нашей стране, так и за рубежом. По своей постановке данная задача относится к классу многомерных многоэкстремальных задач дискретной оптимизации, эффективные точные методы решения которых существуют только для частных случаев.

Значительная практическая ценность рассматриваемого класса задач обуславливает необходимость создания программного обеспечения процесса решения, основанного на современном математическом инструментарии.

В данной статье выделены новые свойства математической модели задачи, на основании чего построена эффективная модификация точного метода решения задачи и улучшена оценка вычислительной сложности алгоритма решения.

Рассмотрим следующую оптимизационную задачу геометрического проектирования [1]. Пусть есть конечный набор $R = \{R_i\}, i = \overline{1, N}$ ориентированных прямоугольников, заданных в арифметическом евклидовом пространстве E^2 , и полубесконечная полоса $S_0 = \{(x, y) \in E^2 \mid x \in [0, z], y \in [0, W], W = \text{const}, z = \text{var}\}$.

Для каждого объекта R_i известны его метрические характеристики (a_i, b_i) . Положение R_i в общей системе координат XOY , связанной с областью S_0 , задается вектором параметров размещения $u_i = (x_i, y_i)$, который определяет начало его собственной системы координат $X_iO_iY_i$ (рис. 1).

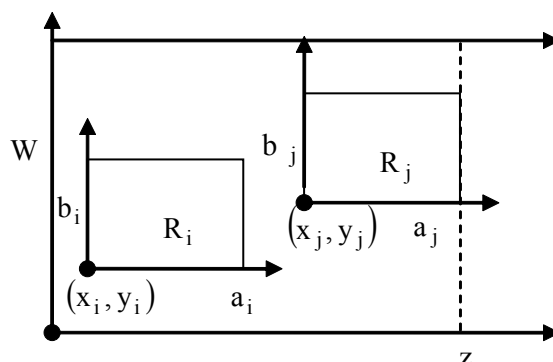


Рис. 1. Графическое представление постановки задачи

Задача размещения прямоугольников в области S_0 такова: необходимо разместить N объектов без взаимных наложений в полосе S_0 , так, чтобы длина занятой части полосы z была минимальной.

Анализ предыдущих исследований. Теория и методы оптимизационного геометрического проектирования развиваются учеными известной отечественной научной школы Ю.Г.Стояна. В частности, изучению особенностей оптимизационной задачи размещения прямоугольных объектов, которая является парадигмой теории геометрического проектирования, посвящены работы [1-5].

Для класса задач размещения прямоугольников в прямоугольной области разработаны методы определения глобального минимума функции цели задачи [3,4], основанные на схеме метода ветвей и границ [6], и методы поиска локального минимума функции цели задачи, основанные на применении метода последовательно-одиночного размещения [1], симплекс-метода [7] или метода набора активных неравенств [8].

Математическая модель задачи размещения имеет вид:

$$\text{Найти: } u^* = \arg \min_{u \in D \subset E^{2N+1}} z \quad (1)$$

где $u = (u_1, u_2, \dots, u_N, z) = (x_1, y_1, x_2, y_2, \dots, x_N, y_N, z)$, $D = D^1 \cap D^2$ – множество допустимых решений задачи.

Подобласть D^1 , сформированная условиями размещения набора объектов R в S_0 , описывается системой $F_0(u) \leq 0$ линейных неравенств:

$$F_0(u) := \{f_{0i}^h(u_i)\}, h = 1, \dots, 4; i = 1, 2, \dots, N, \quad (2)$$

здесь $f_{0i}^1(u_i) = x_i - z + a_i$, $f_{0i}^2(u_i) = y_i - W + b_i$, $f_{0i}^3(u_i) = -x_i$, $f_{0i}^4(u_i) = -y_i$.

Подобласть D^2 , определяемая условиями взаимного попарного непересечения объектов R_i и R_j , $i, j = 1, 2, \dots, N, i \neq j$, задается системой $F(u)$ наборов $F_{ij}(u_i, u_j) \leq 0$ линейных неравенств вида:

$$F_{ij}(u_i, u_j) \leq 0 := \langle f_{ij}^k(u_i, u_j) \leq 0, k = 1, \dots, 4; i, j = 1, 2, \dots, N, i \neq j, \quad (3)$$

где

$$\begin{aligned} f_{ij}^1(u_i, u_j) &= x_j - x_i + a_j, f_{ij}^2(u_i, u_j) = x_i - x_j + a_i, \\ f_{ij}^3(u_i, u_j) &= y_j - y_i + b_j, f_{ij}^4(u_i, u_j) = y_i - y_j + b_i. \end{aligned} \quad (4)$$

Количество M ограничений задачи (1)-(3) составляет $4N + 2N(N-1)/2$. Далее будем использовать обозначение вектора u для соответствующей ему точки в пространстве E^{2N+1} .

Известны [3,4] такие свойства задачи (1)-(3).

Свойство 1. Многогранная область D является несвязным замкнутым и при $N > 1$ невыпуклым точечным множеством. Компоненты связности области допустимых решений D невыпуклые, в общем случае многосвязные множества. Граница области $\Gamma = \text{Fr}D$ – кусочно-линейная.

Свойство 2. Глобальный минимум z^* функции цели достигается в вершине u^* области допустимых решений D , которая задается системой $(2N+1)$ линейных уравнений $\mathfrak{Z}^*(u) = 0$, т.е. набором $\mathfrak{Z}^*(u)$ активных в точке u^* ограничений систем (2)-(3).

Свойство 3. Область D может быть представлена в виде объединения конечного числа выпуклых подмножеств:

$$D = \bigcup_{q=1}^Q D_q, \quad (5)$$

где $D_q \subset E^{2N+1}$ – многогранное выпуклое множество, $Q = 4^{N(N-1)/2}$.

При этом область D_q описывается системой $F_0(u)$ линейных неравенств вида (2) и $(N-1)N/2$ неравенств вида (4) – по одному из каждого набора неравенств вида (3) для каждой пары объектов.

Свойство 4. Покрывание (5) не является разбиением, поэтому для некоторых точек u области D имеет место такое соотношение:

$$u \in D_{Q_1} = \bigcap_{q=1}^{Q_1} D_q, \quad G_1 < Q. \quad (6)$$

Свойство 5. В силу выделенных особенностей функции цели и области допустимых решений решение задачи (1)-(3) сводится к решению $4^{N(N-1)}$ задач линейного программирования вида

$$u^* = \arg \min_{u \in D_q \subset D} Z, \quad (7)$$

$$D_q : \begin{cases} F_0(u) \leq 0, \\ f_{ij}^k(u_i, u_j) \leq 0, k \in \{1, \dots, 4\}, i, j \in \{1, \dots, N\}, i \neq j. \end{cases} \quad (8)$$

Итак, несвязность области допустимых решений D задачи и необходимость перебора конечного числа систем линейных неравенств (систем линейных уравнений) обусловили применение методов дискретной оптимизации, в частности, метода ветвей и границ.

Приведем сравнение двух схем точного метода решения, которые составляют основу предлагаемой в данной работе модификации метода решения задачи.

В [4] предложено дерево решений A^1 для формирования и перебора подмножеств D_q , т.е. систем неравенств вида (8) (рис.2,а). Такое дерево имеет $(N-1)N/2$ уровней. Уровни дерева A^1 упорядочены таким образом:

$$D = (p_1, p_2, \dots, p_h, \dots, p_{(N-1)N/2}) = (\{1,2\}, \{1,3\}, \dots, \{1, N\}, \{2,3\}, \dots, \{2, N\}, \{3,4\}, \dots, \{N-1, N\}),$$

где в фигурных скобках указаны номера объектов размещения, условия непересечения которых рассматриваются на уровне $h=1,2,\dots,(N-1)N/2$ дерева решений, отвечающем порядковому номеру компоненты p_h вектора P . Корень дерева A_0^1 соответствует системе неравенств $F_0(u) \leq 0$ (2).

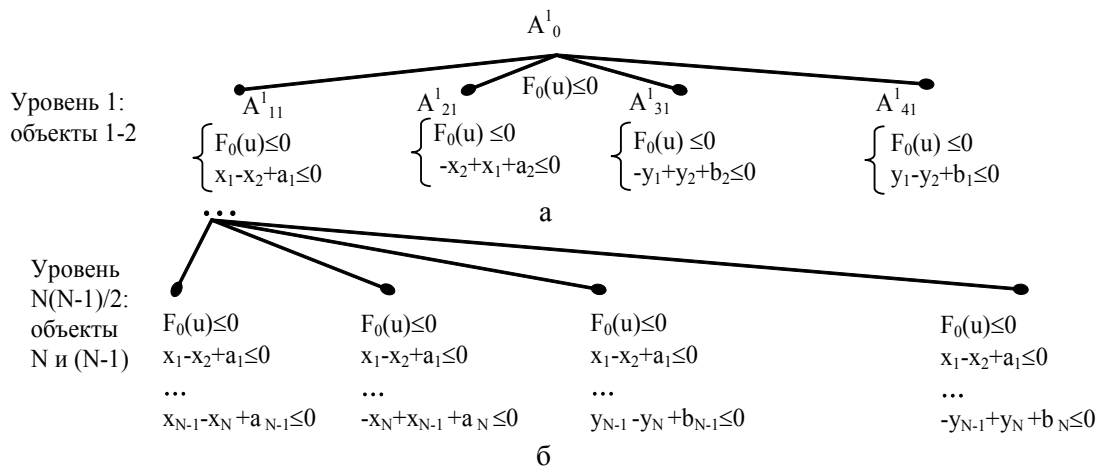


Рис. 2. Иллюстрация дерева решений A^1

На каждой вершине $A_{kh}^1, k=1,2,\dots,4$ h -го уровня дерева решений формируется система неравенств, состоящая из системы неравенств, полученной на предыдущем $(h-1)$ -м уровне дерева решений, и неравенства $f_{ij}^k(u_i, u_j) \leq 0$ из набора $F_{ij}(u_i, u_j) \leq 0$ неравенств, который отвечает уровню h дерева решений. Последний уровень дерева содержит $4^{N(N-1)/2}$ вершин, которые соответствуют системам неравенств, описывающим всевозможные по-

добласти D_q области допустимых решений D . На каждой области D_q необходимо решить задачу линейного программирования для определения локального минимума u_q^* функции цели задачи (1)-(2).

В [3] предложено дерево A^2 для задачи (1)-(2), с помощью которого осуществляется формирование и перебор всевозможных систем $\mathfrak{Z}(u) = 0$ линейных уравнений – множества активных ограничений – (рис. 2,б), в результате чего определяется система $\mathfrak{Z}^*(u) = 0$. Дерево A^2 имеет $2N+1$ уровень – по числу независимых переменных задачи. На каждом уровне дерева решений к системе уравнений, сформированной на вершине дерева решений более высокого уровня, добавляется уравнение, содержащее независимую переменную задачи (1)-(3) с ненулевым коэффициентом. Последний уровень дерева A^2 содержит $2N^{2N}$ вершин, которые соответствуют системам $\mathfrak{Z}_{\text{fin}}(u) = 0$, состоящим из $2N$ уравнений.

Отметим, что дерево решений A^1 в силу свойства 3 содержит значительное число вершин, на которых формируются подобласти, имеющие одну и ту же оптимальную вершину u^* , т.е. система $\mathfrak{Z}^*(u) = 0$ переопределена. Аналогично, дерево решений A^2 содержит множество вершин, определяющих системы уравнений $\mathfrak{Z}(u) = 0$, решения которых – точки области D – совпадают. Кроме того, на вершинах дерева решений A^1 формируются системы неравенств, определяющие внутренние подобласти $D_q \subset \text{int } D$ области допустимых решений D .

Анализ особенностей области D

Рассмотрим следующие дополнительные особенности математической модели задачи (1)-(3) и области допустимых решений D , сформулированные в виде свойств и утверждений.

Благодаря невыпуклости области D системы линейных уравнений $\mathfrak{Z}_{\text{fin}}(u) = 0$ могут описывать вершины области D , граничные точки области D , внутренние точки области D , даже если это вершины подобласти D_q . Кроме того, система $\mathfrak{Z}(u) = 0$ может описывать точки u^{out} вне области D : $u^{\text{out}} \notin D$ и вообще быть несовместной. Больше того, на промежуточных уровнях дерева A^2 система $\mathfrak{Z}(u) = 0$ может описывать линейные многообразия размерности больше 0 – прямые, плоскости, гиперплоскости, т.е. ребра области D и грани области D различной размерности. Обозначим $V = \{u_1^V, u_2^V, \dots, u_{N_V}^V\}$ – множество вершин области D , $O = \{u_1^{\text{out}}, u_2^{\text{out}}, \dots, u_{N_O}^{\text{out}}\}$ – множество точек вне области D , задаваемых системами вида $\mathfrak{Z}(u) = 0$, $I = \{u_1^{\text{int}}, u_2^{\text{int}}, \dots, u_{N_I}^{\text{int}}\}$ – множество внутренних точек области D , описываемых системами $\mathfrak{Z}(u) = 0$, $|I| = N_I, |V| = N_V, |O| = N_O$.

При этом для дальнейшего рассмотрения представляют интерес только граничные точки области D , больше того – множество V вершин области D .

Утверждение 1. На системе $\mathfrak{Z}^*(u) = 0$, которая задает оптимальное решение u^* , имеет место биекция Ψ вида

$$x_i \leftrightarrow \mathfrak{Z}_k^* = 0, \mathfrak{Z}_k^* \ni \{-x_i + x_1 + a_1 = 0, k \in \{1, 2, \dots, 2N + 1\}, l \in \{1, 2, \dots, N\}; -x_i = 0\}, \quad (9)$$

$$y_i \leftrightarrow \mathfrak{Z}_k^* = 0, \mathfrak{Z}_k^* \ni \{-y_i + y_1 + b_1 = 0, k \in \{1, 2, \dots, 2N + 1\}, l \in \{1, 2, \dots, N\}; -y = 0\}, \quad (10)$$

$$z \leftrightarrow \mathfrak{Z}_k^* = 0, \mathfrak{Z}_k^* = -z + x_1 + a_1, k \in \{1, 2, \dots, 2N + 1\}, l \in \{1, 2, \dots, N\}. \quad (11)$$

Выполнение условий утверждения 1 означает, что каждой компоненте – независимой переменной задачи (1)-(3) – оптимального решения u^* можно поставить в соответствие уравнение системы $\mathfrak{Z}^*(u) = 0$, которое содержит эту переменную со знаком „-“.

Доказательство первой части утверждения 1 проведем от противного. Пусть формула (9) неверна. Тогда существует ненулевой шаг $(x_i + \alpha p)$ в любом направлении, удовлетворяющий данное ограничение, т.е. оставляющий точку в области D . Так как функция цели $z = \max_{i=1,2,\dots,N} (x_i + a_i)$, это допустимое перемещение может привести к уменьшению значения

z , т.е. система $\mathfrak{Z}^*(u) = 0$ не определяет в общем случае оптимальную точку, что противоречит условию утверждения.

Доказательство второй части утверждения 1 (формула (8)) опирается на тот факт, что движение вдоль удерживающих направлений вида $y_1 - y_i + b_1 = 0$ или $y_i - y_1 + b_i = 0$ не приведет к изменению функции цели, так как проводится в гиперплоскости ортогональной нормали функции цели.

Третья часть утверждения 1 (формула (11)) очевидна.

Модификация схемы точного метода решения задачи (1)-(3)

Основная идея модификации состоит в определении принципиальной возможности организации немотононного пошагового процесса оптимизации (с неизбежными возвратами, упорядоченными по дереву решений A^1 , но конечное число раз), т.е. процесса перехода вида

$$\mathfrak{Z}_h(u) = 0 \rightarrow \mathfrak{Z}_{h+1}(u) = 0,$$

где множество функций $\mathfrak{Z}_{h+1}(u)$ имеет вид:

$$\mathfrak{Z}_{h+1}(u) = \left\{ \mathfrak{Z}_h(u) / f_f^1(u_i, u_j) \wedge (f_{h+1}^k(u_i, u_n) \vee f_{h+1}^k(u_t, u_j)) \right\}, \quad k, l \in \{1, \dots, 4\}; i, j, n, t \in \{1, 2, \dots, N\}, \quad (12)$$

$$i \neq j, i \neq n, t \neq j, f \in \{1, 2, \dots, 2N\}.$$

из недопустимой начальной точки $u_{out} \notin D$, которая является, например, оптимальным решением релаксованой задачи вида

$$u_{out}^* = \arg \min_{u \in D^1 \subset E^{2N+1}} z \quad (13)$$

$$D^1 := F_0(u) \leq 0, \quad (14)$$

к некоторой допустимой точке $u \in FrD$, более того, использование правил отсечения, описанных ниже, позволяет удовлетворить условие: $u \in V$.

Итак, согласно (12) на каждом шаге процесса решения происходит подстановка одного активного ограничения $f_{h+1}^k(u_i, u_n)$ или $f_{h+1}^k(u_t, u_j)$, $k, l \in \{1, \dots, 4\}; i, j, n, t \in \{1, 2, \dots, N\}$, в соответствии с порядком прохода по дереву решений A^1 вместо определенного уравнения $f_f^1(u_i, u_j) = 0$ в текущей системе $\mathfrak{Z}_h(u) = 0$.

Отметим важный факт. На основе утверждения 2 на каждом шаге алгоритма известно, какое именно ограничение становится активным, а на основе утверждения 1 на каждом шаге процесса известен номер f ограничения $f_f^1(u_i, u_j) = 0$, $l \in \{1, \dots, 4\}, f \in \{1, 2, \dots, h\}$, которое подлежит замене на $(h+1)$ -м уровне дерева решений.

Утверждение 2. Каждое новое ограничение, генерируемое по дереву A^1 , должно быть активным.

Доказательство необходимости такого подхода состоит в непрерывном отслеживании исключительно точек $u \in FrD$, так как множество граничных точек области D содержит множество $V = \{u_1^V, u_2^V, \dots, u_{N_v}^V\}$ вершин области, в том числе и оптимальную вершину.

Добавление каждого нового ограничения: $f_{h+1}^k(u_i, u_n) = 0$ или $f_{h+1}^k(u_t, u_j) = 0$ из набора ограничений (3) по дереву решений A^1 означает выполнение условия касания пары (i, n) или (t, j) объектов размещения соответственно, т.е. пошаговое приближение к границе области D .

Для поддержки процесса решения разработан набор правил отсечения, основные из них таковы:

Правило 1 (Условие несовместности). Если текущей является вершина A_{kh}^1 , $k \in \{1, \dots, 4\}$ уровня $h \geq N$ дерева A^1 , на котором рассматриваются условия касания объектов (i, j) , то при выполнении следующих условий:

при $k=1$ активны вершины A_{i-1}^1 и A_{2j-1}^1 ; при $k=2$ активны вершины A_{2i-1}^1 и A_{1j-1}^1 ;

при $k=3$ активны вершины A_{3i-1}^1 и A_{4j-1}^1 ; при $k=4$ активны вершины A_{4i-1}^1 и A_{3j-1}^1 ,

$h > j > i$

вершина A_{kh}^1 является концевой.

Правило 2 (Отсечение внутренних точек). Пусть текущей является вершина A_{kh}^1 , $k \in \{1, \dots, 4\}$ уровня h дерева A^1 , на которой построена система активных ограничений $\mathfrak{Z}_h(u) = 0$.

Если ограничение $f_{ij}^k(u_i, u_j) = 0$, $k \in \{1, \dots, 4\}$, $i, j \in \{1, \dots, N\}$, $i \neq j$, задающее условия касания объектов (i, j) , принадлежит системе $\mathfrak{Z}_h(u) = 0$, то при выполнении следующих условий:

при $k \in \{1, 2\}$ $u_i + b_i < u_j$ или $u_j + b_j < u_i$;

при $k \in \{3, 4\}$ $x_i + a_i < x_j$ или $x_j + a_j < x_i$

вершина A_{kh}^1 является концевой.

Правило 3 (Проактивное отсечение). Если текущим является уровень $h \geq N$ дерева A^1 , на котором рассматриваются условия касания $\langle f_{ij}^k(u_i, u_j) = 0$, $k = 1, \dots, 4$ объектов (i, j) , то при выполнении соответствующих условий непересечения уровень $h \geq N$ дерева A^1 исключается из рассмотрения.

Результаты численных экспериментов

Данный подход был реализован программно на Object Pascal 6.0 в программной среде Delphi. Рассматривались задачи размещения различных наборов прямоугольников ($N \leq 10$) с различными метрическими характеристиками и количеством объектов размещения. Время решения – $O(1 \text{ мин})$.

Усредненные предварительные результаты сведены в таблице. При этом N – количество объектов размещения, N_{all} – общее количество просмотренных вершин дерева решений A^1 , N_I – количество построенных систем уравнений, описывающих внутренние точки области D , N_O – количество вершин дерева решений A^1 , соответствующих точкам вне области D , N_{UN} – количество несовместных систем (такие вершины дерева считались концевыми), N_V – количество вершин области D , N_{opt} – номер первой вершины дерева решений, соответствующей оптимальной системе уравнений $\mathfrak{Z}^*(u) = 0$. Отметим, что приведенная статистика представляет теоретический интерес, так как, во-первых, учитывались вершины промежуточных уровней дерева A^1 , а во-вторых, рассматривались задачи без учета ограничений вида $f_{oi}^1(u_i) \leq 0$, $f_{oi}^2(u_i) \leq 0$, т.е. по сути проводился полный перебор вершин дерева решений.

N	N_{all}	N_I (%)	N_O	N_{UN}	N_V	N_{opt}
3	80	32 (~38%)	19	4	16	7
4	2391	1670 (~70%)	383	167	191	233
5	102800	83500 (~80%)	11200	6900	1200	980
8	$7 \cdot 10^{16}$	$6,12 \cdot 10^{16}$ (~86%)	$1,08 \cdot 10^{16}$	73950	9100	43000

Выводы

Проведен анализ структуры области допустимых решений задачи размещения прямоугольников, выделены дополнительные особенности, на основе которых разработана модификация точного метода решения задачи, состоящая в осуществлении обхода дерева решений, которое обеспечивает пошаговый переход из точки вне области допустимых решений в граничную точку области (вершину) с последующим усеченным перебором вершин. Предложенная модификация позволяет существенно сократить число просматриваемых элементов области допустимых решений задачи. Данный подход программно реализован. В дальнейшем предусматривается проведение численных экспериментов с наборами данных практической размерности.

Список литературы: 1. Стоян Ю.Г., Яковлев С.В. Математические модели и оптимизационные методы геометрического проектирования. К.: Наук. думка, 1986. 268 с. 2. Новожилова М.В., Попельнюх Н.О. Розв'язання задачі оптимізації ресурсів проекту при точних вихідних даних// Вісник ЖДТУ/ Технічні науки. 2006. № 4 (39). С. 225-230. 3. Stoyan, Yu. G., Novozhilova, M.V. Non-guillotine placement of rectangles into a strip of given width / Pesquisa Operacional. 1999. Vol. 19, N 2. 145 p. 4. Магач С.А. Методы решения экстремальных задач размещения многоугольных геометрических объектов в полосе: Автореф. дисс. канд. физ.-мат. наук. М., 1984. 20 с. 5. Dyckhoff H, Scheithauer G, Terno J. Cutting and packing. In: Dell'Amico M, Maffioli F, Martello S (eds) Annotated bibliographies in combinatorial optimization. Wiley, Chichester, Chapt. 22, 1997. P. 393-412. 6. Таха Х.А. Введение в исследование операций. М.: Вильямс, 2001. 912с. 7. Петров Е.Г., Новожилова М.В., Гребенник І.В. Методи і засоби прийняття рішень у соціально-економічних системах. К.: Техніка, 2003. 240с. 8. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация. М.: Мир, 1985. 252с.

Поступила в редколлегию 27.11.2008

Чуб Игорь Андреевич, канд. техн. наук, доцент, докторант Университета гражданской защиты Украины. Научные интересы: математическое моделирование, прикладная геометрия, методы оптимизации. Адрес: Украина, 61023, Харьков, ул. Чернышевского, 94, тел.: 707-34-90, e-mail: chubia@apbu.edu.ua

Новожилова Марина Владимировна, д-р физ.-мат. наук, профессор, зав. кафедрой компьютерного моделирования и информационных технологий Харьковского государственного технического университета строительства и архитектуры. Адрес: Украина, 61000, Харьков, ул. Сумская, 40, тел.: 706-20-49, e-mail: novozhilova@kstuca.kharkov.ua.

УДК 519.677

С.А. ТАЯНОВ, В.А. ТАЯНОВ

МЕТОДИКА КЛАСТЕРИЗАЦІЇ ЗОБРАЖЕНЬ ДЛЯ ЇХ КОМПРЕСІЇ НА ОСНОВІ КОМПОНЕНТНОГО АНАЛІЗУ

Пропонується нова методика кластеризації зображень для їх подальшого адаптивного стиску. Проводиться дослідження ефективності стискання зображень шляхом використання двох різних методів кластеризації та стиску кластерів з допомогою адаптивної методики на основі перетворення Карунена-Лоева.

1. Вступ

В багатьох системах для зберігання графічної інформації часто виникає необхідність стискання зображень, причому не тільки індивідуальних зображень, а також і їх груп для того, щоб отримати кращі ступені компресії. Це досягається шляхом зменшення інформації, що міститься в заголовках, оскільки ця інформація буде спільною для всієї серії зображень.

Суть більшості алгоритмів для стискання зображень полягає в діленні зображення на окремі блоки (кластери) та в подальшому стисканню цих блоків за допомогою різних алгоритмів. Тому перспективним є не тільки розробка алгоритмів для стиску зображень, а й розробка нових методик кластеризації зображень.

В даній роботі пропонується методика кластеризації зображень та алгоритм для покращення ступеня компресії зображення на основі перетворення Карунена-Лоева (ПКЛ).

Використання ПКЛ для стискання зображень є дуже ефективним, оскільки воно мінімізує середньоквадратичну похибку при використанні часткового набору базисних функцій у

розкладі. Однак використання цього алгоритму для стискання цілого зображення неефективне на практиці, тому що воно вимагає збереження великої матриці перетворення разом зі стиснутими даними. Використання кластеризації та стискання окремих кластерів зображення на основі ПКЛ є доволі перспективним, а тому актуальним є розробка нових адаптивних алгоритмів для кластерного стиску зображень на основі ПКЛ. Застосування матриці ПКЛ для серії зображень також підвищить ступінь стиску для індивідуальних зображень за рахунок спільної службової інформації (матриця ПКЛ), оскільки практичні результати [1] показують, що ефективність ПКЛ для серії зображень незначно зменшується порівняно з одним зображенням і вона є завжди більшою, ніж в дискретного косинусного перетворення, яке застосовується в багатьох алгоритмах стиску зображень.

2. Аналіз останніх досліджень

Кластеризація зображення та стискання окремих кластерів використовується в одному з найбільш популярних графічних форматів JPEG [2] та JPEG 2000 [3]. Також використання вейвлет-перетворення та автоматичної сегментації дозволили створити найефективніший на сьогоднішній день формат DjVu [4].

Що стосується стиску на основі ПКЛ, то зокрема відомі роботи, присвячені компресії спектра зображень одного розміру на основі попередньої кластеризації, наприклад [5]. Ще одна область, де себе добре зарекомендував кластерний ПКЛ, – це область цифрового стиску відеоінформації [6] та аудіоінформації [7]. Детальний огляд проблем кластеризації поданий у [8] та [9].

Мета даного дослідження – розробка нових методик кластеризації зображень для їх подальшого адаптивного стиску на основі ПКЛ.

3. Кластеризація та подальше використання ПКЛ

Для зображення розміру $m \times n$ розбиваємо матрицю на прямокутні кластери розміром 8×8 та нумеруємо зліва направо та зверху вниз. Також нумеруємо елемент кожного кластера зліва-направо та зверху вниз, тобто лівий верхній елемент i -го кластера буде позначатися як S_{i1} , а правий нижній елемент – як S_{i64} .

В результаті отримуємо $\frac{m \times n}{8 \times 8}$ кластерів з 64 елементів кожний. Якщо S_i – i -й кластер, A – матриця зображення, A_{ij} – елемент матриці A , тоді коваріаційна матриця для всіх кластерів S буде виглядати так:

$$\mathbf{R} = \frac{8 \times 8}{m \times n} \sum_{i=1}^{\frac{m \times n}{8 \times 8}} (S_i - \bar{x})(S_i - \bar{x})^T, \quad (1)$$

де $\bar{x} = \frac{1}{m \cdot n} \sum_{i=1}^m \sum_{j=1}^n A_{ij}$ – середнє значення матриці A .

З коваріаційної матриці отримаємо матрицю власних векторів, розв'язавши рівняння на власні значення λ_j виду

$$\mathbf{R}\varphi_j = \lambda_j\varphi_j, \quad (2)$$

яка складається з 64 власних векторів, кожний з яких складається з 64 елементів, тут φ_j – j -й власний вектор.

Знаходимо вектор коефіцієнтів розкладу за власними векторами для кожного кластера:

$$\mathbf{c}_i = \mathbf{\Phi}^{-1}\mathbf{S}_i, \quad (3)$$

де $\mathbf{\Phi}$ – матриця власних векторів, яка складена з векторів φ_j .

Методика стискання зображень за допомогою ПКЛ полягає в тому, що якщо в кожному векторі \mathbf{c}_i матриці коефіцієнтів розкладу \mathbf{C} виділити тільки певну кількість найбільш суттєвих значень за амплітудою і тільки їх зберігати і враховувати, то можна зменшити кількість інформації, що зберігається. Крім того, необхідно зберігати значення матриці

власних векторів. Але якщо кількість пікселів у зображенні буде на декілька порядків більшою, ніж кількість значень в матриці власних векторів, то нею можна знехтувати при розрахунку коефіцієнта компресії.

Для відновлення зображення потрібно відновити кожний кластер зображення за допомогою зворотного ПКЛ:

$$S'_i = \Phi' c'_i, \quad (4)$$

де c'_i , Φ' – модифіковані матриця коефіцієнтів розкладу та матриця власних векторів, отримані в результаті стиснення (відкидання менш суттєвих значень). Після того з кластерів відновлюється зображення.

Запропонований адаптивний алгоритм стиску полягає в тому, що для різних кластерів вибираємо різну кількість коефіцієнтів розкладу для забезпечення постійного заданого SNR по кожному кластеру та певним чином обмежуємо максимальну кількість коефіцієнтів розкладу.

Блок-схема цього алгоритму наведена на рис. 1. Вихідними даними алгоритму служать: заданий SNR, визначена з (2) матриця власних векторів Φ , матриця коефіцієнтів розкладу c , яка знайдена за формулою (3), розмір матриці зображення $m \times n$ та коефіцієнт допуску d . В блоці 3 встановлюється початкова кількість коефіцієнтів розкладу для i -го кластера $n=1$, в блоці 4 встановлюється SNR при відновленні i -го кластера ($SNR_DEF=0$). В блоці 5 порівнюється заданий SNR та SNR_DEF i -го кластера при його відновленні з допомогою n коефіцієнтів розкладу, а також перевіряється умова (5) допустимої кількості коефіцієнтів розкладу $n \leq d \cdot \bar{N}$, де \bar{N} – середнє значення кількості коефіцієнтів розкладу серед всіх кластерів, яке необхідне для забезпечення заданого SNR. Для того щоб знайти \bar{N} , необхідно виконати весь алгоритм блок-схеми (див. рис.1) без умови (5), тобто блок 5 буде мати умову тільки $SNR_DEF \geq SNR$.

Визначення \bar{N} проводиться в блоці 13. Якщо умови в блоці 5 виконуються, то вибирається наступний кластер. Якщо ж ні, то для n коефіцієнтів розкладу формується матриця коефіцієнтів розкладу c' з матриці c та матриця власних векторів Φ' з матриці Φ (блок 6).

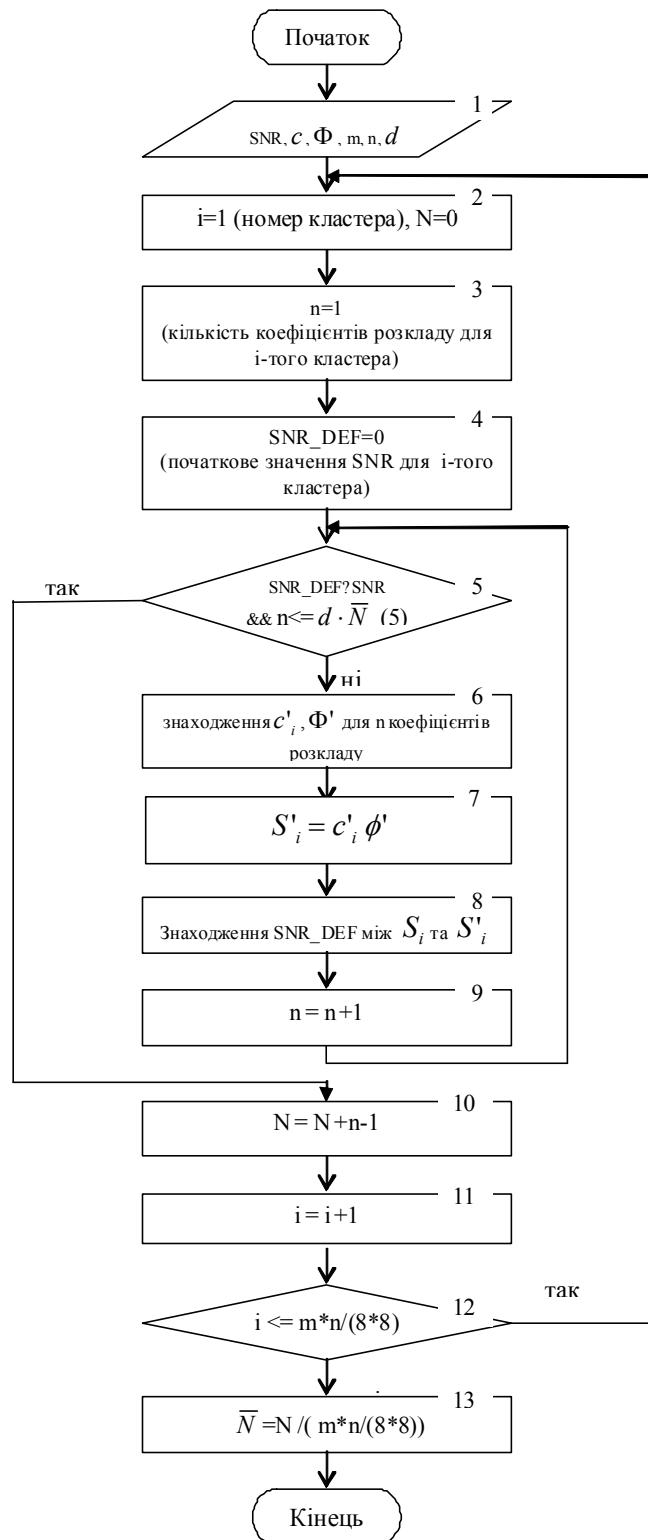


Рис. 1. Блок-схема адаптивного алгоритму для стиснення кластерів на основі ПКЛ

За формулою (4) виконується зворотне ПКЛ для i -го кластера (блок 7), в результаті якого відновлюється i -й кластер S'_i . Далі в блоці 8 знаходиться SNR_DEF між оригінальним кластером S_i та відновленим S'_i . Після того в блоці 9 збільшуємо кількість коефіцієнтів розкладу для i -го кластера $n_i = n_i + 1$ і блоки 6-9 виконуються знову. Цикл (блоки 5-9) виконується до моменту досягнення заданого SNR для i -го кластера (поки SNR_DEF < SNR). Блоки з 3 по 9 виконуються для всіх $\frac{m \times n}{8 \times 8}$ кластерів. В блоці 10 проводиться підсумовування всіх коефіцієнтів розкладу. В блоці 12 відбувається перехід до наступного кластера.

На рис. 2 показані результати моделювання залежності SNR від коефіцієнта стиску K для різних значень максимальної кількості коефіцієнтів розкладу N_{max} . Розглядаються 4 випадки: $N_{max} = \bar{N}$ ($d=1$), $N_{max} = 2\bar{N}$ ($d=2$), $N_{max} = 1.5\bar{N}$ ($d=1.5$) та $N_{max} = 64$ (64 – повна кількість коефіцієнтів розкладу), тобто обмеження на кількість коефіцієнтів відсутнє. Як видно з графіка, в діапазоні $K < 3.5$ найбільший SNR буде при $N_{max} = 2\bar{N}$ та коли обмежень на кількість коефіцієнтів розкладу немає ($N_{max} = 64$). При $K > 3.5$ найбільший ефект досягається, коли $N_{max} = \bar{N}$, але SNR для випадків, коли $N_{max} = 2\bar{N}$ та $N_{max} = 1.5\bar{N}$, не дуже сильно від нього відрізняється. Тому найбільш ефективний SNR на всьому діапазоні буде при $N_{max} = 2\bar{N}$ ($d=2$).

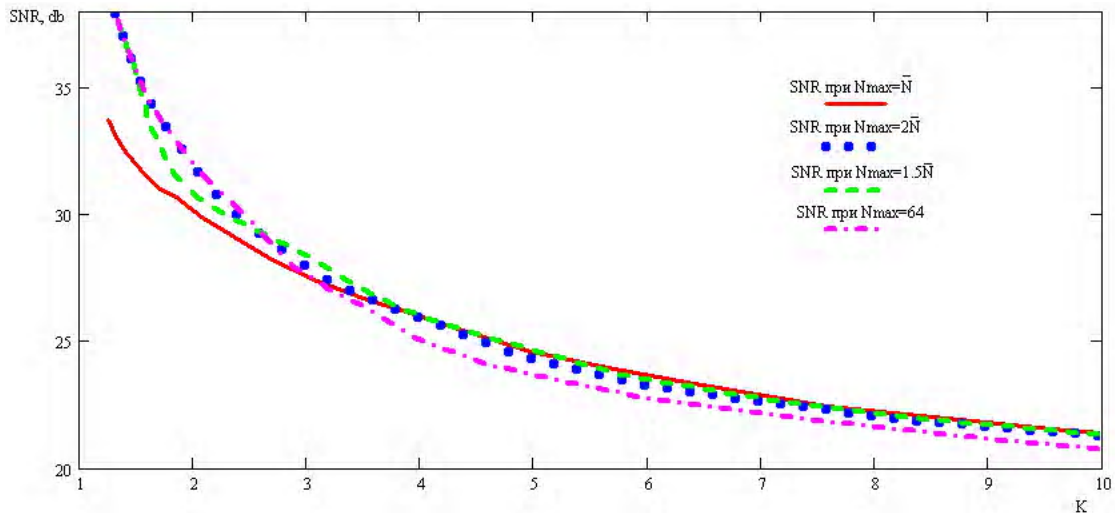


Рис.2. Залежність SNR від коефіцієнта стиску K для різних значень максимальної кількості коефіцієнтів розкладу N_{max}

На рис. 3 показано запропонований алгоритм кластеризації зображень, який може бути застосований для стиснення серії зображень. Суть алгоритму полягає в тому, що зображення розбивається на блоки 8×8 зліва-направо, зверху-вниз, як і в попередньому випадку. Але при цьому одним кластером вважається 8 поряд розташованих по горизонталі блоків (в принципі можна вибирати блоки і по вертикалі). Так, для i -го кластера номери елементів першого з восьми блоків будуть $S_{i1}, S_{i2}, \dots, S_{i64}$, для другого $S_{i65}, S_{i66}, \dots, S_{i128}$ і так далі, аж до останнього, як показано на рис. 3. Таким чином, загальна кількість коефіцієнтів для кожного кластера дорівнює 512.

З виразу (1) отримуємо коваріаційну матрицю для вибраних кластерів розміром 512×512 . Після того можна знайти матрицю власних векторів Φ . Якщо для стиснення кластерів застосувати алгоритм, зображений на рис. 1, то умова (12) буде виглядати як $i \leq \frac{mn}{512}$, а вираз (13) запишеться у вигляді $\bar{N} = \frac{512N}{mn}$. На рис. 4 представлені результати моделюван-

ня залежності SNR від коефіцієнта стиску K для значень максимальної кількості коефіцієнтів розкладу $N_{\max} = 512$ (максимальна кількість коефіцієнтів не обмежується) та $N_{\max} = 2\bar{N}$ для запропонованого алгоритму кластеризації.

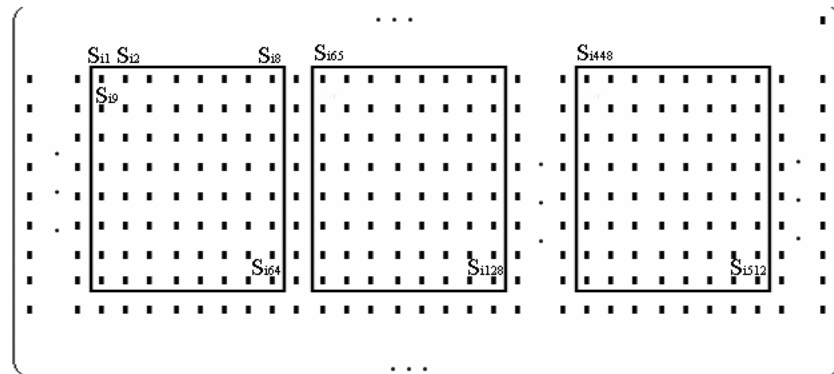


Рис. 3. Схема запропонованого алгоритму кластеризації зображень

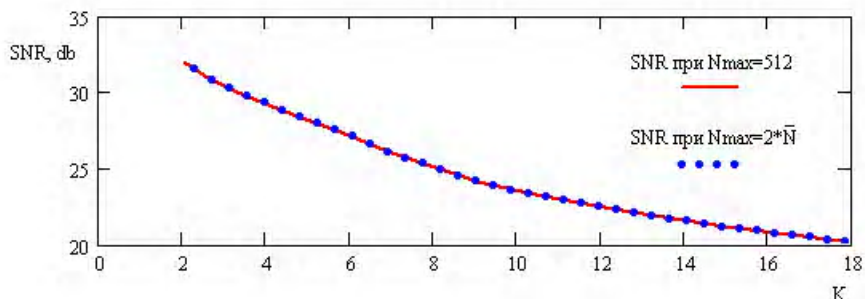


Рис. 4. Залежність SNR від коефіцієнта стиску K для різних значень максимальної кількості коефіцієнтів розкладу N_{\max} для кластерів розміром 512

З рис. 4 можна побачити, що при необмеженій кількості коефіцієнтів розкладу ПКЛ та при $N_{\max} = 2\bar{N}$ – значення SNR при певних коефіцієнтах K фактично збігаються. Таким чином, умову (5) в блок-схемі на рис. 1 для даного алгоритму можна не застосовувати. Якщо порівняти результати, які наведені на рис. 2, з результатами на рис. 4, то можна побачити, що SNR для запропонованого алгоритму кластеризації є кращим на 10-12% для коефіцієнтів стиску $K \leq 10..12$. Цей алгоритм можна застосовувати для великої серії зображень, тому що отримуємо достатньо велику матрицю власних векторів (512×512), яка має зберігатися разом зі стиснутими даними зневолює кращі результати стиску порівняно із застосуванням традиційних кластерів 8×8 та разом з тим збільшення кількості зображень у серії тільки незначно погіршує значення сигнал-шум при відновленні, як показано в [1]. У випадку серії зображень коваріаційна матриця знаходиться на основі кластерів всіх зображень серії.

Висновок

Запропонована нова методика кластеризації зображень для їх подальшого стиску та алгоритм для подальшого стиску на основі ПКЛ. Проведено дослідження ефективності стискання зображень на основі використання двох різних методів кластеризації та адаптивної методики на основі ПКЛ.

Дослідження показали, що запропоновані алгоритми дозволяють досягти більшого коефіцієнта компресії для заданого SNR, ніж традиційні алгоритми. Ефективність наведеного алгоритму зростає при застосуванні його для компресії серій зображень.

Список літератури: 1. Lee J. Optimized quadtree for karhunen-loeve transform in multispectral image coding”, Image Processing, IEEE Transactions on 8. 1999. P. 453-461. 2. JPEG: ITU-T Rec. T.81-ISO/IEC. No. 10918-1, “Information Technology – Digital compression and Coding of Continuous-Tone Still Images”, 1993. 3. Taubman D.S. and Marcellin M.W. JPEG2000: Fundamentals, Standards and Practice. Kluwer Academic Publishers, Boston, 2002. 4. Technical Papers from AT&T Labs: Електронний ресурс. Режим

доступу: <http://djvuzone.org/techpapers/index.html>. **5.** *Saghri J., Tescher A., and Reagan J.* Principal Transform coding of multispectral imagery // Signal Processing Magazine, IEEE, pp. 32-43, 2005. **6.** *Z. Wen, Z. Liu, M. Cohen, J. Li, K. Zheng, and T. Huang.* Low bit-rate video streaming for face-to-face teleconference // Multimedia and Expo, 2004. ICME '04. 2004 IEEE International conference, pp. 1631-1634 Vol.3, 2004. **7.** *Sergey Tayanov, Vitalij Tayanov.* The error analysis of speech signal computer compression by Karhunen-Loeve transform // Proceedings of the XI Polish-Ukrainian Conference on "CAD in Machinery Design. Implementation and Educational Problems." 2003. P. 113-119. **8.** *Jain A. and Dubes R.* Algorithms for clustering Data, Prentice Hall, 1988. **9.** *Jain A., Murty M., and Flynn P.* Data clustering: A review // ACM Computing Surveys 31, 1999.

Надійшла до редколегії 07.12.2008

Таянов Сергій Анатолійович, канд. техн. наук, доцент кафедри АКМНУ "Львівська політехніка". Наукові інтереси: обробка сигналів та зображень. Адреса: Україна, 79013, Львів, вул. С. Бандери, 12, тел. 258-25-98, e-mail: stayanov@polynet.lviv.ua.

Таянов Віталій Анатолійович, канд. техн. наук, наук. співр. ФМІ ім. Г.В. Карпенка НАН України. Наукові інтереси: математичні методи розпізнавання образів. Адреса: Україна, 79601, Львів, вул. Наукова, 5а, тел. 229-65-30, e-mail: vtayanov@ipm.lviv.ua.

УДК 004.652.4+004.827

Н.Б. ШАХОВСЬКА, Д.І. УГРИН

ТЕХНОЛОГІЯ ETL В ІНТЕГРАЦІЇ ДАНИХ ТУРИСТИЧНОГО БІЗНЕСУ

Розглядається одна із технологій консолідації даних – ETL. Будуються алгоритми для різних етапів виконання ETL та показуються вузькі місця цієї технології.

Вступ

Дослідження інтеграції даних туристичного бізнесу дає змогу оцінити розвиток технічних засобів, технологій, додатків і продуктів з метою подальшого прийняття ефективних рішень стосовно розвитку галузі. У фокусі нового дослідження перебуває не тільки роль інтеграції даних у проектах по створенню сховищ, але й технологічна база керування метаданими.

Засоби ETL – це технологія управління метаданими. На сьогодні існують два типи розв'язання цієї задачі: доступ до різних баз даних за допомогою SQL; використання інструментів ETL, що витягують дані і поміщають їх у сховища, минувши логіку застосувань.

ETL орієнтована на бази даних, сховища, вітрини або операційні сховища даних та виконує процес проектування процедур витягання, завантаження і перетворення даних (ETL – extract, transform and load). На цьому етапі інтеграції вирішуються проблеми сумісності даних з різних джерел. Разом з аналізом і стандартизацією інформації – це один з найтрудомісткіших і відповідальних етапів, а також один з найбільш витратних за часом.

1. Актуальність роботи

Мета роботи – інтеграція у просторі даних туристичної сфери.

Тема дослідження проблеми інтеграції даних в туристичному бізнесі є актуальною, оскільки методи інтеграції даних, що використовуються, не можуть до кінця реалізувати весь обсяг виконуваного процесу, а саме інтегрують лише частину спільних для різних застосувань даних.

Інтеграція даних дозволить:

- одержати повну інформацію про клієнтів;
- виконати одержання поточних фінансових даних для обов'язкової звітності й виконання вимог законодавства;
- консолідувати поточну інформацію з декількох джерел.

Технологія ETL в інтеграції даних виявляється найкориснішою в тих випадках, коли необхідно створити сховище даних, що містить добре документовані і надійні дані для історичного аналізу, наприклад, для аналізу часових рядів або багатовимірних запитів. Вона

також використовується для інтеграції ключових довідкових даних. Технологія ETL незамінна для таких завдань, як видалення дублікатів, здійснення процесів перевірки якості даних і т.ін. Ці інструменти також використовуються для створення окремих вітрин даних, що обслуговують конкретний відділ або бізнес-процес, або призначених для яких-небудь довгострокових цілей. Інструменти ETL дають користувачу можливість запустити процеси, що повторюються, для більшої злагодженості дій і можливості їх багаторазового використання. Такі процеси включають створення точних технічних метаданих, що підтримують загальну цілісність середовища business intelligence (BI).

З активним розповсюдженням Internet процеси ETL стали все ширше використовуватися і для підтримки Web-застосувань. Наприклад, туристичний оператор може використовувати засоби ETL для «закачування» у Web-систему даних, необхідних для перевірки стану опрацювання замовлень, з внутрішніх програм. Програми, що реалізують підхід ETL, стають важливими компонентами багатьох ініціатив, пов'язаних з електронною комерцією, зокрема застосувань, що підтримують взаємодію між партнерами бізнесу, а також між виробником і споживачами.

Стандартизація послідовності операцій при завантаженні сховищ даних, з огляду на важливість і вартість багатьох рішень ETL, дозволить уникнути повторення помилок, зроблених у попередніх розробках.

Крім того, досвід розробки ETL виявив загальні частини ETL-процесів при завантаженні різномірних джерел, що дозволяє говорити про однаковість підходу до розробки ETL для джерел даних довільного походження.

2. Постановка задачі та аналіз літературних джерел

Інтеграція даних дозволяє одержати єдину цілісну базу даних з різних джерел. Вона може бути описана за допомогою моделі, що включає додатки, продукти, технології й методи:

застосування – це рішення, створені розробниками відповідно до вимог клієнтів, які використовують одні або більше продуктів інтеграції даних;

продукти – це готові комерційні рішення, що підтримують одну або більше технологій інтеграції даних;

технології реалізують одні або більше методів інтеграції даних;

методи – це підходи до інтеграції даних, незалежні від технологій.

Враховуючи модель інтеграції даних у сховищах даних, можна виділити її переваги використання:

- інтеграція поточних та історичних значень даних;
- об'єднання даних з розрізнених джерел;
- створення надійної платформи даних для аналітичних цілей;
- забезпечення однорідності даних в організації;
- полегшення впровадження корпоративних стандартів даних без зміни існуючих операційних систем;
- забезпечення широкої історичної картини й можливостей для аналізу тенденцій;
- інтеграція даних будь-якого ETL-додатка полягає у своєчасному наданні даних його користувачам.

Інструменти маніпуляції даними ETL дозволяють переносити інформацію з одного середовища в інше, не створюючи спеціальних програм для відкриття файлів, читання записів, об'єднання даних і інших подібних операцій. Застосування ETL витягають інформацію з вихідної бази даних, перетворюють її у формат, підтримуваний базою даних, а потім завантажують у неї перетворену інформацію.

Схема загального опису ETL-технології (рис.1) показує, що при одержанні вихідної бази даних з вхідних даних бази потрібно пройти стадії витягання, перетворення та завантаження.

3. Основний матеріал

ETL полягає у виконанні таких укрупнених етапів:

1. Підготовка і отримання вхідних даних з декількох джерел.
2. Визначення вимог до вхідних даних і критеріїв перетворення.
3. Аналіз, фільтрація та перетворення вхідних даних.
4. Завантаження таблиці фактів.
5. Аналіз та верифікація сховища даних.

Ці етапи ETL-технології можна подати у вигляді блок-схем стосовно кожного пункту інтеграції даних. Розглянемо підготовку і отримання вхідних даних з декількох джерел. Сформуємо кілька таблиць, які містять однотипні дані, отримані з різних джерел інформації, зображені в різних форматах та структурах (MS Access, Visual Foxpro, dBase, Ms Excel, структурований текст тощо).

В основному джерела даних, що братимуть участь в інтеграції, містять однотипну інформацію, яка має спільний характер, але відрізняється складом, способами подання і форматами. На основі цих метаданих може бути утворена інтегрована таблиця, яка буде виконувати функції оперативного сховища даних.

Алгоритм підготовки і отримання вхідних даних з декількох джерел поданий на рис. 2.

Наступний крок визначає вимоги до вхідних даних і критеріїв перетворення, тобто для попередньо отриманих і збережених у оперативному сховищі даних розробимо запити на основі таких норм (рис. 3).

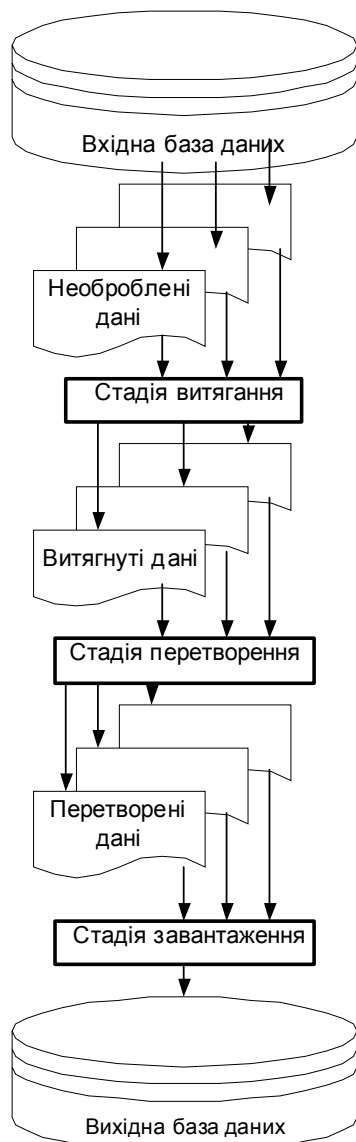


Рис. 1. Концептуальна схема загального опису ETL – технології

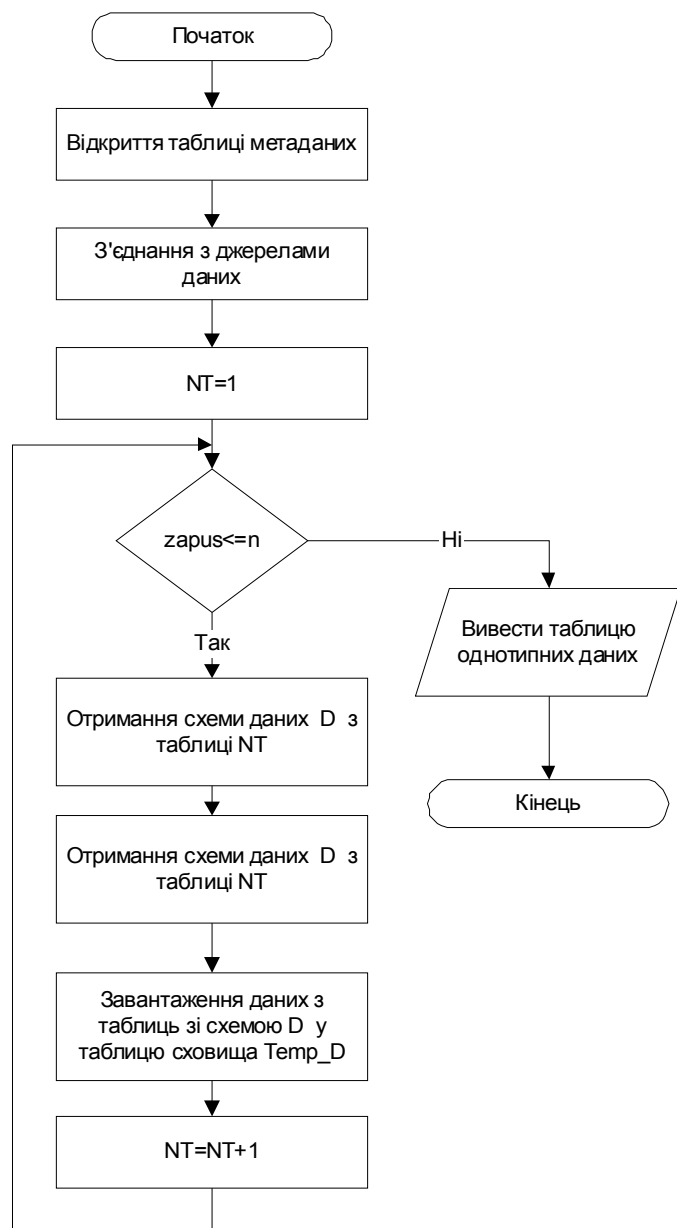


Рис. 2. Блок-схема формування однотипних даних

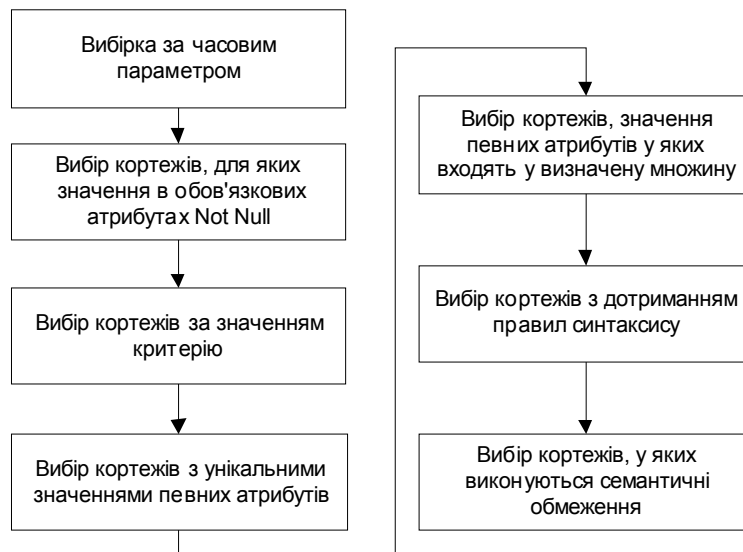


Рис. 3. Блок-схема системи норм і критеріїв

Виконавши другий крок ETL – технології, проведемо третій етап роботи, що носить назву: аналіз, фільтрація та перетворення вхідних даних. Сформуємо та виконаємо стосовно таблиці оперативного сховища даних запити на вилучення або зміну даних, які не відповідають критеріям, побудованим на другому кроці. Основними з таких критеріїв можуть бути:

– вилучення даних, які не містять обов'язкових значень і внаслідок цього не можуть бути застосовані у сховищі даних;

– виправлення некоректних чи помилкових значень (наприклад, коли у джерела даних внесено не коди підрозділів, а назви).

Алгоритм аналізу, фільтрації та перетворення вхідних даних.

1. За метаданими визначити структури даних.

2. Визначити, де в атрибутах замість foreign key використовується описове значення.

3. Якщо описове значення є в довідковій таблиці, замінити його на значення зовнішнього ключа.

4. Вилучити кортежі, у яких нема обов'язкових значень.

5. Додати дату завантаження.

Приклад підпрограми заміни описових значень на значення зовнішнього ключа, реалізованої на VBA:

```

Public Sub update_fact()
Dim str As String
Dim rs As Recordset

Dim str1 As String
Dim rs1 As Recordset
'вибираємо всі значення назв підрозділів
Set rs = CurrentDb.Execute("select distinct dept_name from documents1")
While Not rs.EOF
' визначаємо код підрозділу за назвою
str1 = "select id from dept where name=" & rs!dept_name
Set rs1 = CurrentDb.execute(str)
' замінюємо назву на знайдений код
str1 = "update documents1 set dept_name=" & rs1!id & " where dept_name=" & rs!dept_name
rs.MoveNext
Wend

End Sub
  
```

На четвертому кроці виконаємо завантаження таблиці фактів. Побудуємо для неї блок-схему та опишемо алгоритм (рис. 4).

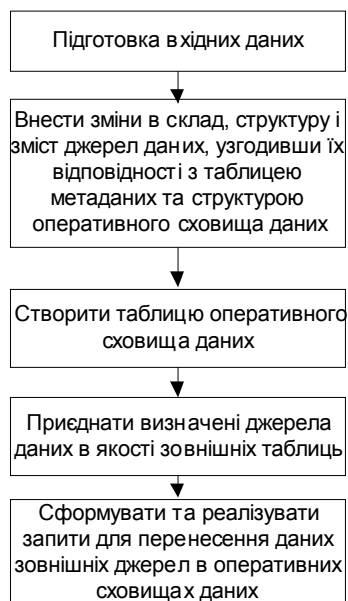


Рис. 4. Блок-схема завантаження таблиці фактів

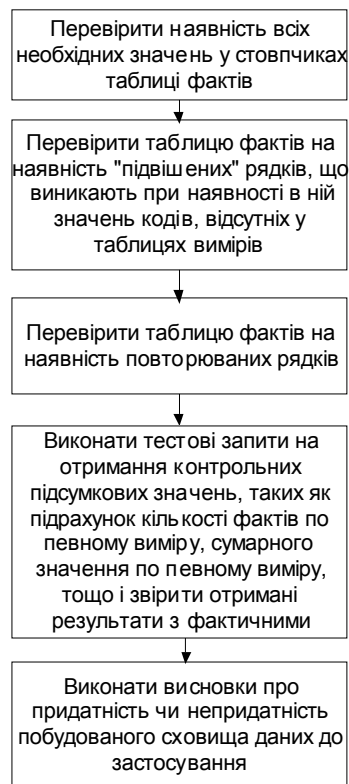


Рис. 5. Блок-схема аналізу та верифікації сховища даних

На п'ятому, завершальному кроці, проведемо аналіз та верифікацію сховища даних, тобто перевіримо та обґрунтуємо його працездатність (рис. 5).

Використовуючи наведену методичку, можна означити порядок та засоби добування, узгодження, інтеграції даних, створення оперативних сховищ даних та завантаження їх у центральне сховище.

Приклад реалізації перевірки усіх записів:

А) Рахуємо кількість даних в джерелах

```
SELECT Count(id)
FROM documents1;
```

і т.д.

Б) Додаємо отримані кількості.

В) Рахуємо кількість вставлених у таблицю сховища записів (дата внесення цих записів рівна сьогоднішній):

```
SELECT real_date, Count(id)
FROM document
GROUP BY real_date
HAVING (real_date=Date());
```

Якщо суми рівні, то ETL пройшов без перешкод.

Висновки

Розглянуто одну із технологій інтеграції даних – ETL, метою якої є створення корпоративного сховища даних. *Наукова новизна* дослідження полягає у порівнянні підходів до реалізації консолідації та обґрунтуванні потреби створення репозитарія метаданих. *Практична цінність* полягає у побудові алгоритмів консолідації. *Подальші дослідження* стосуватимуться формалізації методів інтеграції даних та пошуку неструктурованих, напівструктурованих та строго структурованих даних.

Список літератури: 1. *Колін Уайт* (Colin White). Интеграция данных: использование технологий ETL, EAI и EII для создания интегрированной корпорации (Data Integration: Using ETL, EAI, and EII Tools to Create an Integrated Enterprise). Ноябрь 2005. 2. Особенности построения хранилищ данных. <http://citforum.uar.net/seminars/cis99/sch.shtml/>. 3. Интеграция данных и Хранилища. –2005, <http://citecity.ru/12101/>. 4. Огляд технологій інтеграції інформаційних систем, 2006, <http://www.microsoft.com/Ukraine/Government/Analytics/IntegrationTechnologies/Overview.msp>. 5. *Кэтэрин Дрюэк* (Katherine Drewek). “Хранилища данных: сходство и различия подходов Билла Инмона и Ральфа Кимболла”, 2005, <http://www.b-eye-network.com/view/743>. 6. *Dan Linstedt*. Data Vaulttm overview the next evolution in data modeling. – 2005, <http://www.tdan.com/i021hy01.htm/>.

Надійшла до редколегії 06.04.2008

Шаховська Наталя Богданівна, канд. техн. наук, доцент кафедри інформаційних систем та мереж НУ «Львівська політехніка». Наукові інтереси: сховища даних, простори даних, методи інтеграції та опрацювання даних. Адреса: Україна, 79013, Львів, вул. С. Бандери, 12.

Утрин Дмитро Ілліч, аспірант НУ «Львівська політехніка». Наукові інтереси: простори даних в сфері туризму, методи опрацювання даних. Адреса: Україна, 79013, Львів, вул. С. Бандери, 12.

УДК 615.85

Н.Г. ДОЛГОВА, М.В. НОВОЖИЛОВА, О.И. СИНЕЛЬНИКОВА

МЕТОД ОЦЕНКИ АЛЬТЕРНАТИВНЫХ ВАРИАНТОВ ФУНКЦИОНАЛЬНОГО ЗОНИРОВАНИЯ ТЕРРИТОРИИ ГОРОДА

Предлагается метод оценки альтернативных вариантов развития территорий города, который позволяет лицу, принимающему решение (ЛПР), провести комплексный анализ населенного пункта, определить ресурсный потенциал и выделить направления его стратегического планирования.

1. Актуальность исследований

В течение последних десятилетий градостроительные и санитарно-гигиенические нормативы являлись определяющими при планировании и проектировании городских территорий. Эти нормативы определяли в основном требования не к городу как территориально-целостному образованию, а к отдельным его районам, различным по функциям – промышленным зонам, селитебным территориям, инженерно-транспортным коридорам и т.д.

На современном этапе развития экономики Украины становится очевидным, что реализация имеющегося потенциала города должна осуществляться на основе его комплексного анализа как единой функциональной системы. При этом уже на этапе планирования необходимо оценить инвестиционную привлекательность городских территорий в целях предотвращения возможного ущерба от их нерационального использования.

Механизм получения такой оценки предполагает разработку соответствующего формального аппарата и инструментальных средств анализа больших объемов слабоформализованной, зачастую неструктурированной информации о городской территории как объекте инвестирования.

2. Теоретическая база исследований

Анализ отечественных и зарубежных публикаций, посвященных проблеме создания и обоснования генеральных планов городских территорий, позволил выделить такие основные научные направления. Задачи размещения объектов промышленного, гражданского и жилого строительства в территориальном аспекте рассматриваются в работах Э.Б.Алаева, Н.Н. Барановского, Э.П. Григорьева, Э.Г. Петрова, И.С. Степанова [1,2]. Методические основы оценки инвестиционной привлекательности территорий разработаны в трудах А.Шахназарова, И.Ройзмана, К.Гусевой, Н.Марковой.

Концептуальным вопросам создания программного обеспечения компьютерного проектирования объектов строительства, в том числе компьютерных средств визуализации и поддержки принятия решений, посвящены работы таких ученых как Л.Э.Ванда, А.А.Гусак, Н.И. Ильин, Э.П. Григорьев [3,4].

Анализ функциональных особенностей существующих программных продуктов, таких как AutoCAD Architecture, AutoCAD Revit Architecture Suite, Autodesk Revit Architecture и ряда других на их основе показал, что возможность оценки экономической целесообразности размещения объектов строительства, а также инвестиционной привлекательности территорий как результата размещения рассматриваемых объектов в них отсутствует.

Таким образом, для повышения эффективности решения задач генерального планирования и проектирования сложных объектов, таких как объекты недвижимости (ОН), характеризующихся существенной неопределенностью предпроектной информации, актуально создание системы поддержки принятия инвестиционных решений (СППИР) [5]. Разработка и внедрение СППИР в деятельность субъектов государственного и муниципального управления иностранных и национальных инвесторов приводит к необходимости применения математических методов и создания математических моделей принятия решений, позволяющих комплексно и всесторонне анализировать проблемные ситуации, характерные для этапа инициации инвестиционного проекта строительства объектов недвижимости (предпроектный этап) [6].

3. Постановка задачи

Пусть задана территория города, разделенная на n районов. В качестве единицы территориального деления в работе принят бытовой район города¹.

Каждый бытовой район может быть описан набором свойств:

$$P = (p_1, p_2, \dots, p_1, \dots, p_L). \quad (1)$$

Частные свойства p_l имеют различный функциональный смысл, размерности, интервалы возможных значений, например, среди частных свойств такие: близость к метро, деловой центр города, ставки капитализации для различных типов недвижимости и так далее. Поэтому в рамках данной работы реализован подход, позволяющий получить количественные оценки частных свойств p_l – критерии k_l [7].

Пусть k_l^j – количественная оценка l -го частного свойства для j -го бытового района.

Для решения задачи приведения критериев k_l^j к изоморфному виду в работе использована нормализация и построена функция локальной полезности f_l^j критерия k_l^j вида [8]:

$$f_l^j = \frac{k_l^j - \min_{j \in J} k_l^j}{\max_{j \in J} k_l^j - \min_{j \in J} k_l^j}, \quad l = \overline{1, L}, \quad (2)$$

где J – множество индексов, $j \in J$, $|J| = n$.

Полученные оценки j -го бытового района образуют вектор $F_j = (f_1^j, f_2^j, \dots, f_1^j, \dots, f_L^j)$, $j = \overline{1, n}$. Нормализация частных критериев по всему списку бытовых районов необходима еще и для того, чтобы районы были сравнимы между собой в контексте всего города.

Множество F_j построенных количественных оценок свойств района образует систему частных критериев, которые удовлетворяют основным требованиям: полнота, минимальность, операциональность, декомпозируемость, измеримость.

В контексте данной работы свойства $P_j = (p_1^j, p_2^j, \dots, p_1^j, \dots, p_L^j)$ (и, соответственно, частные критерии k_l) условно разделены на группы (тип градостроительной зоны, социально-демографические показатели, экономическая база района, транспортная инфраструктура и уровень развития рынка недвижимости). Пусть M – число рассматриваемых групп свойств.

¹ Бытовой район – селитебная зона города со сложившейся социально-бытовой и транспортной инфраструктурой, включающая прилегающие территории промышленных и рекреационных зон.

Обозначим $\{K_g\}_{g=\overline{1,M}}$ – множество групп критериев. В свою очередь каждая g -я группа состоит из ряда частных критериев k_r^g , $r = \overline{1, m_g}$, где $m_g = |K_g|$, $g = \overline{1, M}$. При этом $L = \sum_{g=1}^M m_g$.

Имеется также множество $A = (A_1, A_2, \dots, A_i, \dots, A_N)$ возможных вариантов – альтернатив – функционального развития территорий города. Функциональное развитие территории подразумевает строительство объектов определенного типа недвижимости и/или проведение реконструкции существующей застройки.

Тогда задача сводится к определению для каждого бытового района подмножества альтернатив $A_i^j \subset A$ допустимых, а на подмножестве A_i^j – наиболее эффективных вариантов функционального развития.

Обобщенная оценка y_i^j соответствия свойств j -го бытового района определенной альтернативе A_i имеет вид: $y_i^j = \varphi(W_i, F_j)$, где $W_i = (w_1^i, w_2^i, \dots, w_1^i, \dots, w_L^i)$ – вектор оценок значимости частных критериев для выбора i -й альтернативы; $\varphi(\cdot)$ – некоторая функция свертки [6].

В работе исследовалась аддитивная функция свертки с последующей нормировкой вида:

$$y_i^j = \frac{\sum_{l=1}^L w_l^i f_l^j}{\sum_{i=1}^N \sum_{l=1}^L w_l^i f_l^j}. \quad (3)$$

Таким образом, задачу выбора наиболее перспективного функционального развития для каждого j -го бытового района можно записать следующим образом.

Определить:

$$A_j^* = \{A_{i0} \in A : y_{i0}^j = \max_i y_i^j, i = \overline{1, N}\}, \quad (4)$$

где A_j^* – множество альтернатив для j -го бытового района, соответствующих перспективным вариантам развития, для которых значение оценки y_i^j соответствия свойств j -го бытового района i -й альтернативе максимальное среди оценок для всех альтернатив.

4. Общая схема метода решения задачи

Метод решения задачи выбора (4) для каждого бытового района наиболее перспективных альтернатив развития содержит следующие этапы.

Этап 1. Проводим количественную оценку свойств (1) для каждого j -го бытового района, получаем оценки \tilde{f}_l^j для каждого l -го свойства для j -го бытового района.

Этап 2. Выполняем нормализацию (2) и получаем набор векторов $F_j = (f_1^j, f_2^j, \dots, f_1^j, \dots, f_L^j)$, $j = \overline{1, n}$.

Этап 3. Определяем оценки значимости частных критериев k_l для каждой альтернативы A_i .

Этап 4. Вычисляем оценки степени соответствия бытового района альтернативам y_i^j , $j = \overline{1, n}$, $i = \overline{1, N}$, согласно (3), т.е. для каждого бытового района получаем оценки степени соответствия каждой альтернативе.

Этап 5. Задаем интервал значений оценок y_i^j , в соответствии с которым осуществляем выбор подмножества наиболее перспективных альтернатив для каждого бытового района, что является решением задачи (4).

5. Метод получения оценок значимости частных критериев

Для выполнения этапа 3 в работе адаптирован бальный метод получения оценок $W_i = (w_1^i, w_2^i, \dots, w_1^i, \dots, w_L^i)$ значимости частных критериев k_l для каждой альтернативы A_i , $i = \overline{1, N}$, которые составляют матрицу $W = \|w_l^i\|$, $i = \overline{1, N}$, $l = \overline{1, L}$.

В случае недостатка данных выбор наиболее перспективных вариантов может осуществляться на основании как полной группы критериев, так и отдельных групп, например градостроительные особенности, уровень развития рынка недвижимости и др.

Оценка значимости частных критериев на альтернативы для решения поставленной задачи на основании статистического анализа затруднена, что связано с отсутствием достаточных объемов статистических данных, а особенно ретроспективных данных о рынке недвижимости. Данный факт обуславливает использование методики экспертного оценивания [7].

Результаты опроса экспертов с последующей оценкой степени согласованности и в случае необходимости коррекцией сведены в табл. 1.

Таблица 1
Система критериев оценки альтернатив

Группы критериев \ Альтернативы	K_1 – градостроительные особенности					...	K_g		...	K_M			
	k_1^1	k_2^1	k_r^1	..	k_{m1}^1	..	k_1^g	..	k_{mg}^g	..	k_1^M	..	k_{mM}^M
A_1	b_{11}	b_{12}											
A_2	b_{21}	b_{22}											
...													
A_i			b_{ir}				b_{il}						
...													
A_N					b_{Nm1}								b_{NL}

В табл. 1 b_{ir} – экспертная оценка влияния r -го частного критерия на i -ю альтернативу, при этом оценки могут принимать значения от « $-b_{max}$ » до « b_{max} », $b_{max} \in R^+$ (в настоящей работе $b_{max} = 5$).

Метод предполагает анализ значимости групп критериев.

Оценки λ'_{ig} соответствия g -й группы максимальному значению для i -й альтернативы определяются по формуле:

$$\lambda'_{ig} = \frac{\sum_{r=1}^{m_g} |b_{ir}|}{m_g \cdot b_{max}}$$

Нормировка полученных оценок λ'_{ig} выполняется по формуле:

$$\lambda_{ig} = \frac{\lambda'_{ig}}{\sum_{g=1}^M \lambda'_{ig}}$$

Таким образом, получена значимость g -й группы критериев для каждой i -й альтернативы.

Следующий шаг метода – анализ значимости частных критериев внутри каждой группы.

Оценки θ'_{ir} соответствия балла b_{ir} максимальному значению вычисляются по формуле:

$$\theta'_{ir} = \frac{b_{ir}}{b_{\max}}, \quad r = \overline{1, m_g}, \quad g = \overline{1, M}.$$

Нормировка полученных оценок θ'_{ir} внутри каждой g -й группы критериев имеет вид:

$$\theta_{ir} = \frac{\theta'_{ir}}{\sum_{r=1}^{m_g} \theta'_{ir}}, \quad r = \overline{1, m_g}, \quad g = \overline{1, M}.$$

Для каждой g -й группы критериев получаем матрицу оценок значимости всех r -х частных критериев, входящих в g -ю группу, на каждую i -ю альтернативу – $\Theta_g = \|\theta_{ir}\|$, $i = \overline{1, N}$, $g = \overline{1, M}$, $r = \overline{1, m_g}$. Каждую i -ю строку матрицы Θ_g умножаем на соответствующее значение коэффициента значимости g -й группы на i -ю альтернативу – λ'_{ig} , $i = \overline{1, N}$, $g = \overline{1, M}$, получая тем самым подматрицу \tilde{W}_g матрицы $\tilde{W} = \|\tilde{w}_i\|$ оценок значимости частных критериев на альтернативы: $\tilde{W} = (\tilde{W}_1, \tilde{W}_2, \dots, \tilde{W}_g, \dots, \tilde{W}_M)$.

Компоненты важности альтернатив в рамках каждого критерия вычисляются по формуле:

$$w_{il} = \frac{\tilde{w}_{il}}{\sum_{i=1}^N \tilde{w}_{il}}.$$

Таким образом, получена матрица $W = \|w_{il}\|$, $i = \overline{1, N}$, $l = \overline{1, L}$ оценок приоритетности альтернатив с учетом внутренней структуры значимости частных критериев.

6. Анализ научных и практических результатов

Оценка эффективности предложенного подхода проведена на основе анализа развития города Харькова. В качестве альтернатив были выбраны: торговая недвижимость, офисная недвижимость, многоэтажное жилье, малоэтажное жилье, гостиничная недвижимость, промышленная недвижимость и складская недвижимость ($N=7$). По каждому бытовому району города Харькова ($n=58$) были собраны данные соответственно по всем признакам с последующей количественной оценкой и получена система критериев, состоящая из 52 частных критериев ($L=52$, $M=5$ – количество групп).

В результате обработки были получены оценки перспективности развития всех альтернатив для каждого бытового района. В табл. 2 приведены только наиболее привлекательные бытовые районы для каждой альтернативы.

На рис.1 приведены результаты оценки перспективности развития альтернативы – многоэтажное жилье. Наиболее темным цветом отражены районы, где целесообразнее развивать указанную альтернативу.

Представленный подход оценки наиболее перспективного варианта функционального развития бытового района реализован в рамках разработки СППИР «Девелопмент». Общая структура данной информационной системы приведена на рис. 2.

Рассматриваемый метод реализован в виде функционального модуля блока 3.2.2, при этом данные, необходимые для проведения анализа, хранятся в общей базе данных

Таблица 2
Альтернативы развития районов

Альтернатива	Название бытового района
Торговая недвижимость	Лесопарк, Немышля
Офисная недвижимость	Гончаровка, м. Научная
Многоэтажное жилье	Лесопарк, Северная Салтовка
Малоэтажное жилье	Немышля, Шатиловка
Гостиничная недвижимость	Лесопарк, пос. Жуковского
Промышленная недвижимость	Диканевка, Гончаровка
Складская недвижимость	Диканевка, Аэропорт



Рис. 1

– блок 3.1.2, доступ к которым обеспечивается реализацией запросов блока 3.1.1.

Результаты обработки данных предоставляются лицу, принимающему решения (блок 5), в следующих формах: таблица оценок степени соответствия свойств бытового района альтернативам, списки наиболее перспективных альтернатив для каждого района, графическое отображение на карте города результатов оценок для каждой альтернативы.

На рис.3 приведено одно из окон интерфейса функционального блока, реализующего представленный метод анализа перспектив развития. В данном окне можно просматривать результаты оценки степени соответствия признаков бытового района, информация о котором указана в нижней части окна.



Рис. 2

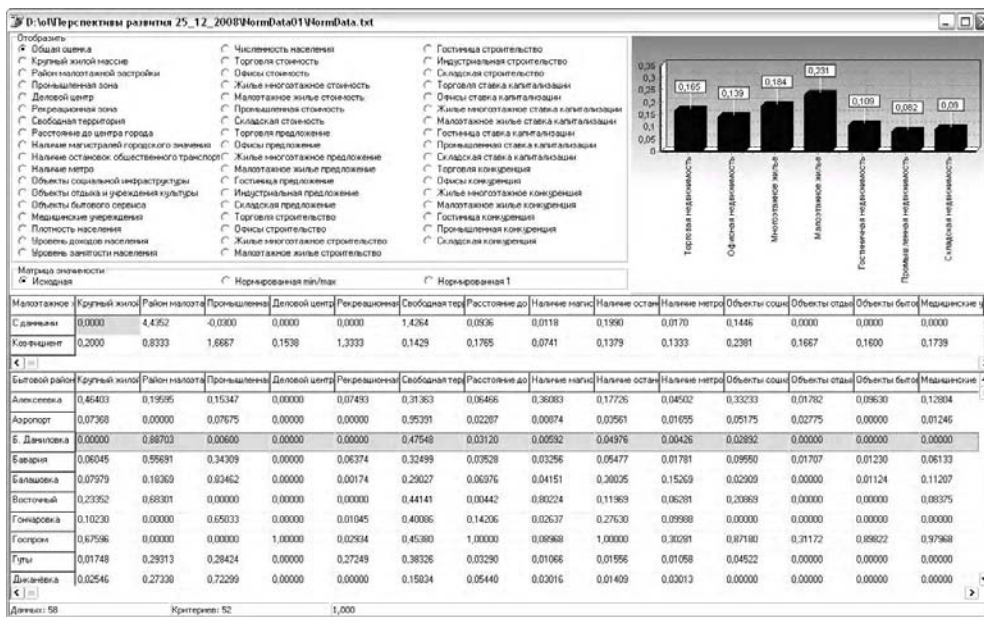


Рис. 3

Выводы

Предлагаемый метод позволяет получить для территориальной единицы (город, район города, выделенный земельный участок) оценки перспективности развития того или иного типа недвижимости в условиях изменяющегося рынка недвижимости. Также метод может быть использован при определении основных технических характеристик и территориального размещения объекта недвижимости в ходе реализации девелоперского проекта.

Список литературы: 1. *Петров Э.Г.* Алгоритм размещения нескольких предприятий обслуживания одной специализации в АСУ жилищным строительством. М.: Мысль, 1995. 2. *Григорьев Э.П.* Теория и практика машинного проектирования объектов строительства. М., 1974. 248 с. 3. *Гусаков А.А., Ильин Н.И., Эдели Х.* Экспертные системы в проектировании и управлении строительством. М.: Стройиздат, 1995. 463 с. 4. *Ильин Н.И.* Системный подход в строительстве. М.: Стройиздат, 1994. 215 с. 5. *Долгова Н.Г.* Модель системы поддержки принятия решений на рынке недвижимости // Сборник научных статей международной конференции "Компьютерное моделирование - 2007". СПб: Питер, 2007. 6. *Мазур И.И., Шатилов В.Д., Ольдерогге Н.Г.* Девелопмент. М.: Экономика, 2004. 359 с. 7. *Новожилова М.В., Долгова Н.Г.* Критерии определения привлекательности районов города для реализации девелоперских проектов // Научный вісник будівництва. Вип. 38. Харків: ХОТВ АБУ, 2006. 8. *Петров Е.Г., Новожилова М.В., Гребеннік І.В.* Методи і засоби прийняття рішень у соціально-економічних системах. К.: Техніка, 2003. 240с. 9. *Литвак Б.Г.* Экспертные информации: методы получения и анализа. М.: Радио и связь, 1982. 184с.

Поступила в редколлегию 17.12.2008

Долгова Наталья Геннадьевна, аспирантка кафедры компьютерного моделирования и информационных технологий Харьковского государственного университета строительства и архитектуры. Научные интересы: теория принятия решений, автоматизированное проектирование. Адрес: Украина, 61000, Харьков, ул. Сумская, 40, тел. +38(067)7837492, e-mail: Ngt_d@mail.ru.

Новожилова Марина Владимировна, д-р физ.-мат. наук, профессор, зав. кафедрой компьютерного моделирования и информационных технологий Харьковского государственного технического университета строительства и архитектуры. Адрес: Украина, 61000, Харьков, ул. Сумская, 40, тел.: 706-20-49, e-mail: novozhilova@kstuca.kharkov.ua.

Синельникова Ольга Игоревна, канд. техн. наук, старший преподаватель кафедры прикладной математики ХНУРЭ. Научные интересы: системный анализ и теория принятия решений при управлении социально-техническими системами. Адрес: Украина, 61000, Харьков, ул. Кирова, 5, кв. 120, тел./факс: +38(097)6550969.

ПРОИЗВОДИТЕЛЬНОСТЬ БУФЕРНЫХ АСИНХРОННЫХ МНОГОСТУПЕНЧАТЫХ СЕТЕЙ С ПРОИЗВОЛЬНЫМ ЧИСЛОМ ПРИОРИТЕТНЫХ МОДУЛЕЙ ПАМЯТИ

Синтезируется математическая модель асинхронной многоступенчатой коммутирующей сети, которая применяется для оценки пропускной способности и времени отклика сети. Модель расширяется для случая произвольного числа приоритетных модулей памяти и случайно распределенного размера пакетов. Рассматривается влияние длины буфера на общую производительность системы.

Общая постановка проблемы

Многоступенчатые коммутирующие сети используются в качестве коммутирующей фабрики во многих коммерческих и экспериментальных многопроцессорных системах [1]. Производительность коммутирующей сети во многом определяет производительность системы в целом. Чтобы достичь оптимального соотношения скорости системы к ее стоимости, архитектору необходимо изучить влияние различных вариантов конфигурации сети на производительность многопроцессорной системы. Зная время отклика сети и ее пропускную способность, можно сделать вывод о производительности будущей многопроцессорной системы.

Анализ исследований и публикаций

Существующие результаты получены на основе математических моделей, построенных с использованием, преимущественно, вероятностных методов [2]. Такие модели могут быть применены только для многопроцессорных SIMD-систем с синхронно работающими процессорами [3].

Однако наиболее значительный интерес представляют MIMD-системы с работающими независимо друг от друга процессорами. Количество процессоров у таких систем нередко превышает десятки тысяч.

Для оценки производительности асинхронных коммутирующих сетей были построены математические модели с использованием теории очередей [3]. Недостаток предложенной в [3] модели состоит в ограниченности области ее применения. Модель применима только к системам с детерминированным временем коммутации пакетов и с не более чем одним приоритетным модулем памяти.

Цель исследования: разработать программный инструментарий для оценки производительности асинхронной буферной многоступенчатой сети.

Задача исследования: 1) синтезировать математическую модель многоступенчатой коммутирующей сети, которая позволяет оценить пропускную способность и время отклика асинхронной коммутирующей сети до начала имплементации. Модель должна быть применима к системам с недетерминированным временем коммутации пакетов и произвольным числом приоритетных модулей памяти; 2) реализовать модель в виде программного обеспечения.

Функционирование многоступенчатых коммутирующих сетей

Многоступенчатая коммутирующая сеть состоит из множества переключающих элементов, организованных в виде последовательных ступеней (рис. 1). Процессорные элементы (PE) генерируют пакеты независимо друг от друга. Данное предположение справедливо для MIMD-систем, где отдельный процессор управляется собственным потоком команд.

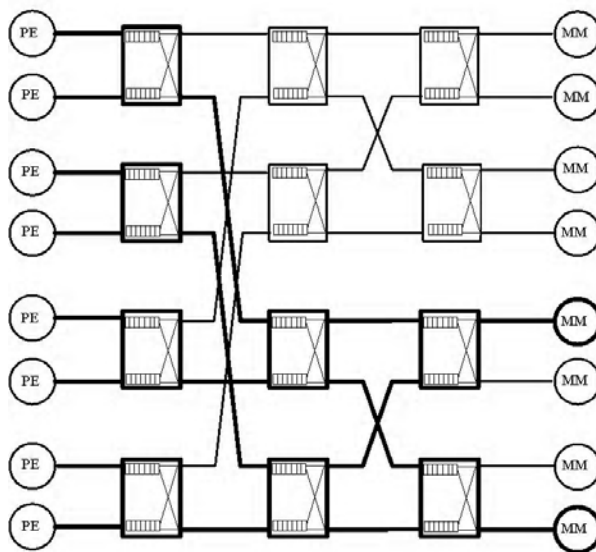


Рис. 1. Многоступенчатая коммутирующая сеть 8×8

Чтобы не потерять входящие пакеты в случае высокой интенсивности трафика, каждый переключатель имеет буфер для временного хранения нескольких пакетов. Далее будет проанализировано влияние размера буфера на производительность многоступенчатой сети.

Математическая модель

Модель строится на основе следующих предположений:

- 1) процессорные элементы генерируют пакеты независимо друг от друга с интенсивностью λ согласно распределению Пуассона;
- 2) время, которое требуется переключателю для перенаправления пакета на один из своих выходных каналов, пропорционально размеру пакета;
- 3) размер пакетов распределен согласно закону Эрланга;
- 4) каждый пакет равновероятно перенаправляется к любому модулю памяти;
- 5) в случае необходимости перенаправить более одного пакета на один выходной канал один из пакетов блокируется.

Большинство существующих моделей предполагают наличие трафика, равномерно распределенного между всеми модулями памяти. Данное предположение выражено в пункте 4. После построения упрощенной модели данное ограничение будет снято для получения более общей математической модели.

Введем следующие обозначения:

- 1) λ – интенсивность генерации пакетов;
- 2) μ – интенсивность перенаправления пакетов на выходные каналы;
- 3) $P_k^{(L)}$ – вероятность того, что в буфере размера L находится k пакетов;
- 4) $P_{k,s}^{(L)}$ – вероятность того, что в буфере размера L находится k пакетов и пакет, который перенаправляется на один из выходных каналов, находится в фазе s . $1 \leq s \leq r$.

$$\text{Очевидно, что } P_k^{(L)} = \sum_{s=1}^r P_{k,s}^{(L)}.$$

Путь, который проходит пакет, удобно представить в виде последовательно соединенных друг с другом очередей. Уровень выходного потока предыдущей очереди равен уровню входного потока последующей очереди.

После прохождения первой ступени поток более не подчиняется условиям Пуассона по причине влияния ненулевого времени переключения пакета. Однако имитационные эксперименты показали, что погрешность, вызванная аппроксимацией потока к пуассоновскому, пренебрежительно мала.

Введем дополнительные обозначения:

- 1) λ_i – интенсивность входного потока ступени i ;
- 2) $P_k^{(L)}(i)$ – вероятность нахождения k пакетов в буфере на ступени i ;
- 3) x_i – вероятность блокировки пакета на ступени i .

Пусть τ_i – случайная величина, которая представляет собой длину интервала времени прошедшего между отправками двух последовательных пакетов. Пусть ϕ – событие, обозначающее опустошение буфера после отправки последнего пакета. Тогда

$$f_{\tau_i}(t) = P_0^{(L)}(i) \cdot f_{\tau_i|\phi}(t) + (1 - P_0^{(L)}(i)) \cdot f_{\tau_i|\bar{\phi}}(t),$$

где $f_{\tau_i}(t)$ – функция распределения случайной величины τ_i . Используя свойства линейных операторов, имеем: $E[\tau_i] = P_0^{(L)}(i) \cdot E[\tau_i|\phi] + (1 - P_0^{(L)}(i)) \cdot E[\tau_i|\bar{\phi}]$. Далее находим, что

$$E[\tau_i] = P_0^{(L)}(i) \cdot \left(\frac{1}{\lambda_i} + \frac{1}{\mu} \right) + (1 - P_0^{(L)}(i)) \frac{1}{\mu}.$$

В случае, когда последний отправленный пакет оставляет за собой пустой буфер, время следующей отправки пакета будет состоять из двух составляющих: времени, необходимого для поступления следующего пакета, и времени, необходимого для перенаправления пакета.

Так как интенсивность входного потока равна интенсивности выходного потока предыдущей ступени, имеем: $\lambda_{i+1} = \frac{1}{E[\tau_i]} = \frac{(1 - x_i)\lambda_i\mu}{P_0^{(L)}(i)\mu + (1 - x_i)\lambda_i}$.

Используя теорему Литла, получаем: $E[T_i] = \frac{\sum_{k=1}^{L+1} k \cdot P_k^{(L)}(i)}{\lambda_i}$. Среднее время отклика сети рассчитывается как время пребывания пакета на ступени i , просуммированное по всем i . Пропускная способность сети рассчитывается как λ_i последней ступени.

Найдем выражение для $P_k^{(L)}(i)$. Воспользуемся результатами, приведенными в [4]:

$$P_{1,s} = \frac{\lambda}{\Gamma\mu} \left(1 + \frac{\lambda}{\Gamma\mu}\right)^{s-1} \cdot P_0, \quad P_{n,s} = \frac{\lambda}{\Gamma\mu} \left(1 + \frac{\lambda}{\Gamma\mu}\right)^{s-1} \left(\sum_{i=1}^r P_{n-1,i} - \sum_{i=1}^{s-1} \left(\frac{1}{1 + \frac{\lambda}{\Gamma\mu}}\right)^i \cdot P_{n-1,i} \right),$$

$$P_{N,s} = \frac{\lambda}{\Gamma\mu} \cdot \sum_{i=s}^r P_{N-1,i},$$

где $P_{n,s}$ – вероятность того, что в буфере находятся n пакетов и пакет в процессе переключения находится в фазе s . $1 \leq n \leq N$, $1 \leq s \leq r$.

Используя эти результаты, имеем: $P_k^{(L)} = \sum_{s=1}^r P_{k,s}$.

Приоритетные модули памяти

В многопроцессорных системах поток пакетов распределяется между приоритетными модулями памяти. Это происходит вследствие того, что наиболее важные данные находятся на одном или нескольких модулях памяти. В результате происходит насыщение фрагмента многоступенчатой сети и общее падение производительности системы [5]. Для анализа производительности сети при работе с неравномерным трафиком расширим описанную выше модель.

Рассмотрим случай единственного приоритетного модуля памяти MM_h . Методом половинного деления разобьем группу модулей памяти на подгруппы, пока в некоторой подгруппе не останется один модуль памяти MM_h . Процессоры генерируют поток пакетов независимо друг от друга. Некоторая часть h_0 сгенерированных пакетов направляется к приоритетному модулю памяти. Остальные пакеты, равномерно распределены между всеми модулями памяти.

Пусть процессоры генерируют пакеты с интенсивностью λ_0 .

Этап 1. Обозначим через λ_1 – интенсивность потока после прохождения первой ступени. Поток разделяется на два потока: один поток содержит пакеты как равномерно распределенные между всеми модулями памяти, так и предназначенные для приоритетного модуля памяти; второй поток содержит исключительно пакеты равномерно распределенные между всеми модулями памяти. Пусть λ_1^h – интенсивность первого потока, λ_1^r – интенсивность второго потока.

Этап 2. Находим λ_1^h и λ_1^r . $\lambda_1^h = \lambda_1(1 - h_0) + 2h_0\lambda_1$, $\lambda_1^r = \lambda_1(1 - h_0)$. Перепишем выражение для λ_1^h . $\lambda_1^h = \lambda_1^h(1 - h_1) + \lambda_1^h h_1$. Отсюда $h_1 = \frac{\lambda_1^h - \lambda_1(1 - h_0)}{\lambda_1^h}$.

Этап 3. После прохождения второй ступени поток с пакетами для приоритетного модуля памяти расщепляется на два потока: λ_2^h и λ_2^r . Находим λ_2^h и λ_2^r , $\lambda_2^h = \lambda_2(1 - h_1) + 2h_1\lambda_2$, $\lambda_2^r = \lambda_2(1 - h_1)$. Получаем $\lambda_2^h = \lambda_2^h(1 - h_2) + \lambda_2^h h_2$, где $h_2 = \frac{\lambda_2^h - \lambda_2(1 - h_1)}{\lambda_2^h}$.

Повторяем этап 3 до тех пор, пока получим интенсивность потока пакетов на последней ступени сети λ_N^h .

Процедуру выполняем для каждой подгруппы модулей памяти. Пропускная способность сети равняется арифметическому среднему величин λ_N^h , найденных для каждой подгруппы модулей памяти.

Применим данный метод для нахождения пропускной способности сети 8×8 с двумя приоритетными модулями памяти (см. рис. 1).

Разбиваем модули памяти на подгруппы: $M_0 : \{MM_0, MM_1, MM_2, MM_3\}$, $M_1 : \{MM_4\}$, $M_2 : \{MM_5\}$, $M_3 : \{MM_6\}$, $M_4 : \{MM_7\}$. Модули памяти MM_4 и MM_7 из групп M_1 и M_4 являются приоритетными с долей пакетов h_0^4 и h_0^7 соответственно. Оставшаяся часть пакетов $(1 - h_0^4 - h_0^7)$ равномерно распределена между всеми модулями памяти. Необходимо найти λ_3 для каждой подгруппы:

$$1) M_0: \quad \lambda_1^{M_0} = \lambda_1(1 - h_0^4 - h_0^7), \quad \lambda_2^{M_0} = \lambda_2, \quad \lambda_3^{M_0} = \lambda_3.$$

$$2) M_1: \quad \lambda_1^{M_1} = \lambda_1(1 - h_0^4 - h_0^7) + 2h_0^4\lambda_1 + 2h_0^7\lambda_1;$$

$$\lambda_1^{M_1} = \lambda_1^{M_1}(1 - h_1^4 - h_1^7) + \lambda_1^{M_1}h_1^4 + \lambda_1^{M_1}h_1^7,$$

$$\text{где } h_1^4 = \frac{\lambda_1^{M_1} - \lambda_1(1 - h_0^4 - h_0^7) - 2h_0^7\lambda_1}{\lambda_1^{M_1}}, \quad h_1^7 = \frac{\lambda_1^{M_1} - \lambda_1(1 - h_0^4 - h_0^7) - 2h_0^4\lambda_1}{\lambda_1^{M_1}};$$

$$\lambda_2^{M_1} = \lambda_2(1 - h_1^4 - h_1^7) + 2h_1^4\lambda_2; \quad \lambda_2^{M_1} = \lambda_2^{M_1}c_2^{M_1} + \lambda_2^{M_1}h_2^4,$$

$$\text{где } h_2^4 = \frac{\lambda_2^{M_1} - \lambda_2(1 - h_1^4 - h_1^7)}{\lambda_2^{M_1}}, c_2^{M_1} = \frac{\lambda_2^{M_1} - \lambda_2^{M_1} h_2^4}{\lambda_2^{M_1}}; \lambda_3^{M_1} = 2h_2^4 \lambda_3.$$

$$3) M_2: \quad \lambda_3^{M_2} = 2c_2^{M_1} \lambda_3.$$

$$4) M_3: \quad \lambda_2^{M_3} = \lambda_2(1 - h_1^4 - h_1^7) + 2h_1^7 \lambda_2;$$

$$\lambda_2^{M_3} = \lambda_2^{M_3} c_2^{M_3} + \lambda_2^{M_3} h_2^7,$$

$$\text{где } h_2^7 = \frac{\lambda_2^{M_3} - \lambda_2(1 - h_1^4 - h_1^7)}{\lambda_2^{M_3}}, c_2^{M_3} = \frac{\lambda_2^{M_3} - \lambda_2^{M_3} h_2^4}{\lambda_2^{M_3}}; \lambda_3^{M_3} = 2c_2^{M_3} \lambda_3.$$

$$5) M_4: \quad \lambda_3^{M_4} = 2h_2^7 \lambda_3.$$

$$6) \lambda = \frac{4\lambda_3^{M_0} + \lambda_3^{M_1} + \lambda_3^{M_2} + \lambda_3^{M_4}}{8}.$$

Анализ производительности

Используем математическую модель для получения некоторых показателей производительности многоступенчатой коммутирующей сети, а именно: *пропускной способности, времени отклика и системной производительности.*

Влияние размера буфера на пропускную способность сети показано на рис. 2. При размере буфера в две ячейки насыщение сети происходит при входной интенсивности в 0.6. Дальнейшее увеличение интенсивности входного потока не приводит к увеличению интенсивности потока на выходе по причине блокировки пакетов.

При размере буфера в пять ячеек насыщение сети происходит медленнее. Если величина интенсивности входного потока пакетов для проектируемой системы не планируется быть больше 0.5, дальнейшее увеличение размера буфера не приведет к росту пропускной способности сети.

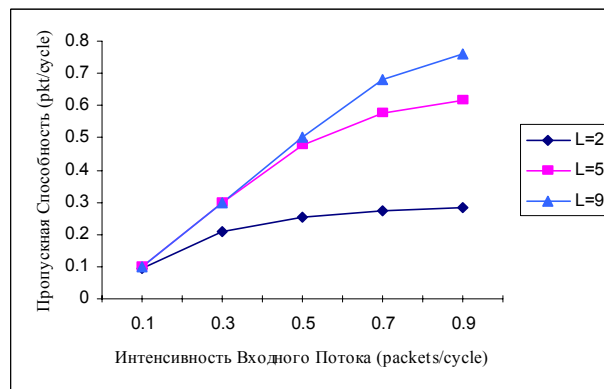


Рис. 2. Пропускная способность многоступенчатой сети 1024×1024

Для системы с размером буфера в девять ячеек увеличение интенсивности входного потока после 0.5 дает лишь незначительный прирост пропускной способности. Вместе с тем, время отклика сети увеличивается значительно (рис. 3).

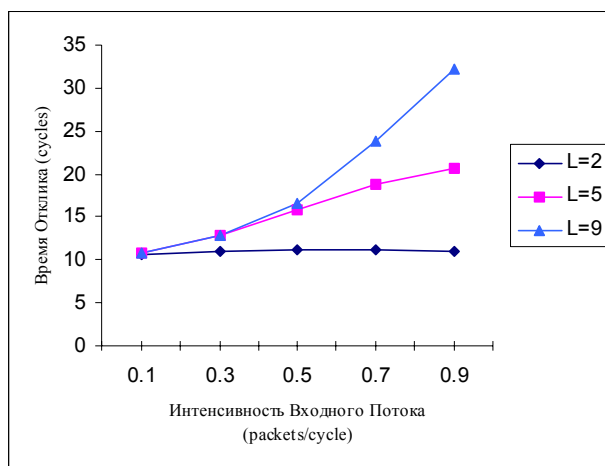


Рис. 3. Время отклика многоступенчатой сети 1024×1024

Отдельно взятые пропускная способность и время отклика не являются достаточными показателями производительности системы. При возрастании пропускной способности и насыщении сети также увеличивается время отклика сети. Определим системную производительность как отношение пропускной способности к времени отклика сети.

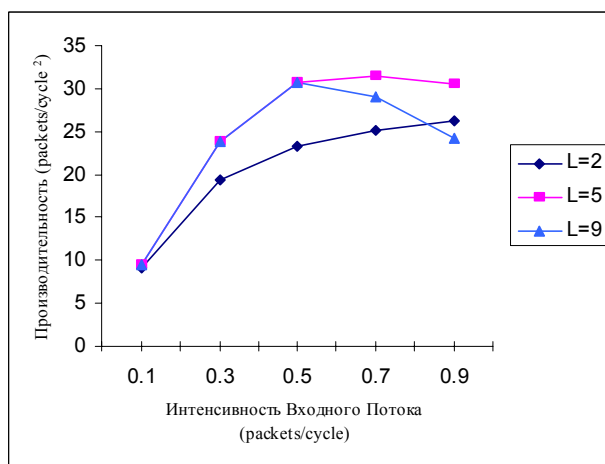


Рис. 4. Производительность многоступенчатой сети 1024×1024

Из графика для системы с буфером на девять ячеек (рис. 4) делаем вывод, что размер буфера в девять ячеек оправдан для систем, ориентированных исключительно на пропускную способность, для которых время отклика не критично.

Выводы

Основные результаты. Разработан программный пакет, который позволяет оценить производительность асинхронной многопроцессорной системы и оптимизировать ее путем подбора оптимальных размеров буферов многоступенчатой коммутационной сети.

Сравнение с лучшими аналогами. Применение реализованной в виде программного пакета математической модели не ограничено случаем единственного приоритетного модуля памяти и детерминированным временем коммутации пакета. Данный программный пакет также применим для случая произвольного числа приоритетных модулей памяти и недетерминированным временем коммутации.

Научная новизна. В данной работе обобщена модель, ранее предложенная в [3]. Впервые получена модель, которая применима к асинхронным многоступенчатым сетям с случайным временем коммутации, распределенным по закону Эрланга, и с произвольным числом приоритетных модулей памяти.

Практическая значимость. Программный комплекс позволяет ускорить разработку MIMD-системы благодаря возможностям расчета ключевых показателей производительности многоступенчатой сети на ранних этапах проектирования. Благодаря предварительной оценке производительности будущей коммутирующей сети, сокращаются затраты на имплементацию пробной версии сети, что уменьшает общую стоимость многопроцессорной системы.

Список литературы: 1. *J. Konicek, T. Tilton, et al*, The organization of the Cedar system // Proc. Int' Conf. Parallel Processing, Aug. 1991. P. 49-56. 2. *Y. Mun and H.Y. Youn*. Performance analysis of finite buffered multistage interconnection networks // IEEE Trans. Computers, Feb. 1994. P. 153-162. 3. *Prasant Mohapatra, Chita R. Das*. Performance Analysis of Finite-Buffered Asynchronous Multistage Interconnection Networks, IEEE Transactions on Parallel and Distributed Systems, Vol. 7, No. 1. 1996. P. 18–25. 4. El-Paoumy, On Truncated Erlangian Queuing Systems with State-Dependent Service Rate, Balking and Reneging, Applied Mathematical Sciences, Vol. 2. 2008. P. 1161-1167. 5. *Dickey S.R. and Percus O.E*. Performance differences among combining switch architectures, Proc. I&Z, Conf. on Parallel Processing, Aug. 1992. Vol. 1. P. 110-117.

Поступила в редколлегию 15.12.2008

Евграфов Вячеслав Николаевич, аспирант кафедры ПМ ХНУРЭ. Научные интересы: многопроцессорные системы, многоступенчатые сети. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 8(063)95-82-985.

УДК 681.325.53

Н.Я. КАКУРИН, С.Н. КОВАЛЕНКО, Ю.В. ЛОПУХИН, А.Н. МАКАРЕНКО

СПОСОБ ПОВЫШЕНИЯ БЫСТРОДЕЙСТВИЯ ПРЕОБРАЗОВАТЕЛЕЙ КОДОВ НА СЧЕТЧИКАХ

Предлагается способ повышения быстродействия преобразователей кодов на счетчиках с помощью параллельной стратегии использования шагов преобразования. Разрабатывается метод расчета числа тактов преобразования и программное обеспечение для его реализации.

1. Постановка задачи

Одношаговый преобразователь кодов по методу накопления эквивалентов обладает небольшими аппаратными затратами, но требует K тактов на преобразование числа [1]. При более высоких требованиях к быстродействию применяют двухшаговые, трехшаговые и другие преобразователи кодов с большим числом различных преобразований шагов [2-5].

Для максимального числа тактов преобразования $N_1^{дп}$ ПК дробных чисел в работах [1,2] была получена система формул

$$\begin{aligned} N_1^{дп} &= K-1; \\ N_2^{дп} &= \lfloor (K-1)/a \rfloor + a - 1; \\ N_3^{дп} &= \lfloor (K-1)/b \rfloor + \lfloor (b-1)/a \rfloor + a - 1; \\ N_4^{дп} &= \lfloor (K-1)/c \rfloor + \lfloor (c-1)/b \rfloor + \lfloor (b-1)/a \rfloor + a - 1; \end{aligned} \quad (1)$$

где K — основание системы счисления на входе; a, b, c — соответственно второй, третий и четвертый шаги преобразования (первый шаг преобразования равен 1); $N_1^{дп}, N_2^{дп}, N_3^{дп}, N_4^{дп}$ — максимальное число тактов преобразования соответственно одношагового, двухшагового, трехшагового и четырехшагового ПК.

В процессе преобразования применяется стратегия последовательного использования шагов преобразования, т.е. показание разрядных счетчиков вначале уменьшается на величину наибольшего шага. Затем, если во всех разрядах значения цифр оказываются меньше величины наибольшего шага, происходит переход на меньший шаг; затем на последнем этапе преобразования происходит переход на первый шаг, равный 1, и преобразование выполняется до полного обнуления всех разрядных счетчиков.

Увеличение числа различных шагов преобразования целесообразно, если уменьшается максимальное число тактов преобразования. Из всех возможных значений шагов преобра-

зования целесообразно выбирать их по возможности большими и равными степени двойки. В этом случае аппаратные затраты на построение других узлов ПК, таких как кодовые шифраторы (субтракторы) и дешифраторы нуля, а также дешифраторы превышения будут иметь меньшее значение. Эффект увеличения быстродействия для $K=3-12$ в двухшаговом ПК значителен и по сравнению с одношаговым составляет 1,33-1,71. При более высоких требованиях быстродействия следует использовать трехшаговый ПК.

2. Структура и функционирование трехшагового ПК

Рассмотрим структуру и функционирование трехшагового преобразователя кодов целых чисел (рис.1) с последовательным использованием шагов преобразования.

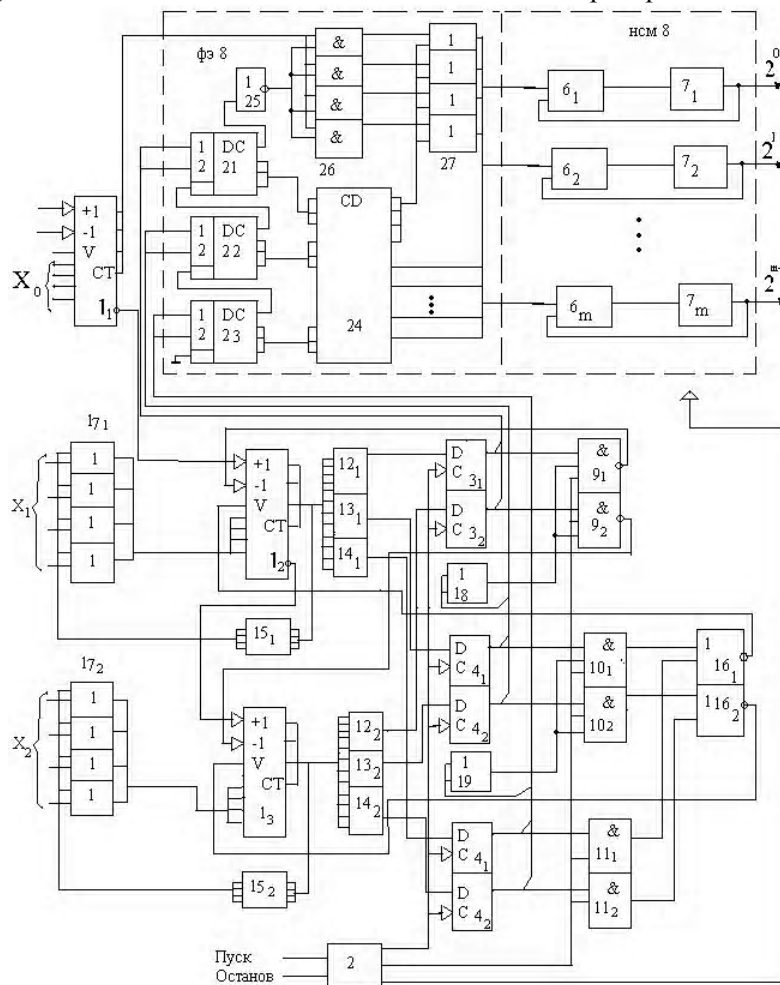


Рис. 1. Структура трехшагового ПК

ПК содержит группу разрядных счетчиков 1, генератор импульсов 2 (блок управления), содержащий прямой П, прямые задержанные ПЗ1 и ПЗ2 И выходы, первую группу триггеров 3 состояния, вторую группу триггеров 4 состояния, третью группу триггеров 5 состояния, комбинационный двоичный сумматор 6 и регистр 7 результата, образующие в совокупности накапливающий сумматор 8, первую группу элементов И-НЕ 9, вторую и третью группы элементов И 10, И 11, группу дешифраторов нуля 12, первую группу дешифраторов превышения 13, вторую группу дешифраторов превышения 14, группу шифраторов 15, первую группу элементов ИЛИ-НЕ 16, $(n-1)$ - группу элементов ИЛИ 17, первый элемент ИЛИ-НЕ 18, второй элемент ИЛИ-НЕ 19, формирователь эквивалента 20, включающий в свой состав первый 21, второй 22, третий 23 дешифраторы и шифратор 24.

В состав формирователя эквивалентов (ФЭ) 20 также входит инвертор НЕ 25, группа элементов И 26, группа элементов ИЛИ 27. Первая группа триггеров 3 фиксирует ненулевое значение счетчиков соответствующих разрядов. Вторая группа триггеров состояния 4

фиксирует значение счетчиков соответствующих разрядов, превышающее заданное число и, например 1. Третья группа триггеров состояния 5 фиксирует значение счетчиков соответствующих разрядов, превышающее заданное число a , например 3. Шифраторы 15 реализуют функцию:

$$y = \begin{cases} x - b; x \geq b; \\ x - a; a \leq x < b; \\ x; x < a, \end{cases} \quad (2)$$

где x – входной код; a – параметр (второй шаг преобразования), в частном случае, равный 2; b – параметр (третий шаг преобразования), в частном случае, равный 4.

Цепи инициирования и сброса на рис.1 не показаны. Так как в конкретном случае $n=3$; $K=12$, то диапазон изменения входного кода $0 - (12^3 - 1)$. Код состояния триггеров как первой группы 3, так и второй группы 4, а также третьей группы 5 имеет $2^2 = 4$ значения от 00 до 11. В данном ПК двоично- K -ичного кода в двоичный код ФЭ 20, выполненный в виде последовательного соединения первого 21, второго 22 и третьего 23 дешифраторов, реализует функцию:

$$S = \begin{cases} KC_1 + K^2C_2; E_1 = E_2 = D_1 = D_2 = 0; \\ aKD_1 + aK^2D_2; E_1 = E_2 = 0 \text{ и } D_1, D_2 \neq 0; \\ bKE_1 + bK^2E_2; E_1, E_2 \neq 0; \end{cases} \quad (3)$$

где C_1C_2 – значения разрядов двоичного кода триггеров состояний первой группы 3; D_1D_2 – значения разрядов двоичного кода триггеров состояния второй группы 4; E_1E_2 – значения разрядов двоичного кода триггеров состояния третьей группы 5.

Каждый из C_m ($m = 1,2$) разрядов кода триггеров 3 первой группы состояний разрядных счетчиков соответствует наличию (1) или отсутствию (0) информации в соответствующем старшем разряде преобразуемого кода, а разряд D_m ($m = 1,2$) триггеров 4 второй группы состояний разрядных счетчиков равен 1, если соответствующий разряд преобразуемого кода имеет значение $x_i \geq a$ ($1 < a < b$), в противном случае $D_m = 0$; разряд E_m ($m = 1,2$) триггеров 5 третьей группы состояний разрядных счетчиков равен 1, если соответствующий разряд преобразуемого кода имеет значение $x_i \geq b$ ($a < b \leq K - 1$), в противном случае $E_m = 0$. ФЭ 20 преобразует вначале двоичный код E_1, E_2 триггеров состояния третьей группы, затем при $E_1 = E_2 = 0$ и $D_1, D_2 \neq 0$ – двоичный код D_1, D_2 триггеров состояния второй группы, затем при $E_1 = E_2 = D_1 = D_2 = 0$ и $C_1, C_2 \neq 0$ – двоичный код C_1, C_2 триггеров состояний первой группы и при $E_1 = E_2 = D_1 = D_2 = C_1 = C_2 = 00$ выполняет трансляцию (передачу) двоично- K -ичного кода младшего разряда в двоичный код эквивалента на выходе.

Преобразование числа выполняется по принципу последовательного использования шагов; вначале третий шаг b ; затем второй шаг a , наконец, шаг 1. При поступлении импульсов генератора 2 с выхода П информация о состоянии разрядных счетчиков заносится в регистры состояний, ФЭ выдает код эквивалента, соответствующий состояниям регистров 3,4,5; затем из состояний разрядных счетчиков 1 вычитается значение текущего шага (по заднему фронту 0-1 импульса с выхода 11,31. И, наконец, по заднему фронту 1-0 импульса с выхода ПЗ2 выполняется прибавление эквивалента в накапливающий сумматор 8. Описанная последовательность действий повторяется циклически до обнуления всех старших разрядных счетчиков.

Рассмотрим пример преобразования числа 7926_{10} из двоично-десятичной системы в двоичную в трехшаговом ПК последовательного типа с набором шагов 1,2,4 (табл. 1.).

Таблица 1

Используемый шаг	Номер такта	Состояние разрядных счетчиков		Коды состояний триггеров			Величина эквивалента	Значение сумматора	
		Начало такта	Конец такта	Третий регистр	Второй регистр	Первый регистр		Начало такта	Конец такта
		N	$x_4x_3x_2x_1$	$x_4x_3x_2x_1$	$E_3E_2E_1$	$D_3D_2D_1$		$C_3C_2C_1$	ФЭ
	0	7926	7926	000	000	000	0000	0000	0000
4	1	7926	3526	110	111	111	4400	0000	4400
4	2	3526	3126	010	111	111	400	4400	4800
2	3	3126	1106	000	101	111	2020	4800	6820
1	4	1106	0006	000	000	110	1100	6820	7920
	5	0006	0006	000	000	000	6	7920	7926

3. Проектирование формирователей эквивалентов для трехшаговых преобразователей последовательного типа

Методику проектирования ФЭ рассмотрим на примере трехразрядного трехшагового ПК (табл. 2).

Таблица 2

Номер строки	Коды состояний триггеров групп			Общий вид эквивалента	Десятичный эквивалент	Двоичный код эквивалента		
	Третьей	Второй	Первой			$Y_{10}Y_9$	$Y_8Y_7Y_6Y_5$	$Y_4Y_3Y_2Y_1$
Z_i	E_2E_1	D_2D_1	C_2C_1	S	S_{10}	$Y_{10}Y_9$	$Y_8Y_7Y_6Y_5$	$Y_4Y_3Y_2Y_1$
0	00	00	00	x_0	x_0	Трансляция тетрады x_0		
1	00	00	01	K	12	00	0000	1100
2	00	00	10	K^2	144	00	1001	0000
3	00	00	11	$K^2 + K$	156	00	1001	1100
4	00	01	X1	aK	24	00	0001	1000
5	00	10	1X	aK^2	288	01	0010	0000
6	00	11	11	$aK^2 + aK$	312	01	0011	1000
7	01	X1	X1	bK	48	00	0011	0000
8	10	1X	1X	bK^2	576	10	0100	0000
9	11	11	11	$bK^2 + bK$	624	10	0111	0000

Шифраторы 15 формируют выходное слово, меньшее на 4 единицы по отношению к входному слову при $x_i \geq 4$ (если $a=4$), меньшее на 2 единицы по отношению к входному слову при $2 \leq x_i < 4$ (если $a=2$) и равное входному слову при $x_i < 2$.

ФЭ 20 при $n=3$ и $K=12$ реализует функцию:

$$S = \begin{cases} 12C_1 + 144C_2; E_1 = E_2 = D_1 = D_2 = 0; \\ 24D_1 + 288D_2; E_1 = E_2 = 0; D_1, D_2 \neq 0; \\ 48E_1 + 576E_2; E_1, E_2 \neq 0. \end{cases} \quad (4)$$

Преобразование двоичных кодов триггеров состояний первой 3, второй 4 и третьей 5 групп соответствует табл. 2.

ФЭ для ПК последовательного типа реализуем на основе трех последовательно включенных ДШ 21-23, выходы группы из четырех элементов И 26 поступают на соответствующие входы четырех элементов ИЛИ 27, вторые входы которых соединены с выходами четырех младших разрядов шифратора 24 (рис. 1).

Функции выходов шифратора 24 в целях сокращения аппаратных затрат реализуют с учетом общих частей по следующим выражениям:

$$y_1 = y_2 = 0; y_7 = y_{10} = E_2; y_9 = \overline{Z_5} \overline{Z_6} = M_1, M_2 = \overline{Z_4} \overline{Z_6}; M_3 = \overline{Z_7} \overline{Z_9}; M_4 = y_8 = \overline{Z_2} \overline{Z_3}, \\ M_5 = y_3 = \overline{Z_1} \overline{Z_3}; y_4 = M_2 \cup M_5; y_5 = M_2 \cup M_3 \cup M_4; y_6 = M_1 \cup M_3. \quad (5)$$

В целях минимизации числа корпусов ЛЭ для реализации разрядных функций ФЭ можно использовать ПЛМ типа К556РТ1, с параметрами $s, t, q (16, 8, 48)$.

Максимальное число шагов преобразования в трехшаговом ПК последовательного типа определим путем подсчета числа тактов преобразования на всех четырех этапах. Первый этап – это использование шага a, b , второй – шага a , третий – шага 1 и четвертый этап – трансляция младшей цифры x_0 . При наличии в одном из старших разрядов $x_m (m = \overline{2, n})$ максимального значения цифры $x_m = K - 1$ потребуется на первом этапе $\lceil (K-1)/b \rceil$ шагов преобразования, где $\lceil \rceil$ означает округление до меньшего целого. Затем, если хотя бы одна из оставшихся старших цифр $a \leq x_m < b$, потребуется еще $\lceil (b-1)/a \rceil$ тактов преобразования на втором этапе. Если после второго этапа останется цифра $x_m = a - 1$, то на третьем этапе потребуется $a - 1$ такт для обнуления разрядного счетчика, хранящего эту цифру.

И на четвертом этапе будет затрачен еще один такт на трансляцию младшей цифры.

Следовательно, $N_3 = \lceil (K-1)/b \rceil + \lceil (b-1)/a \rceil + a - 1 + 1 = \lceil (K-1)/b \rceil + \lceil (b-1)/a \rceil + a$.

4. Повышение быстродействия ПК параллельного типа

Структура трехшагового ПК параллельного типа содержит вентили опроса, каждый из которых управляется через инвертор от триггера соответствующего разряда старшего регистра состояний (рис. 2).

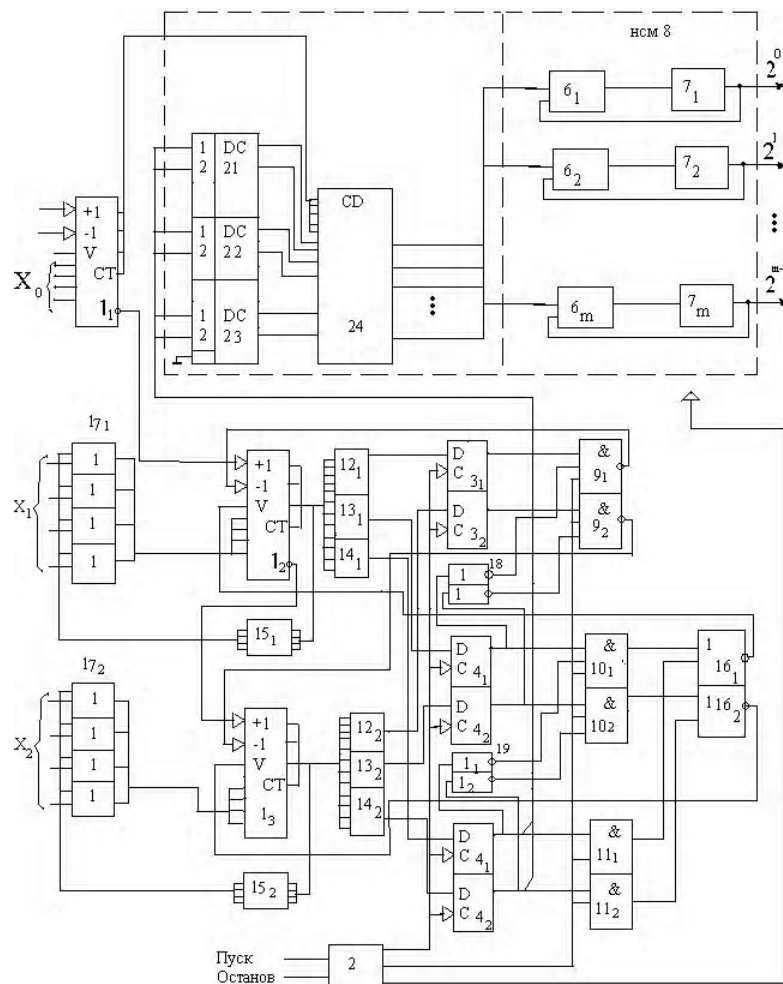


Рис. 2. Структура трехшагового ПК с параллельным использованием шагов преобразования

Для трехшагового ПК параллельного типа вначале составляют таблицу закона функционирования ФЭ (табл.3).

Таблица 3

Номер строки	Коды состояний триггеров групп			Общий вид эквивалента	Десятичный эквивалент	Двоичный код эквивалента		
	Третьей	Второй	Первой			$Y_{10}Y_9$	$Y_8Y_7Y_6Y_5$	$Y_4Y_3Y_2Y_1$
Z_i	E_2E_1	D_2D_1	C_2C_1	S	S_{10}	$Y_{10}Y_9$	$Y_8Y_7Y_6Y_5$	$Y_4Y_3Y_2Y_1$
0	00	00	00	x_0	x_0	Трансляция тетрады x_0		
1	00	00	01	K	12	00	0000	1100
2	00	00	10	K^2	144	00	1001	0000
3	00	00	11	$K^2 + K$	156	00	1001	1100
4	00	01	01	aK	24	00	0001	1000
5	00	10	11	$K^2 + aK$	168	00	1010	1000
6	00	10	10	aK^2	288	01	0010	0000
7	00	10	11	$aK^2 + K$	300	01	0010	1100
8	00	11	11	$aK^2 + aK$	312	01	0011	1000
9	01	01	01	bK	48	00	0011	0000
10	01	01	11	$K^2 + bK$	192	00	1100	0000
11	01	11	11	$aK^2 + bK$	336	01	0101	0000
12	10	10	10	bK^2	576	10	0100	0000
13	10	10	11	$bK^2 + K$	588	10	0100	1100
14	10	11	11	$bK^2 + aK$	600	10	0101	1000
15	11	11	11	$bK^2 + bK$	624	10	0111	0000

Затем находят запись разрядных функций шифратора:

$$\begin{aligned}
 y_1 &= y_2 = 0; y_9 = \overline{Z_6} \overline{Z_7} \overline{Z_8} \overline{Z_{11}}; y_{10} = E_2; y_8 = \overline{Z_5} \overline{Z_{10}}; \\
 y_7 &= \overline{Z_{10}} \overline{Z_{11}} \overline{Z_{12}} \overline{Z_{13}} \overline{Z_{14}} \overline{Z_{15}}; y_6 = \overline{Z_5} \overline{Z_6} \overline{Z_7} \overline{Z_8} \overline{Z_9} \overline{Z_{15}}; \\
 y_5 &= \overline{Z_2} \overline{Z_3} \overline{Z_4} \overline{Z_8} \overline{Z_9} \overline{Z_{11}} \overline{Z_{14}} \overline{Z_{15}}; \\
 y_4 &= \overline{Z_1} \overline{Z_3} \overline{Z_4} \overline{Z_5} \overline{Z_7} \overline{Z_8} \overline{Z_{13}} \overline{Z_{14}}; y_3 = \overline{Z_1} \overline{Z_3} \overline{Z_7} \overline{Z_{13}};
 \end{aligned} \tag{6}$$

ФЭ для трехшагового ПК с параллельным использованием шагов реализует функции:

$$\begin{aligned}
 Y_4 &= \overline{z_1} / \overline{z_3} / \overline{z_4} / \overline{z_5} / \overline{z_7} / \overline{z_8} / \overline{z_{13}} / \overline{z_{14}}; Y_{10} = \overline{z_{12}} / \overline{z_{13}} / \overline{z_{14}} / \overline{z_{15}}; \\
 Y_5 &= \overline{z_2} / \overline{z_3} / \overline{z_4} / \overline{z_8} / \overline{z_9} / \overline{z_{11}} / \overline{z_{14}} / \overline{z_{15}}; Y_6 = \overline{z_5} / \overline{z_6} / \overline{z_7} / \overline{z_8} / \overline{z_9} / \overline{z_{15}}; Y_1 = 0; Y_2 = 0; \\
 Y_3 &= \overline{z_7} / \overline{z_{13}}; \\
 Y_7 &= \overline{z_{10}} / \overline{z_{11}} / \overline{z_{12}} / \overline{z_{13}} / \overline{z_{14}} / \overline{z_{15}}; Y_8 = \overline{z_2} / \overline{z_3} / \overline{z_5} / \overline{z_{10}}; Y_9 = \overline{z_6} / \overline{z_7} / \overline{z_8} / \overline{z_{11}}.
 \end{aligned} \tag{7}$$

Сравнение табл. 3 и табл. 2 показывает, что количество строк в табл.3 равно 16, а в табл.2 — 10. Особенностью ФЭ для ПК параллельного типа является отказ от принципа последовательного соединения стролируемых дешифраторов (в схеме без параллельного использования шагов) и применение ряда вентилях (схем И), позволяющих выделить определенное подмножество входных наборов (рис.3). Как следует из рис. 3, в схеме ФЭ используется один ДШ 2-4 и четыре ДШ 1-2, девять схем И-НЕ и ИЛИ-НЕ. Таким образом, количество аппаратных затрат при параллельном использовании шагов преобразования небольшое (15-20% при увеличении быстродействия на 66,7%)

5. Метод расчета числа тактов преобразования для различных стратегий

Для расчета числа тактов преобразований ПК, как с последовательной, так и с параллельной стратегиями использования шагов преобразования можно применить метод моделирования преобразования с помощью таблиц (табл.4; табл.5), характеризующих состояние разрядных счетчиков в соответствующий такт.

Таблица 4

Такт	Набор шагов 1,7,8
0	98765 43210
1	10054 32100
2	00043 21000
3	00032 10000
4	00001 00000
5	00010 00000
6	00000 00000

Таблица 5

Такт	Набор шагов 1,2,6
0	98765 43210
1	32103 21000
2	10001 00000
3	00000 00000

В табл.4 и 5 приведены примеры преобразований для параллельной стратегии использования шагов преобразования. Из табл.5 следует, что набор шагов 1,2,6 позволяет в два раза увеличить быстродействие (три такта преобразования вместо шести для набора шагов 1,7,8). Для моделирования всех возможных случаев преобразования исходное число разрядов принимаем равным K (в каждом из разрядов значение цифры равно $K-1, K-2, \dots, 1, 0$). Для определения максимального числа тактов преобразования трехшагового ПК с последовательным использованием шагов преобразования следует применять формулы (1) или разработанное авторами программное средство МАТРИЦА.

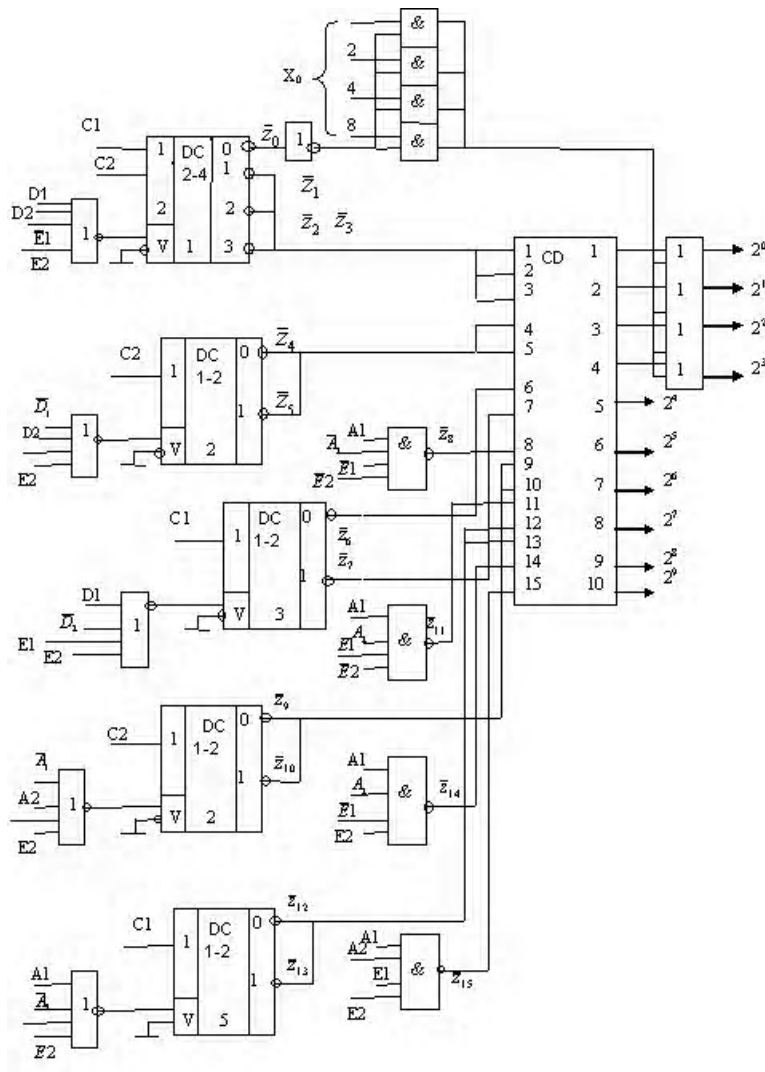


Рис. 3. Структура формирователя эквивалента ПК параллельного типа

Для нахождения оптимального значения шагов b и a , при котором число тактов преобразования N'_3 , трехшагового ПК минимально используем программу МАТРИЦА (для ПК с последовательным использованием шагов). Задав исходное значение основание K , получим матрицу всевозможных значений N'_3 . Для $K=10$ получим матрицу:

a	2	3	4	5	6	7	8
b	5						
3	4	5					
4	4	4	5				
5	4	4	5	6			
6	5	5	5	6	7		
7	5	5	5	6	7	8	
8	6	5	6	6	7	8	9
9							

Программа МАТРИЦА рассчитывает максимальное число тактов преобразования, но для конкретно заданного исходного числа конкретный результат можно получить только путем моделирования методом таблиц. Сравнительные результаты числа тактов преобразования последовательной и параллельной стратегии приведены в табл.6.

Таблица 6

i	1,a,b	N_3	N'_3	i	1,a,b	N_3	N'_3
1	1,2,3	3	5	15	1,4,6	4	5
2	1,2,4	4	5	16	1,4,7	3	5
3	1,2,5	3	4	17	1,4,8	4	5
4	1,2,6	3	4	18	1,4,9	4	6
5	1,2,7	3	4	19	1,5,6	4	6
6	1,2,8	4	5	20	1,5,7	4	6
7	1,2,9	4	6	21	1,5,8	4	6
8	1,3,4	3	4	22	1,5,9	4	6
9	1,3,5	3	4	23	1,6,7	5	7
10	1,3,6	3	4	24	1,6,8	5	7
11	1,3,7	3	5	25	1,6,9	5	7
12	1,3,8	3	5	26	1,7,8	6	8
13	1,3,9	4	5	27	1,7,9	6	8
14	1,4,5	4	5	28	1,8,9	7	9

6. Математические модели, описывающие ФЭ в ПК параллельного типа

Функционирование трехшагового многоблочного формирователя эквивалентов (ФЭ) с параллельной стратегией использования шагов преобразования описывается выражением:

$$S_m = \sum_{i=(m-1)P+1}^{i=mP} \gamma_i(h) K^{i-1} \cdot R_i(h), (i = 1, \overline{mP}), \quad (9)$$

где m – номер блока ($m = \overline{1, M}$); i – номер разряда в блоке m ; P – число разрядов в блоке

($P = \frac{n}{M}$); n – число входных преобразуемых разрядов; h – текущий шаг преобразований; M

– число блоков в разбиении ($M = \frac{n}{p}$); C_i – состояние i -го триггера регистра состояний.

Коэффициент $\gamma_i(h)$ при степени оснований K^{i-1} может принимать три значения в зависимости от значений D_i и C_i триггеров i -го разряда соответственно старшего и младшего регистров состояний на этапе (такте) преобразования h :

$$\gamma_i(h) = \begin{cases} b, E_i \neq 0; D_i \neq 0; C_i \neq 0; \\ a, D_i \neq 0; C_i \neq 0; \\ 1, D_i = 0; C_i \neq 0; \\ 0, D_i = 0; C_i = 0. \end{cases} \quad (10)$$

Коэффициент $R_i(h)$ учитывает вхождение компоненты i -го разряда (ее вклад) в выражение для S_m блока m и равен:

$$R_i(h) = \begin{cases} 0, E_i = D_i = C_i = 0; \\ 1, E_i \vee D_i \vee C_i = 1. \end{cases} \quad (11)$$

В зависимости от номера текущего такта преобразования $h(h = \overline{0, N})$ значения коэффициентов $\gamma_i(h)$ и $R_i(h)$ могут изменяться, т.е. являются динамически изменяющимися, что относится также и к величине S_m .

Процесс преобразования заканчивается, если на каком-либо из тактов преобразования h состояния всех триггеров D_i и C_i всех блоков оказываются равными нулю. В случае разбиения шестиразрядного ПК и ФЭ на 6 блоков имеем следующие выражения для $S_1 - S_6$:

$$\begin{aligned} S_1 &= \gamma_1(h)K^0R_1(h); S_2 = \gamma_2(h)K^1R_2(h); S_3 = \gamma_3(h)K^2R_3(h); \\ S_4 &= \gamma_4(h)K^3R_4(h); S_5 = \gamma_5(h)K^4R_5(h); S_6 = \gamma_6(h)K^5R_6(h). \end{aligned} \quad (12)$$

Число блоков разбиения (декомпозиции) M ФЭ и ПК на блоке является делителем числа входных разрядов n . В этом случае все блоки содержат одинаковое число разрядов. Так, при $n=6$ делителями являются числа 6,3,2,1 ($M=1,2,3,6$).

При $M=3$ получим следующие выражения для S_m :

$$\begin{aligned} S_1 &= \sum_{i=1}^{i=2} \gamma_i(h)K^{i-1}R_i(h) = \gamma_2(h)K^1R_2(h) + \gamma_1(h)K^0R_1(h), \\ S_2 &= \sum_{i=3}^{i=4} \gamma_i(h)K^{i-1}R_i(h) = \gamma_4(h)K^3R_4(h) + \gamma_3(h)K^2R_3(h), \\ S_3 &= \sum_{i=5}^{i=6} \gamma_i(h)K^{i-1}R_i(h) = \gamma_6(h)K^5R_6(h) + \gamma_5(h)K^4R_5(h). \end{aligned} \quad (13)$$

При $M=2$ выражения для вычисления значений S_m ФЭ преобразуются к виду:

$$\begin{aligned} S_1 &= \sum_{i=1}^{i=3} \gamma_i(h)K^{i-1}R_i(h) = \gamma_3(h)K^2R_3(h) + \gamma_2(h)K^1R_2(h) + \gamma_1(h)K^0R_1(h), \\ S_2 &= \sum_{i=6}^{i=6} \gamma_i(h)K^{i-1}R_i(h) = \gamma_6(h)K^5R_6(h) + \gamma_4(h)K^3R_4(h) + \gamma_3(h)K^2R_3(h). \end{aligned} \quad (14)$$

Наконец, при $M=1$ получим выражение:

$$\begin{aligned} S_1 &= \sum_{i=1}^{i=6} \gamma_i(h)K^{i-1}R_i(h) = \gamma_6(h)K^5R_6(h) + \gamma_4(h)K^3R_4(h) + \gamma_3(h)K^2R_3(h) + \\ &+ \gamma_2(h)K^1R_2(h) + \gamma_1(h)K^0R_1(h). \end{aligned} \quad (15)$$

Следует отметить, что если M является делителем числа n , то все блоки разбиения будут идентичными и иметь одинаковое число разрядов в блоке. Если это условие не выполняется, то последний блок разбиения будет иметь меньшее число разрядов по сравнению с предыдущими.

Так, при $n=8$ и $M=3$ будет два блока по три разряда (S_1 и S_2) и один блок (третий) с двумя разрядами. Функционирование ФЭ этих блоков описывается следующими формулами:

$$S_1 = \sum_{i=1}^{i=3} \gamma_i(h) K^{i-1} R_i(h) = \gamma_3(h) K^2 R_3(h) + \gamma_2(h) K^1 R_2(h) + \gamma_1(h) K^0 R_1(h),$$

$$S_2 = \sum_{i=4}^{i=6} \gamma_i(h) K^{i-1} R_i(h) = \gamma_6(h) K^5 R_6(h) + \gamma_5(h) K^4 R_5(h) + \gamma_4(h) K^3 R_4(h), \quad (16)$$

$$S_3 = \sum_{i=7}^{i=8} \gamma_i(h) K^{i-1} R_i(h) = \gamma_8(h) K^7 R_8(h) + \gamma_7(h) K^6 R_7(h).$$

Выводы

Научная новизна выполненного исследования заключается в следующем:

1. Предложен способ повышения быстродействия трехшаговых преобразователей кодов на базе счетчиков, функционирующих по методу накопления эквивалентов и использующих параллельную стратегию шагов преобразования.

2. Проведена сравнительная оценка последовательной и параллельной стратегии использования шагов преобразования и разработана методика расчета максимального числа тактов преобразования для обеих стратегий.

Практическая значимость исследования состоит в возможности построения более быстродействующих ПК в методике расчета быстродействия и программном обеспечении для его реализации.

Список литературы: 1. А.С. 1126946 5G06F 5/02. Преобразователь двоично-К-ичного кода в двоичный код / А.Н. Слобожанин // Открытия, изобретения. 1984. №44. С.250. 2. А.С. 1647908 5H03M 7/12. Преобразователь двоично-К-ичного кода в двоичный код / Н.Я. Какурин, Ю.К. Кирьяков, А.Н. Макаренко // Там же. 1991. №17. С. 262-263. 3. А.С. 1783618 5G06F 5/02. Преобразователь двоично-К-ичного кода в двоичный код / Н.Я. Какурин, А.Н. Макаренко, Д.Ю. Исхаков, В.А. Толмацкий // Открытия, изобретения. 1984. №44. С.250. 4. Патент України 58165 H03M 7/12 Перетворювач коду Фібоначчі в двійковий код / М.Я. Какурін, І.В. Купрейчик, Г.М. Макаренко // Промислова власність. 2003. №7. С.4. 149 5. Какурин Н.Я., Макаренко А.Н., Старчевский Д.Л. Проектирование алгоритмов функционирования преобразователей двоично-десятичных кодов последовательного типа. Часть 1. Проектирование алгоритмов преобразования // АСУ и приборы автоматики. 2004. Вып. 128. С.76-87.

Поступила в редколлегию 14.12.2008

Какурин Николай Яковлевич, канд. техн. наук, профессор кафедры автоматизации проектирования вычислительной техники ХНУРЭ. Научные интересы: прикладная теория цифровых автоматов, автоматизация проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

Коваленко Сергей Николаевич, соискатель кафедры АПВТ ХНУРЭ. Научные интересы: цифровые датчики, устройства преобразования кодов, автоматизация проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

Лопухин Юрий Владимирович, ст. преподаватель кафедры АПВТ ХНУРЭ. Научные интересы: проектирование программного обеспечения, автоматизации проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

Макаренко Анна Николаевна, канд. техн. наук, доцент кафедры высшей математики и информационных технологий Харьковского банковского института. Научные интересы: дискретная математика, анализ и синтез преобразователей код-код. Адрес: Украина, 61074, Харьков, пр.Победы, 65, тел. 33-80-564.

ПЕРСПЕКТИВИ ЗАСТОСУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ТА ІДЕНТИФІКАЦІЇ КОРИСТУВАЧІВ КОМП'ЮТЕРА, ОПИСАНОЇ НА ОСНОВІ ДИСКРЕТНОЇ МОДЕЛІ

На основі дискретної моделі описується підхід до побудови автоматизованої системи розпізнавання користувачів комп'ютера у випадку, коли дискретна модель рекурентно зв'язує часові затримки при введенні інформації з клавіатури комп'ютера у реальному діапазоні часу. Доцільність розробленого підходу ілюструється при реалізації автоматизованої процедури ідентифікації користувачів комп'ютера на основі детермінованого та імовірнісного методів та відзначаються перспективні напрями застосування створеної системи.

1. Постановка задачі

При створенні нових реальних пристроїв, дослідженні невивчених фізичних явищ чи процесів, побудові систем розпізнавання та ідентифікації, що мають бажані характеристики інформаційного сигналу або невідомі характеристики, які підлягають вивченню, доцільно провести комп'ютерне моделювання та аналіз, створивши адекватні математичної моделі об'єкта, що розробляється чи вивчається. Такий підхід вимагає значно менших часових і технічних засобів порівняно з фізичним експериментом, особливо на попередній стадії розробки, за відсутності достовірної апріорної інформації.

Останнім часом в нелінійній динаміці широко застосовуються дискретні моделі систем [5-9], для яких дискретність закладена в природі самого об'єкта досліджень, а не є наслідком дискретизації неперервної системи. Доцільність використання дискретних по своїй природі моделей пояснюється такими їх особливостями:

- простотою математичного опису в порівнянні з неперервними моделями;
- наявністю суттєво ширшого спектру динамічних режимів, порівняно з відомими моделями;
- нескінченною вимірністю, що дозволяє моделювати кожен нову гармоніку процесу шляхом її введення у вектор змінних стану, тоді як для неперервних систем для вирішення цієї задачі необхідно підвищувати розмірність системи;
- відсутністю необхідності визначення кроку дискретизації, оцінки локальної і глобальної похибок чисельних методів, областей стійкості та синхронізації;
- кращою адаптованістю до постановки комп'ютерного експерименту, порівняно з неперервними моделями.

Власне моделі, дискретні за своєю природою, застосовні як до побудови пристроїв, що мають бажані режими, так і до розпізнавання та ідентифікації таких режимів у системах зі складною динамікою і поведінкою, що дозволяє підвищити ефективність їх роботи.

При такій постановці задачі актуальною є проблема розроблення надійних підходів до встановлення пріоритету первинних ознак, що формуються в процесі розпізнавання в реальному режимі часу.

Метою даної статті є опис автоматизованої комп'ютерної системи розпізнавання та достовірної ідентифікації об'єктів і явищ зі складною динамічною природою для забезпечення достовірного якісного та автоматизованого процесу розпізнавання та ідентифікації досліджуваних систем. В роботі також визначено перспективні напрями розвитку систем, розпізнавання складних динамічних систем на основі дискретних моделей та напрями їх доцільного застосування.

2. Виклад основного матеріалу

2.1. Аналіз основних результатів. При розробленні систем розпізнавання об'єктів і явищ та їх достовірної ідентифікації необхідний системний підхід, суть якого полягає у

формуванні первинних ознак про об'єкт розпізнавання, встановленні їх пріоритету та у виборі або розробленні та реалізації надійних критеріїв розпізнавання і достовірної ідентифікації об'єктів та процесів.

Перші дослідження у галузі розпізнавання в нашій країні проводилися А.А. Харкевичем [14] – одним з основоположників та фундаторів теорії інформації та сигналів. Значний внесок у розвиток теорії розпізнавання зробили В.М. Глушков, В.С. Міхалевич, О.Г. Івахненко, Ю.І. Журавльов, Я.З. Ципкін, В.І. Васильєв. Серед зарубіжних вчених слід згадати роботу Ф. Розенблатта, який у 1957 р. запропонував машину, що навчалася розпізнавати образи і називалася перцептроном (в перекладі з англійської мови “to percept” – сприймати). Це була найпростіша модель діяльності людського мозку. Значний вклад у подальший розвиток теорії розпізнавання образів зробили У. Гарднер, Р. Дуда, Г. Себастьян, Дж. Ту, К. Фу, П. Харт, С. Ватанабе та інші.

Перші роботи з розпізнавання образів було присвячено теорії і практиці побудови читальних автоматів (під образом розумівся знак, зображення, буква або цифра). Математичним апаратом для розв'язання задач розпізнавання з моменту їх виникнення була теорія статистичних розв'язків [15].

На сьогоднішній день результати теорії статистичних розв'язків стали базою для побудови алгоритмів розпізнавання, які забезпечували віднесення об'єкта до його класу на підставі експериментальних апостеріорних даних, що характеризують об'єкт, та апіорних даних, що описують класи об'єктів. Пізніше математичний апарат розширився завдяки використанню методів алгебри логіки і деяких розділів прикладної математики, теорії інформації, математичного програмування, системотехніки і системного аналізу [16,17].

Незважаючи на те, що методи і алгоритми розпізнавання все більшою мірою стають невід'ємною складовою таких прикладних галузей природознавства, як медична і технічна діагностика, ідентифікація складних коливних динамічних процесів і явищ, екологічний моніторинг та соціальна інформатика, метеорологічне прогнозування і геологічна розвідка, локаційні засоби спостереження та системи введення і виведення текстової, графічної та мовної інформації в комп'ютер [9], інтелектуальні системи прийняття рішень, в літературі – як вітчизняній, так і в зарубіжній – системний підхід до задач розпізнавання поки що не став домінуючим.

Сьогодні, як і півстоліття тому, проблема розпізнавання значною мірою ототожнюється з побудовою оптимальних алгоритмів розпізнавання та дослідженням умов, які дозволяють реалізувати такий алгоритм. Теоретичні дослідження орієнтуються на розв'язання хоча й важливих, але часткових задач здебільш прикладного характеру. До таких задач у першу чергу треба віднести задачі достовірного розпізнавання, суть яких зводиться до поділу простору ознак, мовою яких описуються об'єкти чи процеси розпізнавання, на області, що відповідають класам цих об'єктів, тобто до вибору найкращих границь (правил) розділення класів. Але розв'язання цих задач можливе лише тоді, коли апіорі відомі класи об'єктів і ознаки, мовою яких описуються розпізнавані об'єкти та їх класи. Однак розробник системи розпізнавання, як правило, не володіє цією інформацією. Навіть у найпростіших випадках розпізнавання букв алфавіту, відбитків пальців, слів мови, екстремумів та особливих точок функцій (де не виникає питання про класи) їх інформативні ознаки та апаратура для їх визначення невідомі – це є предметом нетрадиційних досліджень.

Виникає питання про причини такої уваги до задачі опису класів мовою ознак і побудови оптимальних алгоритмів розпізнавання.

Перша причина полягає в тому, що ці задачі порівняно легко піддаються формальному і аналітичному розв'язанню, що й визначає їх привабливість для дослідників. Друга причина полягає у тому, що значна частина дослідників обмежує свою діяльність лише теоретичними дослідженнями. Третя проблема в тому, що традиційно вважається, що системи розпізнавання є автономними. У деяких часткових задачах це виправдано, хоча в загальному випадку таке формулювання питання не є правомірним. Адже і в системах технічної та медичної діагностики, в автоматизованих системах управління виробництвом, розпізнавання дефектів механізмів і машин, визначення діагнозу пацієнта, розпізнавання складних коливних режимів, класифікація виробничих ситуацій не є самоціллю. Їх розпізнавання необхідне для отримання вихідної інформації для підсистеми управління з метою прийняття

керівних рішень, адекватних результатам розпізнавання невідомих об'єктів, явищ, ситуацій, станів.

Можна стверджувати, що достовірне розпізнавання конкретних ситуацій не є достатньою умовою досягнення потенціально можливої ефективності системи управління. Але це необхідна умова. Важко уявити, що лікар, який поставив неправильний діагноз, знайде правильний метод лікування чи не виявлення нестійких коливних режимів забезпечить надійну роботу технічного пристрою.

При розробці будь-яких систем розпізнавання необхідний системний підхід, суть якого полягає в тому, щоб в умовах неминучих фінансових і технічних обмежень система розпізнавання забезпечила системі управління реалізацію потенціально можливої ефективності. Вибору чи створенню критеріїв розпізнавання повинна передувати процедура вимірювання первинних ознак про процес розпізнавання, встановлення пріоритету цих ознак та їх впливу на інтегральні характеристики досліджуваного процесу чи об'єкта. З математичної точки зору опис такої системи має забезпечувати мінімальну похибку розпізнавання та достовірну ідентифікацію об'єкта розпізнавання за певними ознаками та критеріями прийняття рішення

2.2.Метод ідентифікації користувача шляхом виділення первинних дискретних інформативних ознак. Суть методу полягає у тому, щоб забезпечити процедуру розпізнавання конкретного користувача при його роботі за клавіатурою комп'ютера. Деякі загальні міркування щодо створення такої системи подані в роботі [17].

Очевидно, для організації процесу розпізнавання у пам'ять комп'ютера необхідно ввести текст (зразок) кожного із об'єктів розпізнавання. При відсутності зразка об'єкт не розпізнається або пропонується створити новий клас об'єктів шляхом завдання зразка почерку (це можна використати для забезпечення санкціонованого доступу до ресурсів комп'ютера). Паралельно при створенні зразка за рукомоторними ознаками об'єкта формується його інформаційна модель шляхом визначення функцій розподілу часових затримок при введенні інформації в комп'ютер. Як первинні ознаки про об'єкт використано різні часові затримки при роботі об'єкта з клавіатурою комп'ютера. Встановити пріоритет кожної із первинних ознак можна експериментальним шляхом, що запропоновано в роботі [19]. При ідентифікації об'єкта знову реалізуємо процедуру вибору або розроблення критеріїв прийняття рішення і на основі цих критеріїв [15] і приймаємо рішення про віднесення об'єкта до певного класу. У випадку неоднозначного рішення можна застосувати функції відстані (детермінований підхід) і однозначно обрати клас (з найменшим середньоквадратичним відхиленням ознак). Відзначимо, що різні інформаційні ознаки можуть мати різний пріоритет, який також можна встановити експериментально. З метою підвищення ефективності системи доцільно відсікати недетерміновані хаотичні рухи руки особи шляхом попередньої фільтрації інформації, що вводиться користувачем в режимі реального часу, створюючи тим самим неперервні послідовності (набори) символів.

У роботі [11-13] сформульовано і проаналізовано велику кількість характеристик. Наведено лише найбільш інформативні та доступні для швидкого формування. Отже, для побудови системи розпізнавання особи за її рукомоторними реакціями було обрано такі характеристики:

1) відносна девіація паузи перед клавішею – розподіл відносних відхилень паузи перед даним клавішем до середнього значення паузи перед всіма клавішами у даній неперервній послідовності набору:

$$\text{DevB} = \frac{t_i - t_{cp}}{t_{cp}} \cdot 100\% , \quad (1)$$

де t_i – тривалість паузи перед i -м клавішем; t_{cp} – середня тривалість паузи перед клавішами в послідовності набраного тексту;

2) відносна девіація утримання клавіша – розподіл відносних відхилень тривалості утримання натиснутим даного клавіша до середньої тривалості його утримання у даній неперервній послідовності:

$$\text{DevP} = \frac{t_i - t_{cp}}{t_{cp}} \cdot 100\% \quad (2)$$

Приклад даного розподілу зображено на рисунку. На осі абсцис відкладено відносні відхилення у відсотках, а на осі ординат – відносну частоту попадань у відповідний інтервал відхилень;

3) відносна девіація паузи після клавіша – аналогічна попередній характеристиці:

$$\text{DevA} = \frac{t_i - t_{cp}}{t_{cp}} \cdot 100\% \quad (3)$$

4) відношення величини паузи перед клавішем до тривалості його утримання;

5) відношення величини паузи перед клавішем до величини паузи після нього;

6) відношення величини паузи після клавіша до тривалості його утримання;

7) розподіл частот використання клавіш зміни регістру.

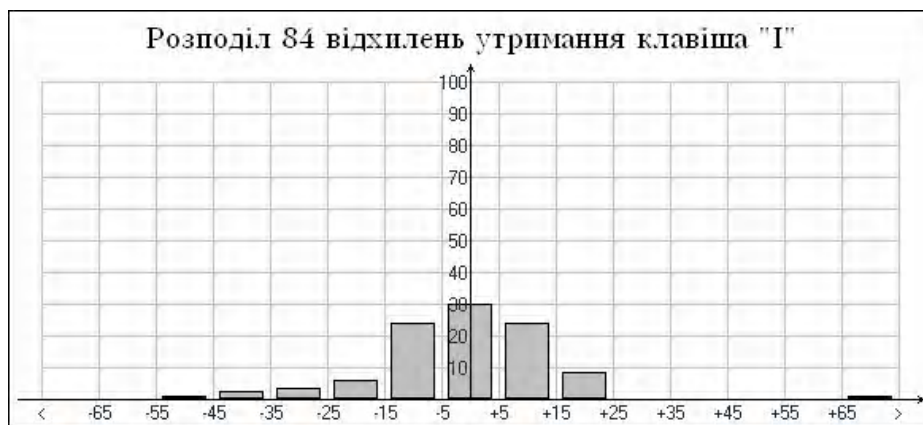
Всього в роботі [11] розглянуто 18 характеристик, але найбільш інформативними є наведені вище.

Характеристики 1-6 формуються для кожного клавіша, що був задіяний у наборі. Щоби спростити балансування важливості характеристик, при побудові системи прийнято рішення об'єднати перші шість характеристик у групи, оскільки це значно зменшує їх кількість (а в межах групи можна розгадати їх як еквівалентні). На спосіб групування характеристик безпосередньо впливає обраний метод їх зіставлення.

У першому варіанті побудови системи розпізнавання для реалізації процедури ідентифікації було використано функції відстані. Оскільки вага характеристик кожної групи могла бути різною, то відстані обчислювались окремо по кожній з груп характеристик. Відстань між класами Ω і Z в межах кожної групи характеристик обчислюється за формулою середнього квадратичного відхилення:

$$\text{Dist}(\lambda) = \sqrt{\frac{1}{n} \sum_{i=1}^n (m_i^{\Omega} - m_i^Z)^2}, \quad (4)$$

де m_i – середнє значення вибірки i -ї характеристики даної групи класу Ω ; $\text{Dist}(\lambda)$ – відстань між класами за групою характеристик λ .



Розподіл відносних девіацій утримання заданого клавіша натиснутим

Відстані вимірюються між середніми значеннями, оскільки середнє може бути оцінено вже після відносно невеликої кількості дослідів (10-20), що є важливим для зменшення об'єму тексту, який набирається об'єктами розпізнавання.

Групи характеристик 1-6 не еквівалентні за якістю рішень, що приймаються на їх основі. Перед об'єднанням результатів для прийняття рішення по розпізнаванню необхідно збалансувати ваги груп між собою. Баланс характеристик здійснений обернено пропорційно ймовірностям допустити помилку другого роду (коли два об'єкти різних класів розпізнаються як

такі, що належать до одного класу) по кожній з груп характеристик зокрема: $1/p_1: 1/p_2: 1/p_3: 1/p_4: 1/p_5: 1/p_6: 1/p_7$. Експериментально було отримано відношення ваг груп 1-7 як 4:12:8:6:5:2:6 відповідно. Недоліком системи на основі функції відстані є те, що вона принципово не може визначити ймовірність правильності або неправильності рішення по розпізнаванню. Кожний сторонній користувач буде схожий на того чи іншого зареєстрованого користувача системи.

Розроблений другий варіант побудови системи розпізнавання базується на використанні методу довірчих інтервалів. Для перевірки гіпотези про належність пари об'єктів одному класу перевіряються гіпотези про рівність середніх значень розподілів [17] всіх характеристик кожної групи. Для цього обчислюються значення середнього a та вибіркового стандарту s за формулами:

$$a = \frac{1}{n} \sum_{i=1}^n x_i, \quad (5)$$

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - a)^2}. \quad (6)$$

Для розрахунку довірчих інтервалів враховується закон розподілу середнього значення:

$$f(x_{cp}) = \frac{\sqrt{n}}{\sigma\sqrt{2\pi}} e^{-\frac{n}{2\sigma^2}(x_{cp}-x_0)}. \quad (7)$$

Для вибірок малого об'єму оцінка середнього значення уточнюється за допомогою розподілу Стьюдента [18], за яким розподілена величина $u = \frac{a-m}{s_{cp}}$. Його густина розподілу задається формулою:

$$S(u, n) = \frac{\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n}{2})\sqrt{\pi n}} \left(1 + \frac{u^2}{n}\right)^{-\frac{n+1}{2}}. \quad (8)$$

Нехай при порівнянні пари відповідних розподілів ми допускаємо помилку першого роду P_α , а всього порівнюємо N таких пар. Отже, логічно припустити, що інтегральна характеристика групи класу і об'єкта збігається з ймовірністю $\geq 1 - P_\alpha$, якщо кількість непідтверджених гіпотез N_α не перевищує числа $P_\alpha N$, у іншому випадку вважаємо, що об'єкт не належить класу. Такого типу (так/ні) результат ми отримаємо для кожної з шести груп характеристик. Як і у випадку системи на основі функції відстані, ці групи не еквівалентні по якості рішень, що приймаються на їх основі. Кожна з них має свою ймовірність помилки другого роду.

На основі практичних експериментів по розпізнаванню з кожною групою характеристик досліджувалися помилки другого роду. Окремі гіпотези система перевіряла з рівнем значущості $\alpha = 0,05$. Так були отримані ймовірності помилок другого роду 35, 13, 20, 27, 32, 78%. Отримані ймовірності помилок були отримані для порівняно невеликої кількості експериментів по ідентифікації (105 експериментів). Для великої групи людей ймовірності помилок можуть дещо відрізнятись від наведених.

При тестуванні розробленої системи на досліджуваних об'єктах була допущена лише одна помилка на 22 проведених розпізнаваннях (запропоновано два схожих на об'єкт класи, один серед яких був правильний).

Наближена оцінка помилки прийняти об'єкт одного класу за об'єкт іншого не перевищує 35% при наявності 112-ти зареєстрованих у системі класів.

Система розпізнає зареєстрованого користувача після набору ним 5-8 речень по 60 знаків кожне, тобто після введення 300-500 знаків. При достатній кваліфікації користувача

(швидкість набору тексту 200 знаків за хвилину) система розпізнає користувача, який набирає замість завдання довільний текст. В ряді випадків зареєстровано розпізнавання особи при наборі тексту англійською мовою. Це характерно для висококваліфікованого користувача (швидкість набору тексту більше 300 символів за хвилину), коли ймовірність хаотичних рухів руки від усталеного часового режиму є малою ймовірною.

2.3. Підхід до опису системи розпізнавання користувача комп'ютера на основі дискретної моделі. Запропонований у роботі підхід для побудови дискретних моделей коливних процесів зі складною структурою у роботах [6-8] можна застосувати до опису коливної системи будь-якої природи за умови, що її стани характеризуються дискретними ознаками. Для довільного числа ознак N маємо N - вимірний вектор змінних стану, а матрицю переходу станів A будемо таким чином, щоб її визначник дорівнював одиниці. Найпростіше це можна зробити, якщо $N - 2$ рядки матриці мають одиниці на головній діагоналі, а позадіагональні елементи дорівнюють нулю. При цьому останні два рядки цієї матриці є комбінацією гармонічних функцій початкової фази φ :

$$A(\varphi) = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & \cos \varphi & \sin \varphi \\ 0 & 0 & 0 & \dots & 0 & -\sin \varphi & \cos \varphi \end{pmatrix}. \quad (9)$$

Тоді амплітуді коливань відповідатиме середньоквадратичне значення N -вимірного вектора змінних стану, яке може бути обчислене із завданням конкретного набору функцій f . Як засвідчує аналіз дискретної моделі в роботі [20], з введенням в матрицю переходу станів (9) N - мірного вектора станів період при цьому знову може бути оцінений на основі формули (10). Ефективність такого підходу до опису наведеної в попередньому розділі комп'ютерної системи розпізнавання користувача комп'ютера за його рукомоторними реакціями, які визначаються різними часовими інтервалами (час утримання клавіші, тривалість паузи перед натисканням клавіші, тривалість паузи після натисканням клавіші) як абсолютних, так і віднесених до їх середнього значення, або одного часового інтервалу до іншого, підтверджена результатами комп'ютерного моделювання.

На основі запропонованого підходу реалізована в середовищі DELPHI комп'ютерна система розпізнавання користувача комп'ютера за його рукомоторними діями. У реальному режимі часу в процесі набору користувачем заданого тексту відбувається формування функцій розподілу різних часових затримок, які апроксимуються нормальним законом розподілу. На основі співставлення біжучих значень математичних сподівань і дисперсій для кожного із сформованих розподілів з апіорі заданими зразками ідентифікується той чи інший користувач. Ефективність такої системи не перевищує 65 % при реєстрації всіх часових ознак.

Для підвищення ефективності розробленої системи запропоновано описувати її у вигляді системи дискретних рівнянь шостого порядку відповідно до сформованих значень дискретних ознак (часових затримок). Вибір базових функцій для опису такої системи розпізнавання є проблематичним, оскільки це мають бути імовірнісні функції розподілу, які відповідно до рукомоторних дій користувача мають передбачати появу тієї чи іншої літери на клавіатурі комп'ютера і прогнозувати величину часової затримки при її натисканні чи величину паузи до і після натискання. Але незалежно від вигляду цих базових функцій у випадку опису процесу у вигляді дискретної моделі, коли за ознаки вибрати відношення девіацій часу утримання до паузи перед клавішею та відношення девіацій паузи до часу утримання клавіші, максимальна інформативність яких підтверджена результатами комп'ютерного моделювання, оцінку періоду повторення слідування літер на клавіатурі можна отримати за формулою:

$$T = \frac{2\pi}{\varphi} \quad (10)$$

Якщо виходити з реального середнього часу утримання клавіші 0,3 с, то з урахуванням пауз до і після її утримання період набору літер не перевищуватиме 1 с, що відповідає початковій фазі коливань 2π . Отже, при введенні в алгоритм розпізнавання блоку формування неперервної послідовності літер, коли в реальному режимі часу відсікаються будь-які хаотичні рухи (випадкова неуважність, механічна затримка, натискання кількох клавіш, вимушена пауза тощо), ефективність такої системи ідентифікації користувача значно зростає. Як показали результати статистичних випробувань за наявності 200 користувачів у базі даних похибка розпізнавання не перевищувала 5%.

2.4. Перспективи розвитку та застосування дискретних моделей коливних систем до аналізу динаміки складних об'єктів. Результати проведеного аналізу комп'ютерної системи ідентифікації користувача комп'ютера підтверджують доцільність використання дискретних моделей до розв'язання широкого класу прикладних проблем, пов'язаних з розпізнаванням складних динамічних режимів, що мають місце в об'єктах коливної природи. Завжди, коли з апріорних міркувань можна визначити елементи матриці переходу станів для двох змінних, то коливну систему будь-якого порядку можна подати у дискретному вигляді [7], використовуючи для запису матриці станів подання (9). Застосування цього підходу до опису системи ідентифікації користувача комп'ютера підвищило достовірність розпізнавання в 1,4 рази. Вдалий вибір для функцій зміни амплітуд часових затримок дозволяє не лише ефективно реалізовувати процедуру розпізнавання, але й аналізувати психофізіологічний стан користувача комп'ютера і передбачати появу того чи іншого слова на екрані монітора. Таким чином, ця система може бути ефективно застосована і до розв'язання задач медичної діагностики при створенні біометричних вимірювальних систем.

Видається доцільним застосування описаного підходу до побудови системи розпізнавання рукописних літер, алгоритм та архітектура якої на основі структурного підходу описані в роботах [21-24]. Очевидно, тут слід виходити не зі структури написання літери, а будувати систему розпізнавання виходячи з напрямку руху руки (рух зверху вниз і в зворотному напрямку, рух зліва направо і в зворотному напрямку) та з часу написання літери. Оцінивши математичне сподівання часу написання кожної літери, яке характерне для кожного користувача, можна реалізувати процедуру розпізнавання. З математичної точки зору, з врахуванням напрямку руху руки це буде дискретна система 12 порядку. Виходячи з (10) можна стверджувати, що при формуванні неперервних послідовностей літер час написання має бути кратний цілому числу.

Цей інтервально-часовий підхід до формування первинних інформативних ознак про об'єкт чи процес дослідження орієнтований на використання дискретних моделей, з успіхом може бути застосований до побудови систем захисту інформації, медичної діагностики, біометричних систем, розв'язання транспортних задач, опрацювання потоків даних та створення інтелектуальних баз знань.

3. Висновки. Описано метод до формування первинних ознак при побудові комп'ютерної системи розпізнавання користувачів комп'ютера, запропоновано алгоритм та спосіб реалізації такої системи, відзначені особливості її функціонування та запропоновані шляхи підвищення точності розпізнавання та забезпечення достовірності ідентифікації користувачів комп'ютера на основі встановлення пріоритету первинних дискретних ознак та використання дискретних моделей, що по суті справи зв'язують тривалості пауз з часом утримання клавіш при введенні інформації з клавіатури комп'ютера. Відзначено основні напрямки розвитку автоматизованих систем розпізнавання об'єктів та процесів, побудованих на основі дискретних моделей, та вказані сфери їх доцільного застосування.

Список літератури: 1. Андронов А. А., Вит А. А., Хайкин С. Е. Теория колебаний. М.: Наука, 1981. 400 с. 2. Бутенин Н.В., Неймарк И., Фуфаев Н.А. Введение в теорию нелинейных колебаний. М.: Наука, 1976. 354 с. 3. Ван-дер-Полю. Нелинейная теория электрических цепей. М.: Связь, 1935. 186 с. 4. Видадь П. Нелинейные импульсные системы. М.: Энергия, 1974. 336с. 5. Динамика одномерных отображений / А.Н. Шарковский, С.Ф. Коляда, А.Г. Сивак, В.В. Федоренко. Киев: Наук. думка, 1989. 216с. 6. Заяць В.М. Построение и анализ модели дискретной колебательной системы // Кибернетика и системный анализ, 2000. № 5. С. 161-165. 7. Заяць В.М. Моделі дискретних коливних систем // Комп'ютерні технології друкарства. 1998. С.37-38. 8. Заяць В.М. Аналіз динаміки та умов стійкості дискретних моделей коливних систем // Вісник НУ „Львівська політехніка” “Інформаційні технології та мережі”. 2004. №519. С. 132-142. 9. Шустер Г. Детерминированный хаос: Введение: Пер. с англ. М.: Мир, 1988. 240 с. 10. Zayats V. Chaos searching algorithm for second order oscillatory system // Proc. International Conf. “TCSET – 2002”. Lviv – Slavsk. 2002. P. 97 - 98. 11. Заяць В.М., Улицький О.О. Алгоритмічне та програмне забезпечення системи розпізнавання людини за її рукомоторними реакціями. Вісник ДУ „Львівська політехніка” “Комп'ютерна інженерія та інформаційні технології”. 2000. № 392. С.73-76. 12. Заяць В.М. Підхід до опису системи розпізнавання користувача комп'ютера // Комп'ютерні технології друкарства. 2006. С. 46-53. 13. Заяць В.М., Заяць М.М. Математичний опис системи розпізнавання користувача комп'ютера // 36. “Фізико-математичне моделювання та інформаційні технології”. Львів. 2005. Вип. 1. С. 146-152. 14. Харкевич А.А. Опознание образов // Радиотехника. 1959. Том 14. С. 15-19. 15. Фукунага К. Введение в статистическую теорию распознавания. М.: Наука, 1979. 512 с. 16. Горелик А.Л., Скрипник В.А. Методы распознавания. М.: Высшая школа, 1989. 232 с. 17. Дуда Р., Харт П. Распознавание образов и анализ сцен. М.: Мир, 1976. 512 с. 18. Березин И.С., Жидков Н.П. Методы вычислений. М.: Физматгиздат, 1962. 639 с. 19. Заяць В.М., Шокира О. Визначення пріоритету детермінованих ознак при побудові системи розпізнавання об'єктів. Збірник. праць науково-практичної конф. ЛДІНТУ імені В. Чорновола “Математичне моделювання складних систем”. 2007. С.135-137. 20. Заяць В.М. Приведення неперервної автоколивної системи до дискретної моделі та спрощення її аналізу // Відбір і обробка інформації. 2005. Вип. 23 (99). С. 35-39. 21. Алексеев А., Заяць В., Иванов Д. Алгоритм розпізнавання символів на основі структурного підходу // Вісник НУ „Львівська політехніка” „Комп'ютерна інженерія та інформаційні технології”. 2002. № 468. С.129-133. 22. Заяць В.М., Иванов Д.О. Проект системи розпізнавання рукописного тексту // Вісник НУ „Львівська політехніка” „Комп'ютерна інженерія та інформаційні технології”. Львів. 2003. № 481. С. 78-83. 23. Заяць В.М., Иванов Д.О. Архітектура подієорієнтованих систем на прикладі системи розпізнавання рукописного тексту // Вісник НУ „Львівська політехніка” „Комп'ютерна інженерія та інформаційні технології”. Львів. 2004. № 530. С. 78-83. 24. Zayats V., Ivanov D. Structural method of handwritten text recognition // Pros. International Conf. “The experience of designing and application of CAD systems in microelectronics”. Lviv-Polyana. 2005. P. 493-494.

Надійшла до редколегії 05.07.2008

Заяць Василь Михайлович, д-р техн. наук, академік АНПРЕ, завідувач кафедри інформаційно - комп'ютерних технологій і систем Львівського державного інституту новітніх технологій та управління імені В. Чорновола. Наукові інтереси: комп'ютерне і математичне моделювання, розпізнавання та ідентифікація об'єктів складної динамічної природи. Адреса: Україна, 79000, Львів, вул. Генерала Чупринки, 130, тел. 258-91-82, 258-91-91 email: zvm01@rambler.ru.

СИСТЕМОЛОГИЧЕСКИЙ АНАЛИЗ ПРОБЛЕМЫ ПРИНЯТИЯ РЕШЕНИЙ В УСЛОВИЯХ МНОГОКРИТЕРИАЛЬНОСТИ И НЕОПРЕДЕЛЕННОСТИ

Анализируются особенности проблем принятия решений в условиях многокритериальности и интервальной целевой функции. Показывается, что многокритериальность является дополнительным источником неопределенности. Рассматриваются общие подходы к решению проблемы многокритериальной оптимизации в условиях интервальной неопределенности.

1. Введение

Процедура принятия решений является обязательным этапом любой целенаправленной деятельности. Именно поэтому проблема формализации этой процедуры, превращения ее из субъективного искусства в научно-обоснованную методологию уже долгие годы находится в поле зрения научной общественности. Интерес к проблеме особенно обострился в последние годы, что связано, с одной стороны, с непрерывно повышающимися требованиями к эффективности и оптимальности принимаемых решений, а с другой – с широкой компьютеризацией и информатизацией всех сфер интеллектуальной человеческой деятельности. В этих условиях уровень формализации процессов принятия решений в значительной степени определяют перспективы создания и развития «интеллектуальных» информационных систем, как инструментальной базы дальнейшей и более глубокой автоматизации интеллектуальной деятельности.

В общем случае процесс принятия решений можно структурировать на следующие этапы:

- формирование и анализ цели;
- выделение допустимого множества альтернативных путей ее достижения (решений);
- определение метрики, в которой производится оценивание эффективности альтернативных решений;
- оптимизация, т.е. выбор в рамках заданной метрики экстремального решения.

Все перечисленные этапы являются необходимыми, но, не умаляя важности любого из них, следует отметить особую концептуальную важность этапа оценивания, т.е. формирования критериев оценки эффективности решений.

На текущий момент основные успехи общей теории принятия решений связаны с концепцией скалярного критерия оценки эффективности. На этой концепции полностью базируется теория математического программирования, изучающая методы и инструментальные средства решения задач безусловной и условной оптимизации, теория игр, методы принятия решений в условиях риска и неопределенности.

Вместе с этим, как показано В.М. Глушковым [1], необходимыми условиями эффективности решений являются требования своевременности, полноты (комплексности) и оптимальности. Комплексность решения предполагает, что при его принятии необходимо как можно более полно учитывать различные аспекты последствий принятого решения, характер взаимосвязей внутри системы и с внешней средой. Первое порождает необходимость учета набора частных критериев эффективности, каждый из которых характеризует «частное» качество, а их совокупность достаточно полно характеризует качество решения в целом. Второе приводит за счет НЕ-факторов [2], таких как НЕполнота знаний, НЕточности формального описания модели, НЕвозможности количественного измерения или НЕточности измерений и т.д. к большей или меньшей, в зависимости от конкретной ситуации, неопределенности исходной информации.

Таким образом, усилия должны быть направлены на развитие теории принятия решений в условиях многокритериальности и неопределенности, с учетом того, что эти проблемы

взаимосвязаны, так как многокритериальность порождает неопределенность, а неопределенность – многокритериальность решения.

2. Основная часть

Постановка задачи многокритериальной оптимизации в условиях неопределенности.

Будем полагать, что задано множество допустимых решений X . Каждое решение $x_i \in X$, $i = \overline{1, n}$ характеризуется кортежем разнородных частных критериев $K = \langle k_j(x_i) \rangle$, $j = \overline{1, m}$. Это означает, что функциональная зависимость $k_j(x) = f_j(x)$ предполагается известной. Тогда оптимальным будет решение

$$x^0 = \arg \operatorname{extr}_{x \in X} \langle k_j(x) \rangle, \quad \forall j = \overline{1, m}. \quad (1)$$

В общем случае, т.е. если множество X содержит противоречивые решения (область компромиссов), задача (1) является некорректной по Адамару [3]. Это означает, что решения, для которого все частные критерии одновременно достигают экстремальных значений на множестве X не существует.

Общий подход к решению некорректных задач заключается в их регуляризации [4], т.е. трансформации их в условно корректные, путем учета некоторой дополнительной, зачастую эвристической информации. Эта процедура является проблемно-ориентированной, т.е. зависит от семантики формального описания задачи. Общий подход к регуляризации задачи многокритериальной оптимизации заключается в ее скаляризации, т.е. трансформации в задачу скалярной оптимизации или иерархическую последовательность таких задач. Широко известны такие методы скаляризации как принцип главного критерия, функционально-стоимостный анализ, последовательная оптимизация, анализ иерархий, теория полезности [5]. Реализация всех перечисленных методов связана с использованием дополнительной эвристической субъективной информации, носителем которой является человек – лицо, принимающее решение (ЛПР), эксперт и т.д.

Особое место среди перечисленных методов регуляризации занимает теория полезности [6], основой которой является гипотеза о существовании на ограниченном множестве альтернатив X относительной обобщенной количественной оценки их «качества» – полезности $P(x)$. Отсюда следует, что если заданы альтернативные решения $x_1, x_2 \in X$, и x_1 предпочтительней x_2 , то из этого следует

$$x_1 \succ x_2 \Leftrightarrow P(x_1) > P(x_2). \quad (2)$$

Таким образом, полезность $P(x)$ является скалярной количественной оценкой, учитывающей все разнородные частные критерии $k_i(x)$, $i = \overline{1, n}$

$$P(x) = F[\lambda, k_i(x)]. \quad (3)$$

где λ – кортеж коэффициентов, приводящих разнородные частные критерии к изоморфному виду.

Функцию полезности (3) можно привести к виду [7]

$$P(x) = F[A, k_i^H(x)], \quad (4)$$

здесь $k_i^H(x)$ – нормализованные, т.е. приведенные к безразмерному виду, с ограниченным интервалом возможных значений $[0, 1]$ и одинаковым направлением доминирования частных критериев; $A = \langle a_i \rangle$, $i = \overline{1, n}$ – кортеж безразмерных весовых коэффициентов, для которых выполняется условие

$$0 \leq a_i \leq 1, \quad \forall i = \overline{1, n}, \quad \sum_{i=1}^n a_i = 1. \quad (5)$$

Нормализация частных критериев проводится по формуле

$$k_i^H(x) = \frac{k_i(x) - k_i^{HX}(x)}{k_i^{HL}(x) - k_i^{HX}(x)}, \quad (6)$$

где $k_i^{HL}(x)$, $k_i^{HX}(x)$ – соответственно наилучшее и наихудшее значение частных критериев на данном множестве альтернатив X .

Согласно теореме [8] и последующему ее обобщению Д. Габором [9] любую функцию многих переменных можно представить в виде полинома Колмогора-Габора вида

$$P(x) = \sum a_i k_i(x) + \sum_{i=1}^n \sum_{j=1}^n a_{ij} k_i(x) k_j(x) + \dots \quad (7)$$

С учетом сказанного, задача многокритериальной оптимизации (1) трансформируется в корректную задачу скалярной оптимизации вида

$$x^0 = \arg \operatorname{extr}_{x \in X} P(x) \quad (8)$$

Для конструктивного решения этой задачи необходимо синтезировать модель вычисления полезности, т.е. идентифицировать структуру и параметры полинома (7). Особенность заключается в том, что принятие решений является субъективной интеллектуальной процедурой. Это означает, что единственным источником информации, необходимой для идентификации модели (7), является ЛПР. Для получения этой информации в настоящее время широко используется интроспективный подход, который заключается в том, что носитель информации побуждается к осознанию, структуризации, количественному оцениванию своих знаний и сообщению результатов внешнему наблюдателю – когнитологу. Такая процедура известна как методология экспертного оценивания.

Недостаток методологии экспертного оценивания заключается в том, что в силу особенностей процесса мышления, заключающихся в том, что человек в каждый момент времени может обрабатывать ограниченный объем информации, сложные процессы приходится структурировать на иерархически упорядоченных последовательностях задач экспертного оценивания. При этом на каждом этапе оценивания, особенно если необходимы количественные оценки, возникает интервальная погрешность, которая накапливается вплоть до потери адекватности модели и воспроизводимости результатов.

Вместе с этим известно, что специалисты достаточно уверенно и воспроизводимо решают задачу качественного ранжирования множества допустимых альтернатив, указывая лучшую из них или устанавливая отношение строгого или нестрогого порядка в терминах «лучше», «хуже», «одинаково».

Это открывает перспективы развития альтернативного экспертному оцениванию метода идентификации модели функции полезности, основанного на идеях компараторной идентификации [7].

Метод заключается в том, что ЛПР анализирует множество допустимых альтернативных решений X и указывает лучшее решение $x^1 \in X$. Это означает, что для него $x^1 \succ x_i$, $\forall i = 1, n-1$, где n – мощность множества допустимых решений. Тогда согласно теории полезности (2), можно записать систему неравенств вида

$$P(x_i) - P(x^1) \leq 0, \quad \forall i = 1, n-1. \quad (9)$$

Анализ системы неравенств (9) позволит решить задачу структурно-параметрической идентификации функции полезности в рамках полинома (7). Трудность заключается в том, что система неравенств (9) определяет не единственное, а область возможных решений. Для определения единственного решения вводятся различные регуляризирующие правила: чебышевской точки, среднего решения и другие [7,10,11].

Независимо от метода идентификации функции полезности $P(x)$ ее параметры содержат интервальную неопределенность. Источником такой неопределенности является разброс субъективных мнений экспертов при экспертном оценивании или мощность множества возможных решений при компараторной идентификации.

Другими источниками неопределенности модели принятых решений (8) являются НЕ-факторы [2] перечисленные выше. В результате модель принятия решений принимает вид

$$x^0 = \arg \max_{x \in X} F[\bar{A}, \bar{k}_i(x)], \quad (10)$$

где знаком «-» обозначены величины, содержащие интервальную неопределенность.

Под интервальной неопределенностью в дальнейшем будем понимать скалярные величины, для которых на числовой оси заданы левая и правая границы, определяющие интервал возможных значений и некоторая информация о предпочтительности значений внутри интервала.

Если границы интервала совпадают или он пренебрежимо мал, то величина считается детерминированной. В зависимости от вида и формы представления информации о предпочтительности значений внутри интервала можно выделить вероятностную, нечеткую и интервальную неопределенность. Вероятностная неопределенность задается в общем случае функцией плотности распределения вероятности и статистическими параметрами: дисперсией, математическим ожиданием и т.д. Достоверное определение этих характеристик требует очень большого (в пределе бесконечного) объема статистически однородных наблюдений. Практически всегда оперируют с ограниченным объемом наблюдений, на котором можно получить только оценки статистических характеристик, содержащие большую или меньшую, в зависимости от объема статистической выборки, интервальную неопределенность.

Иногда указанную неопределенность ЛПП или эксперты могут устранить с помощью своих знаний, опыта решения и анализа подобных задач и т.д. Например, принимаются допущения о нормальном законе распределения случайной величины, о значении математического ожидания и т.д. В этом случае целесообразно говорить об эвристической стохастической неопределенности, в отличие от объективной.

Другим видом неопределенности является нечеткость, представляющая собой субъективную эвристическую формализацию на основе теории нечетких множеств вербальных высказываний типа «приблизительно», «около», «большое значение» и т.д. Формализация заключается в задании ЛПП или экспертами границ множества возможных значений переменной и функции принадлежности, характеризующей степень принадлежности нечеткому множеству.

Наконец, если информация о предпочтительности значений внутри интервала полностью отсутствует, будем называть такую неопределенность интервальной.

Для многокритериальных ситуаций эффективность характеризуется скалярной многофакторной оценкой полезности $P(x)$ решения. По определению функция интервальных переменных является интервальной величиной. Это означает, что функция полезности для каждого решения $x_i \in X$, $\forall i = \overline{1, n}$ является интервальной. При этом, в зависимости от формы задания неопределенных параметров, характеризующих внешнюю независимую среду, возможны две формы представления эффективности решения.

Если множество возможных состояний внешней среды S задано неявно с помощью интервальных ограничений, то результаты анализа эффективности допустимых решений представляют обычно в экстенсивной форме. Это означает, что для каждого решения $x_i \in X$ определяется интервальное значение эффективности и на нем задается некоторая функция предпочтительности возможных значений. Например, для случая стохастической неопределенности внешней среды вычисляется функция плотности распределения значений полезности для каждого допустимого решения.

Если множество состояний S задано в явном виде (перечислением), то результаты анализа эффективности решений представляются в нормальной форме. В этом случае для каждого фиксированного состояния внешней среды определяются оценки эффективности каждого из допустимых решений и предпочтительности его реализации. Для случая стохастической неопределенности внешней среды S это позволяет определить функции распределения вероятностей для допустимого множества решений при фиксированном состоянии внешней среды S_j .

Таким образом, в первом случае каждое решение $x_i \in X$, $i = \overline{1, n}$, характеризуется набором пар $\langle p(x_i, s_j), f(x_i, s_j) \rangle$, $j = \overline{1, m}$, где первый элемент – условная вероятность реализации S_j состояния внешней среды для решения x_i , а во втором – каждое состояние S_j характеризуется набором кортежей $\langle p(s_j), f(x_i) \rangle$.

Если множества решений X и состояний внешней среды S счетные, указанные выше оценки представляются в виде матрицы платежей. Вычисления элементов матриц производятся или аналитически с помощью арифметики определенной для всех трех перечисленных видов неопределенности, или численным моделированием методом Монте-Карло.

Выбор эффективного решения для экстенсивной формы представления исходной информации связан с решением задачи сравнения интервальных величин (эта задача не тривиальна, если интервалы пересекаются)[12]. Для нормальной формы задача выбора эффективного решения сводится к выбору экстремального компромиссного решения по двум противоположным критериям: эффективности и вероятности реализации конкретного состояния внешней среды, т.е. к двухкритериальной оптимизационной задаче.

Для решения такой оптимизационной задачи введем понятие риска.

Пусть на основании анализа матрицы платежей выбрано конкретное решение, эффективность которого равна $p(x_k, s_k)$, где индексом «к» обозначены конкретные решения $x_i \in X$, и состояние внешней среды $s_k \in S$. В силу неопределенности внешняя среда может реализовать любое возможное состояние, отличающееся от s_k , как неблагоприятное, что приведет к потере эффективности (уменьшению прибыли, увеличению затрат ресурсов, времени и т.д.) по сравнению с расчетным значением, так и благоприятное, при котором, если бы было выбрано соответствующее решение, можно было получить дополнительный эффект. Таким образом, при принятии решений в условиях неопределенности существует риск потери эффекта, будем называть его негативным риском и обозначим R_1 , и риск упустить возможный эффект (негативный риск R_2). Очевидно, что указанные риски имеют для ЛПП различную «ценность». В первом случае это прямые экономические потери, вплоть до катастрофических, а во втором – сожаления по поводу упущенной возможности получить дополнительный эффект.

Таким образом, любое решение, принимаемое в условиях неопределенности, характеризуется тремя противоречивыми показателями (частными критериями): эффективностью $p(x)$, негативным R_1 и позитивным R_2 рисками. Это означает, что задача принятия решения в условиях неопределенности является типичной задачей многокритериальной оптимизации вида (1) и ее решение основывается на общей методологии, рассмотренной выше в данной статье.

В частности, как и в общем случае, возникает проблема скаляризации оценки качества решения. Для этого могут быть использованы традиционные методы: главного критерия, последовательной оптимизации, функционально-стоимостного анализа, обобщенной многофакторной оценки. Все известные в настоящее время критерии принятия решений в условиях неопределенности могут быть интерпретированы в рамках перечисленных методов. Для примера укажем, что на принципе главного критерия базируются критерий максимального математического ожидания эффективности, минимальной дисперсии, критерий минимума среднего риска Вальда. Типичным критерием, основанным на обобщенной оценке, является критерий Гурвица и т.д.

3. Выводы

Проведен системологический анализ проблемы принятия решений в условиях многокритериальности и неопределенности. Показано, что задачи многокритериальной оптимизации порождают неопределенность. Рассмотрены источники неопределенности, классификация видов интервальной неопределенности.

Сформулирована общая постановка задачи многокритериальной оптимизации в условиях неопределенности. Проведен анализ и показано, что неопределенность приводит к многокритериальности и для решения задач принятия решений в условиях неопределенности необходимо применять методологию многокритериальной оптимизации.

Список литературы: 1. Глушков В.М. Введение в АСУ. Киев: Техника, 1974. 320 с. 2. Нариньяни А.С. Нефакторы: неоднозначность (доформальное исследование) // Новости искусственного интеллекта. 2003. №5. С.58-69. 3. Математический энциклопедический словарь / Под ред. Ю.В.Прохорова. М.: Сов. энциклопедия, 1988. 848 с. 4. Тихонов А.И., Арсенин В.Я. Методы решения некорректных задач. М.: Наука, 1986. 288 с. 5. Петров Е.Г., Новожилова М.В., Гребенник Г.В. Методи і засоби прийняття рішень у соціально-економічних системах. К.: Техніка, 2004. 256 с. 6. Нейман Дж., Моргенштейн О. Теория игр и экономическое поведение. М.: Наука, 1970. 124 с. 7. Овезгельдыев А.О., Петров Э.Г., Петров К.Э. Синтез и идентификация моделей многофакторного оценивания и оптимизации / Под общей редакцией Э.Г.Петрова К.: Наук. думка, 2002. 64с. 8. Колмогоров А.Н. О представлении непрерывных функций

нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения // Докл. АН СССР. 1957. Т. 114, № 5. С. 953 – 956. **9.** Ивахненко А.Г. Метод группового учета аргументов – конкурент метода стохастической аппроксимации. Автоматика. 1968. №3. С. 57-73. **10.** Овезгельдыев А.О., Петров К.Э. Компараторная идентификация параметров линейных моделей многофакторного оценивания // Радиоэлектроника и информатика. 1998. №2. С. 41-43. **11.** Петров К.Э. Структурно-параметрическая идентификация модели многофакторного оценивания с применением аппарата искусственных нейронных сетей // Інформаційні технології та комп'ютерна інженерія. 2007. №1(8). С. 35-42. **12.** Овезгельдыев А.О., Петров К.Э. Оценка ранжирования альтернатив в условиях интервальной неопределенности // Кибернетика и системный анализ. 2005. №4. С. 148-153.

Поступила в редколлегию 12.12.2008

Лысенко Эдуард Викторович, д-р техн. наук, профессор каф. системотехники ХНУРЭ. Научные интересы: теория принятия решений, новые информационные технологии. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (8057)702-04-29, e-mail: ST@kture.kharkov.ua.

Пономаренко Владимир Петрович, председатель правления ОАО «Объединение Стеклопластик», г. Северодонецк, асп. каф. системотехники ХНУРЭ. Научные интересы: системы управления и принятия решений, теория принятия решений. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (8057)702-04-29, e-mail: ST@kture.kharkov.ua.

Писклакова Валентина Петровна, канд. техн. наук, доцент каф. системотехники ХНУРЭ. Научные интересы: математическое моделирование, теория принятия решений. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (8057)702-04-29, e-mail: ST@kture.kharkov.ua.

УДК 004.89

М.Ю. ВИШНЯК, С.С. ДОВГАНЬ

РАСШИРЕНИЕ ФУНКЦИЙ АСУ: УПРАВЛЕНИЕ ЗНАНИЯМИ

Управление знаниями становится необходимой функцией АСУ современного предприятия. Постоянно возрастающее влияние человеческого фактора на положение предприятия на рынке делает знание важнейшим ресурсом. Знание рассматривается как состояние системы самоуправления человека. Формулируется и формализуется критерий степени освоения знания. Предлагается не ограничиваться технологическими процедурами управления знаниями, а рассматривать весь цикл менеджмента. Описания знаний являются важной составляющей управления знаниями. Для повышения эффективности оперирования с описаниями их следует структурировать.

1. Введение

Последние 20-30 лет происходят кардинальные изменения в мировой экономической среде. Рождается и так называемая «новая экономика», экономика знаний, которая характеризуется следующими признаками [1–4]:

– интенсивное и часто непредсказуемое изменение рыночного спроса: компании должны научиться адаптироваться к происходящему и уметь быстро обучаться;

– усиливающаяся угроза потери конкурентоспособности, как следствие всевозрастающей скорости изменений, вызывает необходимость постоянного обновления продукции, что требует высокого уровня инноваций, который может быть обеспечен только при наиболее полном использовании знаний;

– глобализация и территориальное удаление подразделений предприятий требует использования информационных технологий для эффективного взаимодействия при совместной работе.

Теперь умение получить знания и эффективно ими распорядиться в значительной степени определяет результативность предприятия.

Способность эффективно использовать и развивать знания, воплощать их в новые технологии, изделия и услуги превращается в важнейший фактор выживания в условиях новой экономической среды. Возрастающий экономический вес знаний и их ведущая роль для развития компании увеличивают их значимость. Еще философ Ф. Бэкон (1561 – 1626)

сформулировал роль знаний, утверждая, что мы можем столько, сколько знаем, т.е. чем большими знаниями владеет человек, тем шире его возможности. Это утверждение справедливо и для предприятий, и для государств. Не случайно самыми богатыми являются страны с капиталом в форме хорошего образования и человеческих ресурсов, а самые бедные опираются только на природное сырье. Есть примеры стран с богатейшими природными ресурсами, но большая часть их населения бедна. Сами по себе природные ресурсы не являются определяющим фактором конкурентоспособности при отсутствии знаний [5].

По мере роста значения знаний в бизнесе и осознания их важности как стратегического ресурса компании становятся заинтересованными в управлении знаниями, потому что тем самым они осуществляют инвестиции в создание, сохранность и использование своих важнейших рыночных преимуществ. Но теория управления знаниями находится пока на начальной стадии становления, а знания как объект управления малоизучены [6-9].

Цель исследования состоит в развитии методологии построения системы управления знаниями как подсистемы АСУ современного предприятия.

Задачи исследования: Уточнить понятия «знание» и «управление знаниями». Формализовать критерий оценки знаний. Предложить обобщенную схему управления знаниями. Сформулировать задачу структурирования знаний.

2. Особенности знаний

Понятие «знание» используется в разных контекстах и является достаточно расплывчатым. На сегодня все еще трудно дать точное определение знаний. А без точного определения нет однозначности понимания, что является ресурсом и чем управлять.

БСЭ дает следующее толкование: «Знание – проверенный практикой результат познания действительности, вернее, ее отражение в сознании человека». Интерпретация знаний, как «совокупности сведений, образующих целостное описание, соответствующее некоторому уровню осведомленности об описываемом вопросе, предмете, проблеме и т.д.», дана в толковом словаре по искусственному интеллекту [10]. Среди специалистов в области управления знаниями наиболее общее определение дано Гавриловой Т.А. [11]: «Знания – это основные закономерности предметной области, позволяющие человеку решать конкретные производственные, научные и другие задачи».

Связать знания и действия предлагает М. Витцель в [12]: «Знание – это накопленные предпосылки для действия. ... Действие без знания становится невозможным». А также М. Желены в [5] определяет, что «знание – это целенаправленное координирование действия. Его единственное доказательство или способ демонстрации заключается в достижении цели». Эта точка зрения интересна тем, что согласно ей знания предопределяют действие, а действие, как известно, порождает результат, следовательно, результат обусловлен знаниями, и знания можно оценивать по полученному с их помощью результату.

Знания, как отображение объективной реальности, которыми оперирует человек, можно считать ментальной моделью мира (включающей и человеческий опыт). Определим знания через их основные свойства, а именно: знания не материальны, находятся внутри человека; определяют представления человека о внешнем мире; отражают и аккумулируют человеческий опыт; определяют возможности человека получать конкретные результаты.

Они обладают следующей спецификой: утрачиваются, если не используются, а при использовании совершенствуются; обладают уникальностью и слабой воспроизводимостью.

Знания, полученные в процессе деятельности предприятия, уникальны и трудны для копирования: при отсутствии должного опыта невозможно его воспроизводство конкурентами. Чтобы получить аналогичные знания, конкуренты должны обладать аналогичным опытом, либо они должны существенно ускорить свое обучение, что достаточно трудно выполнить даже при больших инвестициях.

Знания можно рассматривать как состояние системы самоуправления человека. В соответствии с подходом [13] определим эту систему как:

$$S \subset X \times Y,$$

$$\begin{aligned} \bar{\rho} &= \{\rho_t : Z_t \times X_t \rightarrow Y_t \ \& \ t \in T\}, \\ \bar{o} &= \{o_{tt'} : Z_t \times X_t \times Y_t \rightarrow Z_{t'} \ \& \ t, t' \in T \ \& \ t' > t\}, \\ \bar{\varphi} &= \{\varphi_{tt'} : Z_t \rightarrow Z_{t'} \ \& \ t, t' \in T \ \& \ t' > t\}, \\ T &= \{t : t \geq 0\}, \end{aligned}$$

где S – система; X – входной объект (информация, которую человек воспринимает); Y – выходной объект (принятые решения, выполненные действия и т.п.); Z – состояние (знание, которым обладает человек); $\bar{\rho}$ – семейство реакций системы S ; \bar{o} – семейство функций перехода состояний (изменение знаний в процессе деятельности); $\bar{\varphi}$ – семейство функций перехода состояний (изменение знаний при размышлениях); T – множество моментов времени.

Модель отображает то, что знания за определенные моменты времени определяют результат деятельности – реакция системы ρ , изменяются в процессе деятельности – функция перехода o и при размышлениях – функция перехода φ . Опыт человека (практика) сильно подвержен влиянию знаний: человек рассматривает мир через «призму» текущего знания (рис. 1).

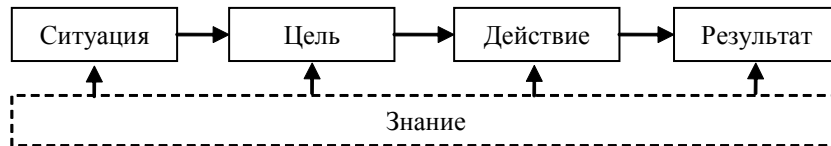


Рис. 1. Схема элементарного акта практики

Ситуация подталкивает к текущей цели, цель порождает действие, а действие приводит к результату, причем знания влияют на все составляющие практики: анализ и классификацию ситуации, постановку цели, выбор и выполнение действия, интерпретацию результата. Такое положение вещей, когда практика опирается только на текущее знание, может увести в сторону от получения желаемых результатов. Поэтому обучаемый должен получать дополнительные знания, позволяющие ему правильно строить свою практику и обучаться на ней.

Знания находятся внутри человека и присутствуют в нем, пока человек жив. Передавать знания другому человеку можно различными путями. Для аккумуляции и передачи знаний (как один из путей), как правило, используются текстовые и графические описания (рис.2). Но данные описания не есть знание [14].

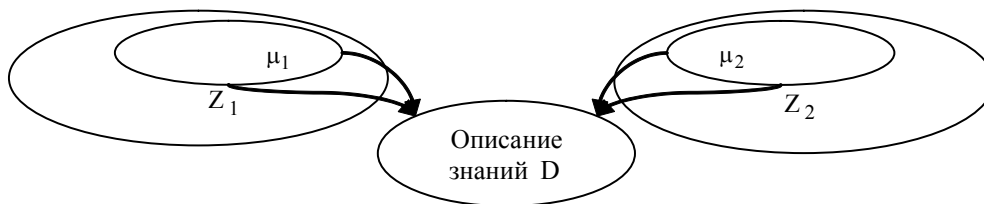


Рис. 2. Передача знаний через описание

В этом способе передачи заложены две погрешности, получающиеся при составлении описания и при его освоении [14].

Задача освоения знания по описанию состоит в достижении наиболее полного соответствия μ_1, μ_2 , т.е.

$$\|\mu_1 - \mu_2\| \rightarrow \min,$$

где μ_1 – знания (смысл), закладываемые автором описания; μ_2 – знания (смысл), получаемые обучаемым.

Каждому конкретному результату $R \in Y$ соответствует необходимое знание $\mu \in Z$. Степень освоения конкретного знания $\|\mu_1 - \mu_2\|$ для получения результата R_1 будем определять как

$$\|\mu_1 - \mu_2\| \approx \alpha * \|R_1 - R_2\| \quad R_1 \in Y_1, R_2 \in Y_2,$$

где α – масштабный коэффициент (некоторое число).

Процесс обучения должен продолжаться, пока не будет достигнут приемлемый уровень совпадения эталонного результата R_1 и результата, который получает обучаемый R_2 :

$$\|R_1 - R_2\| \leq \Delta,$$

где Δ – достаточная точность результата.

Освоение знания по описанию нуждается в сравнении результата, получаемого обучаемым, с некоторым эталонным. Поскольку описание не есть само знание, то должна еще присутствовать методика обучения по описанию. Таким образом, полное описание конкретного знания должно включать: описание знаний, необходимых для получения результата; описание получаемого результата; методику освоения знаний по его описанию.

3. Управление знаниями

Знания становятся объектом управления. Поэтому АСУ должны включать в свой состав функцию управления этим новым объектом.

Компании, которые используют управление знаниями в своей практике, способны:

- интенсивнее распространять ценные знания внутри компании, передавать их от сотрудника к сотруднику;
- гораздо быстрее принимать эффективные и качественные управленческие решения в результате получения своевременного доступа к релевантным знаниям;
- обновлять свою продукцию и вносить инновации в предоставляемые услуги за счет возможности коллективного формирования идей и сокращения затрат на возможное дублирование работ;
- быстрее вводить в курс новых сотрудников, которые на первых порах могут использовать имеющиеся в компании знания и осуществлять непрерывное повышение квалификации работников;
- предоставлять партнерам (потребителям, поставщикам, подрядчикам, клиентам) в дополнение к своим основным услугам доступ к накопленным знаниям, включая консалтинг и обучение.

В качестве объекта управления знаниями определим корпоративную компетенцию. Корпоративная компетенция – это совокупные знания всех сотрудников компании. Она определяется: умением производить; умением управлять; умением действовать на рынке.

Поскольку знания характеризуются полученным с их помощью результатом, то уровень корпоративной компетенции можно оценивать через: результаты деятельности компании; стоимость компании; уровень применяемых технологий, оборудования и т.п.

Цель управления знаниями состоит в поддержании и развитии корпоративной компетенции (совокупных знаний компании).

В литературе управление знаниями отождествляют с процессами, касающимися работы с источниками знаний, описаниями знаний и получателями знаний, причем стандартной схемы управления знаниями не существует, для каждой компании эта схема индивидуальна. В результате анализа существующих схем процесса управления знаниями (рис. 3а, б) были выделены основные процессы, которые в той или иной степени должны присутствовать в любой схеме управления знаниями [15-18].

Основные процессы, которые должны присутствовать в любой схеме управления знаниями – извлечение, хранение, распространение, применение знаний, представлены на рис. 4.

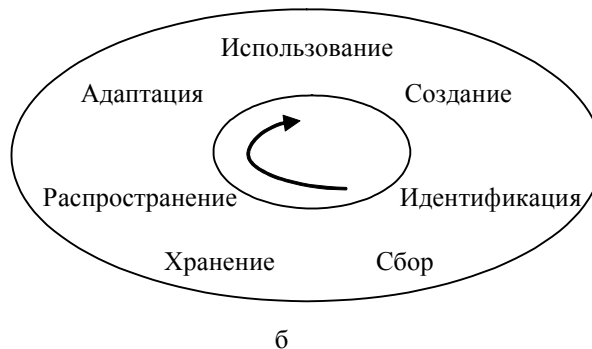
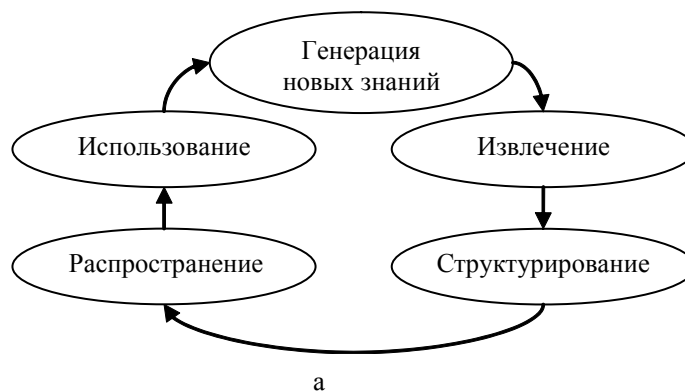


Рис. 3. Схемы процесса управления знаниями

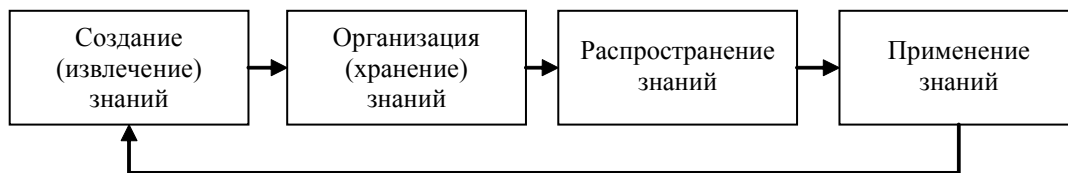


Рис. 4. Основные процессы управления знаниями

Это только технологические процессы. Управление знаниями должно включать все компоненты управления (весь цикл менеджмента): постановка целей, планирование, организация, сопровождение (выполнение), контроль, анализ и принятие решений (регулирование) (рис.5).

Цели управления знаниями формулируются в терминах желаемых результатов деятельности предприятия. Кроме этого, существует постоянно присутствующая цель, не выраженная через результаты деятельности, – это сохранение текущей компетенции компании. Эта цель была одним из основных мотивов появления управления знаниями. Когда сотрудник покидает компанию, он уносит с собой все свои знания, т.е. часть компетенции компании. Поэтому задача сохранения уровня компетенции компании с помощью технологии управления знаниями на сегодняшний день является актуальной.

Технологические процедуры – это процедуры, представленные в укрупненном виде на рис. 4. Назначение и функциональное наполнение остальных составляющих схемы (см. рис.5) соответствует общепринятому в менеджменте.

4. Структурирование описаний знаний

В управлении знаниями предполагается передача знаний как от человека к человеку (семинары, наставничество и т.п.), так и с использованием описаний знаний. Описания знаний позволяют также накапливать и тиражировать знания. Для повышения эффективности работы с описаниями знаний их рекомендуется структурировать.

Структурирование описаний знаний вызвано существующей потребностью:

– в быстром приобретении нужных знаний индивидуумом (существует много описаний знаний, но получение нужного требует больших усилий);

– в накапливании знаний в целях передачи, а также обобщений и получения нового знания.



Рис. 5. Схема управления знаниями

Компьютер стал мощным инструментом и его используют в следующих аспектах:

– знания вкладывают в компьютер для того, чтобы он выполнял действия по работе с информацией. Здесь знаниям необходимо придавать форму, подходящую для компьютерных операций;

– использовать компьютер как средство быстрого поиска и освоения знаний, необходимых человеку для выполнения действий. Здесь нужны формы представления, удобные для манипулирования фрагментами описаний знаний, методологии освоения и использования знаний.

Структурирование описаний знаний включает:

- выделение типов элементов (фрагментов) знаний;
- разработку шаблонов для каждого типа;
- установление взаимосвязей (отношений) между элементами одного типа и между элементами различных типов;
- классификацию знаний (по предметным областям и т.п.).

Структурированные описания знаний необходимы для формирования блоков необходимых описаний конкретному получателю в соответствии с его текущими запросами (требованиями), т.е. необходимо построить следующую процедуру структурирования описания знаний и процедуру восстановления необходимого блока знаний:

$$\psi_S : D_I \rightarrow D_S,$$

$$\psi_L : G_C \times D_S \rightarrow D_O,$$

где D_I – исходное описание знаний, полученное каким-либо способом от источника знаний;

D_S – структурированное описание знаний; G_C – потребности конкретных пользователей;

D_O – требуемый блок описания знаний.

Типами элементов описаний знаний для каждой предметной области могут быть [19]:

- понятия, определения и описания понятий;
- отношения между понятиями (структуры);
- решаемые задачи, структуры задач;

- процедуры, процессы, методы; знания, определяющие область применения процедур, рекомендации по применению, знания, на которых базируются процедуры и методы;
- правила (причинно-следственные связи);
- установленные факты;
- концепции (основные идеи, цели), парадигмы; предпосылки и основания для концепций и парадигм;
- примеры;
- динамика области (отображение исторического аспекта);
- инструменты, применяемые для решения задач из предметной области, и характеристики этих инструментов, технологии применения, ссылки на области, лежащие в основе создания инструментов;
- процедуры приобретения знаний человеком из этой предметной области (самообразование, самообразование, дистанционное академическое образование и т.п., т.е. упражнения, методические указания, процедуры самоконтроля, рекомендации контактов с преподавателями, обучающие игры и т.п.);
- мотивационные знания для этой предметной области.

По мере развития и получения новых знаний этот список может расширяться.

5. Заключение

Термин «управление знаниями» появился в англоязычной литературе как «knowledge management», т.е. менеджмент, а полный цикл менеджмента в публикациях по управлению знаниями не рассматривался, что вносило некоторую степень неопределенности. Кроме того, не проводилось различий между знаниями и описаниями знаний.

Научная новизна – дано определение знаний через список его релевантных свойств, предложено рассматривать знание как состояние системы самоуправления человека, сформулирован и формализован критерий степени освоения знания.

Определение знания, как состояния системы и того, что знание проявляется через результат, позволило сформулировать критерий степени освоения (обладания) конкретным знанием.

Введение понятия «описание знаний» позволяет определить более точно роль компьютера как инструмента аккумуляции и передачи знаний.

Практическая значимость – обоснована необходимость включения управления знаниями в функциональный состав АСУ предприятием, предложен расширенный цикл, который охватывает все функции управления знаниями, что позволяет строить более эффективные системы.

Введенные критерий степени освоения знания и понятие «описание знаний» дают возможность лучше понять знание как объект оперирования и строить соответствующие процедуры.

Все изложенные в работе положения используются авторами для дальнейшего развития теории управления знаниями и разработки новых функций современных АСУ.

Список литературы: 1. *Экономика знаний* / В.В. Глухов, С.Б. Коробко, Т.В. Маринина. СПб.: Питер, 2003. 527 с. 2. *Макаров В.Л.* Экономика знаний: уроки для России // Проблемы экономики переходного общества. 2004. №2. С. 57-63. 3. *Стрелец И.А.* Новая экономика и информационные технологии. М.: Экзамен, 2003. 254 с. 4. *Тельнов Ю.Ф.* Интеллектуальные информационные системы в экономике. М.: СИНТЕГ, 2002. 316 с. 5. *Желены М.* Знание против информации: В кн. Информационные технологии в бизнесе / Под ред. М. Желены. СПб: Питер, 2002. С. 211-218. 6. *Башмаков А.И., Башмаков И.А.* Интеллектуальные информационные технологии. М.: МГТУ им. Н.Э. Баумана, 2005. 304 с. 7. *Нонака И., Такеучи Х.* Компания – создатель знания. Зарождения и развитие инноваций в японских фирмах / Перевод с английского. М.: Олимп-Бизнес, 2003. 384 с. 8. *Махлун Ф.* Производство и распространение знаний в США. М.: Прогресс, 1996. 296 с. 9. *Беспалов П.В., Гапоненко А.Л., Корниенко В.И.* Интеллектуальный капитал – стратегический потенциал организации. М.: Соц. отношения, 2003. 178 с. 10. *Толковый словарь по искусственному интеллекту* / Авторы-составители А.И. Аверкин, М.Г. Гаазе-Рапопорт, Д.А. Поспелов. М.: Радио и связь, 1992. 256 с. 11. *Частиков А.П., Гаврилова Т.А., Белов Д.Л.* Разработка экспертных систем. Среда CLIPS. СПб.: БХВ-Петербург, 2003. 608 с. 12. *Витцель М.* Знание, определение понятия: В кн. Информационные технологии в бизнесе / Под ред. М. Желены. СПб: Питер, 2002. С. 198-210. 13. *Месарович М., Такахара Я.* Общая теория систем: математические основы. М.: Мир, 1978. 311 с. 14. *Гаврилова Т.А., Червинская К.Р.* Извлечение и структурирование знаний для экспертных

систем. М.: Радио и связь, 1992. 200 с. **15.** Кантерев А.И. Менеджмент знаний: от теории к технологиям. М.: Либерия-Бибинформ, 2005. 295 с. **16.** Knowledge-технологии в консалтинге и управлении предприятием / А.В. Вебер, А.Д. Данилов, С.И. Шифрин / Под ред. М.В. Финкова. СПб.: Наука и техника, 2003. 172 с. **17.** Wiig, K.M. People-focused knowledge management: how effective decision making leads to corporate success. Boston: Butterworth-Heinemann, 2004. 392 p. **18.** O'Dell C. The Executive's Role in Knowledge Management. Houston: APQC Publications, 2004. 129 p. **19.** Вишняк М.Ю. Особенности принятия решений по управлению в организационных сетях // Тезисы докладов Международной научной конференции «Теория и техника передачи, приема и обработки информации». 2003. №1. С. 371-372.

Поступила в редколлегию 24.12.2008

Вишняк Михаил Юрьевич, канд. техн. наук, доцент кафедры системотехники ХНУРЭ. Научные интересы: современные технологии управления организационными сетями. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (057) 702-10-06, (057) 345-12-57, (095) 887-44-09, E-mail: mvishnyak95@gmail.com.

Довгань Светлана Сергеевна, аспирантка кафедры системотехники ХНУРЭ. Научные интересы: исследование интеллектуальных систем, поддерживающих технологию управления знаниями. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (057) 702-10-06, (050) 228-63-88, E-mail: insolent@mail.ru.

УДК 519.7

ОМРИ КАРИМ

ПОСТРОЕНИЕ АЛГОРИТМА РАСПОЗНАВАНИЯ ОТПЕЧАТКОВ ПАЛЬЦЕВ ДЛЯ СИСТЕМЫ КОНТРОЛЯ ДОСТУПА

Разрабатывается алгоритм системы контроля доступа на основе отпечатков пальцев. Многолетний опыт применения и интенсивные разработки метода идентификации по отпечаткам пальцев привели к тому, что в настоящее время он рассматривается как наиболее надежный способ идентификации личности. Анализируются существующие методы распознавания и разрабатывается структурная схема системы контроля доступа на основе отпечатков пальцев.

1. Введение

За последние годы интерес к биометрической идентификации значительно вырос, так как с развитием современных технологий появилась возможность быстрого и точного определения уникальных физиологических данных человека. Такого рода идентификация нашла широкое применение в современных биометрических системах контроля и управления доступом (СКУД), которые обладают рядом существенных преимуществ перед традиционными СКУД, основанными на использовании идентификационных ключей (магнитных карт, электронных ключей и пр.) [1, 2]. Перечислим главные преимущества биометрических СКУД:

- исключается возможность несанкционированного использования ключей;
- обеспечивается высокая степень защиты от имитации;
- отпадает необходимость обязательного ношения ключей;
- исключается влияние человеческого фактора (потеря или порча ключа, забывчивость, передача ключей третьим лицам и т.п.);
- отсутствуют затраты на изготовление новых ключей, замену или восстановление существующих.

Наибольшее распространение в биометрических СКУД получил дактилоскопический метод (анализ отпечатков пальцев), который уже длительное время успешно используется в криминалистике. Выбор именно этого метода является разумным компромиссом между требованиями по достаточной надёжности, экономичности и скорости идентификации.

Дактилоскопический метод идентификации базируется на том, что не существует людей с одинаковыми отпечатками пальцев (даже у близнецов они различны), а папиллярный узор, сформировавшись на эмбриональном уровне, не меняется с возрастом и восстанавливается после кожных повреждений. Кожа человека имеет определённые температурные и

электрические характеристики, поэтому дактилоскопические данные могут быть получены не только оптическим, но и термическим и ёмкостным способами или на основе различных сочетаний этих трёх способов. Такое разнообразие способов реализации делает дактилоскопический метод идентификации более доступным, универсальным и актуальным [3, 4].

Целью исследования является разработка алгоритма распознавания на базе известных методов сканирования отпечатков пальцев для систем контроля доступа. Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать аналогичные автоматизированные системы контроля доступа;
- определить основные подходы к построению подобного рода систем;
- провести анализ существующих методов распознавания личности по отпечаткам пальцев.

2. Анализ методов идентификации по отпечаткам пальцев

Идентификация по отпечаткам пальцев – на сегодня самая распространенная биометрическая технология. По данным International Biometric Group, доля систем распознавания по отпечаткам пальцев составляет 52% от всех используемых в мире биометрических систем.

Во второй половине XX века в связи с появлением новых технических возможностей распознавание по отпечаткам пальцев начало выходить за рамки использования только в криминалистике и нашло свое применение в самых различных областях информационных технологий; в первую очередь такими областями стали:

- системы управления доступом;
- информационная безопасность (доступ в сеть, вход на ПК);
- учет рабочего времени и регистрация посетителей;
- системы голосования;
- проведение электронных платежей;
- аутентификация на Web-ресурсах;
- различные социальные проекты, где требуется идентификация людей (благотворительные акции и т. д.);
- проекты гражданской идентификации (пересечение государственных границ, выдача виз на посещение страны и т.п.).

В автоматизированном распознавании отпечатков пальцев, в отличие от традиционной дактилоскопии, возникает гораздо меньше проблем, связанных с различными внешними факторами, влияющими на сам процесс распознавания.

Качество получаемого со сканера изображения папиллярного узора пальца является одним из основных критериев, от которого зависит избираемый алгоритм формирования свертки отпечатка пальца и в конечном итоге идентификации человека.

В настоящее время выделяют три класса алгоритмов сравнения отпечатков пальцев:

1. Корреляционное сравнение – два изображения отпечатка пальца накладываются друг на друга и подсчитывается корреляция (по уровню интенсивности) между соответствующими пикселями, вычисленная для различных выравниваний изображений относительно друг друга (например, путем различных смещений и вращений). По соответствующему коэффициенту принимается решение об идентичности отпечатков.

Вследствие сложности и длительности работы данного алгоритма, особенно при решении задач идентификации (сравнение «один-ко-многим») – системы, построенные с его использованием, сейчас практически не используются.

2. Сравнение по особым точкам – по одному или нескольким изображениям отпечатков пальцев со сканера формируется шаблон, представляющий собой двухмерную поверхность, на которой выделены конечные точки и точки ветвления. При сравнении – на отсканированном изображении отпечатка также выделяются эти точки, их карта сравнивается с шаблоном и по количеству совпавших точек принимается решение по идентичности отпечатков.

В работе алгоритмов данного класса также используются механизмы корреляционного сравнения, но при сравнении положения каждой из предположительно соответствующих друг другу точек (рис. 1).

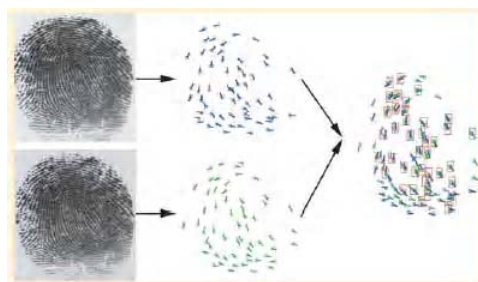


Рис. 1. Сравнение двух отпечатков пальцев по особым точкам

В силу простоты реализации и скорости работы – алгоритмы данного класса являются наиболее распространенными. Единственным существенным недостатком данного метода сравнения являются достаточно высокие требования к качеству получаемого изображения (около 500 dpi).

3. Сравнение по узору – в данном алгоритме сравнения используются непосредственно особенности строения папиллярного узора на поверхности пальцев. Полученное со сканера изображение отпечатка пальца разбивается на множество мелких ячеек, как показано на рис. 2 (размер ячеек зависит от требуемой точности).

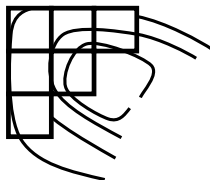


Рис. 2. Разбитие папиллярного рисунка на ячейки

Кроме этого, существует еще ряд методов распознавания отпечатков пальцев.

Метод выявления ключевых точек. Каждый отпечаток пальца состоит из определенного количества борозд и полосок. Полосы – это приподнятые части кожного покрова, борозды – впалые части. Полосы составляют так называемые ключевые точки: края полос (там, где полосы заканчиваются) и раздвоения – там, где они разветвляются (рис. 3).



Рис. 3. Регистрация отпечатка по набору ключевых точек

Во время регистрации ключевые точки располагаются в определенном месте, а их расположение относительно друг друга и их направление регистрируются. На основе этих данных создается образец (шаблон) – информация, которая впоследствии будет использована для удостоверения личности пользователя.

Запатентованный метод сопоставления узоров Precise Biometrics во время регистрации отпечатка определяет наличие упомянутых выше дополнительных характеристик. Небольшие участки отпечатка и расстояние между ними извлекаются из общей картины с целью максимально увеличить количество уникальной информации. Наиболее значимы участки вокруг ключевых точек и участки с небольшим радиусом изгиба. Основная структура и уникальные комбинации полос также являются ценными данными.

Метод Precise BioMatch использует как методы выделения ключевых точек, так и алгоритмы сопоставления узоров. Объединение двух разных технологий позволяет Precise BioMatch более эффективно работать с различными типами изображений, даже с отпечат-

ками низкого качества. Например, отпечаток пальца с несколькими ключевыми точками или отпечатки с размытым рисунком могут помешать пользователю при регистрации, однако смешанная технология, используемая алгоритмом Precise BioMatch, в данном случае будет иметь преимущество.

3. Алгоритм сканирования отпечатка пальца

Рассмотрим алгоритм считывания данных с фотодатчика (рис. 4, а).

При получении изображения мы работаем с регистром PixelData, который расположен по адресу 0x48.

Как видно из рис. 5, данный регистр состоит из двух системных бит (SOF и DataValid) и шести бит Pixel Data.

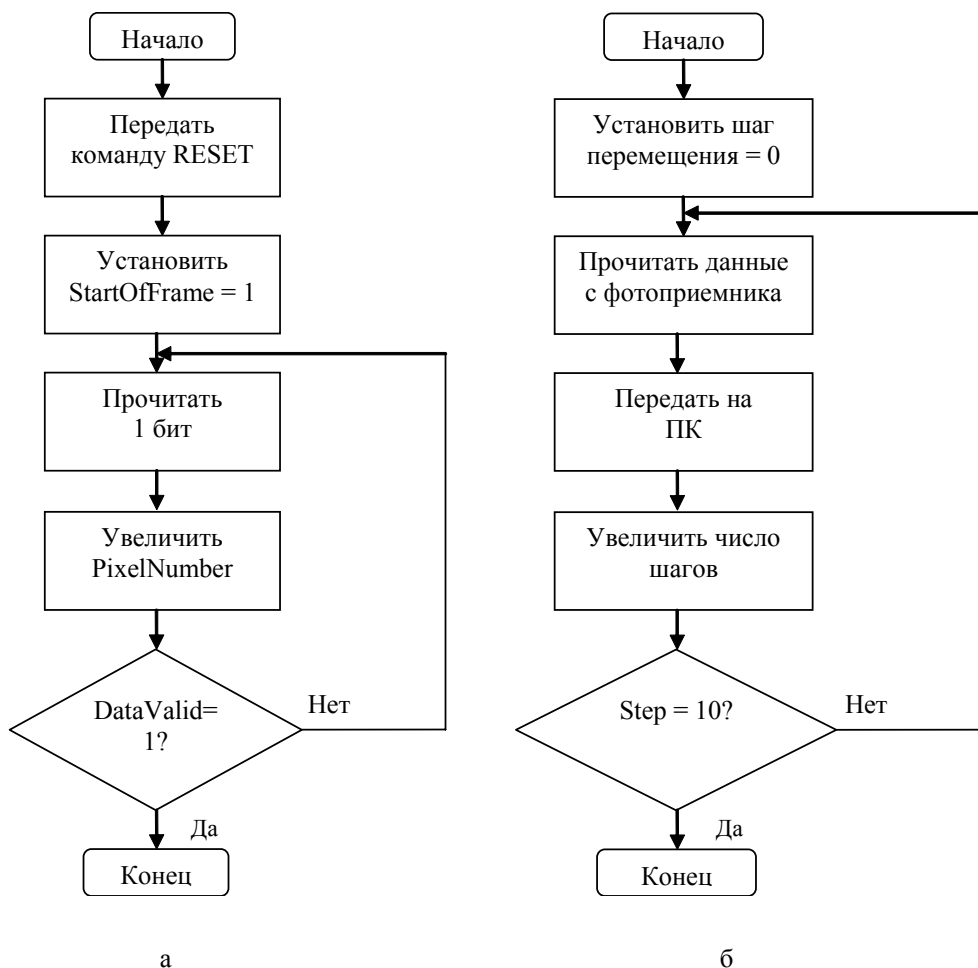


Рис. 4. Алгоритм считывания изображения (а) и алгоритм сканирования отпечатка пальца (б)

Bit	7	6	5	4	3	2	1	0
Field	SOF	Data_Valid	PD ₅	PD ₄	PD ₃	PD ₂	PD ₁	PD ₀

Рис. 5. Регистр PixelData

Бит SOF (StartOfFrame) принимает значение «1» после процедуры инициализации. Бит DataValid переходит в единичное состояние после считывания последнего бита изображения. Шесть бит данных кодируют значение яркости текущего пикселя по шкале от 0 до 63. При этом минимальное значение равно 0, а максимальное – 63.

Для считывания изображения, которое находится под фотодатчиком ADNS-2610, необходимо выполнить следующие операции:

- произвести процедуру инициализации;
- установить бит SOF (StartOfFrame) в состояние «1»;
- прочитать значение яркости для первого бита изображения;
- увеличить счетчик пикселей на 1;
- проверить состояние бита DataValid. Если оно равно 1, то операцию чтения можно прекратить. Если нет, то выполнить чтение следующего бита изображения.

На рис. 6 показан пример полученного изображения.

Для сканирования полного отпечатка пальца необходимо перемещение фотодатчика как по оси x , так и по оси y . Для этого необходимо разрабатывать сложный механизм перемещения, что существенно удорожит данную конструкцию.

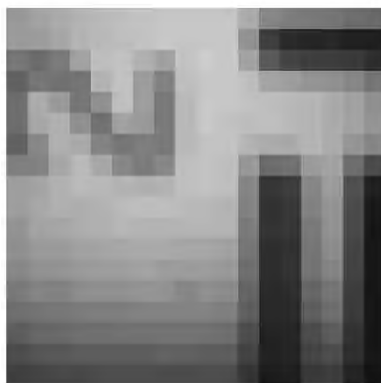


Рис. 6. Пример полученного изображения

Проведенные экспериментальные исследования показали, что при небольшом количестве хранимых изображений отпечатков пальцев (до 32) можно отказаться от перемещения датчика по одной из осей и сканировать только узкую полоску, шириной 5 мм. При этом полученного рисунка будет достаточно для однозначной идентификации личности.

При обработке наборов признаков происходит определение степени согласования каждого эталона набора с изображением отпечатка. Полученные вследствие этого данные суммируются с учетом весовых коэффициентов и запоминаются для дальнейшего использования. После обработки всех наборов эталонов происходит поиск наибольшей величины степени согласования. По номеру этого набора находится описание распознаваемого изображения.

В случае, если максимальная вероятность не превышает 50%, то изображение считается не распознанным. Тогда можно перейти в режим обучения для классификации подобного изображения. В режиме обучения новые изображения отпечатков пальцев сканируются, описываются и присваиваются определенному сотруднику. После этого полученные данные заносятся в базу данных.

4. Выводы

В результате исследования был разработан новый алгоритм сканирования отпечатка пальца.

Научная и практическая значимость. На основе синтеза известных методов распознавания отпечатков пальцев получен надежный алгоритм распознавания и управления системой контроля доступа, реализуемый на базе ПК.

Список литературы: 1. Андрианов В.И., Соколов А.В. Охранные системы для дома и офиса. СПб.: БХВ-Петербург; Арлит. 2002. 304 с. 2. Магауенов Р.Г. Системы охранной сигнализации: основы теории и принципы построения: Учебное пособие. М.: Горячая линия – Телеком, 2004. 367 с. 3. Горелик А.Л., Скрипник В.А. Методы распознавания. М.: Высшая школа, 1984. 4. FBI Integrated Automated Fingerprint Identification System (IAFIS), USA Federal Bureau of Investigation, Web: <http://www.fbi.gov/hq/cjisd/iafis/iafisbuilds.htm>

Поступила в редколлегию 15.12.2008

Омри Карим, студент гр. ПЗАСм-08-1, кафедра программного обеспечения ЭВМ, ХНУРЭ. Научные интересы: методы распознавания, искусственный интеллект. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-14-46.

МЕТОДИКА ФОРМИРОВАНИЯ МАТЕМАТИЧЕСКОЙ МОДЕЛИ СУММАТОРА В ФОРМАТЕ ТРОИЧНОЙ СИСТЕМЫ СЧИСЛЕНИЯ

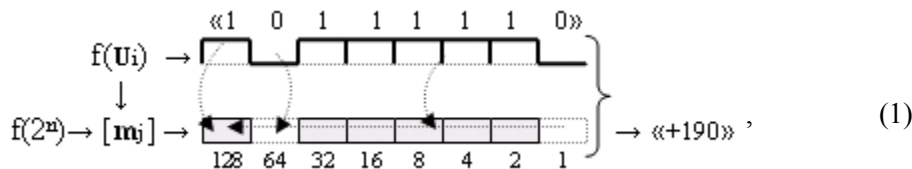
Предлагается метод выполнения арифметических операций, в частности процессов суммирования и вычитания, в позиционно-знаковых кодах. Данный метод позволяет минимизировать сквозные переносы и сформировать параллельные структуры сумматоров с локальными переносами (перенос только в соседний разряд) независимо от разрядности самого сумматора.

1. Введение

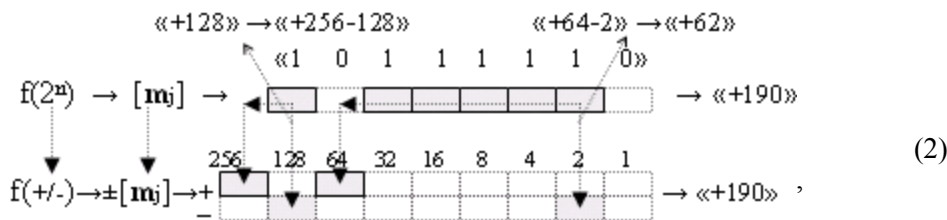
Одним из методов повышения быстродействия сумматоров является метод ускоренного переноса [1,2], в котором все разряды сумматора разбивают на группы и в каждой из них формируют общую функциональную структуру переноса. Но и этот метод не приводит к структуре параллельного суммирования, поскольку с увеличением разрядности сумматора совокупность общих функциональных структур переноса по существу реализует последовательную функциональную структуру переноса. Существенное повышение быстродействия арифметических операций и в частности процесса суммирования для создания процессорных структур нового поколения остается актуальной проблемой.

2. Постановка задачи

Если анализировать арифметические процессы преобразования аргументов не на «экономическом» уровне, который формализован до уровня арифметических аргументов «0» и «1» [1-3], а на уровне аналоговых сигналов, т.е. тех реальных процессов, которые происходят в функциональной структуре арифметического устройства, то аргументы аналоговых сигналов необходимо формализовать. При этом введенная формализация должна быть по существу логическим развитием существующей формализации в плане понимания логики преобразования аргументов в реальных арифметических структурах. В качестве перехода от аргументов реального сигнала $f(U_i)$ к формализованным аргументам $[m_j]$ может быть графоаналитическая структура (1).

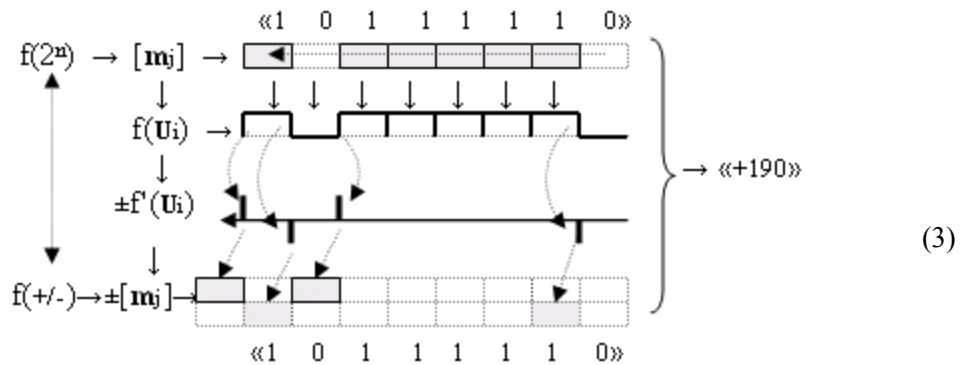


где $\boxed{}$ и $\boxed{}$ – формализованные аргументы $[m_j]$ соответственно активного и неактивного уровня аналоговых сигналов, структура которых эквивалентна как коду двоичной системы, так и совокупности цифр десятичной системы счисления. При этом переход от структуры аргументов аналоговых сигналов (1) в двоичной системе счисления $f(2^n)$ к структуре аргументов аналоговых сигналов в позиционно-знаковой системе счисления $f(+/-)$ можно записать в виде графоаналитического перехода (2).



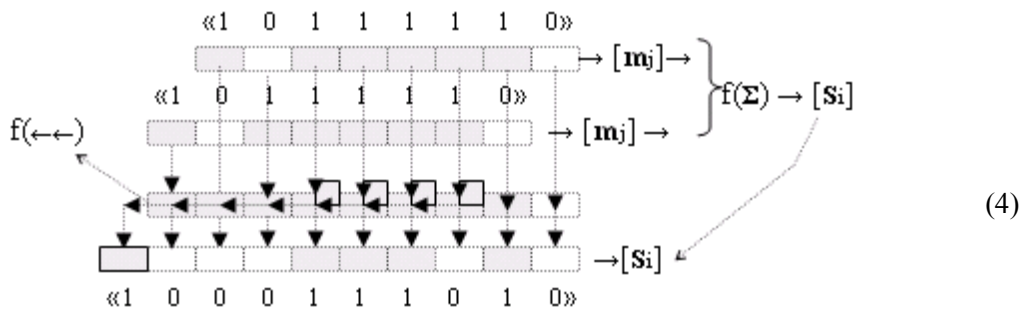
где $\boxed{}$ и $\boxed{}$ – положительный и условно отрицательный аргументы позиционно-знаковой системы счисления $f(+/-)$ [4]. Следует отметить, что только совмещение трех форм формализованной записи информационного содержания арифметического аргумента «+190»

(2) позволило сформировать позиционно-знаковую систему счисления $f(+/-)$, которая формируется в результате логического дифференцирования d/dn (3) аргументов $[m_j]$, эквивалентны структуре аналоговых сигналов $f(U_i)$:

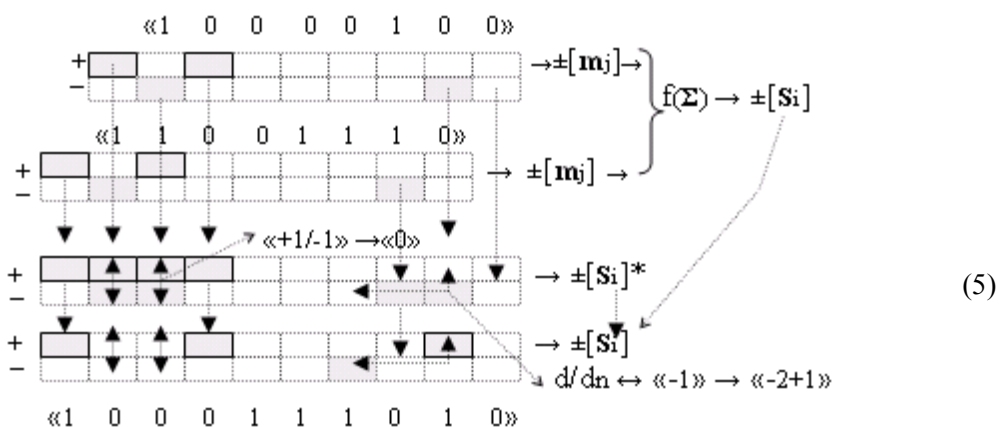


Сформированная позиционно-знаковая система счисления $f(+/-)$, которая по существу эквивалентна троичной системе счисления $(+1, 0, -1)$, позволяет существенно повысить быстродействие процессов суммирования и умножения. Сравнительный анализ арифметического процесса суммирования в двоичной $f(2)$ и позиционно-знаковой системах счисления $f(+/-)$ целесообразно провести в графоаналитическом виде.

В двоичной системе счисления $f(2^n) > [m_j] > 2 \cdot [m_j] > \langle 10111110 \rangle$:



В позиционно-знаковой системе счисления $f(+/-) > \pm[m_j] > \pm 2 \cdot [m_j]$:



Из сопоставительного анализа процессов суммирования аргументов (4) и (5) следует, что с одной стороны только при выполнении операции суммирования в позиционно-знаковой системе счисления $f(+/-)$ может быть получено высокое быстродействие, поскольку многофункциональная операция сквозного переноса $f(\leftarrow\leftarrow)$ выполняется в системе аргументов $f(+/-)$ косвенным путем, посредством удаления активных аргументов, которые эквивалентны логическому нулю $\langle +1/-1 \rangle \rightarrow \langle 0 \rangle$. С другой стороны, операция удаления

активных аргументов, которые эквивалентны логическому нулю «+1/-1» → «0», является операцией вычитания аргументов в системе $f(2)$. Поэтому можно предположить, что операции суммирования и вычитания аргументов в системе аргументов $f(2)$ являются частными реализациями операции суммирования в системе аргументов $f(+/-)$.

3. Цель исследования

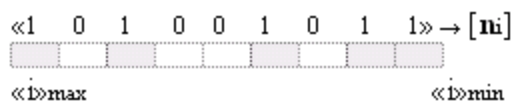
Целью исследования является, с одной стороны, сформировать методику суммирования в позиционно-знаковой системе аргументов $f(+/-)$. С другой – показать на формализованном уровне, что только позиционно-знаковая система счисления $f(+/-)$ может адекватно отражать арифметические процессы суммирования и вычитания во всем объективно существующем арифметическом пространстве суммирования и вычитания аргументов.

4. Методика формирования позиционно-знаковой системы счисления $f(+/-)$ с оптимальной и условно-оптимальной структурой аргументов

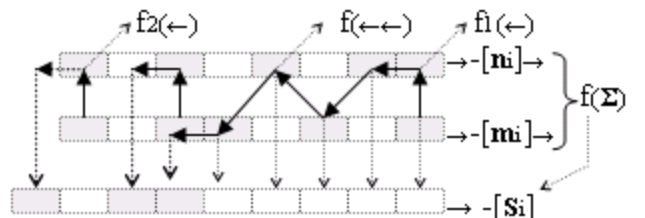
Поскольку структура аргументов в реальной действительности – это электрические напряжения, а последовательность электрических напряжений, которая соответствует логической композиции аргументов, например, $[n_i] \rightarrow \langle 101001011 \rangle$, во временном интервале имеет графоаналитическую интерпретацию вида



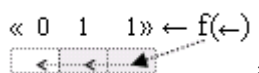
то имеет смысл графоаналитическую интерпретацию композиции аргументов в обычной записи $[n_i] \rightarrow \langle 101001011 \rangle$ записывать в виде



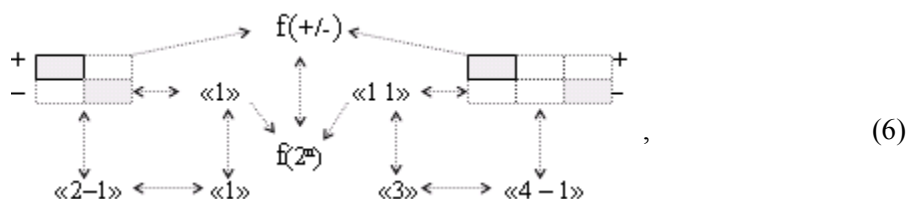
В связи с введенной формализацией логических аргументов проанализируем графоаналитическую структуру процесса суммирования



где $f1(\leftarrow)$ и $f1(\leftarrow)$ – функции локального переноса аргумента; $f(\leftarrow\leftarrow)$ – функция сквозного переносов аргумента. С одной стороны, анализ возникновения сквозного переноса $f(\leftarrow\leftarrow)$ показывает, что он формируется только в тех случаях, когда возникает структура аргументов вида



т.е. при наличии последовательно расположенных логических аргументов «11» и функции локального переноса в младшем разряде. С другой стороны, если проанализировать функциональную структуру вычитания, например,



где $f(+/-)$ – структура логических аргументов позиционно-знаковая система счисления с неполным вычитанием аргументов, которые представлены в двоичной системе $f(2^n)$.

Из анализа графоаналитической структуры (6) следует, что двоичная система $f(2^n)$ может быть оптимизирована путем исключения причин, которые формируют сквозные переносы $f(\leftarrow\leftarrow)$, поскольку любую последовательность логических аргументов вида «11» можно представить, как систему

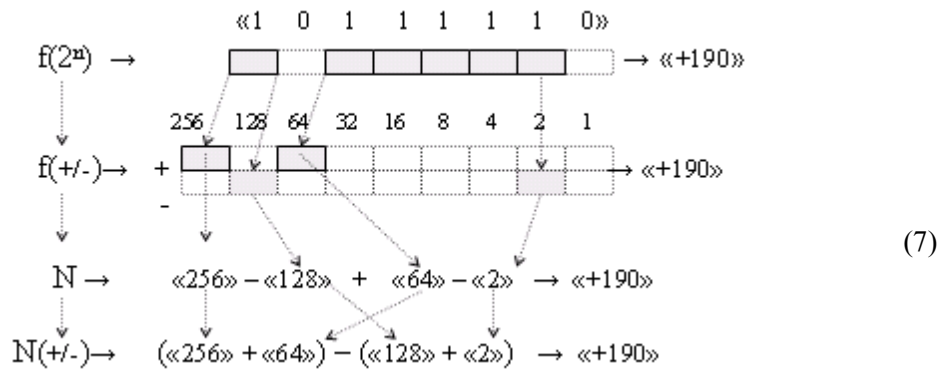
$$\langle\langle +11 \rangle\rangle \rightarrow \langle\langle +100 - 001 \rangle\rangle,$$

при этом любое число в двоичной системе может быть представлено в виде суммарной последовательности разности двух чисел

$$N \rightarrow \sum [f(2^{n_1}) - f(2^{n_2})] \rightarrow f(+/-),$$

где n_1 и n_2 – наперед заданные дискретно изменяющиеся числовые последовательности. Другими словами, одно и то же число может быть представлено различными структурами логических аргументов. Из сказанного выше следует, что позиционно-знаковая система счисления $f(+/-)$ является оптимальной для выполнения арифметических операций, поскольку отсутствуют причины, вызывающие сквозной перенос.

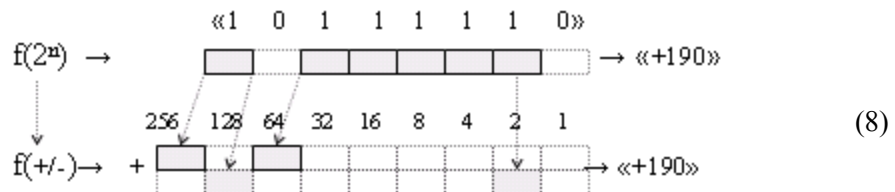
Пример 1. Проанализируем структуру аргументов $[ni] \rightarrow \langle\langle 101111110 \rangle\rangle$, записав ее в виде графоаналитической последовательности:



Анализ графоаналитической структуры (7) показывает, что любое число в двоичной системе счисления $f(2^n)$ может быть преобразовано в позиционно-знаковую систему счисления $f(+/-)$, которая несет в себе знаковую информацию «+» и «-». При этом система счисления $f(+/-) \rightarrow N(+/-)$ по существу является системой двух композиций аргументов двоичной системы $+f(2^n)$ и $-f(2^n)$ логические аргументы «1»,



в которой по своему позиционному положению разделены логическими аргументами «0», что и исключает возникновение сквозного переноса. Из графоаналитической структуры (7) также следует, что положительная составляющая аргументов $+f(2^n)$ системы $f(+/-)$ имеет увеличенную разрядность по сравнению с двоичной системой $f(2^n)$ на один старший разряд. Формализуя процесс преобразования в структуре (7), его можно представить в виде

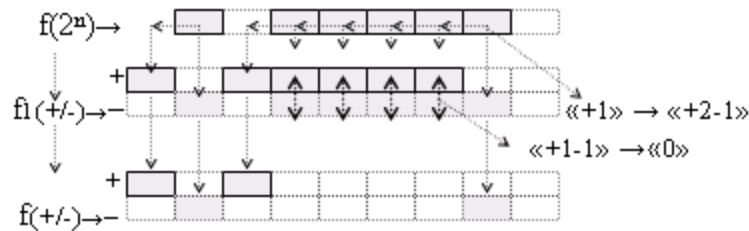


Из сформированного процесса перехода (8) от двоичной системы $f(2^n)$ к позиционно-знаковой системе $f(+/-)$ следует правило преобразования.

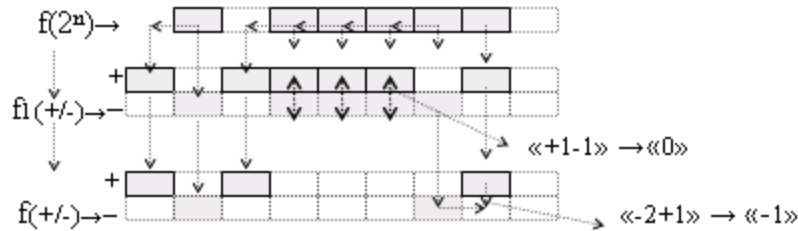
Правило. В процессе преобразования $f(2^n) \rightarrow f(+/-)$, независимо от количества логических аргументов в последовательности $\langle 1 \dots 1 \rangle$ выполняют перенос логического аргумента «1» старшего и младшего разрядов последовательности $\langle 1 \dots 1 \rangle$ независимо от количества логических аргументов «1» в последовательности. При этом перенос старшего разряда последовательности $\langle 1 \dots 1 \rangle$ выполняют в старший разряд положительной составляющей $f(+/-)$, а младшего – в одноименный разряд отрицательной составляющей $f(+/-)$. Из сформулированного правила следуют следствия.

Следствие 1. Любая архитектурная композиция логических аргументов в позиционно-знаковой системе счисления $f(+/-)$ увеличена на один разряд по сравнению с двоичной $f(2^n)$ системой счисления.

Следствие 2. Преобразование двоичной системы $f(2^n)$ в позиционно-знаковую систему $f(+/-)$ для логической последовательности аргументов $\langle 1 \dots 1 \rangle$ может быть осуществлено как для всего блока $\langle 1 \dots 1 \rangle$, так и для каждого логического аргумента «1» в отдельности:



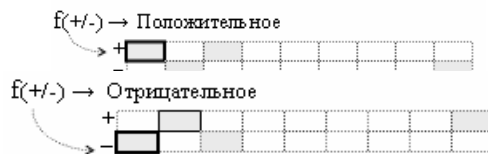
Следствие 3. Преобразование двоичной системы $f(2^n)$ в позиционно-знаковую систему $f(+/-)$ для логической последовательности аргументов $\langle 1 \dots 1 \rangle$ может быть осуществлено как для всего блока $\langle 1 \dots 1 \rangle$, так и для каждого логического аргумента «1» в отдельности, кроме последнего:



Данное свойство основано на применении аксиомы преобразования $\langle -2+1 \rangle \rightarrow \langle -1 \rangle$.

На основании сказанного выше следует вывод, что синтезированная система счисления $f(+/-)$ является производной двоичной $f(2^n)$ системы счисления, которая является позиционно-знаковой или биполярной, а поскольку система биполярная, то для того, чтобы получить из положительного числа в системе $f(+/-)$ отрицательное, достаточно аргументы в структуре «+» и «-».

Пример 2. $\langle 256 \rangle + \langle 64 \rangle - \langle 128 + 2 \rangle > \langle +190 \rangle$:



Определение 1. Оптимальным кодом системы логических аргументов $f(+/-)$ вида



будем называть такую системную последовательность «0» и «1», у которой две последовательно расположенные «1» разделены как минимум одним «0», а число логических «1» в + и – равно. При этом разность между положительной и отрицательной комбинацией аргументов равно значению аргумента в двоичной системе счисления $f(2^n)$.

Свойство 1. Сумма логических аргументов «1» оптимального кода всегда четная.

Свойство 2. Оптимальный код – код с функциональным знаком, т.е. если аргумент $|+f(2^n)| > |-f(2^n)|$, то оптимальный код положителен, а если $|+f(2^n)| < |-f(2^n)| \rightarrow$ оптимальный код отрицательный.

Введем определение.

Определение 2. Условно оптимальным кодом системы аргументов вида



будем называть такую систему $f^n(+/-)$, у которой нарушается четность, но сохраняется чередование логических аргументов «1», как минимум, через два «0».

Аксиома 1. Оптимальный код и совокупность условно-оптимальных кодов могут формировать одно и то же двоичное число.

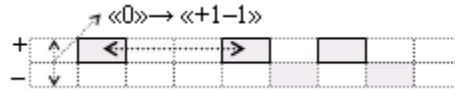
На основании сказанного выше можно сформулировать ряд утверждений.

Утверждение 1. Из числового множества обязательно найдется соответствующая пара $+f(2^n)$ и $-f(2^n)$, которая удовлетворит условию условной оптимальности кода.

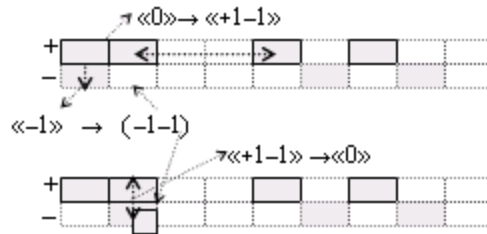
Утверждение 2. При формировании условно-оптимального кода сквозных переносов через разряд можно избежать.

Аксиома 2. В условно-оптимальном коде «заем» логической «1» может быть осуществлен у логического «0» предыдущего разряда.

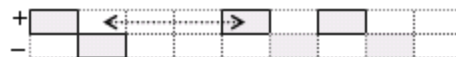
Пример 3.



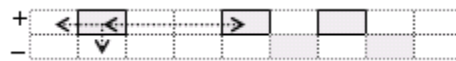
Введем в предыдущий разряд логический «0» $\rightarrow (+1-1)$:



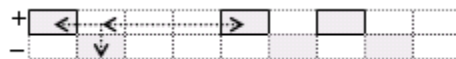
Выполняя указанные действия, получаем структуру вида



Аксиома 3. (Обратная). Любая логическая «1» в оптимальном и условно-оптимальном коде может быть преобразована или представлена как $\langle 1 \rangle \rightarrow \langle 2 - 1 \rangle$, поскольку каждый предыдущий разряд по весу в два раза больше предыдущего:



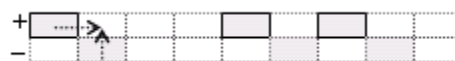
В результате выполнения указанных действий получаем структуру вида



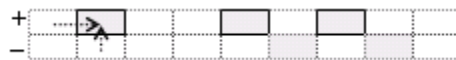
Аксиома 4. (Обратного знака). Любая логическая «отрицательная» «1» в оптимальном и условно-оптимальном коде может быть преобразована или представлена как

$$\langle -1 \rangle \rightarrow \langle -2 + 1 \rangle,$$

поскольку каждый предыдущий разряд по весу в два раза больше предыдущего:



В результате выполнения указанных действий получаем структуру вида



Утверждение 3. Двоичный код не является оптимальным для выполнения арифметических операций.

Данное утверждение обосновывается тем, что в двоичной системе счисления нельзя локализовать перенос до одного разряда, что в конечном счете снижает быстродействие арифметических операций повышенной разрядности.

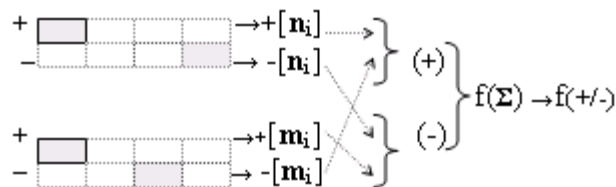
Утверждение 4. Любой не оптимальный код может быть локально преобразован в оптимальный или условно-оптимальный код.

Следует особо отметить, что сформированная система счисления по существу является многовариантной системой прямого кода со знаком, в которой отсутствуют проблемы с «+0» и «-0» и переполнением, а знак результата преобразования формируется из соотношения: если (+) > (-) → число положительное; (+) < (-) → число отрицательное.

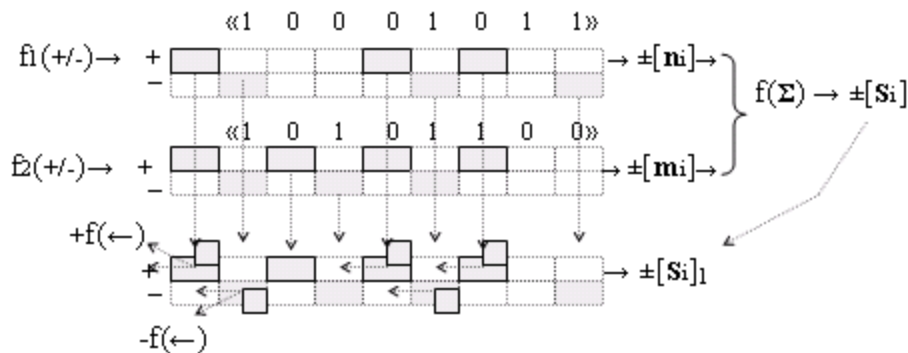
5. Методика формирования арифметических операций суммирования и вычитания

Поскольку структура логических аргументов в системе счисления $f(+/-)$ несет также знаковую информацию, то графоаналитический процесс арифметических действий может быть представлен в виде:

Сумма (сложение одноименных по знаку разрядов)



Пример 4. Сумма для аргументов $f_1(+/-) \rightarrow [n_i]$ и $f_2(+/-) \rightarrow \pm [m_i]$, которые эквивалентны аргументам двоичной системы $\pm [n_i] \rightarrow \langle 10001011 \rangle$ и $\pm [m_i] \rightarrow \langle 1010100 \rangle$:

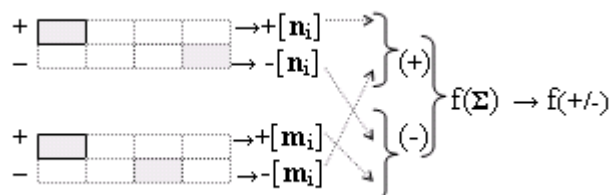


Анализ примера 4 показывает, что при суммировании цифровых аргументов в системе счисления $f(+/-)$ любых структур аргументов возникают только локальные переносы аргументов в очередной старший разряд.

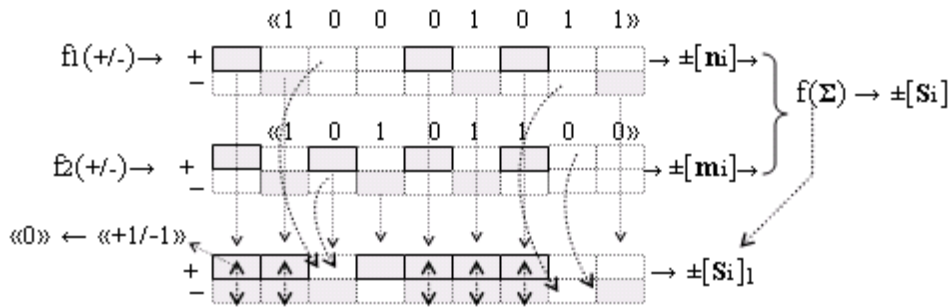
Разность (вычитание)

(сложение разноименных по знаку разрядов)

Из отрицательной структуры аргументов вычитают положительную структуру аргументов:



Пример 5. Разность аргументов $f_1(+/-) \rightarrow \pm[n_i]$ и $f_2(+/-) \rightarrow \pm[m_i]$, которые эквивалентны аргументам двоичной системы $\pm[n_i] \rightarrow \langle 10001011 \rangle$ и $\pm[m_i] \rightarrow \langle 10101100 \rangle$:



Легко заметить, что в оптимальной системе $f(+/-)$ процесс суммирования и вычитания реализуется не только с локальным переносом (без сквозного переноса), т.е. перенос осуществляется только в соседний разряд, но и не возникает неопределенность вида

При суммировании

$\langle + \rangle$ и $\langle - \rangle$ } > либо $\langle + \rangle$, либо $\langle - \rangle$.

При вычитании

$\langle + \rangle$ и $\langle + \rangle$ } > либо $\langle + \rangle$, либо $\langle - \rangle$.

$\langle - \rangle$ и $\langle - \rangle$ }

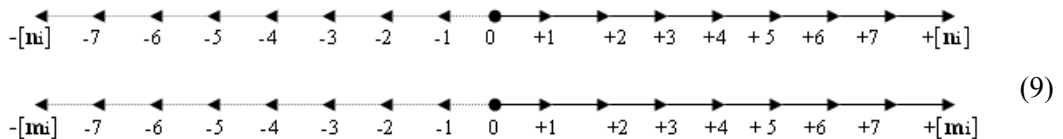
На основании сказанного выше сформируем последовательность действий, которые должны быть включены в каждый разряд параллельного сумматора, при суммировании в оптимальных и условно-оптимальных двоичных кодах:

\leftarrow – «1» перенос в соседний разряд; \leftarrow – «+1/-1» удаление логического нуля «+1-1» > «0»; \leftarrow – «1,1» корректировка логической последовательности аргументов, которые нарушают условие оптимальности кода; \leftarrow – «+1/-1» удаление логического нуля на втором этапе преобразования «+1-1» \rightarrow «0».

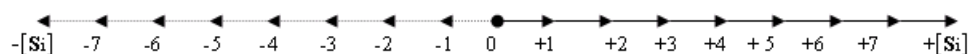
Сформированная последовательность действий при выполнении операции суммирования аргументов слагаемых позволяет увеличить быстродействие сумматора с ускоренным переносом как минимум в два раза.

6. Векторная графоаналитическая интерпретация процесса суммирования и вычитания аргументов

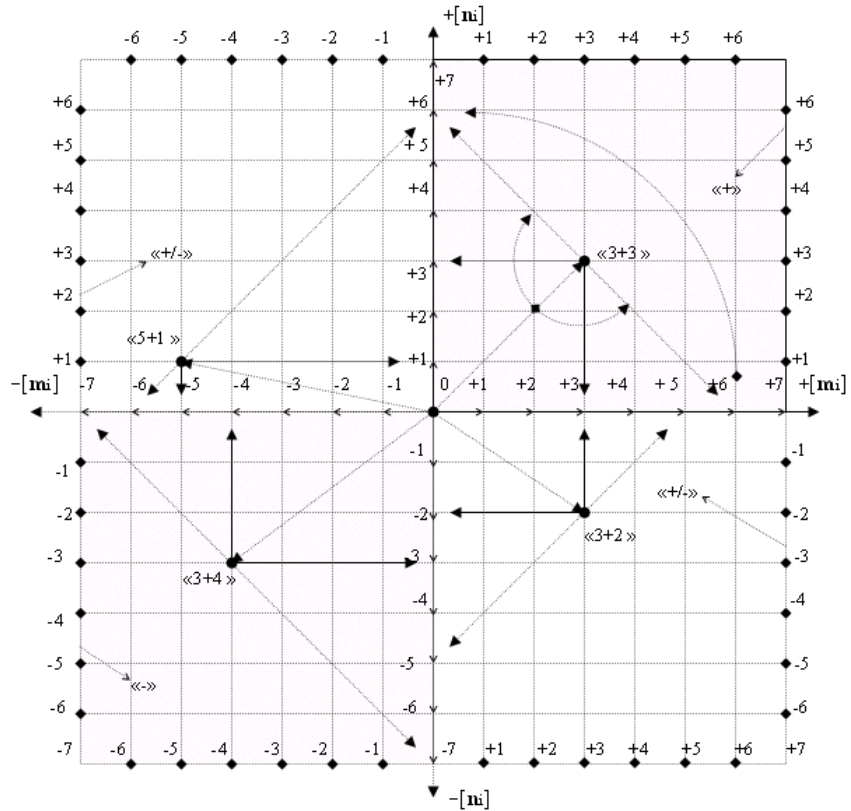
С одной стороны, если исходить из того, что в арифметических устройствах аргументы, например, $[n_i]$ и $[m_i]$ – это логические аналоговые сигналы, то имеет смысл формализовать их, а точнее изображать в виде направленных векторов (9). При этом длина векторов



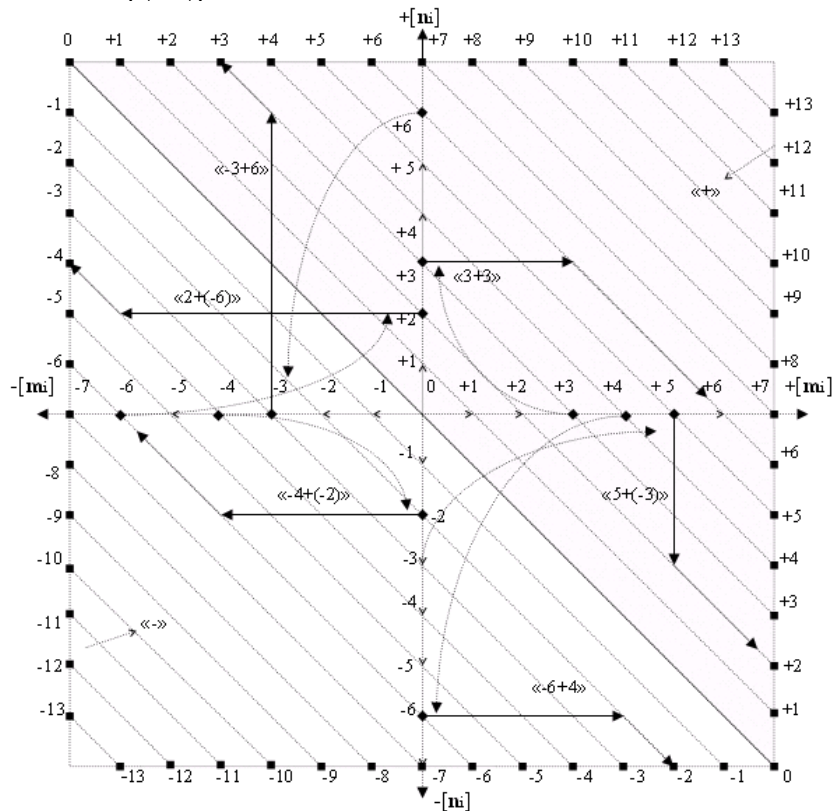
должна соответствовать информационной составляющей аргументов $[n_i]$ и $[m_i]$, которые имеют свои динамические диапазоны возможных независимых изменений (9). С другой стороны, аргументы, как аналоговые сигналы, в процессе изменения формируют производный вектор, для данной ситуации – это аргументы суммы $\pm[S_i]$, которые предварительно также изобразим в виде векторов:



Далее, для понимания динамики изменения одного аргумента аналоговых сигналов в интервале другого аргумента, который функционально с ним связан, существует Декартова система координат:

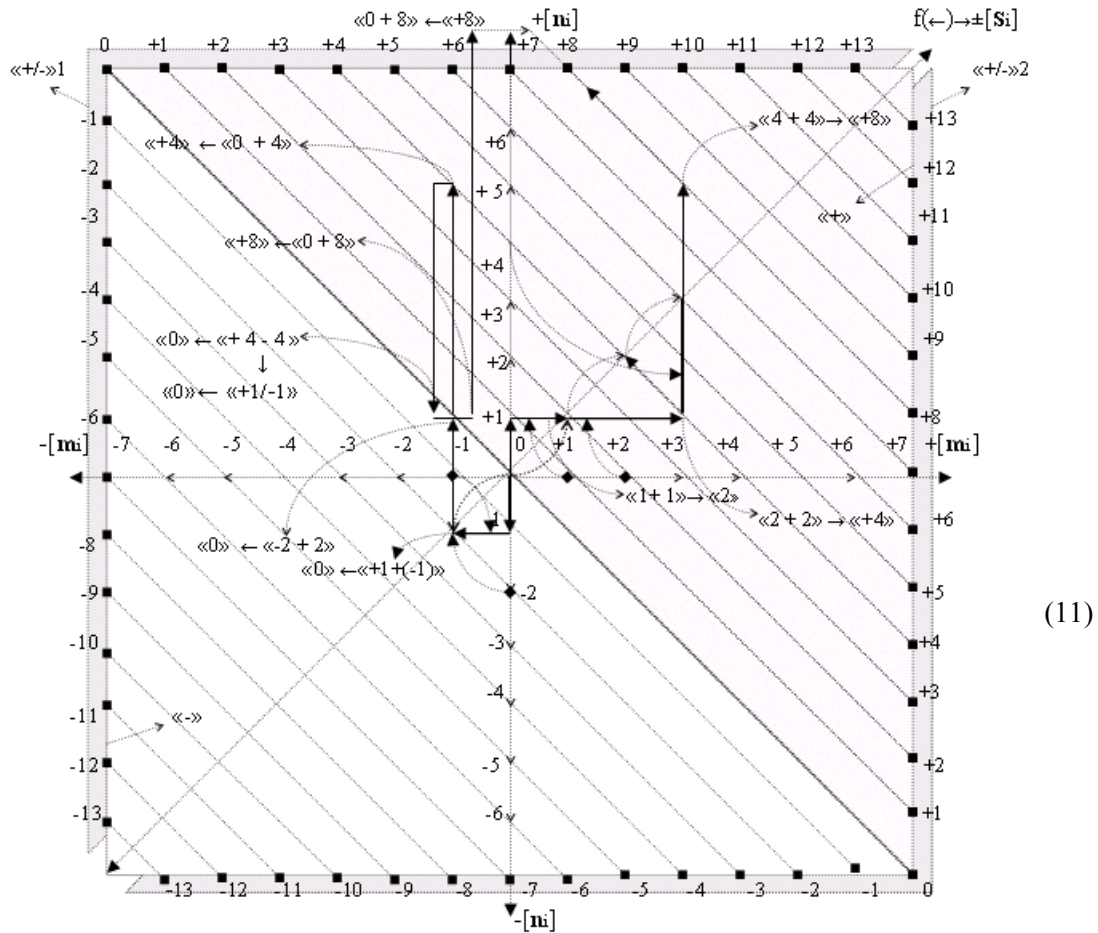


Анализ смыслового содержания процесса суммирования аргументов слагаемых в Декартовой системе координат показывает, что вектор результирующей суммы только косвенным образом указывает на его информационный параметр. Поэтому для решения поставленной задачи скорректируем Декартову систему координат применительно к процессу суммирования и вычитания аргументов слагаемых $[ni]$ и $[mi]$ и изобразим ее в виде графоаналитической структуры:



(10)

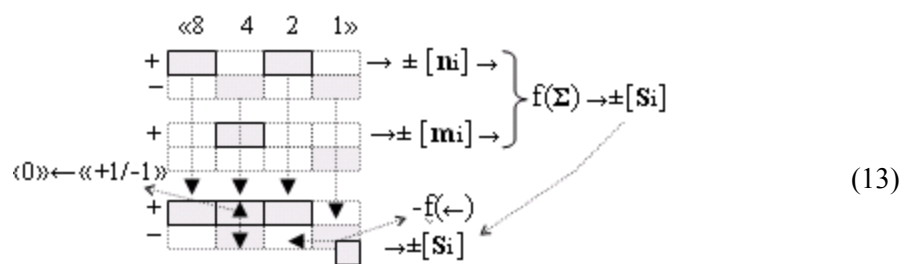
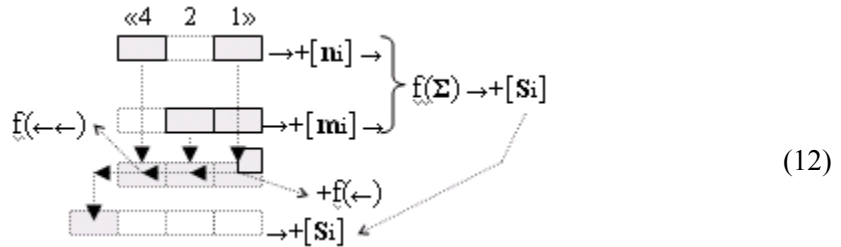
Из анализа графоаналитической структуры (10) следует, что она представлена в виде взаимно-ортогональных эквипотенциальных уровней слагаемых аргументов $[n_i]$ и $[m_i]$. При этом эквипотенциальные уровни структуры (10) аргументов $[n_i]$ и $[m_i]$ функционально разделены на три знаковые области «+», «-» и «+/-». Однако если выбрать в системе (10) координаты на пересечении эквипотенциальных уровней слагаемых аргументов $[n_i]$ и $[m_i]$ и провести диагональ, то эта диагональ точно укажет арифметическую сумму двух векторов $[n_i]$ и $[m_i]$ без учета их знаков, т.е. независимо от расположения их в знаковых системах «+», «-» и «+/-». На основании этих соображений Декартова система координат может быть преобразована в арифметическую систему эквипотенциальных уровней, а для этого введем арифметические уровни результирующей суммы, которые будут проходить через дискретные значения направленных векторов $[n_i]$ и $[m_i]$. В результате формируем арифметическую систему:



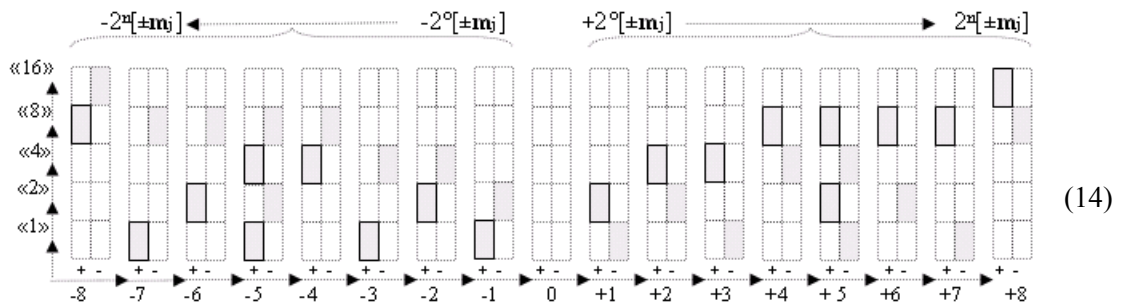
Из анализа арифметических действий графоаналитической структуры (11) следует, что, с одной стороны, процесс суммирования и вычитания аргументов слагаемых $[n_i]$ и $[m_i]$ на качественно ином уровне является векторной структурой. При этом логика формирования такой структуры не зависит от знака слагаемого и заключается в последовательном совмещении начала последующего вектора с концом предыдущего вектора, ориентация эквивалентна исходным знаковым осям соответствующего слагаемого $[n_i]$ и $[m_i]$. Следовательно, суммирование и вычитание аргументов слагаемых по существу является одним технологическим процессом последовательного переноса аргументов слагаемых в арифметическом пространстве. С другой стороны, в графоаналитической структуре (11) все арифметическое пространство разделено нулевым аргументом эквипотенциального уровня на две функциональные области – положительную «+» и условно-отрицательную «-» результирующую сумму. При этом эквипотенциальные уровни, которые проведены через одноименные аргументы двух слагаемых $\pm[n_i]$ и $\pm[m_i]$, являются аргументами суммы $\pm[S_i]$.

Из анализа приведенных в графоаналитической структуре (11) конкретных аргументов, которые эквивалентны десятичной системе счисления $\pm f(10)$, следует, что процесс суммиро-

вания аргументов любой другой системы счисления будет аналогичен, поскольку в арифметической системе координат (11) была формализована как форма информации, так и сама информация. В качестве адекватного соответствия синтезированной арифметической системы координат (10) процессу суммирования аргументов различных систем счисления сформируем графоаналитическую структуру суммирования аргументов слагаемых $[n_i]$ и $[m_i]$ двоичной системы счисления $f(2^n)$ и позиционно-знаковой системы счисления $f(+/-)$:



Из анализа графоаналитических структур (12) и (13) суммирования аргументов в двух системах счисления $f(2^n)$ и $f(+/-)$ следует, что общей операцией у них является перенос двойного аргумента $f(<)$. Если процессы суммирования аргументов (12) и (13) перенести в арифметическое пространство (14), то в нем может быть проведен вектор $\pm f(<)$ локальных переносов двойных аргументов, который функционально связан с ортогональными осями аргументов слагаемых $\pm[n_i]$ и $\pm[m_i]$:

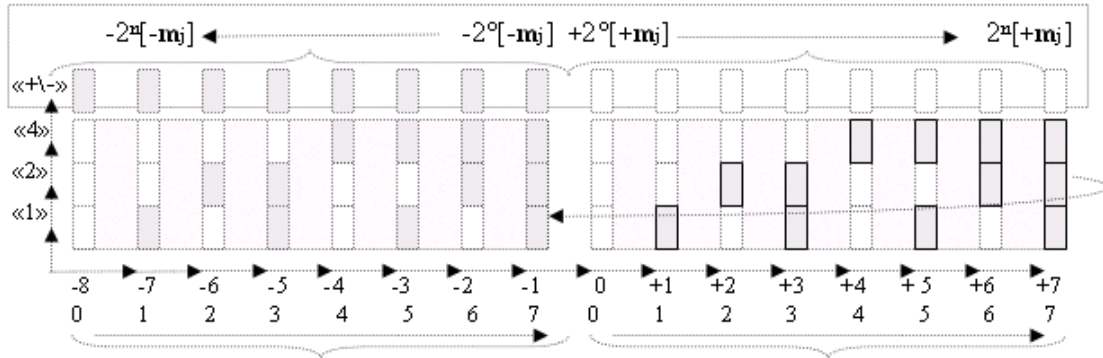


При этом, с одной стороны, вектор $\pm f(<)$ локальных переносов двойных аргументов эквивалентен результирующей сумме $\pm[S_i]$, с другой – вектор $\pm f(<)$ локальных переносов двойных аргументов по существу делит арифметическое пространство на две функциональные структуры $\langle +/ - \rangle_1$ и $\langle +/ - \rangle_2$, смысловое содержание которых заключается в том, что при многократном переходе из одной функциональной структуры $\langle +/ - \rangle_1$ в другую $\langle +/ - \rangle_2$ формируются сквозные переносы $f(\leftarrow\leftarrow)$.

Из сравнительного анализа векторных структур двух процессов суммирования (12) и (13), аргументы слагаемых которых представлены в двоичной системе счисления $f(2^n)$ и позиционно-знаковой системе счисления $f(+/-)$, можно сделать вывод, что арифметические действия в позиционно-знаковой системе счисления $f(+/-)$ полностью соответствуют всему арифметическому пространству. Следовательно, система счисления соответствует объективной реальности, поскольку арифметические векторы могут быть расположены в арифметическом пространстве без ограничения, а арифметические действия двоичной системы $f(2^n)$ относительны. Другими словами, только арифметические действия в позиционно-знаковой системе счисления $f(+/-)$ адекватно соответствуют объективно существующему арифметическому пространству, что, в конечном счете, позволяет повысить быстродействие процесса суммирования аргументов слагаемых.

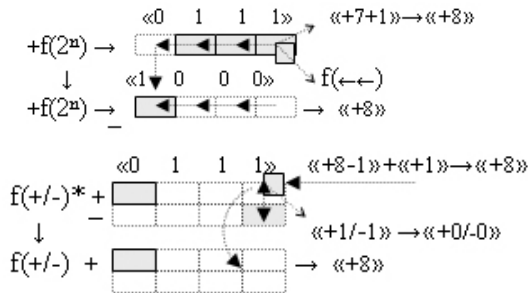
7. Графоаналитический метод анализа логики формирования двоичной системы $f(2^n)$, посредством которой выполняют арифметические преобразования аргументов

В основе арифметических процессов преобразования аргументов в арифметических устройствах лежит двоичная система счисления $f(2^n)$ в «дополнительном коде»:

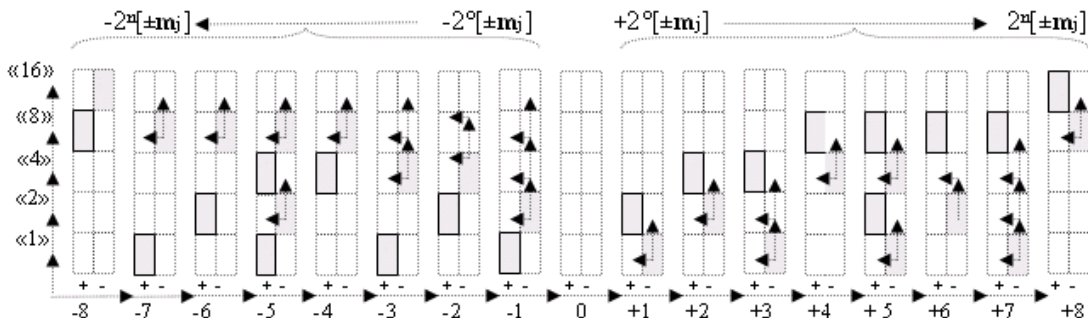


В данном случае дело имеем с частной реализацией позиционно-знаковой системы счисления $f(+/-)$, которая является лучшей реализацией системы счисления в арифметических устройствах.

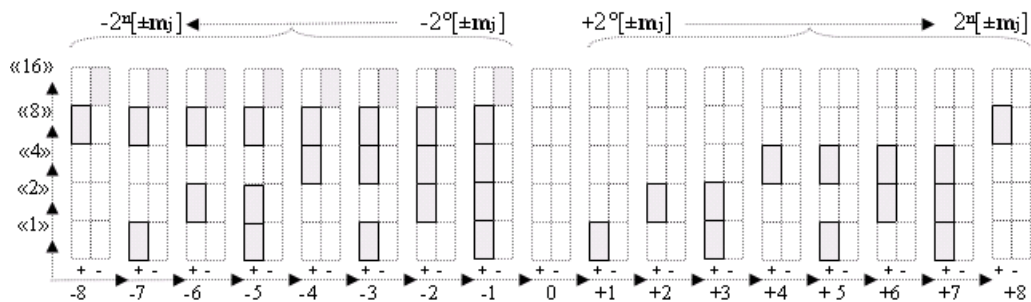
Следует особо отметить, что, с одной стороны, позиционно-знаковая система счисления $f(+/-)$ соответствует категории объективной реальности, поскольку представляет собой устойчивые системы противоположностей в виде положительных и условно-отрицательных аргументов электрических сигналов. С другой стороны, позиционно-знаковая система счисления $f(+/-)$ в виде графоаналитической структуры минимизирована, а точнее оптимизирована не относительно, а применительно к выполнению арифметических действий суммирования и вычитания, поскольку все структуры аргументов находятся в состоянии предварительно введенных переносов, что является принципиально важным качеством структур аргументов системы, которая исключает сквозные переносы $f(\leftarrow\leftarrow)$ при суммировании и вычитании аргументов, например, для положительного аргумента $\langle\langle +7 \rangle\rangle \rightarrow f(2^n)$ в двоичной системе счисления и $\langle\langle +7 \rangle\rangle \rightarrow f(+/-)$ в позиционно-знаковой системе счисления:



Из сопоставительного анализа графоаналитических структур суммирования аргументов слагаемых следует, что «дополнительный код» может быть обоснован путем арифметического преобразования позиционно-знаковой системы $f(+/-)$. При этом формирование производной структуры в виде «дополнительного кода» осуществляется с помощью арифметической аксиомы $\langle\langle -1 \rangle\rangle \rightarrow \langle\langle -2+1 \rangle\rangle$ и $\langle\langle +1 \rangle\rangle \rightarrow \langle\langle +(-2) \rangle\rangle$ ко всем аргументам:



Результатом такого применения аксиомы является двоичной системы счисления $f(2^n)$ в «дополнительном коде», структуры аргументов, которые лишены предварительно введенных потенциально возможных локальных и сквозных переносов, при суммировании которых в структуре «дополнительного кода» локальные и сквозные переносы вновь необходимо вводить:



Из анализа полученных результатов следует вывод, что система счисления в «дополнительном коде» является частным случаем позиционно-знаковой системы счисления $f(+/-)$.

8. Выводы

1. Позиционно-знаковая система счисления $f(+/-)$ соответствует объективной реальности и является минимизированной структурой для выполнения операции суммирования. 2. Повышение быстродействия арифметических устройств сумматоров и умножителей непосредственно связано с применением аргументов, эквивалентных позиционно-знаковой системе счисления. 3. Сформированное арифметическое пространство позволяет на качественном уровне анализировать различные арифметические процессы преобразования аргументов. 2. На векторном уровне процесс суммирования и вычитания аргументов – это единый технологический процесс формирования результирующего вектора суммы при суммировании и вычитании аргументов слагаемых и включает одну операцию последовательного соединения векторов слагаемых аргументов.

Список литературы: 1. Сигорский В.П. Математический аппарат инженера. К.: Техніка 1975. 768 с. 2. Карцев М.А., Брик В.А. Вычислительные системы и синхронная арифметика. М.: Радио и связь, 1981. 360 с. 3. Дж. Уэйкерли. Проектирование цифровых устройств. Том 1 и 2. М.: Посмаркет, 2002. 1087 с. 4. Преобразователь двоичного кода в позиционно-знаковый код. А.С. № 1438005 СССР./ Л.П. Петренко. – Оpubл. 1988, БИ № 42. 4 с. 5. Рябенкий В.М., Петренко Л.П. Принципы формирования математической модели сумматора в позиционно-знаковой системе счисления // Проблемы информационных технологий. Сборник № 3 ХНТУ. Херсон, 2007. С.51-57. 6. Рябенкий В.М., Петренко Л.П., Принципы формирования математической модели элементов, схем и устройств вычислительной техники и систем управления // Проблемы информационных технологий. Сборник № 3 ХНТУ. Херсон. 2007. С.57-61. 7. Рябенкий В.М., Петренко Л.П. Параллельный сумматор. Патент № 23363 UA Оpubл. 2007, № 7. 3 с. 8. Коваленко И.И., Рябенкий В.М., Петренко Л.П. Методика формирования арифметических процессов в позиционно-знаковой системе счисления // Сборник научных трудов НУК. 2007. С.113-122.

Поступила в редколлегию 15.12.2008

Кривуля Геннадий, Федорович, д-р техн. наук, профессор, зав. каф АПВТ ХНУРЭ. Научные интересы: автоматизированное проектирование и техническая диагностика компьютерных систем и сетей. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.

Рябенкий Владимир Михайлович, д-р техн. наук, профессор, зав. кафедрой Теоретической электротехники и электронных систем национального университета кораблестроения. Научные интересы: цифровые и микропроцессорные системы реального времени, компьютерное управление в электроэнергетике. Адрес: Украина, 54001, Николаев, пр. Героев Сталинграда, 9, тел 44 - 16 - 49.

Петренко Лев Петрович, ст. преподаватель кафедры теоретической электротехники и электронных систем национального университета Кораблестроения. Научные интересы: проектирование матричных цифровых систем управления, специализированные процессоры. Адрес: Украина, 54001, Николаев, пр. Героев Сталинграда, 9, тел 44 - 16 - 49.

РЕАЛИЗАЦИЯ МОДЕЛИ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В ОБЛАСТИ СЕРВИСНОГО ОБСЛУЖИВАНИЯ БАНКОМАТОВ

Рассматриваются основные подходы к построению модели системы поддержки принятия решений в специфичной области сервисного обслуживания банкоматов путем использования нейронных сетей. Приводится пример построения нейронной сети для реализации модели заданной системы.

1. Введение

Современный человек живет в мире техники, это касается всех сторон его существования - работы, дома и отдыха. Соответственно производство и продажа техники с каждым годом увеличиваются и в плане количества, и в плане расширения ассортимента. Но любая техника имеет свойство отказывать, что является нормальным. Надежной считается не та техника, которая не ломается, такой просто не бывает, а та, которую можно починить. Поэтому техническое обслуживание, которое в течение длительного времени не рассматривалось в качестве элемента экономической эффективности, сегодня является важнейшей областью, позволяющей эффективно использовать ресурсы предприятия. Хорошо организованная сервисная служба, которая поддерживает высокую степень надежности техники, может оказаться тем ключевым элементом, наличие или отсутствие которого означает успех или неудачу.

2. Постановка задачи

Эффективность сервисной службы во многом зависит от степени автоматизации ее работы.

Цель исследования – раскрыть содержание проблемы управления ресурсами в сфере сервисного обслуживания банкоматов и разработать механизм управления ресурсами предприятия в современных условиях хозяйствования. Такой механизм должен решать задачи координации поставки запасных частей и комплектующих, регулирования отпускных цен на запасные части для дилеров, финансовых расчетов с авторизованными Сервисными Центрами, контроля качества и сроков выполнения ремонтов.

3. Преимущества нейронных сетей для решения рассматриваемой задачи

На практике в сфере сервисного обслуживания банкоматов часто встречаются задачи, связанные с наблюдением случайных величин. Для подобных задач не удастся построить детерминированные модели, поэтому применяется принципиально иной, вероятностный подход. Параметры вероятностных моделей - это распределения случайных величин, их средние значения, дисперсии и т.д. Как правило, эти параметры изначально неизвестны, а для их оценки используются статистические методы, применяемые к выборкам наблюдаемых значений (историческим данным) [5].

К сожалению, классические методики оказываются малоэффективными во многих практических задачах. Это связано с тем, что невозможно достаточно полно описать реальность с помощью небольшого числа параметров модели, либо расчет модели требует слишком много времени и вычислительных ресурсов. В частности, рассмотрим одну из основных задач, решаемую в области сервисного обслуживания банкоматов - оптимальное распределение ресурсов и инвестиций, вкладываемых в развитие такого бизнеса.

В такой задаче ни одна из функций не известна точно - известны лишь приблизительные или ожидаемые значения используемых ресурсов (как материальных, так и человеческих). А однозначно определить размер ожидаемой прибыли в результате проведения ремонтных работ тем более тяжело. Для того чтобы избавиться от неопределенности, мы вынуждены зафиксировать функции, теряя при этом в точности описание задачи.

Детерминированный алгоритм для поиска оптимального решения (симплекс-метод) применим только в том случае, если все данные функции линейны. В реальных задачах оценки используемых ресурсов это условие не выполняется. Хотя данные функции можно аппроксимировать линейными, решение в этом случае будет далеким от оптимального.

Если одна из функций нелинейная, то симплекс-метод неприменим, и остается два традиционных пути решения этой задачи.

Первый путь – использовать метод градиентного спуска для поиска максимума прибыли. В данном случае область определения функции прибыли имеет сложную форму, а сама функция – несколько локальных максимумов, поэтому градиентный метод может привести к неоптимальному решению. Второй путь – провести полный перебор вариантов распределения ресурсов. Если каждая из 10 функций задана в 100 точках, то придется проверить около 10^{20} вариантов, что потребует не менее нескольких месяцев работы современного компьютера.

Отметим также, что статистические методы хорошо развиты только для одномерных случайных величин. Если же мы хотим учитывать несколько взаимосвязанных факторов, то придется обратиться к построению многомерной статистической модели. Однако такие модели либо предполагают гауссовское распределение наблюдений (что не выполняется на практике), либо не обоснованы теоретически [5].

Таким образом, приходим к необходимости применения усовершенствованного математического аппарата.

Применение нейросетевой технологии уместно в случаях, когда формализация процесса решения трудна или вообще невозможна. Они являются очень мощным инструментом моделирования, поскольку нелинейны по своей природе. Как уже было сказано, линейное моделирование долгое время являлось основным в большинстве областей, поскольку для него существует большое число методов оптимизации. Однако для задачи анализа прибыли и распределения ресурсов предположение о линейности исходных данных в подавляющем большинстве случаев оказывается неверным. Кроме того, для нейронных сетей не существует проблемы «проклятия размерности», не позволяющей моделировать линейные зависимости от большого числа переменных.

Нейронная сеть применяется в первую очередь тогда, когда неизвестен точный вид связи между входом и выходом. Достаточно лишь точно знать, что связь между входными и выходными данными существует. При этом сама зависимость будет выведена в процессе обучения нейронной сети [2].

Подводя итог сказанному выше, можно сделать вывод, что по сравнению с традиционными технологиями нейронные сети обладают следующими преимуществами:

- Универсальность. Нейронные сети не зависят от свойств входных данных, для них не существует требования к определенному типу распределения исходных данных либо требования к линейности целевых функций.

- Не существует проблемы «проклятия размерности». Они способны моделировать зависимости в случае большого числа переменных.

- В отличие от статистических исследований не требуют большого объема данных.

- Ускоряют процесс нахождения зависимости за счет одновременной обработки данных всеми нейронами.

Несмотря на то, что нейронные сети способны решить практически любые задачи, во многих случаях их применение не является целесообразным. Для большинства задач более эффективным оказывается применение других математических моделей. Нейронные сети особенно хорошо зарекомендовали себя при решении задач классификации, прогнозирования, кодирования и декодирования информации [3].

4. Элементы построения нейронной сети

В работе [1] было введено понятие юнита (Unit) как минимального обучаемого элемента, способного самостоятельно обрабатывать информацию в гранулированном виде, и кластера (Cluster), как совокупности таких элементов.

Модель юнита, согласно [1], может быть представлена набором множеств

$$M_c = \{X, W_c, H_x, S_c, H_y, BF, y\}, \quad (1)$$

где $X = \{x_0, \dots, x_n\}$ – множество входных параметров – микроситуаций; $W_c = \{w_0, \dots, w_v\}$ – множество регулируемых весов (min); $H_x = \{h_{x1}, \dots, h_{xq}\}$ – множество скрытых входных параметров, соответствующих информационным гранулам на входах; $H_y = \{h_{y1}, \dots, h_{ye}\}$ – множество скрытых выходных параметров, соответствующих информационным гранулам на выходах; $S_c = (s_{c1}, \dots, s_{cr})$ – множество связей скрытых входных и выходных информационных гранул, определяющих цепочку преобразований гранул при активизации этой связи; $BF = (bf_1, \dots, bf_k)$ – множество базисных функций; y – выходной параметр юнита.

Модель кластера, согласно [1], также может быть представлена с помощью (1), где $X=(X_1 \vee X_2 \vee \dots \vee X_M)$ представляет собой совокупность входов юнитов, входящих в кластер; W_N, H_x, H_y – множества весов и скрытых параметров кластера; S_N – множество связей кластера, объединяющее связи юнитов; BF – набор базисных функций активаторных элементов; скалярный выход y заменен на векторный Y – объединенное множество выходных параметров кластера.

Получаем, что юнит является универсальным преобразователем информации (адаптивным аппроксиматором), имеющим n -входов и один выход. Он соответствует в модельном плане биологическому нейрону с его сетью синапсов, через которые организуются связи с другими нейронами. В свою очередь кластер с частичным объединением юнитов можно считать однослойной сетью клеток с входами, параллельно подведенными к каждой клетке (от каждого входа к каждой клетке).

5. Локальные и распределенные сети

Локальная сеть (LN – Local Network) содержит связанные юниты и кластеры. Она рассматривается как самостоятельный модуль обработки образов. Информационная модель локальной сети может быть представлена набором множеств

$$M_{LN} = \{P_X(X, C_X), W_{LN}, S_{LN}, P_Y(Y, C_Y)\}, \quad (2)$$

где X и C_X – вектор образа P_X и связи между его координатами, т.е. множество входных параметров и связи между ними, определяющие входной (преобразуемый сетью) образ P_X ; W_{LN} – множество регулируемых весов связей между юнитами и кластерами, входящими в локальную сеть; S_{LN} – множество связей юнитов и кластеров, входящих в локальную сеть; Y и C_Y – вектор образа P_Y и связи между его координатами, т.е. множество выходных параметров и связей между ними, определяющих выходной (преобразованный сетью) образ P_Y .

Структурно локальная сеть состоит из связанных юнитов и кластеров, а также обучающего, формирующего множество весов связей W_{LN} элементов сети. Наличие разноуровневых компонентов (юнитов и кластеров) в локальной сети позволяет строить достаточно сложные преобразующие структуры с полносвязной и слоистой организацией. Локальные сети с полносвязной организацией используются для ассоциативного запоминания образов. Локальные сети со слоистой организацией предназначены для классификации образов или аппроксимации заданных преобразований образов [2].

Распределенные сети (DN – Distributed Network) имеют кустовую организацию. Они включают несколько локальных сетей (кустов), обрабатывающих образы, между которыми имеют место слабые (одинарные) связи. Так могут реализоваться процессы, определяющие сложное поведение системы. Распределенные сети могут преобразовывать образы замкнуто: образы, проходящие через кусты, возвращаются преобразованные опять на первый куст. Динамический процесс преобразования образов в распределенной сети определяется цепочкой отображений

$$P_{X/t_1} \rightarrow F_{k1}(P_X, P_{Z1}) \rightarrow \dots \rightarrow F(P_X, P_{Zq}) \rightarrow P_{Y/t_2} \rightarrow (OY) \rightarrow P_{S/t_2} \rightarrow P_{X/t_2}, \quad (3)$$

где t_1 и t_2 – моменты времени, соответствующие 1-му и 2-му циклам преобразований; F_{ki} – отображенные образы в i -м кусте; P_S – образ состояния объекта управления (OY); P_X, P_Y, P_Z – входной, выходной и внешний образы распределенной сети.

При этом имеют место слабые связи между кустами, определяемые малыми весами $w_{i,i+1}$.

Такая организация сетей позволяет учесть влияние на процесс управления дополнительных факторов, изменяющих это управление через вводимые в кусты внешние образы P_Z .

6. Структура нейронной сети

Рассмотрим реализацию нейронной сети для системы поддержки принятия решений в области сервисного обслуживания банкоматов. Нейронная сеть должна реализовать нечетко-нейронное отображение входного n -мерного вектора X в выходной m -мерный вектор Y .

Нейронная сеть включает:

- фазификатор (процедура FUZ) - в качестве гранулятора [статья];
- активатор, состоящий из блока ассоциативного вывода и реализующий процедуру взвешенной агрегации (процедура INCL), объединения и памяти весов;
- дефазификатор (процедура DFUZ) - в качестве дегранулятора;
- блок вычисления текущих весов, реализующий процедуру коррекции весов (процедура UPD), и память образов.

Перечисленные процедуры имеют следующее содержание:

$$FUZ = \text{fuz}(x_i, \mu^q(x_i)) \quad (4)$$

– процедура, вычисляющая степень принадлежности переменной x_i к q -й нечеткой грануле этой переменной;

$$INCL = \text{incl}(\ln, w_i, \mu_{x_i}^q; i=1, \dots, I; q=1, \dots, K) \quad (5)$$

– взвешенная агрегация, вычисляющая каждую степень принадлежности переменной y_j к q -й нечеткой грануле этой переменной путем агрегирования степеней принадлежности входных переменных x_i , с дальнейшим умножением результата на вес w_i и объединением; I, K – количество агрегируемых степеней принадлежности и нечетких гранул соответственно; \ln – номер слоя сети, в котором выполняется агрегация;

$$DFUZ = \text{dfuz}(\mu_{y_j}^q, y_j; q=1, \dots, K; j=1, \dots, M) \quad (6)$$

– дефазификация, вычисляющая значение выходного параметра y_j , исходя из степеней принадлежности значения этого параметра к нечетким гранулам его шкалы;

$$UPD = \text{upd}(w_p, \mu_y^*, \mu_y, p) \quad (7)$$

– коррекция веса при обучении; w_p – текущий вес; μ_y^*, μ_y – степени принадлежности правильного и текущего значений выходного параметра y к нечеткой грануле соответственно (правильное значение параметра берется из примера для обучения); p – номер примера; в результате операции находятся веса при обработке $p+1$ примера.

Такой модуль является относительно простым для программной и аппаратной реализации и может обучаться в реальном времени.

7. Модель нейронной сети

Для построения нейронной сети необходимо разработать ее топологию, определить механизм обучения и процедуру тестирования. Кроме того, для обучения нужны входные данные – параметры, описывающие микроситуации [1] предметной области.

Для сферы сервисного обслуживания банкоматов в качестве входных данных используются разработанный автором набор параметров:

- параметры установки банкоматов;
- параметры эксплуатации банкоматов;
- параметры, описывающие состояние узлов и блоков;
- количество зарегистрированных заявок;
- количество просроченных заявок;
- количество невыполненных заявок;
- среднее время в ремонте;
- количество использованных запчастей;
- среднее время выполнения заявки.

На начальной стадии исследования наиболее привлекательным является выбор трех-слойного персептрона и алгоритма обратного распространения в качестве обучающего. Каждый элемент нейронной сети [4] строит взвешенную сумму своих входов с поправкой в

виде слагаемого и затем пропускает эту величину активации через передаточную функцию, и таким образом получается выходное значение этого элемента.

Элементы организованы в послонную топологию с прямой передачей сигнала. Такую сеть легко можно интерпретировать как модель вход-выход, в которой веса и пороговые значения (смещения) являются свободными параметрами модели. Такая сеть может моделировать функцию практически любой степени сложности, причем число слоев и число элементов в каждом слое определяют сложность функции. Определение числа промежуточных слоев и числа элементов в них является вопросом при конструировании, они будут отражены в дальнейших исследованиях. Количество входных и выходных элементов определяется условиями задачи.

Самый известный вариант алгоритма обучения нейронной сети - так называемый алгоритм обратного распространения (back propagation; [2]). В алгоритме обратного распространения вычисляется вектор градиента поверхности ошибок. Этот вектор указывает направление кратчайшего спуска по поверхности из данной точки, поэтому если мы «немного» продвинемся по нему, ошибка уменьшится. Последовательность таких шагов (замедляющаяся по мере приближения ко дну) в конце концов приведет к минимуму того или иного типа. Определенную трудность здесь представляет вопрос о том, какую нужно брать длину шагов [6].

При большой длине шага сходимость будет более быстрой, но имеется опасность перепрыгнуть через решение или уйти в неправильном направлении. Классическим примером такого явления при обучении нейронной сети является ситуация, когда алгоритм очень медленно продвигается по узкому оврагу с крутыми склонами, прыгая с одной его стороны на другую. Напротив, при маленьком шаге, вероятно, будет схвачено верное направление, однако при этом потребуются очень много итераций. На практике величина шага берется пропорциональной крутизне склона (так что алгоритм замедляет ход вблизи минимума) с константой, которая называется скоростью обучения. Правильный выбор скорости обучения зависит от конкретной задачи и обычно осуществляется опытным путем; эта константа может также зависеть от времени, уменьшаясь по мере продвижения алгоритма.

В первом приближении конфигурация разработанной сети следующая:

Общие параметры	Параметры скрытого слоя	Параметры выходного слоя
Количество узлов во входном слое: 13	Коэффициент обучения: 0.25	Коэффициент обучения: 0.01
Число узлов в скрытом слое: 9	Коэффициент инерции: 0.6	Коэффициент инерции: 0
Число узлов в выходном слое: 3	Затухание:0	Затухание:0
Алгоритм обучения: Обратное распространение		
Функция активации: сигмоидальная		

8. Заключение

В результате проведенных исследований моделей и методов, используемых для повышения эффективности, надежности и функциональности создаваемых информационных технологий, был разработан набор входных параметров для исследуемой предметной области. Отметим, что при выборе параметров модели основное внимание уделялось не соответствию расчета выбранных коэффициентов общепринятым методикам анализа ресурсов, а влиянию исходного набора данных на качество обучения сети и как следствие эффективность ее прогнозирования. Проведена сравнительная характеристика методов анализа и оценки распределения ресурсов. Особое внимание было уделено преимуществам и недостаткам нейросетевых методов. Нейронные сети не являются панацеей, во многих случаях более эффективным будет использование традиционных методов статистики. Не-

смотря на это, во многих областях нейронные сети зарекомендовали себя как более рациональное решение и требуются дальнейшие исследования в данной области.

Список литературы: 1. *Станкевич Л.А.* Когнитивные нейробиологические системы управления. Проблемы нейрокибернетики. Материалы XII Международной конференции по нейрокибернетике (Ростов-на-Дону, октябрь 1999), Ростов-на-Дону, 1999. 2. *Haykin S.* (1994). *Neural Networks: A Comprehensive foundation.* Macmillan College Publishing Company Inc. 3. *Widrow B., Rumelhart D.E. and Lehr M.A.* (1994) *Neural networks: Application in industry, business and science, Communications of the ACM* 37 (3) : 93-105 4. *Черноруцкий И.Г.* Методы принятия решений. Л.: ЛПИ, 1989. 5. *Борисов Ю., Кашкаров В., Сорокин С.* Нейросетевые методы обработки информации и средства их программно-аппаратной поддержки / Открытые системы. 1997. №4.

Поступила в редколлегию 11.12.2008

Кузмин Александр Яковлевич, д-р техн. наук, проф. кафедры информатики, начальник инновационно-маркетингового отдела ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел.: 8 (057) 702-15-15, e-mail: kuzy@kture.kharkov.ua.

Головий (Гусарь) Наталья Владимировна, аспирантка кафедры информатики ХНУРЭ. Научные интересы: системный анализ данных. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (057) 702 15 15, e-mail: rica1982@mail.ru.

Ясер Даюб, аспирант кафедры информатики ХНУРЭ. Научные интересы: системный анализ данных. Адрес: Украина, 61166, Харьков, пр.Ленина,14, тел. (057) 702 15 15, e-mail: kuzy@kture.kharkov.ua.

УДК 004.896, 004.932

И.А.ВЕРЕЩАК

МЕТОД НИЗКОУРОВНЕВОЙ ОБРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

Рассматриваются вопросы цифровой обработки графической информации. Описываются разработанные методы фильтрации и сегментации цветных изображений, приводятся результаты их тестирования.

Актуальность. Цифровая обработка графической информации используется во многих областях науки и техники для решения большого спектра задач.

Обработка графической информации способна дать много данных об объектах окружающего мира и единственная проблема, мешающая использовать видеоконтроль, заключается в сложности выбора необходимых методов и этапов преобразования получаемой графической информации.

Постановка задачи

Одна из задач обработки графической информации, требующих решения – это мониторинг пассажиропотока. Вовремя поступающая информация о плотности пассажирского потока способна повысить качество работы систем управления городским пассажирским транспортом.

В общем случае, решая задачу мониторинга пассажиропотока, необходимо:

- произвести предварительную обработку (фильтрацию) изображения,
- перевести графическую информацию из цветовой модели RGB в HSB,
- разложить входное изображение по цветам,
- провести сглаживание изображения,
- определить размеры выделенных областей (площадь),
- определить количество областей, размер которых больше заданного порога.

Методы

При решении данной задачи необходимо разработать или выбрать из существующих такие методы преобразования информации, которые бы позволили выделить на кадре пассажиров и определить их количество. Этот этап как правило называют сегментацией. Зачастую процессу сегментации предшествует процесс фильтрации и сглаживания изображения.

Фильтрация используется как предварительный этап обработки графической информации при выделении объектов на цветном изображении.

Существует большое количество методов фильтрации и сглаживания, к наиболее часто используемым для решения подобных задач относятся:

- метод медианной фильтрации;
- Гаусса;
- SUSAN;
- скользящий фильтр.

Во время исследования и программной реализации методов преобразования графической информации было выявлено несколько недостатков в преобразовании информации, которые были устранены с помощью метода низкоуровневой обработки изображений. Данный метод состоит из таких шагов:

1. Яркость каждого пикселя в круглой маске из 37 пикселей сравнивается с яркостью ядра (центрального пикселя):

$$c(\vec{r}, \vec{r}_0) = \begin{cases} 1, & |I(\vec{r}) - I(\vec{r}_0)| \leq t, \\ 0, & |I(\vec{r}) - I(\vec{r}_0)| \geq t, \end{cases}$$

где \vec{r}_0 – позиция ядра на площади изображения; \vec{r} – позиция каждой точки внутри маски, $I(\vec{r})$ – яркость точки; t – порог разности яркостей; c – результат сравнения.

2. Для каждой точки в середине маски выполняется сравнение яркостей точек после чего производится суммирование (n) всех выходов (c):

$$n(\vec{r}_0) = \sum_{\vec{r}} c(\vec{r}, \vec{r}_0).$$

3. Определяется площадь «близких значений», считается n – количество точек.

4. Сравнивается n с фиксированным порогом g («геометрический порог»), который установлен равным $3n_{\max}/4$, где n_{\max} – максимальное значение, которое может принимать n . Предварительное нахождение границ выполняется по правилу:

$$R(\vec{r}) = \begin{cases} g - n(\vec{r}_0), & n(\vec{r}_0) < g, \\ 0, & n(\vec{r}_0) > g. \end{cases}$$

5. Выявляется шум с помощью g . Если площадь меньше g , то на изображении кроме полезной информации существует импульсный шум.

6. Удаляется импульсный шум. Удаление осуществляется с помощью двумерного медианного фильтра с размером окна 3×3 и центром в точке ядра.

Сравнительная характеристика разработанного метода фильтрации с классическими фильтрами приведена в таблице.

Фильтр	Удаление шума	Сглаживание	Сохранение структуры
Скользящий	–	+	–
Гаусса	+	+	–
Медианный	+	+	–
SUSAN	–	+	+
Разработанный	+	+	+

Проведенные исследования позволили разработать и программно реализовать метод сегментации, основанный на понятии тона.

Данный метод сегментации имеет следующие этапы:

1. Входное изображение представлено в системе RGB, которое необходимо перевести в формат HSB.

2. Необходимо выделить составляющую тона и построить по ней гистограмму $H(h)$, где h – тон. При этом максимуму на гистограмме будет соответствовать тон фона, а второму максимуму $M = H(m)$, где H – тон объекта, который необходимо локализовать. Здесь $H(m_1) = M * k$, $M * k$ – количество точек, достаточное для того, чтобы размер занимаемой ими площади мог быть частью однотонной области объекта.

3. Выбрать диапазон значений тона $[t_1, t_2]$, в который попадают все точки локализуемого объекта. Значения t_1 и t_2 вычисляются по формулам

$$t_1 = \min(m - b_1, m_1), \quad t_2 = \min(m - b_2, m_2),$$

где b_1 и b_2 – максимальный разброс значений тона; m_1 – максимальное значение тона, не превосходящее m .

В результате исследований были установлены приемлемые для большинства изображений параметры: $b_1 = b_2 = 15$, $k = 0,347$.

4. Выделить область изображения, тон точек которой входит в установленный на предыдущем шаге диапазон $[t_1, t_2]$.

Полученное изображение включает области локального шума, которые удаляются с помощью медианного фильтра.

5. Разбить плоскость изображения на прямоугольные сегменты. Все сегменты исходного изображения, которым соответствуют непустые сегменты на изображении, полученном на предыдущем этапе, остаются без изменений; остальные сегменты удаляются.

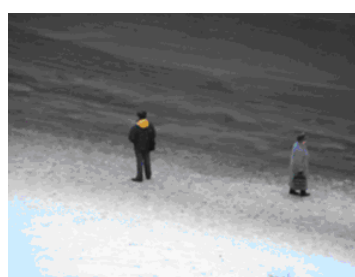
Полученная область является наиболее вероятным местом появления локализуемого объекта.

Далее выделяются контуры полученного изображения, которые включают в себя непосредственно сам объект и незначительные помехи. Данная задача решается любым известным методом оконтуривания.

Пошаговое преобразование графической информации последовательностью методов в которую были включены и разработанные, представлено на рисунке.



Исходный кадр видеоряда



Сглаживание + подавление шума



Сегментация (разложение по цветам)



Сглаживание



Выделение контуров



Выделение областей

Выводы

Проведен анализ методов обработки графической информации, определены и устранены недостатки метода низкоуровневой обработки графической информации. Также разработан метод преобразования графической информации, который позволяет проводить качественное дальнейшее преобразование информации. Данный метод позволяет выделить контуры путем одновременного сглаживания и удаления шума с сохранением структуры графической информации.

Программно реализованы разработанные методы и использованы при построении алгоритма преобразования информации, который может использоваться для решения задачи мониторинга пассажиропотока для диспетчерского управления городским пассажирским транспортом.

Список литературы: 1. Дэвид А. Форсайт, Джин Понс. Компьютерное зрение. Современный подход / Вильямс. 2004. 928 с. 2. Липаев В.В. Системное проектирование сложных программных средств для информационных систем. М.: Синтез. 2002. 268 с. 3. Сергиенко А.Б. Цифровая обработка сигналов. СПб.: Питер. 2003. 608 с.

Поступила в редколлегию 02.12.2008

Верещак Игорь Александрович, заместитель директора Государственного предприятия “Научно-исследовательский институт радиотехнических измерений”. Адрес: Украина, 61054, Харьков, ул. Ак. Павлова, 271, тел. 738-41-12.

РЕФЕРАТИ

УДК 044.03; 681.518:061

Підхід до проектування сховищ даних у інформаційних системах / М.В Євланов, О.В. Сліпченко, В.А. Нікітюк // АСУ та прилади автоматики. 2008. Вип. 145. С. 4-7.

Проаналізовано основні проблеми, що виникають в процесі розробки сховищ даних. Виділено головні підходи до моделювання сховищ даних. Розглянуто підхід, який дозволяє автоматизувати процес проектування схеми сховища даних. Запропоновано метод синтезу схеми сховища даних типу «зірка» виходячи з особливостей представлення схеми джерела даних.

Лл. 4. Бібліогр.: 3 назви.

UDC 044.03; 681.518:061

Approach to developing of data store in information systems / M.V. Ievlanov, E.V. Slipchenko, V.A. Nikituk // Management Information System and Devises. 2008. N 145. P. 4-7.

The basic problems, which is in the process of developing of data stores, was analyzed. Approaches to modeling of data stores is selected. Proposed an approach to automation of process of developing of data store scheme. Proposed method of synthesis for data store scheme type "star", which based on an features of describing of database scheme.

Fig. 4. Ref.: 3 items.

УДК 004.732

Дослідження методів розв'язання задач інжиніринга трафіка в мережах наступного покоління / Г.В. Бабіч, М.В. Кудіна, І.В. Ємельянов // АСУ та прилади автоматики. 2008. Вип. 145. С. 8-13.

Розглянуто процедури інжиніринга трафіка в мережах наступного покоління та засоби забезпечення їх ефективності. Проаналізовано модель інжиніринга трафіка на базі сполучення технології MPLS та протоколів маршрутизації за алгоритмом стану зв'язків з метою виявлення її вузьких місць. Для усунення вказаних вузьких місць запропоновано використання системи AntNet як додаткової структури даних. Сформульовано мету та задачі подальших досліджень для практичної реалізації розширення технології MPLS на базі системи AntNet.

Лл. 1. Бібліогр.: 5 назв.

UDC 004.732

Investigation of traffic engineering methods in the Next Generation Networks / G.V. Babich, M.V. Kudina, I.V. Emelyanov // Management Information System and Devises. 2008. N 145. P. 8-13.

The procedures of Traffic Engineering in NGNs and means of providing of their efficiency have been considered. The Traffic Engineering model based-on a combination of MPLS technology and link-state routing protocols has been analyzed with purpose of identification of its bottlenecks. The usage of AntNet system as an additional data structure has been proposed for indicated bottlenecks elimination. The goal and tasks for further research of practical implementation of enhancement of MPLS technology on the basis of AntNet system have been formulated.

Fig. 1. Ref.: 5 items.

УДК 681.326:519.613

Формальна семантика складних операторів лінійної темпоральної логіки / С.О. Зайченко, В.І. Хаханов // АСУ та прилади автоматики. 2008. Вип. 145. С. 14-28.

Розглянута проблема відсутності чіткості інтерпретації складних операторів лінійної темпоральної логіки при застосуванні до аналізу цифрових систем як частини динамічних методів верифікації. Введені формальні визначення та специфікації ключових обчислювальних процедур робочого циклу верифікації, що орієнтуються на максимальну швидкодію аналізу.

Табл. 2. Лл. 2. Бібліогр.: 8 назв.

UDC 681.326:519.613

Formal semantics of complex linear temporal logic operators / S. O. Zaychenko, V.I. Hahanov // Management Information System and Devises. 2008. N 145. P. 14-28.

This paper considers a problem of interpretation ambiguity for complex linear temporal logic operators applied to the verification of digital systems within dynamic verification methods. Paper introduces formal definitions and specifications for key computational procedures of assertion verification cycle, which target maximum possible analysis performance.

Tab. 2. Fig. 2. Ref.: 8 items.

УДК 517.958:536.71

Чисельне моделювання дифузійних процесів у системах з масивами заглиблених мікроелектродів у формі диску / О.І. Олійник // АСУ та прилади автоматики. 2008. Вип. 145. С.29-39.

Розглянута математична модель для дослідження дифузійних процесів у електрохімічній системі з масивом дискових мікроелектродів, розташованих у заглибленнях. Застосування запропонованого квазіконформного відображення для чисельного розв'язання цієї складної задачі дозволило отримати чисельні результати з необхідною точністю та забезпечило високу швидкість обчислень.

Л. 8. Бібліогр.: 17 назв.

UDK 517.958:536.71

Numerical simulation of diffusion processes at recessed microdisc electrode array systems / A.I. Oleinick // Management Information System and Devises. 2008. N 145. P.29-39.

The mathematical model of diffusion processes in an electrochemical system with a recessed microdisc electrode array was considered. A quasiconformal mapping was derived for the numerical solution of this complex problem. The application of the quasiconformal mapping ensures accuracy of the simulation results as well as speed of the simulation process.

Fig. 8. Ref.: 17 items.

УДК 681.326:519.613

Технології убудованого ремонту компонентів System-in-Package / Є. І. Литвинова // АСУ та прилади автоматики. 2008. Вип. 145. С. 40-48.

Аналіз сучасних технологій убудованого сервісного обслуговування функціональностей цифрової системи у пакеті (SiP) показав «вузькі місця» існуючих стратегій тестування та відновлення працездатності SiP. Визначено напрям подальших досліджень – розробка технологічного підходу до убудованого відновлення працездатності цифрової системи у пакеті в цілях скорочення часу та підвищення надійності відновлення.

Л. 7. Бібліогр.: 31 назва.

UDC 681.326:519.613

Embedded technologies of SiP components repair / E.I. Livinova // Management Information System and Devises. 2008. N 145. P.40-48.

This paper suggests analysis of the modern technologies of embedded F-IP Infrastructure for SiP. The bottle necks of SiP test and repair strategies are shown. The areas of further research is determined, it is development of a technological approach to the embedded repair of digital SiP to decrease repair time and to improve its reliability.

Fig. 7. Ref.: 31 items.

УДК 681.5:69:621:039

Про один метод розв'язання задачі оптимізації планових режимів транспорту і розподілення природного газу у міських газорозподільчих мережах / А.Д. Тев'яшев, Д.О. Золотарьов // АСУ та прилади автоматики. 2008. Вип. 145. С.48-56.

Розглянута математична модель і метод оптимізації роботи міських багаторівневих газорозподільчих мереж (МГРС). Суть методу полягає в мінімізації надлишкових тисків на входах споживачів кожного рівня МГРС. При цьому враховуються границі зміни як тисків на входах споживачів, так і витрат на виходах кожного з регуляторів тиску на кожному з рівнів мережі. Розглянутий алгоритм проілюстрований на прикладі реально існуючої МГРС міста.

Л. 2. Бібліогр.: 9 назв.

UDC 681.5:69:621:039

About one method which solves optimization problems of the scheduled modes of transport and distribution of natural gas in city gas distribution networks / A.D. Tavjashev, D.A. Zolotariov // Management Information System and Devises. 2008. N 145. P.48-56.

In article the mathematical model and a method of optimization of work city multilevel gasdistributive networks (MGDN) is considered. The essence of a method consists in minimization of superfluous pressure on inputs of consumers of each level MGDN. Thus borders of change as pressure upon inputs of consumers, and charges on outputs of each of regulators of pressure on each of levels of a network are considered. The considered algorithm is illustrated on an example really existing cities MGDN.

Fig. 2. Ref.: 9 items.

УДК 519.85

Модифікація точного методу рішення задачі розміщення прямокутних об'єктів/ І.А. Чуб, М.В. Новожилова // АСУ та прилади автоматики. 2008. Вип. 145. С.57-63.

Проведений аналіз структури області припустимих рішень задачі розміщення прямокутників, виділені додаткові особливості, на основі яких розроблена модифікація точного методу розв'язання задачі. У рамках запропонованої модифікації здійснений обхід дерева рішень, що генерує покроковий перехід із початкової точки поза областю припустимих рішень в граничну точку області (вершину) з наступним усіченим перебором вершин області.

Табл. 1. Іл. 2. Бібліогр.: 8 назв.

UDC 519.81

Designing business processes for enterprise in services sphere/ I. A. Chub, M.V. Novozhilova // Management Information System and Devises. 2008. N 145. P.57-63.

The analysis feasible region structure for rectangular placement problem has been performed. The additional properties of feasible region as the base for solution method development have been investigated. The modification allows organizing step-by-step process from unfeasible initial point to the optimal vertex of feasible region.

Tab. 1. Fig. 2. Ref.: 8 items.

УДК 519.677

Методика кластеризації зображень для їх компресії на основі компонентного аналізу / С.А. Таянов, В.А. Таянов // АСУ та прилади автоматики. 2008. Вип. 145. С.63-68.

Запропоновані нові методики кластеризації зображень для їх подальшого адаптивного стиску на основі ПКЛ. Проведене дослідження ефективності стискання зображень на основі використання двох різних методів кластеризації та адаптивної методики з використанням ПКЛ. Дослідження показали, що запропонований алгоритм дозволяє отримати приблизно однакове задане значення SNR для кожного кластера при їх відновленні та досягти більшого коефіцієнта компресії для заданого SNR, ніж традиційні алгоритми. Ефективність наведеного алгоритму зростає при застосуванні його для компресії серій зображень.

Іл. 4. Бібліогр.: 9 назв.

UDC 519.677

Methods of image clustering for image compression on the basis of component analysis / V.A. Tayanov, S.A. Tayanov // Management Information System and Devises. 2008. N 145. P.63-68.

In this paper the new methods both of the image clustering and their adaptive compression on the basis of KLT have been proposed. The effectiveness investigation of the image compression on the basis of both two different clustering methods and adaptive KLT method has been carried out. The research has showed that proposed approach gives the possibility to obtain the approximately equal given SNR value for every cluster using image reconstruction procedure. Also it gives the possibility to reach the larger value of the compression rate for the given SNR than traditional algorithms. The effectiveness of the proposed algorithm grows when one uses it for the image series compression.

Fig. 4. Ref.: 9 items.

УДК 004.652.4+004.827

Технологія ETL в інтеграції даних туристичного бізнесу / Н.Б. Шаховська, Д.І. Угрин // АСУ та прилади автоматики. 2008. Вип. 145. С.68-73.

Системи інтеграції даних і обміну даними традиційно призначаються для підтримки багатьох інших осмислених служб в системах просторів даних. Особливість статті полягає у тому, що в системах інтеграції даних потрібна семантична інтеграція до того, як можуть бути забезпечені які-небудь інші послуги. Тому в статті уведений алгоритм консолідації даних.

Іл. 5. Бібліогр.: 6 назв.

UDC 004.652.4+004.827

ETL integration technology of dataspace in tourism sphere / N.B.Shakhovska, D.I.Ugrin // Management Information System and Devises. 2008. N 145. P.68-73.

Systems of integration of the data and data exchange are traditionally intended for support of many other intelligent services in systems of spaces of the data. Feature of clause will be, that in systems of integration the data semantic integration is necessary before any other services can be provided. Therefore in clause the algorithm of consolidation the data is entered.

Fig. 5. Ref.: 6 items.

УДК 615.85

Метод оцінки альтернативних варіантів функціонального зонування території міста / Н.Г. Долгова, М.В. Новожилова, О.І. Синельникова // АСУ та прилади автоматики. 2008. Вип. 145. С.73-79.

Запропоновано метод оцінки альтернативних варіантів розвитку територій міста, котрий дозволяє провести комплексний аналіз населеного пункту, визначити ресурсний потенціал та виділити напрямки його стратегічного планування. Метод дасть змогу врахувати велику кількість значущих чинників та їх комплексний вплив на розвиток міста.

Табл.2. Іл.3. Бібліогр.: 9 назв.

UDC 615.85

Method of alternative variants of functional zoning of a city area /N.G. Dolgova, M.V. Novogilova, O.I. Sinelnikova // Management Information System and Devises. 2008. N 145. P.73-79.

For planning strategy of cities development, as a rule, corresponds the program of development and a general layout. For this the following features are appropriate: the integrated approach caused by basic necessity to consider deep interrelation of subsystems of a city as a whole; high indefiniteness of the initial data for simulation of analyzed system and environment. Therefore at the given stage of planning it is necessary to have methods and models for the analysis of general development tendencies of system at level of the general, aggregate indicators. Use of the offered estimation method of alternative variants of functional zoning of a city area allows to consider a quantity of significant factors and their complex influence on the city development as a difficult functional system. On the basis of estimation results recommendations can be done for municipal authority and other entities of investment activity for effective use of land resources.

Tab. 2. Fig. 3. Ref.: 9 items.

УДК 004.713

Аналіз швидкодії буферних асинхронних багатоступеневих мереж з довільним числом пріоритетних модулів пам'яті / Євграфов // АСУ та прилади автоматики. 2008. Вип. 145. С.80-86.

Запропонована математична модель асинхронної багатоступеневої комутуючої мережі, яка використовується для знаходження пропускної здатності та часу відповіді мережі. Модель розширюється для випадку довільної кількості пріоритетних модулів пам'яті та випадково розподіленої довжини пакетів. Вивчається вплив ємності буфера на пропускну здатність, час відповіді та загальну швидкодію мережі.

Іл. 4. Бібліогр.: 5 назв.

UDC

Analytic model for performance analysis of buffered asynchronous multistage interconnection networks under non-uniform traffic // Management Information System and Devises. 2008. N 145. P.80-86.

The analytic model for multistage interconnection network has been proposed in this paper. The model is used for evaluation of throughput and delay for asynchronous MIN in case of non-uniform traffic with arbitrary number of hot spots and randomly distributed packet's length. Considered influence of buffer's length on throughput, delay and system power of a MIN.

Fig. 4. Ref.: 5 items.

УДК 681.325.53

Спосіб підвищення швидкодії перетворювачів кодів на лічильниках / М.Я. Какурін, С.М. Коваленко, Ю.В. Лопухін, Г.М.Макаренко // АСУ та прилади автоматики. 2008. Вип. 145. С.86-95.

Запропоновано спосіб підвищення швидкодії перетворювачів кодів на базі лічильників, що використовують метод накопичення еквівалентів, за рахунок паралельного використання кроків перетворення. Розглянуто структуру та функціонування швидкодійного трьохкрокового перетворювача кодів та метод обчислення максимальної кількості кроків перетворення.

Табл. 6. Іл. 3. Бібліогр.: 5 назв.

UDC 681.325.53

The way for rising of the speed of code convertors based on a counters /N. Ya Kakyrin, S.N.Kovalenko, Yu. V. Lopukhin, A.N.Makarenko // Management Information System and Devises. 2008. N 145. P.86-95.

The way for rising of the speed of code convertor based on a counters is offered by the parallel using of a conversions steps/ The structure and functioning of the three-steps code convertor and the method for calculation of the maximal quantity of conversions steps are considered.

Tab. 6. Fig. 3 Ref.: 5 items.

УДК 004.93.1 : 621.314

Перспективи застосування автоматизованої системи розпізнавання та ідентифікації користувачів комп'ютера, описаної на основі дискретної моделі / В.М. Заяць // АСУ та прилади автоматики. 2008. Вип. 145. С.96-103.

На основі дискретної моделі описано системний підхід до побудови автоматизованої системи розпізнавання користувачів комп'ютера у випадку, коли дискретна модель рекурентно зв'язує часові затримки при введенні інформації з клавіатури комп'ютера у реальному діапазоні часу. Доцільність розробленого підходу проілюстровано при реалізації автоматизованої процедури ідентифікації користувачів комп'ютера на основі детермінованого та імовірнісного методів та відзначено перспективні напрями застосування створеної системи.

Л. 1. Бібліогр.: 24 назви.

UDC 004.93.01: 621.314

Prospects of application of the automated system of recognition and authentication of users of computer, described on the basis of discrete model / V.M. Zayats // Management Information System and Devises. 2008. N 145. P.96-103.

On the basis of discrete model systems approach is described to construction of the automated system of recognition of users of computer in the case when the discrete model of recurrent links the temporal delays at entered to information from the keyboard of computer in the real range of time. Expedience of the developed approach is illustrated during realization of the automated procedure of authentication of users of computer on the basis of determined and probabilistic methods and perspective directions of application of the created system are marked.

Fig. 1. Ref.: 24 items.

УДК 518.81

Системологічний аналіз проблеми прийняття рішень в умовах багатокритеріальності та невизначеності / Е.В. Лисенко, В.П. Пономаренко, В.П. Пісклакова // АСУ та прилади автоматики. 2008. Вип. 145. С. 104-109.

Проаналізовані особливості проблем прийняття рішень в умовах багатокритеріальності та інтервальної цільової функції. Показано, що багатокритеріальність є додатковим джерелом невизначеності. Розглянуті загальні підходи до вирішення проблеми багатокритеріальної оптимізації в умовах інтервальної невизначеності.

Бібліогр.: 12 назв.

UDC 518.81

System-logical case study of making decision under the conditions of multicriteriaity and uncertainty / E.V. Lysenko, V.P. Ponomarenko, V.P. Pisklakova // Management Information System and Devises. 2008. N 145. P.104-109.

In this article are analyzed the special features of the problems of making decision under the conditions of multicriteriaity and interval objective function. It is shown that the multicriteriaity is the additional source of uncertainty. Common approaches to the solution of the problem of multicriterional optimization under the conditions of interval uncertainty are examined.

Ref.: 12 item.

УДК 004.89

Розширення функцій АСУ: управління знаннями / М.Ю. Вишняк, С.С. Довгань // АСУ та прилади автоматики. 2008. Вип. 145. С.109-116.

Розглянута інформаційна технологія управління знаннями з точки зору актуальності її використання. Запропоноване поліпшене трактування терміну «знання». Знання розглянуте як стан системи самоуправління людини. Сформований та формалізований критерій ступеня освоєння знань. Проаналізовані існуючі схеми управління знаннями, виділені основні процеси, які є спільними та присутні в кожній з схем.

Л. 5. Бібліогр.: 19 назв.

UDC 004.89

Expansion of MIS functions: knowledge management / M.J. Vishnyak, S.S. Dovgan' // Management Information System and Devises. 2008. N 145. P.109-116.

The knowledge management technology is considered from the point of urgency of using it. The improved definition of term «knowledge» was offered. Knowledge is considered from the point of human self-management system. The degree of knowledge mastering was formed and formalized. Existing schemes of knowledge management were analyzed. The main processes that are common and located in every scheme are extracted.

Fig. 5. Ref.: 19 items.

УДК 519.7

Побудова алгоритму розпізнавання відбитків пальців для системи контролю доступу / Омри Карим // АСУ та прилади автоматики. 2008. Вип. 145. С.116-120.

На основі аналізу основних методів біометричної ідентифікації особистості й синтезу методів розпізнавання відбитків пальців отриманий надійний алгоритм розпізнавання й керування системою контролю доступу. Розроблений алгоритм системи контролю доступу підтримує всі необхідні етапи її роботи: режим обробки зображень зі сканера, режим статистичної обробки й навчання, функції керування периферійними пристроями системи.

Л. 6. Бібліогр.: 4 назви.

UDC 519.7

The fingers prints recognition algorithm construction for access monitoring system / Omri Karim // Management Information System and Devises. 2008. N 145. P.116-120.

On the basis of the person biometric identification analysis basic methods and fingers prints recognition methods synthesis the reliable algorithm of recognition and management of the access monitoring system is received. The developed algorithm of the access monitoring system supports all necessary stages of its work: a mode of the scanner images processing, a mode of statistical processing and training, system remote terminal units management function.

Fig. 4. Ref.: 6 items.

УДК 004.7; 004.8; 007.85

Реалізація моделі системи підтримки прийняття рішень в області сервісного обслуговування банкоматів / О.Я. Кузьомін, Н.В.Головій, Я. Даюб// АСУ та прилади автоматики. 2008. Вип. 145. С.134-139.

Розроблено набір вхідних параметрів для предметної області, також проведена порівняльна характеристика методів аналізу та оцінки розподілення ресурсів. Особливу увагу було приділено дослідженню методів, основаних на нейронних мережах.

Табл. 1. Бібліогр.: 5 назв.

UDC 004.7; 004.8; 007.85

The apparatus application of the neuronics nets for realization of the model of make-decision system in the service operation field of automatic cash terminal / O. Ya. Kuzemin, N.V.Goloviy, Ya. Dayub // Management Information System and Devises. 2008. N 145. P.134-139.

During investigations, wich viewed in article, was worked up set of input characteristics for data domain. Also was conducted comparative description of methods of analysis and evaluation of the resourced allocation. Particularly attention devoted for the research of the methods, which based on neuronics nets.

Tab. 1. Ref.: 5 items.

УДК 004.896, 004.932

Метод низькорівневої обробки графічної інформації / І.О. Верещак // АСУ та прилади автоматики. 2008. Вип. 145. С.139-142.

Розглянуті питання цифрової обробки графічної інформації. Описано розроблені методи фільтрації сегментації кольорових зображень, наведено результати їх тестування.

Табл. 1. Л.1. Бібліогр.: 3 назви.

UDC 004.896, 004.932

Method of low-level treatment of graphic information / I.A. Vereszhak // Management Information System and Devises. 2008. N 145. P.139-142.

The questions of digital treatment of graphic information are considered in the article. The developed methods of filtration and segmentation of the coloured images are described, the results of their testing are resulted.

Tab. 1. Fig. 1. Ref.:3 items.

ПРАВИЛА

оформления рукописей для авторов научно-технического сборника

"АСУ и приборы автоматики"

Формат страницы — А4 (210x297мм), поля: сверху, справа, слева, снизу – 30 мм. Редактор: Pagemaker 6.0, 6,5 (можно, но нежелательно Word), гарнитура Times New Roman Sug, кегль – 11 пунктов, межстрочное расстояние — 110 %, табуляция — 5 мм.

Объем рукописи – до 10 с. (языки: русский, украинский, английский). Содержание должно отражать актуальность исследования, постановку задачи, цель, сущность, научные и практические результаты, сравнение с лучшими аналогами, выводы.

Структура рукописи: заголовок, аннотация, текст, литература, реферат на украинском и английском языках, сведения об авторах.

ОБРАЗЕЦ ОФОРМЛЕНИЯ

УДК 519.713

И.О. ФАМИЛИЯ

НАЗВАНИЕ РУКОПИСИ

Аннотация (абзац 5-10 строк, кегль 10) помещается в начале статьи и содержит информацию о результатах описанных исследований.

Основной текст можно разделять на 2 и более подразделов с заголовками, выделенными полужирным шрифтом, пронумерованными арабскими цифрами, как показано в следующей строке.

1. Название раздела

Рисунки и таблицы (черно-белые, контрастные) помещаются в текст после первой ссылки в виде *переносимых объектов* и отдельно нумеруются, при наличии более одного рисунка (таблицы), арабскими цифрами. Рисунок содержит подрисовочную центрированную подпись (текстовая строка, расположенная вне рисунка, кегль 10) под иллюстрацией, как показано на рис. 1.

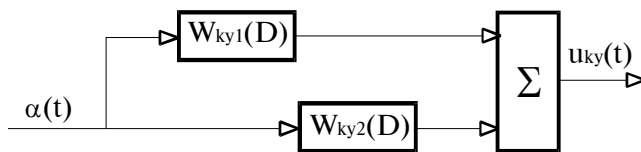


Рисунок 1. Двухзвенная система

Табличный заголовок располагается справа над таблицей, что иллюстрируется табл.1. Редакторы: CorelDraw, Table Editor и др.

Формулы нумеруются при наличии ссылок на них в рукописи. Рекомендуемый кегль формульного набора: обычный (переменная) – 11 пунктов, крупный индекс – 8, мелкий индекс (над- и подиндекс) – 8, крупный символ (основной) – 12, мелкий (индексный) математический символ – 10:

Таблица 1

Шагя	1	2	3	4	5	6
$\Phi_1(1,3)$	1	2	2	4	6	1
$\Phi_1(1,4)$	1	2	2	4	4	8

ра: обычный (переменная) – 11 пунктов, крупный индекс – 8, мелкий индекс (над- и подиндекс) – 8, крупный символ (основной) – 12, мелкий (индексный) математический символ – 10:

$$F_{i+j} = \sum_{i=1}^b F_j^i - \prod_{j=1}^{1+h^2} P_{R_{j+i}} + F^{j-1} + X^{\sum n^k} \quad (1)$$

Формат переменных (желательно не курсивом – без наклона) в тексте и формулах должен быть идентичным. В тексте над- и подиндексы составляют 70 % от кегля, которые рекомендуется опускать (поднимать) на 17 (33) % относительно основной строки.

Список литературы (включает опубликованные источники, на которые имеются ссылки в тексте, заключенные в квадратные скобки) печатается без отступа, кегль 9 пунктов.

Образец окончания текста рукописи (литература, сведения об авторах, реферат) представлен ниже.

Список литературы: 1. *Фамилия И.О.* Название книги. Город: Издательство, 1900. 000 с. 2. *Название сборника / Под ред. И.О. Фамилия.* Город: Издательство, 1900. 000 с. 3. *Фамилия И.О.* Название статьи / / Название журнала. Название серии. 1997. Т. 00, № 00. С. 00-00.

Поступила в редколлегию 00.00.00

Фамилия, имя, отчество, ученая степень, звание, должность и место работы. Научные интересы. Адрес, контактный телефон.

Рефераты на украинском и английском языках. Текст аннотации не должен дублировать реферат.

УДК 000.000.00

Назва статті українською мовою / Ініціали. Прізвище // АСУ та прилади автоматики. 2000. Вип. 00. С. 000-000.

Текст реферату.

Табл. 00. Іл. 00. Бібліогр.: 00 назв.

UDC 000.000.00

Title of paper / Initials. Surname // Management Information System and Devices. All-Ukr. Sci. Interdep. Mag. 2000. N 00. P. 000-000.

Text.

Tab. 00. Fig. 00. Ref.: 00 items.

Представление материалов

Рукопись, реферат, сведения об авторах — в одном файле, *поименованном фамилией первого автора*, на дискете 3,5 дюйма. Твердая копия материалов — для граждан Украины — в одном экземпляре: рукопись, подписанная авторами, рефераты, акт экспертизы, внешняя рецензия, подписанная доктором наук, заявление на имя главного редактора со сведениями об авторах.

Адрес редакции: Украина, 61166, Харьков, просп. Ленина, 14, ХНУРЭ, комната 321, тел. 70-21-326.

E-mail: hahanov@kture.kharkov.ua.

Тематика статей, публикуемых в сборнике:

- Компьютерная инженерия
- Математическое моделирование
- Оптимизация и процессы управления
- Автоматизация проектирования и диагностика
- Информационные интеллектуальные системы
- Проектирование интегральных схем и микросистем
- Компьютерные технологии в образовании

Відповідальний випусковий В.І. Хаханов
Редактор О.П. Гужва
Комп'ютерна верстка Г.В. Хаханова, С.В. Чумаченко

Підп. до друку 27.12.2008. Формат 60x84¹/₈. Умов. друк. арк. .
Обл.-вид. арк. 11,8. Тираж 300 прим.
Зам. № 2-109. Ціна договірна.

Харківський національний університет радіоелектроніки (ХНУРЕ).
Україна, 61166, Харків, просп. Леніна, 14.

Оригінал-макет підготовлено і збірник віддруковано
в навчально-науковому видавничо-поліграфічному центрі ХНУРЕ.
Україна, 61166, Харків, просп. Леніна, 14.