

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 153

Харьков
2010

СОДЕРЖАНИЕ

ХАХАНОВ В.И., ЧУМАЧЕНКО С.В., ТЕСОУРА YVES, ВАСИЛЕНКО В.А., ЩЕРБИН Д.А. ИНФРАСТРУКТУРА ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА НА ОСНОВЕ ВЕКТОРНЫХ ЛОГИЧЕСКИХ ОПЕРАЦИЙ.....	4
БАРКАЛОВ А.А., ТИТАРЕНКО Л.А., ХЕБДА Е.П., СОЛДАТОВ К.А. СИНТЕЗ АВТОМАТА МУРА С ПРЕОБРАЗОВАНИЕМ РАСШИРЕННЫХ КОДОВ СОСТОЯНИЙ.....	12
ХАХАНОВ В.И., ЧУМАЧЕНКО С.В., МИЩЕНКО А.С., ЗАЦАРНЫЙ А.В., ХАХАНОВА Ю.В. ИНФРАСТРУКТУРА ДИАГНОСТИРОВАНИЯ ВРЕДОНОСНЫХ ПРОГРАММ В ИНДИВИДУАЛЬНОМ КИБЕРНЕТИЧЕСКОМ ПРОСТРАНСТВЕ.....	19
БЕСАРАБ В.І., ВОРОПАЄВА Г.О. АНАЛІТИЧНА МОДЕЛЬ ДИСКРЕТНО-БЕЗПЕРЕРВНОЇ СИСТЕМИ З ЗАСТОСУВАННЯМ АПАРАТА МАХ-PLUS АЛГЕБРИ.....	33
ЛЕВЫКИН В.М., ХВОРОСТИНИНА О.С. ОБОБЩЕННАЯ МОДЕЛЬ СИСТЕМЫ СБАЛАНСИРОВАННЫХ ПОКАЗАТЕЛЕЙ.....	40
КАКУРИН Н.Я., ЛОПУХИН Ю.В., ВЕРЕЦА В.В., КАТАСОНОВ В.В., МАКАРЕНКО А.Н. АНАЛИЗ ПРЕОБРАЗОВАТЕЛЕЙ ДВОИЧНО-К-ИЧНОГО КОДА В ДВОИЧНЫЙ КОД ПО МЕТОДУ ДОСЧЕТА.....	45
ЯНКОВСКИЙ А.А., БУГРИЙ А.Н. КРИТЕРИИ ВЫБОРА МЕТОДА БИНАРИЗАЦИИ ПРИ ОБРАБОТКЕ ИЗОБРАЖЕНИЙ ЛАБОРАТОРНЫХ АНАЛИЗОВ.....	53
БАБИЧ А.В., ЕМЕЛЬЯНОВ И.В. ДИСПЕРСИОННЫЙ АНАЛИЗ В ПРОЦЕССАХ ПАССИВНОГО МОНИТОРИНГА И ДИАГНОСТИРОВАНИЯ КОРПОРАТИВНЫХ КОМПЬЮТЕРНЫХ СЕТЕЙ.....	56
МИРОНОВА Н.А., ТАБУНЩИК Г.В. ЭКСПЕРТНАЯ СИСТЕМА ВЫБОРА МОДИФИКАЦИИ МЕТОДА АНАЛИЗА ИЕРАРХИЙ.....	62
ЗАВИЗИСТУП Ю.Ю., ПАРТЫКА С.А. МЕТОД ПРЕОБРАЗОВАНИЯ СТАТИЧЕСКИХ SPT АЛГОРИТМОВ В ДИНАМИЧЕСКИЕ.....	70
ТАЯНОВ В.А. ІМОВІРНІСНІ ТА СТАТИСТИЧНІ ХАРАКТЕРИСТИКИ МОДЕЛІ ВИДІЛЕННЯ НАЙБІЛЬШ СТАБІЛЬНИХ ОБ'ЄКТІВ КЛАСІВ.....	75
РЕФЕРАТИ.....	84
ПРАВИЛА ОФОРМЛЕНИЯ РУКОПИСЕЙ ДЛЯ АВТОРОВ НАУЧНО-ТЕХНИЧЕСКОГО СБОРНИКА.....	88

*В.И. ХАХАНОВ, С.В. ЧУМАЧЕНКО, TIESCOURA YVES, В.А. ВАСИЛЕНКО,
Д.А. ЦЕРБИН*

ИНФРАСТРУКТУРА ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА НА ОСНОВЕ ВЕКТОРНЫХ ЛОГИЧЕСКИХ ОПЕРАЦИЙ

Предлагается аппаратная платформа и технологии для экспертного обслуживания запросов в реальном масштабе времени в виде мультипроцессорной системы на кристалле, ориентированной на анализ логических ассоциативных структур данных для получения точного детерминированного и многозначного решения, валидность (состоятельность) которого оценивается интегральным критерием качества взаимодействия запроса с векторами n -мерного ассоциативного пространства.

1. Введение

Мозгоподобность мультипроцессорной цифровой системы на кристалле есть концепция создания архитектуры и моделей вычислительных процессов, ориентированных на эффективную и быстроедействующую реализацию функциональностей, свойственных мозгу, на основе использования векторных логических операций для решения задач поиска, распознавания и принятия решений.

Существует большой класс логических задач, которые в настоящее время решаются не эффективно на универсальных компьютерах путем использования арифметических операций. Система логических команд является универсальным и полным базисом, а согласно теореме Поста способна описать и решить любую задачу. Однако практические разработчики программно-аппаратных систем уже забыли об изначальной логической сущности компьютера. Они привыкли неэффективно (на 5-10%) использовать мощности системы команд, компиляторов, операционных систем, отдавая компаниям-производителям порядка 90% денежных средств безвозмездно при покупке компьютера. Решение логических задач с помощью арифметического процессора не есть правильный выбор с позиции технологической и математической культуры. В настоящее время существующая платформа цифровых систем на кристаллах предоставляет практически неограниченные возможности для переориентации инфраструктуры моделей и методов на создание специализированных компьютеров с минимальной системой логических векторных операций или команд. Мотивация – создание мультипроцессора, как специализированной системы на кристалле для анализа и синтеза логических ассоциативных отношений, свойственных мозгу. Подтверждением актуализации таких изделий является появление на рынке iPad планшета (толщиной 12 мм и весом 0,68 кг), процессор которого выполнен в виде системы на кристалле Apple A4 на базе многоядерного ARM-процессора Cortex-A9 MPCore с использованием контроллеров памяти и графики.

Наличие аппаратного, быстроедействующего и дешевого специализированного логического вычислителя позволяет эффективно решать интересные для рынка информационных технологий задачи: 1. Анализ и синтез синтаксических и семантических языковых конструкций (реферирование, исправление ошибок, оценивание качества текстов). 2. Распознавание видео- и аудио-образов путем их представления вектором существенных параметров в дискретном пространстве. 3. Сервисное обслуживание сложных технических изделий и восстановление работоспособности в процессе их функционирования. 4. Тестирование знаний и экспертное обслуживание объектов или субъектов для определения их валидности. 5. Идентификация объекта или процесса для принятия решения в условиях неопределенности. 6. Точный поиск заданной вектором параметров информации в Internet, где по запросу пользователя очень часто выдаются два сообщения: отсутствуют данные или слишком много информации, слабо ассоциируемой с входным запросом. Здесь нужна правильная метрика оценивания и валидный запрос. 7. Коррекция текста в процессе его набора, когда автоматически исправляются только тривиальные ошибки, такие как повторение буквы в слове. Можно также корректировать более сложные семантические ошибки,

связанные с неверным окончанием, и предлагать более приемлемые варианты порядка слов в предложении или в его части. Данная задача актуальна для 100% пользователей компьютеров. 8. Более серьезная проблема выбора принадлежит критическим технологиям: целеуказание в истребителе или в автоматической системе посадки лайнера, работающих в реальном масштабе времени в микросекундном диапазоне измерения. 9. Обратной задачей выбора цели в критических технологиях является разведение объектов во времени и в пространстве, например, в диспетчерской службе аэропорта или оптимизация инфраструктуры городского транспорта для исключения коллизий. Практически все упомянутые задачи решаются в реальном масштабе времени, являются сходными по логической структуре процесс-моделей на основе использования ассоциативных таблиц. Для их решения необходима быстродействующая и специализированная аппаратная платформа в виде логического ассоциативного мультипроцессора (LAMP – Logical Associative MultiProcessor), ориентированного на параллельное выполнение процедур поиска, распознавания и принятия решений, оцениваемых путем использования интегрального критерия качества.

Цель – существенное повышение быстродействия процедур поиска, распознавания и принятия решений путем мультипроцессорной реализации параллельных средств обработки аналитических, графовых и табличных форм задания информации для определения детерминированного многозначного решения в n -мерном дискретном булевом пространстве.

Задачи: 1) Актуальность создания мозгоподобных вычислителей. 2) Метрика оценивания векторно-логических решений. 3) Архитектуры структур данных и ассоциативных отношений: таблицы, графы, уравнения. 4) Оптимизация логических структур данных.

Сущность – аппаратное обеспечение экспертного обслуживания запросов в реальном масштабе времени в виде мультипроцессорной системы на кристалле, ориентированной на анализ логических ассоциативных структур данных для получения точного детерминированного и многозначного решения, валидность (состоятельность) которого оценивается интегральным критерием качества взаимодействия запроса с векторами n -мерного ассоциативного пространства.

Объект исследования – аппаратная инфраструктура экспертного обслуживания задач поиска, распознавания и принятия решений в дискретном булевом пространстве на основе использования интегрального критерия качества и иерархических структур данных.

Предмет исследования – аппаратная платформа и технологии для реализации мультипроцессорной системы на кристалле, ориентированной на обслуживание задач поиска, распознавания и выбора решения в ассоциативных информационных структурах путем использования интегрального критерия качества в дискретном булевом пространстве.

Источники. 1. Аппаратные вычислительные изделия, ориентированные на решение логических задач ассоциативного поиска [1-4]. 2. Ассоциативные и логические структуры данных для решения информационных задач [5-8]. 3. Модели и методы дискретного анализа информации [9-12]. 4. Мультипроцессорные модели и средства для решения информационно-логических задач [13-19].

2. Интегральная метрика оценивания решения

При создании аппаратной платформы для информационно-логических задач акцент делается на следующие характеристики: 1) Высокое быстродействие параллельного выполнения минимального множества логических команд. 2) Исключение из процессора мощной системы арифметических вычислений, как функциональностей, несвойственных человеку. 3) Логический секвенсор – элементарный процессор – содержит 4 команды, которые кодируются двумя разрядами. 4) Устройство управления логическим ассоциативным мультипроцессором должно обеспечивать параллельное выполнение задач логического анализа. 5) Каждый секвенсор имеет ассоциативную память, а также регистры для хранения результатов логических вычислений и связи с другими секвенсорами. 6) Компилятор для языка описания аппаратуры или программирования есть внешняя программа по отношению к мультипроцессору, которая обеспечивает квазиоптимальное планирование вычислительного процесса во времени и в пространстве секвенсоров с учетом ограничений на размерность блоков ассоциативной памяти. 7) Память прямого доступа обслуживает мультипроцессор и хранит программу вычислительного процесса, полученную от компиля-

тора, для решения логической задачи. 8) Гибкая инфраструктура ассоциативной памяти обеспечивает размещение таблиц произвольной размерности. 9) GUI (Guide User Interface) предназначен для эффективного и дружественного общения с пользователем в процессе решения логических задач. 10) Точный и экономичный по времени подсчета критерий качества получаемого решения.

В последнем случае речь идет о качестве взаимодействия запроса (входного многозначного, в частности, троичного вектора m) с системой ассоциативных векторов (ассоциаторов), в результате которого должен быть сгенерирован конструктивный ответ в виде одного или нескольких ассоциаторов (A), а также численной характеристики степени принадлежности (функции качества) входного вектора m к найденному решению: $\mu(m \in A)$. Входной вектор $m = (m_1, m_2, \dots, m_i, \dots, m_q)$, $m_i \in \{0, 1, x\}$ и матрица A_i ассоциаторов $A_{ijr} (\in A_{ij} \in A_i \in A) = \{0, 1, x\}$ должны иметь одинаковую размерность, равную q . Далее, для удобства изложения материала, степень принадлежности m -вектора к одному ассоциатору или A -вектору будет обозначаться в виде $\mu(m \in A)$.

Существует всего 5 видов или результатов логического (теоретико-множественного) Δ -взаимодействия (пересечения) двух векторов $m \cap A$, определенных на рис. 1. Они формируют все первичные примитивные варианты реакции обобщенной ПРП-системы (Поиска, Распознавания и Принятия решения) на входное воздействие-запрос.

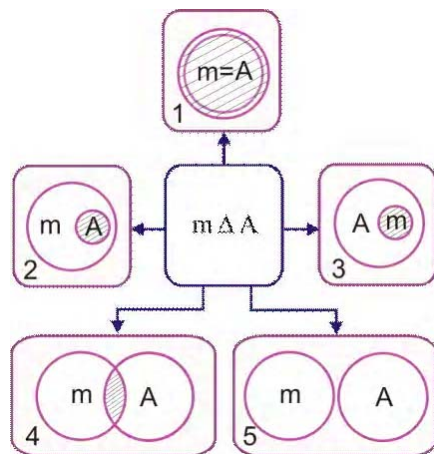


Рис. 1. Результаты пересечения двух векторов

В технологической отрасли знаний – технической диагностике (Design & Test) – указанная последовательность действий трансформируется к маршруту: поиск дефектов, их (распознавание) идентификация, (принятие решения на) восстановление работоспособности. Все три стадии технологического маршрута нуждаются в метрике оценивания решений для выбора оптимального варианта.

Определение. Интегральная теоретико-множественная метрика для оценивания качества запроса есть функция качества взаимодействия многозначных векторов $m \cap A$, которая определяется средней суммой трех нормированных параметров: кодовое расстояние $d(m, A)$, функция принадлежности $\mu(m \in A)$ и эффективность использования входного запроса – функция принадлежности $\mu(A \in m)$:

$$\begin{aligned}
 Q &= \frac{1}{3} [d(m, A) + \mu(m \in A) + \mu(A \in m)], \quad d(m, A) = \frac{1}{n} [n - \text{card}(m_i \cap A_i = \emptyset)]; \\
 \mu(m \in A) &= 2^{\text{card}(m \cap A) - \text{card}(A)} \leftarrow \text{card}(m \cap A) = \\
 &= \text{card}(m_i \cap A_i = x) \& \text{card}(A) = \text{card}(\bigcup_{i=1}^n A_i = x); \\
 \mu(A \in m) &= 2^{\text{card}(m \cap A) - \text{card}(m)} \leftarrow \text{card}(m \cap A) = \\
 &= \text{card}(m_i \cap A_i = x) \& \text{card}(m) = \text{card}(\bigcup_{i=1}^n m_i = x).
 \end{aligned} \tag{1}$$

Пояснения. Нормирование параметров позволяет оценивать уровень взаимодействия векторов в интервале $[0,1]$. Если зафиксировано предельное максимальное значение каждого параметра, равное 1, то векторы равны между собой. Минимальная оценка, $Q = 0$, фиксируется в случае полного несовпадения векторов по всем n координатам. Если мощность пространства вектора m ($m \cap A = m$) равна половине пространства вектора A , то функции принадлежности и качества соответственно равны:

$$\mu(m \in A) = \frac{1}{2}; \mu(A \in m) = 1; d(m, A) = 1; Q(m, A) = \frac{5}{2 \times 3} = \frac{5}{6}.$$

Аналогичное значение будет иметь параметр Q , если мощность пространства вектора A равна половине вектора m . Если мощность пространства пересечения $\text{card}(m \cap A)$ равна половине мощностей пространств векторов A и m , то функции принадлежности имеют значения:

$$\mu(m \in A) = \frac{1}{2}; \mu(A \in m) = \frac{1}{2}; d(m, A) = 1; Q(m, A) = \frac{4}{2 \times 3} = \frac{4}{6} = \frac{2}{3}.$$

Следует также заметить, что если результат пересечения двух векторов равен пустому множеству, то степень двойки от символа «пусто» равна нулю (но не единице): $2^{\text{card}(m \cap A) = \emptyset} = 2^{\emptyset} = 0$. Это действительно означает, что количество общих точек при пересечении двух пространств равно нулю.

3. Процесс-модель поиска, распознавания и принятия решения

Метрика качества, представленная в (1), дает возможность оценивать близость пространственных объектов друг к другу, а также взаимодействие векторных пространств. Практическим примером полезности интегрального критерия качества может служить стрельба по цели, которая иллюстрируется ранее приведенными диаграммами (см. рис. 1) взаимодействия векторов: 1) пуля попала точно в цель и поразила ее полностью; 2) мишень поражена необоснованно большим калибром пули (снаряда); 3) калибра пули недостаточно для поражения крупной цели; 4) неэффективный и неточный выстрел снарядом большого калибра; 5) пуля пролетела мимо мишени. Для решения практических задач взаимодействия $P(m, A)$ интегральный критерий качества дает точную оценку попадания или промаха, а также эффективность использования «калибра оружия».

Аналитическая модель описания и решения логических ассоциативных отношений представлена системой уравнений:

$$\begin{aligned} P(m, A) &= \max_{i=1}^n Q_i(m \Delta A_i); Q(m, A) = (Q_1, Q_2, \dots, Q_i, \dots, Q_n); \\ A &= (A_1, A_2, \dots, A_i, \dots, A_n); Q(m, A_i) = \frac{1}{3} [d(m, A_i) + \mu(m \in A_i) + \mu(A_i \in m)]; \\ Q(m, A_i) &= [0, 1] \vee (1 \leftarrow m = A); \Delta = \{\text{and, or, xor, not, slc, nop}\}; \\ A_i &= (A_{i1}, A_{i2}, \dots, A_{ij}, \dots, A_{is}); A_{ij} = (A_{ij1}, A_{ij2}, \dots, A_{ijr}, \dots, A_{msq}); \\ m &= (m_1, m_2, \dots, m_r, \dots, m_q). \end{aligned} \quad (2)$$

Здесь определены предикаты (высказывания) трех уровней иерархии: 1) Системный уровень функциональности $P(m, A)$ задает не структуры данных, а аналитическую модель вычислительного процесса в виде предиката, максимизирующего интегральный критерий принадлежности в интервале $Q(m, A) = [0, 0 - 1, 0]$, на множестве введенных операций. 2) Система предикатов среднего уровня представляется в виде вершин-таблиц графа $A = (A_1, A_2, \dots, A_i, \dots, A_m)$, логически взаимодействующих между собой. 3) Предикат нижнего уровня задает упорядоченную совокупность вектор-строк ассоциативной таблицы $A_i = (A_{i1}, A_{i2}, \dots, A_{ij}, \dots, A_{is})$, где строка $A_{ij} = (A_{ij1}, A_{ij2}, \dots, A_{ijr}, \dots, A_{msq})$ есть истинное высказывание. Предикат $A_i = (A_{i1}, A_{i2}, \dots, A_{ij}, \dots, A_{is}) = 1$ задается совокупностью ассоциативных векторов, формирующих многозначную таблицу явных решений. Поскольку функционал не имеет постоянных во времени входных и выходных переменных, то данная структура отличается от последовательной машины фон Неймана, задаваемой конечными автоматами Мили и Мура. Ассоциативность или равнозначность всех переменных в векторе $A_{ij} = (A_{ij1}, A_{ij2}, \dots, A_{ijr}, \dots, A_{msq})$ создает равные условия их существования, что означает

инвариантность решения задач прямой и обратной импликации в пространстве $A_i \in A$. Ассоциативный вектор A_{ij} определяет собой явное решение, где каждая переменная задается в конечном, многозначном и дискретном алфавите $A_{ij} \in \{\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_k\} = \beta$. Взаимодействие $P(m, A)$, входного вектора-запроса $m = (m_1, m_2, \dots, m_r, \dots, m_q)$ с графом $A = (A_1, A_2, \dots, A_i, \dots, A_m)$, формирует множество решений с выбором лучшего из них по максимальному критерию качества: $P(m, A) = \max Q_i[m \wedge (A_1 \vee A_2 \vee \dots \vee A_i \vee \dots \vee A_m)]$.

Конкретное взаимодействие вершин графа между собой создает функциональность $A = (A_1, A_2, \dots, A_i, \dots, A_m)$, которая может быть оформлена в следующие структуры: 1) Единственная ассоциативная таблица, содержащая все решения логической задачи в явном виде. Преимущество – максимальное быстродействие параллельного ассоциативного поиска решения по таблице. Недостаток – максимально высокая аппаратная сложность решения задачи. 2) Древовидная (графовая) структура бинарных отношений между предикатами, каждый из которых формирует таблицу истинности для незначительного количества (двух) переменных. Преимущество – максимально низкая аппаратная сложность решения задачи. Недостаток – минимальное быстродействие последовательного ассоциативного поиска решения по дереву. 3) Компромиссная графовая структура логически понятных для пользователя отношений между предикатами, каждый из которых формирует таблицу истинности для логически сильно взаимосвязанных переменных. Преимущество – высокое быстродействие параллельного ассоциативного поиска решений по минимальному числу таблиц. Сравнительно невысокая аппаратная сложность решения задачи. Недостаток – снижение быстродействия из-за последовательной логической обработки графовой структуры решений, найденных в таблицах. Разбиение одной таблицы (ассоциативной памяти) на k частей приводит к уменьшению аппаратных затрат, выраженных в компонентах (лутах) (LUT – Look Up Table) программируемой логической матрицы. Каждая ячейка памяти создается с помощью четырех лутов. В связи с тем, что ассоциативную матрицу можно представить квадратом со стороной n , суммарные аппаратные затраты для реализации памяти системы будут иметь функциональную зависимость от числа разбиений, которая определяется следующим выражением:

$$Z(n) = k \times \frac{1}{4} \times \left(\frac{n}{k}\right)^2 + h = \frac{n^2}{4 \times k} + h, (h = \{n, \text{const}\}) . \quad (3)$$

Второе слагаемое, равное h – есть затраты на общую схему управления системой ассоциативных памяти. Платой за уменьшение аппаратуры является снижение быстродействия обработки структуры памяти или увеличение периода анализа компонентов системы. Функциональная зависимость времени анализа логического ассоциативного графа от числа вершин или разбиений памяти имеет следующий вид:

$$T(n) = \frac{4 \times k}{t_{\text{clk}}} + \frac{4}{t_{\text{clk}}} = \frac{4}{t_{\text{clk}}}(k + 1), (t_{\text{clk}} = \text{const}). \quad (4)$$

Здесь период обработки одной ассоциативной памяти представлен циклом, содержащим 4 синхроимпульса. Число разбиений k пропорционально увеличивает количество тактов в худшем варианте последовательного соединения памяти. Второе слагаемое $\frac{4}{t_{\text{clk}}}$ задает

время, необходимое для подготовки данных на входе системы, а также для их декодирования на выходе вычислительной структуры. Функциональные зависимости аппаратных затрат и времени анализа графа ассоциативных памяти от числа вершин или разбиений представлены на рис. 2.

Построение обобщенной функции эффективности графовой структуры от числа вершин

$$f[Z(n), T(n)] = Z(n) + T(n) = \left(\frac{n^2}{4 \times k} + h\right) + \left(\frac{4}{t_{\text{clk}}}(k + 1)\right) \quad (5)$$

позволяет определить оптимальное разбиение совокупного и наперед заданного объема ассоциативной памяти. В данном случае это есть минимум аддитивной функции, который определяется значением k , обращающим производную функции в нуль. В данном случае

($n = 600$, $h = 200$, $t_{clk} = 4$) оптимальное число разбиений k для матрицы памяти, размерностью 600×600 , равно 4.

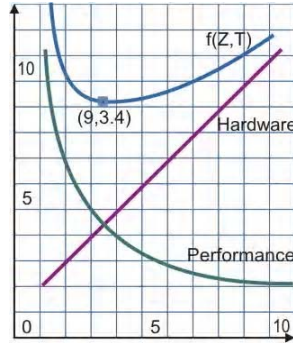


Рис. 2. Функции аппаратуры и времени от числа разбиений

4. Векторно-логический критерий качества решения

Цель введения критерия заключается в использовании только логических операций и исключении арифметических вычислений из процедуры формирования оценки взаимодействия компонентов m и A для существенного повышения быстродействия логического анализа структур информационных данных.

Идея – оценивать решение задачи взаимодействия входного вектора m с ассоциативными таблицами мощностью единиц в векторе качества Q путем использования векторных логических операций. Арифметическая оценка качества взаимодействия может формироваться сложением без усреднения приведенных критериев принадлежности и кодового расстояния, что определяется следующими формулами:

$$Q = d[m, A_{i(j)}] + \mu[m \in A_{i(j)}] + \mu[A_{i(j)} \in m], \quad d(m, A_{i(j)}) = \text{card}\left[m \bigoplus_{i(j)=1}^{n(m)} A_{i(j)} = 1\right];$$

$$\mu(m \in A_{i(j)}) = \text{card}[A_{i(j)} = 1] - \text{card}\left[m \bigwedge_{i(j)=1}^{n(m)} A_{i(j)} = 1\right];$$

$$\mu(A_{i(j)} \in m) = \text{card}[m = 1] - \text{card}\left[m \bigwedge_{i(j)=1}^{n(m)} A_{i(j)} = 1\right].$$
(6)

Первый компонент, составляющий критерий, формирует степень несовпадения n -мерных векторов – кодовое расстояние путем выполнения операции хог, второй и третий определяют степень непринадлежности результата конъюнкции к числу единиц каждого из двух взаимодействующих векторов. Понятия принадлежности и непринадлежности являются взаимодополняющими, но в данном случае технологичнее высчитывать именно непринадлежность. Таким образом, идеальный критерий качества равен нулю, когда два вектора равны между собой. Оценка качества взаимодействия двух двоичных векторов убывает по мере роста критерия от 0 к 1. Чтобы окончательно уйти от арифметических операций при подсчете уже векторного критерия качества, необходимо выражения (6) преобразовать к виду:

$$Q = d(m, A) \vee \mu(\overline{m} \in A) \vee \mu(A \in m), \quad d(m, A) = \overline{m} \oplus A;$$

$$\mu(m \in A) = A \wedge \overline{m} \wedge A; \quad \mu(A \in m) = m \wedge \overline{m} \wedge A.$$
(7)

Здесь критерии представлены уже не числами, а векторами, которые оценивают взаимодействие между компонентами m, A . При этом число нулей в трех оценках есть хорошо, а единицы ухудшают качество взаимодействия. Оптимизация решения логической задачи направлена на минимизацию числа единиц и максимизацию количества нулевых координат в векторах критерия качества. Для сравнения двух оценок необходимо определять мощность единиц в каждом векторе без выполнения операций суммирования. Это можно сделать, например, с помощью регистра [4] уплотнения единиц и их сдвига влево, представленного на рис. 3. Регистр позволяет за один такт выполнить сдвиг влево и уплотнить все единичные координаты n -разрядного двоичного вектора.

После процедуры сжатия номер последнего или правого единичного бита уплотненной серии единиц формирует индекс качества взаимодействия векторов. Для двоичных векторов $m = (1100110011\ 00)$, $A = (0000111101\ 01)$ определение качества их взаимодействия в соответствии с моделью (7) представлено в следующем виде (нулевые координаты отмечены точками):

m	1 1 . . 1 1 . . 1 1 . .
A	 1 1 1 1 . 1 . 1
$\overline{m \wedge A}$	 1 1 . . . 1 . .
$m \wedge A$	1 1 1 1 . . 1 1 1 . 1 1
$d(m, A) = m \oplus A$	1 1 1 1 1 . . 1
$\mu(A \in m) = m \wedge \overline{m \wedge A}$	1 1 1 . . .
$\mu(m \in A) = A \wedge \overline{m \wedge A}$	 1 1 . . . 1
$Q = d(m, A) \vee \mu(m \in A) \vee \mu(A \in m)$	1 1 1 1 1 . . 1
$Q(m, A) = (6/12)$	1 1 1 1 1 1

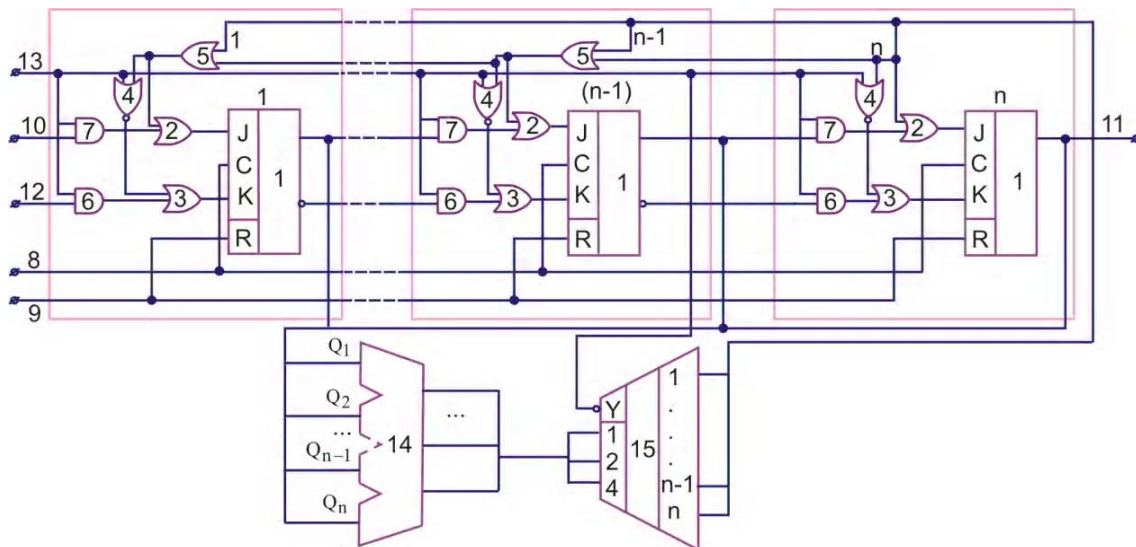


Рис. 3. Регистр уплотнения единиц

Здесь сформирована не только оценка качества взаимодействия векторов, равная $Q(m, A) = (6/12)$, но, что самое главное, единичные координаты строки $Q = d(m, A) \vee \mu(m \in A) \vee \mu(A \in m)$ идентифицируют все те места или позиции, по которым существует некачественное взаимодействие векторов. Другой пример иллюстрирует формирование максимального критерия качества для двоичных векторов $m = (1100001100\ 11)$; $A = (1100001100\ 11)$, имеющих совпадение по всем координатам:

m	1 1 1 1 . . 1 1
A	1 1 1 1 . . 1 1
$\overline{m \wedge A}$	1 1 1 1 . . 1 1
$m \wedge A$	. . 1 1 1 1 . . 1 1 . .
$d(m, A) = m \oplus A$	
$\mu(A \in m) = m \wedge \overline{m \wedge A}$	
$\mu(m \in A) = A \wedge \overline{m \wedge A}$	
$Q = d(m, A) \vee \mu(m \in A) \vee \mu(A \in m)$	
$Q = (0/12)$	

Здесь критерий качества, равный нулю во всех 12 разрядах $Q = (0/12)$, является максимальным или самым лучшим для взаимодействующих векторов $m = (1100001100\ 11)$; $A = (1100001100\ 11)$, поскольку он определен минимальным числом еди-

ниц на двенадцати координатах вектора. Для сравнения двух решений, полученных в результате логического анализа, следует использовать сжатые векторы качества Q , над которыми необходимо выполнить процедуру, включающую следующие векторные операции:

$$Q(m, A) = \begin{cases} Q_1(m, A) \leftarrow Q_1(m, A) \oplus Q_1(m, A) \wedge Q_2(m, A) = 0; \\ Q_2(m, A) \leftarrow Q_1(m, A) \oplus Q_1(m, A) \wedge Q_2(m, A) \neq 0. \end{cases} \quad (8)$$

Для двоичных векторов, представляющих собой критерии качества, выполнена процедура выбора лучшего из них на основе выражения, представленного в (8):

$Q_1(m, A) = (6, 12)$	1 1 1 1 1 1
$Q_2(m, A) = (8, 12)$	1 1 1 1 1 1 1 1
$Q_1(m, A) \wedge Q_2(m, A)$	1 1 1 1 1 1
$Q_1(m, A) \oplus Q_1(m, A) \wedge Q_2(m, A)$	
$Q(m, A) = Q_1(m, A)$	1 1 1 1 1 1

Предложенная модель вычислительного процесса на основе векторных логических операций и разработанные аналитические модели и методы анализа таблиц, а также критерии качества решения позволяют использовать все упомянутое в виде инфраструктуры для поиска квазиоптимального покрытия, диагностирования одиночных и кратных дефектов программных и/или аппаратных блоков. Модель векторных вычислений может служить основой для разработки специализированной мультипроцессорной архитектуры, ориентированной на решение, например, следующих задач технической диагностики: моделирование исправного поведения цифрового устройства, заданного таблицей истинности; определение качества теста по таблице неисправностей; минимизация тестовых наборов; поиск тестовых последовательностей, распознающих дефект; идентификация кратной неисправности, имеющей место в цифровом изделии.

5. Выводы

1. *Научная новизна* представлена новой моделью вычислительного процесса на основе векторных логических операций для решения задач поиска, диагностирования, распознавания образов и принятия решений в векторном дискретном булевом пространстве. Модель ориентирована на достижение высокого быстродействия процедур параллельного векторного логического анализа табличных форм задания информации, в пределах полностью исключая использование арифметических операций, в том числе и для подсчета критерия качества решения.

2. *Практическая значимость* заключается в ориентации предложенной модели, использующей векторные операции, на анализ ассоциативных таблиц с помощью логического мультипроцессора с ограниченной системой команд, что обеспечивает высокое быстродействие параллельной обработки больших массивов информации, представленных в общем случае графовыми структурами ассоциативных матриц или таблиц.

Список литературы: 1. Zorian Y. Test Strategies for System-in-Package / Y. Zorian // Plenary Paper of IEEE East-West Design & Test Symposium (EWDTS'08). Lviv, Ukraine. 2008. 2. Smith L. 3D Packaging Applications, Requirements, Infrastructure and Technologies / L. Smith // Fourth Annual International Wafer-Level Packaging Conference. San Jose, California. September, 2007. 3. The next Step in Assembly and Packaging: System Level Integration in the package (SiP) / Editors: William Chen, W. R. Bottoms, Klaus Pressel, Juergen Wolf // SiP White Paper. International Technology Roadmap for Semiconductors. 2007. P. 17–23. 4. А.с. №1439682. 22.07.88. Регистр сдвига / Какурин Н.Я., Хаханов В.И., Лобода В.Г., Какурина А.Н. 4с. 5. Бондаренко М.Ф. О мозгоподобных ЭВМ / М.Ф. Бондаренко, З.В. Дударь, И.А. Ефимова, В.А. Лещинский, С.Ю. Шабанов-Кушнаренко // Радиоэлектроника и информатика. 2004. № 2. С. 89–105. 6. Бондаренко М.Ф. Об алгебре предикатов / М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко // Бионика интеллекта. 2004. № 1. С. 15–26. 7. Бондаренко М.Ф. Теория интеллекта. Учебник / М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко; Харьков: СМИТ. 2006. 592 с. 8. Бондаренко М.Ф. Модели языка / М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко // Бионика интеллекта. 2004. № 1. С. 27–37. 9. Акритас А. Основы компьютерной алгебры с приложениями: Пер. с англ. / А. Акритас. М.: Мир, 1994. 544 с. 10. Гилл Ф. Практическая оптимизация / Ф. Гилл, У. Мюррей, М. Райт. М.: Мир, 1985. 509 с. 11. Аттетков А.В. Методы оптимизации / А.В. Аттетков, С.В. Галкин, В.С. Зарубин. М.: Изд-во МГТУ им. Н.Э. Баумана, 2003. 440 с. 12. Дегтярев Ю. И. Методы оптимизации: Учебное пособие для вузов / Ю. И. Дегтярев. М.: Сов. радио, 1980. 270 с. 13. Bergeron J. Writing Testbenches Using SystemVerilog / J.

Bergeron // Springer Science and Business Media, Inc. 2006. 414 p. **14.** *Abramovici M.* Digital System Testing and Testable Design / M. Abramovici, M.A. Breuer and A.D. Friedman. Comp. Sc. Press. 1998. 652 p. **15.** Densmore D. A Platform-Based taxonomy for ESL Design / Douglas Densmore, Roberto Passerone, Alberto Sangiovanni-Vincentelli // Design & Test of computers. 2006. P. 359-373. **16.** *Хаханов В.И.* Проектирование и тестирование цифровых систем на кристаллах / В.И. Хаханов, Е.И. Литвинова, О.А. Гузь. Харьков: ХНУРЭ, 2009. 484с. **17.** *Hahanov V.I.* SIGETEST – Test generation and fault simulation for digital design / V.I. Hahanov, D.M. Gorbunov, Y.V. Miroshnichenko, O.V. Melnikova, V.I. Obrizan, E.A. Kamenuka // Proc. of Conf. «Modern SoC Design Technology based on PLD». Kharkov.2003. P. 50-53. **18.** *Автоматизация диагностирования электронных устройств/ Ю.В.Малышенко и др./* Под ред. В.П.Чипулиса. М.: Энергоатомиздат, 1986. 216с. **19.** *Хаханов В.И.* Проектирование и верификация цифровых систем на кристаллах / В.И. Хаханов, И.В. Хаханова, Е.И. Литвинова, О.А. Гузь. Харьков: Новое слово, 2010. 528с.

Поступила в редколлегию 08.12.2010

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. 70-21-326. E-mail: hahanov@kture.kharkov.ua.

Чумаченко Светлана Викторовна, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: математическое моделирование, методы дискретной оптимизации. Увлечения: спорт, путешествия, любительское фото. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326, e-mail: ri@kture.kharkov.ua.

Tiessoua Yves, аспирант кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем и сетей. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: kiu@kture.kharkov.ua.

Василенко Василина Александровна, аспирантка кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем и сетей. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-421. E-mail: kiu@kture.kharkov.ua.

Щербин Дмитрий Александрович, студент факультета КИУ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем и сетей. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-421.

СИНТЕЗ АВТОМАТА МУРА С ПРЕОБРАЗОВАНИЕМ РАСШИРЕННЫХ КОДОВ СОСТОЯНИЙ

Предлагается метод уменьшения числа макроячеек программируемой матричной логики в схеме микропрограммного автомата Мура. Метод основан на представлении кода состояния в виде конкатенации кодов класса псевдоэквивалентных состояний и состояния внутри этого класса. Такой подход позволяет устранить прямую зависимость между состояниями и микрооперациями. Для формирования микроопераций используется специальный преобразователь кода. В результате оптимизируется как блок формирования функций возбуждения, так и блок формирования микроопераций. Приводится пример применения предложенного метода.

1. Введение

Практически каждая цифровая система включает устройство управления, координирующее работу всех ее блоков [1]. Одной из широко используемых моделей устройства управления является микропрограммный автомат (МПА) Мура [2,3]. В настоящее время для реализации схем быстродействующих МПА применяются программируемые логические интегральные схемы (ПЛИС), макроячейки которых включают блоки программируемой матричной логики (ПМЛ) [4,5]. Одной из особенностей ПМЛ является ограниченное число промежуточных шин (термов), что вызывает необходимость минимизации реализуемых функций [6,7]. Схема МПА Мура состоит из блока формирования функций возбуждения памяти (БВП) и блока формирования микроопераций (БМО), которые реализуются на ПМЛ. Известные методы синтеза МПА Мура [2] позволяют оптимизировать аппаратные затраты только в одном из блоков МПА. В настоящей работе предлагается метод, позволяющий оптимизировать число ПМЛ в обоих блоках МПА Мура. Метод основан на

введении дополнительного преобразователя кодов и нестандартном представлении кодов состояний автомата.

Целью исследования является оптимизация схемы МПА Мура за счёт преобразования расширенных кодов состояний в коды наборов микроопераций.

Задача исследования состоит в разработке метода синтеза МПА Мура, позволяющего уменьшить число ПМЛ в схеме автомата.

2. Общее положение и основная идея предлагаемого метода

Пусть алгоритм управления задан в виде граф-схемы алгоритма (ГСА) Γ , отмеченной состояниями автомата Мура [2]. Состояния МПА образуют множество $A = \{a_1, \dots, a_M\}$, а для их кодирования используются внутренние переменные $T_r \in T = \{T_1, \dots, T_R\}$. При этом число внутренних переменных определяется формулой

$$R = \lceil \log_2 M \rceil. \quad (1)$$

В условных вершинах ГСА записаны логические условия (ЛУ) $x_l \in X = \{x_1, \dots, x_L\}$. В операторных вершинах ГСА записаны наборы микроопераций $Y_q \subseteq Y$, где $q = \overline{1, Q}$, а $Y = \{y_1, \dots, y_N\}$ – множество микроопераций.

Представление алгоритма управления в виде ГСА используется для построения прямой структурной таблицы (ПСТ) [2], имеющей столбцы: a_m – исходное состояние МПА; $K(a_m)$ – код состояния $a_m \in A$ разрядности R ; a_s – состояние перехода; $K(a_s)$ – код состояния a_s ; X_h – выходной сигнал, определяющий переход из a_m в a_s ; Φ_h – набор функций возбуждения регистра памяти (RG), определяющий переход из $K(a_m)$ в $K(a_s)$; h – номер строки ПСТ, где $h = \overline{1, N_1}$. В столбце a_m записывается номер микроопераций (НМО) $Y(a_m) = Y_q$, формируемых в состоянии $a_m \in A$. Как правило, для реализации RG используется D-триггер. Поэтому множество функций возбуждения Φ содержит элементы D_r ($r = \overline{1, R}$). Эта таблица является основой для формирования системы функций

$$\Phi = \Phi(T, X), \quad (2)$$

$$Y = Y(T). \quad (3)$$

Система (2) – (3) определяет логическую схему МПА Мура, представленную на рис. 1. В этой схеме блок БВП реализует систему (2), а блок БМО – систему (3). Условимся в дальнейшем обозначать схему на рис. 1 символом U_1 . При этом $U_1(\Gamma_j)$ означает, что автомат Мура U_1 реализует ГСА Γ_j . Сигнал Start используется для установки МПА в начальное состояние, а сигнал Clock – для задания момента изменения содержимого RG.

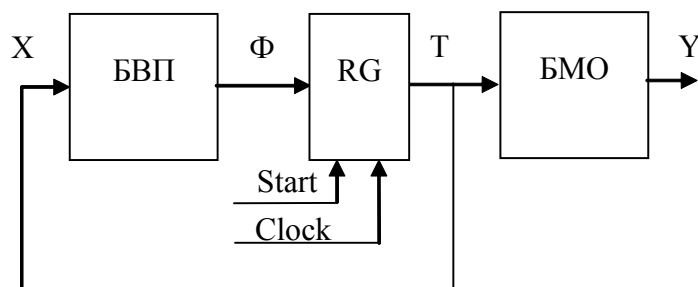


Рис. 1. Структурная схема МПА U_1

В МПА Мура число переходов N значительно превосходит число N_0 переходов эквивалентного МПА Мили [2]. Однако соответствующее кодирование состояний позволяет уменьшить этот параметр, что уменьшает число ПМЛ в блоке БВП [8]. Для уменьшения числа ПМЛ в блоке БМО также используются различные методы кодирования состояний [3,8]. Однако не существует эффективного метода кодирования состояний, который одновременно оптимизирует оба блока.

В настоящей работе предлагается метод синтеза МПА Мура, позволяющий максимально уменьшить число ПМЛ в блоках БВП и БМО. Этот метод заключается в следующем.

Пусть множество A разбивается на I классов псевдоэквивалентных состояний, образующих разбиение $\Pi_A = \{B_1, \dots, B_I\}$. Напомним, что состояния a_i, a_j называются псевдоэквивалентными, если выходы отмеченных ими вершин ГСА связаны со входом одной и той же вершины [3]. Закодируем классы $B_i \in \Pi_A$ кодами $K(B_i)$, имеющими R_1 разрядов:

$$R_1 = \lceil \log_2 I \rceil. \quad (4)$$

Используем для кодирования классов переменные $\tau_r \in \tau$, где $|\tau| = R_1$.

Пусть класс $B_i \in \Pi_A$ содержит M_i состояний и пусть

$$M_0 = \max(M_1, \dots, M_I). \quad (5)$$

Закодируем состояния $a_m \in B_i$ кодом $C(a_m)$ разрядности

$$R_2 = \lceil \log_2 M_0 \rceil \quad (6)$$

и используем для кодирования переменные $T_r \in T$, где $|T| = R_2$.

В этом случае состояние $a_m \in A$ однозначно определяется расширенным кодом

$$K(a_m) = K(B_i) * C(a_m), \quad (7)$$

где $*$ – знак конкатенации, $m = \overline{1, M}$, $i = \overline{1, I}$.

Подставим в соответствие каждому набору $Y_q \subseteq Y$ двоичный код $K(Y_q)$ разрядности

$$R_3 = \lceil \log_2 Q \rceil. \quad (8)$$

Используем для кодирования НМО переменные $z_r \in Z$, где $|Z| = R_3$. В этом случае для интерпретации ГСА Γ предлагается МПА Мура U_2 (рис. 2).

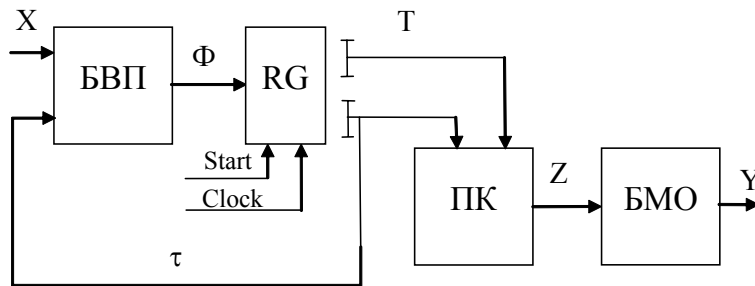


Рис. 2. Структурная схема МПА U_2

3. Метод синтеза автомата Мура U_2

В автомате U_2 блок БВП реализует систему функций

$$\Phi = \Phi(\tau, X), \quad (9)$$

включающую $R_1 + R_2$ элементов. Блок преобразователя кодов (ПК) формирует переменные $z_r \in Z$. Для этого реализуется система функций

$$Z = Z(\tau, T). \quad (10)$$

Блок БМО реализует микрооперации $y_n \in Y$, как функции системы

$$Y = Y(Z). \quad (11)$$

Такой подход даёт следующие преимущества по сравнению с МПА U_1 :

– число термов в системе (9) гарантировано равняется H_0 при любом кодировании классов $B_i \in \Pi_A$;

– кодирование наборов $Y_q \subseteq Y$ может быть выполнено таким образом, чтобы минимизировать число ПМЛ в блоке БМО.

Недостатками являются увеличение числа функций в системе Φ при выполнении условия

$$R_1 + R_2 > R, \quad (12)$$

а также наличие блока ПК. Отметим, что кодирование классов $B_i \in \Pi_A$ и состояний $a_m \in B_i$ можно выполнить так, чтобы уменьшить число ПМЛ в блоке ПК. При выполнении условия

$$R_3 < R \quad (13)$$

уменьшаются аппаратные затраты в блоке БМО, если он реализуется на заказных матрицах [9] либо макроячейках памяти [3].

В настоящей работе предлагается метод синтеза МПА U_2 , включающий следующие этапы:

1. Формирование разбиения Π_A .
2. Формирование наборов микроопераций $Y_q \subseteq Y$.
3. Оптимальное кодирование НМО.
4. Кодирование классов $B_i \in \Pi_A$ и состояний $a_m \in B_i$.
5. Формирование прямой структурной таблицы МПА U_2 .
6. Формирование систем функций (9) – (10).
7. Реализация схемы МПА в заданном элементном базисе.

Рассмотрим пример синтеза МПА $U_2(\Gamma_1)$, где ГСА Γ_1 представлена на рис. 3.

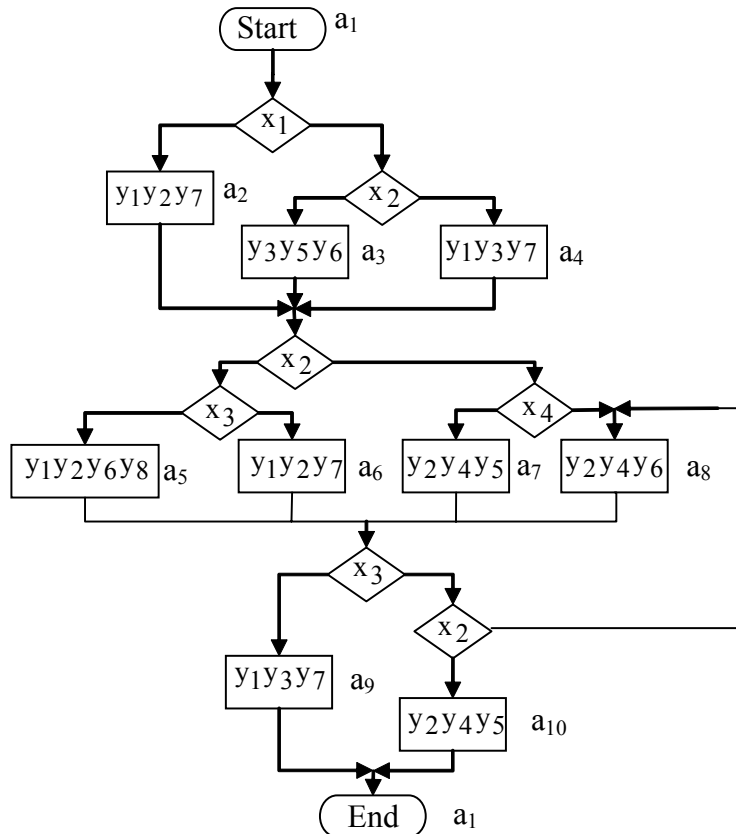


Рис. 3. Граф-схема алгоритма Γ_1 , отмеченная состояниями автомата Мура

Как видно из рис. 3, автомат $U_2(\Gamma_1)$ имеет $M = 10$ состояний, что дает $R = 4$. Исходя из определения псевдоэквивалентных состояний [3], можно найти множество

$\Pi_A = \{B_1, \dots, B_4\}$. Здесь $B_1 = \{a_1\}$, $B_2 = \{a_2, a_3, a_4\}$, $B_3 = \{a_5, \dots, a_8\}$, $B_4 = \{a_9, a_{10}\}$. Итак, $I = 4$, $R_1 = 2$.

В операторных вершинах ГСА Γ_1 находятся $Q = 7$ различных НМО: $Y_1 = Y(a_1) = \emptyset$, $Y_2 = \{y_1, y_2, y_7\}$, $Y_3 = \{y_3, y_5, y_6\}$, $Y_4 = \{y_1, y_3, y_7\}$, $Y_5 = \{y_1, y_2, y_6, y_8\}$, $Y_6 = \{y_2, y_4, y_5\}$, $Y_7 = \{y_2, y_4, y_6\}$. Таким образом, для кодирования НМО достаточно $R_3=3$ переменные и $Z = \{z_1, z_2, z_3\}$. Отметим, что условие (13) в данном случае выполняется.

Будем понимать под «оптимальным» такое кодирование НМО, когда система (11) имеет минимально возможное число термов. На рис. 4 представлен один из вариантов такого кодирования.

$z_1 \backslash z_2 z_3$				
	00	01	11	10
0	Y_1	Y_3	Y_4	*
1	Y_7	Y_6	Y_2	Y_5

Рис. 4. Оптимальные коды наборов микроопераций

С учётом этих кодов имеем систему (14) для автомата $U_2(\Gamma_1)$:

$$\begin{aligned}
 y_1 &= Y_2 \vee Y_4 \vee Y_5 = z_2; y_2 = Y_2 \vee Y_6 \vee Y_5 \vee Y_7 = z_1; \\
 y_3 &= Y_3 \vee Y_4 = \bar{z}_1 z_3; y_4 = Y_6 \vee Y_7 = z_1 \bar{z}_2; \\
 y_5 &= Y_3 \vee Y_6 = \bar{z}_2 z_3; y_6 = Y_3 \vee Y_5 \vee Y_7 = \bar{z}_1 \bar{z}_2 z_3 \vee z_1 \bar{z}_3; \\
 y_7 &= Y_2 \vee Y_4 = z_2 z_3; z_8 = Y_5 = z_2 \bar{z}_3.
 \end{aligned} \tag{14}$$

Классы $B_i \in \Pi_A$ и состояния $a_m \in B_i$ необходимо закодировать так, чтобы уменьшить число термов в системе (10). При этом оптимизируется число ПМЛ в схеме блока ПК. Построим таблицу, отражающую связь между кодами наборов МО и состояниями автомата $U_2(\Gamma_1)$ (табл. 1).

Таблица 1

$K(Y_q)$	001	011	100	101	110	111
a_m	a_3	a_4, a_9	a_8, a_{10}	a_7	a_5	a_2, a_6
Y_q	Y_3	Y_4	Y_7	Y_6	Y_5	Y_2

В каждом столбце табл.1 первый разряд кода $K(Y_q)$ соответствует переменной z_1 , второй – z_2 и третий – z_3 . Это позволяет построить следующую систему функций:

$$\begin{aligned}
 z_1 &= a_2 \vee a_5 \vee a_6 \vee a_7 \vee a_8 \vee a_{10}; \\
 z_2 &= a_2 \vee a_4 \vee a_6 \vee a_9; \\
 z_3 &= a_2 \vee a_3 \vee a_4 \vee a_6 \vee a_7 \vee a_9.
 \end{aligned} \tag{15}$$

Состояния $a_m \in A$ должны быть закодированы так, чтобы у всех $a_m \in B_i$ код $K(B_i)$ совпадал.

Оптимизация системы (15) – это типичная задача символьного кодирования. Методы решения подобных задач можно найти в [1]. Один из вариантов оптимального кодирования классов $B_i \in \Pi_A$ и состояний $a_m \in B_i$ представлен на рис. 5.

Используя коды из рис. 5, можно получить следующую систему (16) из системы (15):

$$\begin{aligned}
 z_1 &= \tau_2 \bar{T}_1 \vee \tau_1 \tau_2 \vee \tau_1 T_1 \bar{T}_2; \\
 z_2 &= \bar{T}_1 T_2 \vee \bar{\tau}_2 T_2; \\
 z_3 &= T_2 \vee \bar{\tau}_1 \tau_2.
 \end{aligned} \tag{16}$$

T_1T_2		00	01	11	10
$\tau_1\tau_2$	00	a_1	*	*	*
	01	*	a_2	a_4	a_3
	11	a_5	a_6	a_7	a_8
	10	*	*	a_9	a_{10}

Рис. 5. Кодирование классов и состояний автомата $U_2(\Gamma_1)$

Если для реализации блока ПК используются ПМЛ, имеющие $q = 3$ терма, то каждая функция $z_r \in Z$ реализуется на одной макроячейке ПМЛ.

Для построения ПСТ необходимо получить систему обобщённых формул перехода [3]. В случае МПА $U_2(\Gamma_1)$ эта система имеет следующий вид:

$$\begin{aligned}
 B_1 &\rightarrow x_1a_2 \vee \bar{x}_1x_2a_3 \vee \bar{x}_1\bar{x}_2a_4; \\
 B_2 &\rightarrow x_2x_3a_5 \vee x_2\bar{x}_3a_6 \vee \bar{x}_2x_4a_7 \vee \bar{x}_2\bar{x}_4a_8; \\
 B_3 &\rightarrow x_3a_9 \vee x_2\bar{x}_3a_{10} \vee \bar{x}_2\bar{x}_3a_8; \\
 B_4 &\rightarrow a_1.
 \end{aligned} \tag{17}$$

Эта система очевидным образом преобразовывается в таблицу переходов (табл. 2).

Таблица 2

B_i	$K(B_i)$	a_s	$K(a_s)$	X_h	Φ_h	h
B_1	00	a_2	0101	x_1	D_2D_4	1
		a_3	0110	$\bar{x}_1x_2$	D_2D_3	2
		a_4	0111	$\bar{x}_1\bar{x}_2$	$D_2D_3D_4$	3
B_2	01	a_5	1100	x_2x_3	D_1D_2	4
		a_6	1101	$x_2\bar{x}_3$	$D_1D_2D_4$	5
		a_7	1111	$\bar{x}_2x_4$	$D_1D_2D_3D_4$	6
		a_8	1110	$\bar{x}_2\bar{x}_4$	$D_1D_2D_3$	7
B_3	11	a_9	1011	x_3	$D_1D_3D_4$	8
		a_{10}	1010	$\bar{x}_3x_2$	D_1D_3	9
		a_8	1110	$\bar{x}_3\bar{x}_2$	$D_1D_2D_3$	10

Из табл. 2 строится система (9), которая затем минимизируется для реализации на макроячейках ПМЛ. Пусть F_h – терм, соответствующий строке h ПСТ. Например, $F_1 = \bar{\tau}_1\bar{\tau}_2x_1$, $F_{10} = \tau_1\tau_2\bar{x}_3\bar{x}_2$. Из табл. 2 имеем, например, $D_1 = F_4 \vee \dots \vee F_{10}$. После минимизации получаем окончательную функцию $D_1 = \tau_2$. Аналогичным образом можно получить остальные функции системы (9).

Последний этап синтеза связан с реализацией схем блоков МПА по полученным системам функций. Этот этап достаточно полно рассмотрен в литературе [6,7]. Имеются промышленные пакеты [4,5], ориентированные на решение подобных задач. В этой связи данный этап в этой статье не рассматривается.

Отметим, что ПСТ автомата $U_1(\Gamma_1)$ имеет $H = 23$ строки. При этом каждая функция $D_r \in D$ будет иметь до 23 термов. Применение предложенного метода уменьшает этот параметр в 2,3 раза. При этом число функций $D_r \in \Phi$ будет одинаковым для $U_1(\Gamma_1)$ и $U_2(\Gamma_2)$ из-за нарушения условия (12).

4. Заключение

Предлагаемый в работе метод синтеза МПА Мура ориентирован на уменьшение аппаратных затрат в схеме автомата. Этот метод позволяет гарантированно уменьшить число макроячеек ПМЛ в схемах формирования функций возбуждения памяти и микроопераций. Однако это связано с введением дополнительного блока ПК, который потребляет некоторые ресурсы кристалла ПЛИС и вносит задержку в схему МПА. Эти недостатки можно уменьшить путем оптимального кодирования расширенных кодов состояний.

Исследование эффективности предложенного метода на стандартных примерах [10] показали, что предложенный метод позволяет уменьшить аппаратные затраты на 37 % (в среднем) для 98,2 % примеров. При этом быстроедействие МПА U_2 практически совпадает с этим параметром МПА U_1 . Это объясняется уменьшением числа уровней в схемах блоков МПА U_2 .

Научная новизна предложенного метода заключается в использовании классов псевдоэквивалентных состояний и кодов состояний в этих классах для уменьшения числа макроячеек ПМЛ в схеме МПА Мура.

Практическая значимость метода заключается в уменьшении аппаратных затрат в схеме МПА, что позволяет получить более дешёвые схемы по сравнению с известными из литературы аналогами.

Дальнейшее направление развития этой работы связано с возможностью применения предложенного метода для базиса FPGA [11] и заказных матриц [12].

Список литературы: 1. *De Micheli G.* Synthesis and Optimization of Digital Circuits. №4.: McGraw-Hill, 1994. 636 p. 2. *Baranov S.* Logic and System Design of Digital Systems. Tallinn: TUT Press, 2008. 267 p. 3. *Barkalov A., Titarenko L.* Logic Synthesis for FSM-Based Control Units. Springer. Lectures Notes in Electrical Engineering. №53 Berlin: Verlag Heidelberg, 2009. 233 p. 4. *Altera Corporation* webpage. <http://www.altera.com> 5. *Xilinx Corporation* webpage. <http://www.xilinx.com> 6. *Соловьев В.В.* Проектирование цифровых систем на основе программируемых логических интегральных схем. М.: Горячая линия – Телеком, 2007. 636 с. 7. *Kania D.* Synteza logiczna przeznaczona dla matrycowych struktur programowalnych typu PAL. Gliwice: Politechnika. 2004. 212 str. 8. *Barkalov A., Titarenko L., Chmielewski S.* Hardware reduction for Moore FSM implemented with CPLD // KEiT PAN, 2009. 212 p. 9. *Barkalov A., Titarenko L., Hebda O.* Matrix implementation of Moore FSM with expansion of coding space // Pomiar Automatyka Kontrola. vol.56. 2010. № 7. 694 – 696 p. 10. *Yang S.* Logic Synthesis and Optimization Benchmarks user guide. Technical report №1991 IWLS-UG Saeyang. – Microelectronics center of North Carolina. 11. *Maxfield C.* The Design Warrior's Guide to FPGAs. Amsterdam: Elsevier, 2004. 541 p. 12. *Smith M.* Application Specific Integrated Circuits. Boston: Addison-Wesley, 1997.

Поступила в редколлегию 22.11.2010

Баркалов Александр Александрович, д-р техн. наук, профессор Института информатики и электроники Университета Зеленогурского. Адрес: ul. prof. Z. Szafrana 2, 65-516 Zielona Góra, e-mail: A.Barkalov@iie.uz.zgora.pl.

Титаренко Лариса Александровна, д-р. техн. наук, профессор Института информатики и электроники Университета Зеленогурского. Адрес: ul. prof. Z. Szafrana 2, 65-516 Zielona Góra, e-mail: L.Titarenko@iie.uz.zgora.pl.

Хебда Елена Петровна, магистр, аспирант Института информатики и электроники Университета Зеленогурского. Адрес: ul. prof. Z. Szafrana 2, 65-516 Zielona Góra, e-mail: O.Shapoval@iie.uz.zgora.pl.

Солдатов Константин Альбертович, магистр, аспирант кафедры компьютерной инженерии Донецкого национального технического университета. Адрес: Украина, 83000, Донецк, ул. Артема, 56, e-mail: K.Soldatov@cs.dgtu.donetsk.ua

*В.И. ХАХАНОВ, С.В. ЧУМАЧЕНКО, А.С. МИЩЕНКО, А.В. ЗАЦАРНЫЙ,
Ю.В. ХАХАНОВА*

ИНФРАСТРУКТУРА ДИАГНОСТИРОВАНИЯ ВРЕДНОСНЫХ ПРОГРАММ В ИНДИВИДУАЛЬНОМ КИБЕРНЕТИЧЕСКОМ ПРОСТРАНСТВЕ

Предлагаются методы и архитектуры параллельного анализа информации, представленной в виде аналитических, графовых и табличных форм ассоциативных отношений для поиска, распознавания, диагностирования деструктивных компонентов и принятия решений в n -мерном векторном кибернетическом индивидуальном пространстве. Рассматриваются векторно-логические процесс-модели актуальных прикладных задач, в том числе – диагностирование спама и восстановление работоспособности программно-аппаратных компонентов компьютерных систем, качество решения которых оценивается неарифметической метрикой взаимодействия булевых векторов. Предлагается концепция саморазвития информационно-компьютерной экосистемы, повторяющая эволюцию функциональностей человека. Представляются оригинальные процесс-модели ассоциативно-логического анализа информации на основе быстродействующего мультипроцессора в n -мерном векторном дискретном пространстве. Разрабатывается алгебраическая структура, регулирующая векторно-матричные преобразования в дискретном булевом пространстве для анализа информации на основе логических операций над ассоциативными данными. Рассматривается применение алгебры векторной логики для создания процесс-моделей актуальных прикладных задач, качество решения которых оценивается интегральной неарифметической метрикой взаимодействия ассоциативных структур. Состоятельность теории, методов, структур данных и процесс-моделей подтверждается рассмотрением многочисленных примеров параллельного анализа, синтеза, поиска и принятия решений в информационном пространстве.

Введение

Проблема создания эффективной инфраструктуры кибернетического пространства (Cyber Space), а также саморазвивающейся информационно-компьютерной экосистемы (ИКЭС) планеты особенно важна для глобальных компаний, таких как Лаборатория Касперского, Google, Microsoft.

Кибернетическое пространство как объект природы также подвержено влиянию деструктивных компонентов, влияющих на работоспособность субъектов, которыми являются компьютеры, системы и сети. Поэтому сейчас и в будущем важной проблемой остается стандартизация пространства и специализация всех взаимодействующих субъектов, включая негативные, как неотъемлемую часть экосистемы. Данная акция есть постоянно действующая во времени, цель которой – не отставать, но на один шаг опережать появление новых вредоносных компонентов, путем создания инфраструктуры кибернетического пространства, обеспечивающей функционирование компьютерной экосистемы планеты и качество жизни каждого человека.

Среди модулей такой инфраструктуры можно выделить диагностирование отказов и спама путем анализа информации, полученной на стадии тестирования и использования специальных методов встроенного поиска спама на основе стандартов граничного сканирования или ассерционной избыточности, ориентированной на обнаружение спама, что позволит идентифицировать и устранять его без применения внешних средств. Таким образом, можно будет обходиться без сложных внешних программ моделирования, тестирования и диагностирования путем прививки каждого электронного письма тестопригодной интеллектуальной избыточностью кода на стадии его создания. При этом следует использовать предикат узнавания, который оперирует не только булевыми, но регистровыми и матричными переменными, что делает его практически значимым в формальной записи уравнений диагноза или распознавания:

$$x^a \approx x \oplus a = 0 \vee \min_i Q_i \rightarrow x \oplus a \oplus Q = 0;$$

$$x^m \approx x \oplus m = 0 \vee \min_i Q_i \rightarrow x \oplus m \oplus Q = 0;$$

$$T \oplus S = Q \approx \begin{matrix} 00 \\ 01 \\ 10 \\ 11 \end{matrix} \begin{bmatrix} 1 & . & 1 & 1 \\ . & 1 & . & . \\ . & 1 & . & 1 \\ 1 & . & . & 1 \end{bmatrix} \Delta \begin{bmatrix} 1 & . & 1 & 1 \\ . & 1 & 1 & 1 \\ . & . & . & . \\ . & . & . & 1 \end{bmatrix} = \begin{bmatrix} . & . & . & . \\ . & . & 1 & 1 \\ . & 1 & . & 1 \\ 1 & . & . & . \end{bmatrix},$$

x^a , x^m – предикатные переменные; a , m – значения переменных; Q_i – оценка узнавания значения переменной; T – тест; S – объект, подлежащий тестированию (программа).

На основе предиката узнавания m -образа любой сложности, природы и формы можно создавать достаточно компактные уравнения предикатов, формирующие интеллектуальные решения в области распознавания образов, принятий решений, тестирования знаний и технических объектов, диагностирования (узнавания) спама среди электронных писем.

В связи с этим предложенная инфраструктура кибернетического пространства, метрика его измерения и процесс-модели анализа и синтеза субъектов дают возможность создавать эффективные решения компьютерных изделий, ориентированных на быстрый поиск, распознавание, диагностирование не только позитивных, но и негативных субъектов. Конкретно, предложенная инфраструктура может решать задачи: 1) Описание многообразия электронного мусора кибернетического пространства. 2) Формализация процессов взаимодействия триады компонентов <программа, спамность, тесты>. 3) Диагностирование и фильтрация электронных писем. 4) Создание и эффективное использование базы спама. 5) Создание быстродействующих интеллектуальных саморазвивающихся средств сервисного обслуживания и защиты кибернетического пространства.

Сегодня чрезвычайно важно обозначить возможные пути для решения проблемы создания инфраструктуры саморазвивающейся компьютерной экосистемы. Эволюция ИКЭС основывается на использовании трех наиболее важных компонентов: фантазия, математика и технология, где субъектом экосистемы выступает саморазвивающийся компьютер (СРК). Основное отличие СРК от современного компьютера заключается в концепции жизненного цикла. Стратегия настоящего компьютера есть обучение или повторение уже пройденного пути. Принципиальная позиция СРК – постоянный поиск новых путей для самосовершенствования (эволюционирования) на основе мирового опыта, скрытого в информационном пространстве. Согласно запрету Геделя, адаптированному для информационного пространства, нельзя создать компьютер, который способен решать любые задачи, формально представленные спецификацией. Тем не менее, принцип Геделя предоставляет методологическую основу эволюции (саморазвития) ИКЭС, которую можно интерпретировать следующим образом. Для информационного пространства всегда можно придумать такую полезную спецификацию, которая не покрывается существующими у человечества решениями, что обуславливает создание нового функционального или технологического компонента для его последующего включения в планетарную библиотеку. ИКЭС имеет возможность повторить эволюцию человечества, только в тысячи раз более быстрыми темпами. Здесь заложены основные принципы эволюции, явно выраженные уже в современной компьютерной индустрии: 1) Стандартизация – самое главное для эволюции и жизненного цикла СРК – рынок не принимает и не понимает нестандартных по интерфейсу решений. 2) Специализация есть повышение эффективности предоставляемых (персонально ориентированных) сервисов изделия, связанных с быстродействием, качеством, затратами, энергосбережением путем оптимизации структуры и функциональных компонентов, покрывающих спецификацию. 3) Повсеместное использование векторно-логического критерия качества решения в задачах генерирования идей, синтеза и анализа. Генерирование – процесс создания новой функциональности. При этом синтез оперирует существующими в информационном пространстве компонентами для создания структуры. Анализ – оценивание полученного решения. 4) Диаграмма Хассе используется для выработки стратегии

оптимизации покрытия функциональностей спецификации библиотечными компонентами или их сочетаниями, принадлежащими информационному пространству. Она согласуется с современной Y-технологией, входящей в состав ESL Design, суть которой – использовать библиотечные компоненты на всех уровнях проектирования изделия для покрытия специфицируемой функциональности в процессе синтеза.

На рис. 1 представлен замкнутый цикл эволюции ИКЭС, который фактически изоморфен спирали развития человечества, накрученной на временную ось.

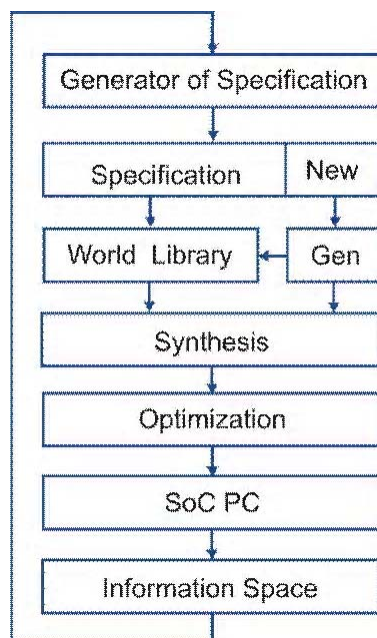


Рис. 1. Цикл ИКЭС

Целью данного исследования является существенное повышение качества индивидуального кибернетического пространства (ИКП) пользователя и уменьшение стоимости эксплуатационных расходов за счет вакцинации ИКП путем добавления в пространство инфраструктуры сервисного обслуживания, включающей библиотеки позитивных и негативных сообщений и обеспечивающей тестирование, диагностирование и устранение вредоносных компонентов электронных писем.

Объект исследования – индивидуальное кибернетическое пространство, представленное информацией, ее носителями и преобразователями, а также деструктивными компонентами, наносящими вред функциональностям, улучшающим качество жизни человека.

Предмет исследования – инфраструктура сервисного обслуживания, включающая библиотеки позитивных и негативных сообщений и встроенную программную избыточность, которая работает в реальном масштабе времени, обеспечивает тестирование, диагностирование и устранение вредоносной и «мусорной» информации электронных писем, описанных в соответствующих библиотеках.

Мотивация: 1) Отсутствие на рынке антиспамной защиты встроенных средств тестирования, диагностирования и удаления вредоносных компонентов, составляющих инфраструктуру сервисного обслуживания, подобной тому, как в цифровых системах на кристаллах существуют стандарты граничного сканирования, а в программных продуктах – ассерционная избыточность, ориентированные на встроенное тестирование дефектов и ошибок с последующим восстановлением работоспособности аппаратных или программных изделий. 2) Наличие теоретических разработок, связанных с технологией алгебрологического векторного анализа информационных данных, ориентированных на высокое быстродействие решения и оценивания задач распознавания, образов, принятия решений и тестирования объектов. 3) Наличие образцовой производственной и маркетинговой инфраструктуры Лаборатории Касперского, способной поддержать проект создания технологии вакцинации электронных сообщений и авторитетно предложить его рынку информационных технологий.

4) Миниатюризация цифровых и телекоммуникационных систем (телефоны смартфоны, планшеты) требует постоянной защиты от массовых и ненужных электронных писем путем внедрения встроенных антиспамных средств, контролирующих информационный обмен.

Задачи: 1) Разработка математического аппарата анализа кибернетического пространства, ориентированного на создание моделей и методов сервисного обслуживания программных продуктов для тестирования, диагностирования и устранения массовых и ненужных электронных писем. 2) Создание типовых процесс-моделей и критериев взаимодействия электронных писем с контентным анализом полезных функциональностей. 3) Разработка технологии анализа структуры программного кода для определения критических точек и установки в них ассерционных операторов наблюдения и управления в процессе его функционирования. 4) Создание инфраструктуры сервисного обслуживания функциональных программ для встроенного тестирования, диагностирования и устранения вредоносных компонентов из программного кода функциональности на основе использования библиотеки спамной информации. 5) Тестирование и верификация встроенной инфраструктуры сервисного обслуживания функциональностей, защищающей программный код от вредоносных компонентов.

1. Эволюция Cyber Space и Internet

Для создания схемы, реализующей полезную функциональность, следует генерировать примитивы $P = \{P_1, P_2, \dots, P_i, \dots, P_n\}$ самого нижнего уровня. Для этого необходимо создавать фильтры $F = \{F_1, F_2, \dots, F_j, \dots, F_m\}$, формирующие таблицы примитивных отношений, взятых из информационного пространства планеты (рис. 2). Имея стандартизованные структуры данных для отдельных порталов и браузеров, доставляющих новые сервисы с более высоким быстродействием, следует ожидать постепенного качественного улучшения всех компонентов Cyber Space. Конечная цель такого взаимного и положительного влияния элементов инфраструктуры кибернетического пространства – выработка единых стандартов по интерфейсам и его превращение в саморазвивающуюся интеллектуальную информационно-компьютерную экосистему. Существенное значение будут иметь первичные фильтры или преобразователи для создания новых стандартизованных примитивов, создающих технологичную инфраструктуру для скоростного драйва по Cyber Space с использованием специализированного неарифметического двигателя (I-Computer). Со временем аморфная, или «мусорная» часть Internet будет уменьшаться, а стандартизованная инфраструктура – увеличиваться. К 2020 году информационное пространство планеты должно принять цивилизованные форматы структур данных со стандартизованными интерфейсами, подобно тому, как это происходило с развитием планетарной инфраструктуры транспортных сообщений с терминалами, отелями, заправками, ориентированными на сервисы, удовлетворяющие любые запросы пользователя.

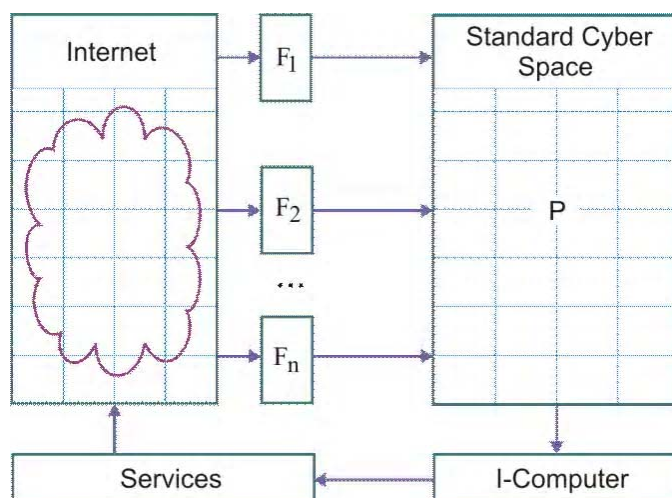


Рис. 2. Эволюция Cyber Space и Internet

В настоящее время отсутствуют стандарты формирования и хранения структур одно-типных данных на сайтах и порталах Internet. Это затрудняет понимание информации поисковыми системами для распознавания и принятия правильных решений. Трудно ожидать, что информационное пространство в одночасье перейдет на рельсы принятых всеми форматов и интерфейсов. Единственным выходом может служить эволюционирование структур данных. Для этого необходимо разрабатывать преобразователи (фильтры) форматов данных. Наличие первичных фильтров позволяет автоматизировать времязатратные процессы создания библиотек базовых примитивов. Например, нетрудно создать генераторы логических отношений (фильтры, см. рис. 2) для каждой части речи русского языка, ориентированные на анализ существующих электронных словарей или признанных эталонных текстов в целях формирования первоначальных таблиц отношений в виде <часть речи> <словообразующие компоненты>. Имея такого рода базовые примитивы, можно автоматически создавать более сложные конструкции и процесс-модели анализа, синтеза предложений, коррекции ошибок, что всегда будет актуально и востребовано на рынке информационно-компьютерных сервисов.

Имея спецификацию, представленную после обработки вербального описания в форме вектора входных и выходных переменных, нетрудно записать стратегию создания новой функциональности как задачу поиска покрытия библиотечными элементами обобщенного вектора $\langle X, Y \rangle$. Общее решение задачи похоже на синтез автоматной модели, определяющей взаимодействие компонентов во времени и в пространстве. Однако многообразие примитивов, заранее не заданных, исключает такую возможность, что означает – необходим переход из строгого детерминизма цифровых автоматов в область эволюционных и квазиоптимальных решений.

Условие задачи: имеется спецификация в виде вектора существенных переменных, которые необходимо покрыть минимальным множеством примитивов из библиотеки и сформировать выходной вектор. Красивое решение задачи синтеза функциональной структуры по спецификации есть ключ к самостоятельному генерированию компьютером новых решений. После этого остается решить только две задачи на пути к созданию интеллекта компьютера – самогенерирование оригинальной функциональности, необходимой для решения задачи покрытия, и спецификаций новых полезных для человека или компьютера сервисов.

2. Интегральная метрика оценивания диагноза

Инфраструктура мозгоподобных алгоритмов для обнаружения спама включает модели, методы и ассоциативно-логические структуры данных, ориентированные на поддержку процессов поиска, распознавания и принятия решений на основе векторных неарифметических операций. Оценка решения задачи определяется векторно-логическим критерием качества взаимодействия запроса (вектора m) с системой ассоциативных векторов (ассоциаторов), в результате которого сгенерируется конструктивный ответ в виде одного или нескольких ассоциаторов, а также, пока еще, численной характеристики степени принадлежности (функции качества) входного вектора m к найденному решению: $\mu(m \in A)$. Входной вектор $m = (m_1, m_2, \dots, m_i, \dots, m_q)$, $m_i \in \{0, 1, x\}$ и матрица A_i ассоциаторов $A_{ijr} (\in A_{ij} \in A_i \in A) = \{0, 1, x\}$ имеют одинаковую размерность, равную q . Далее степень принадлежности m -вектора к вектору A будет обозначаться как $\mu(m \in A)$.

Существует 5 типов теоретико-множественного (логического) Δ -взаимодействия двух векторов $m \cap A$, определенных на рис. 3. Они формируют все примитивные варианты реакции обобщенной ПРП-системы (Поиска, Распознавания и Принятия решения) на входной вектор-запрос. В технологической отрасли знаний – технической диагностике (Design & Test) – указанная последовательность действий изоморфна маршруту: поиск дефектов, их распознавание, принятие решения на восстановление работоспособности. Все три стадии технологического маршрута нуждаются в метрике оценивания решений для выбора оптимального варианта.

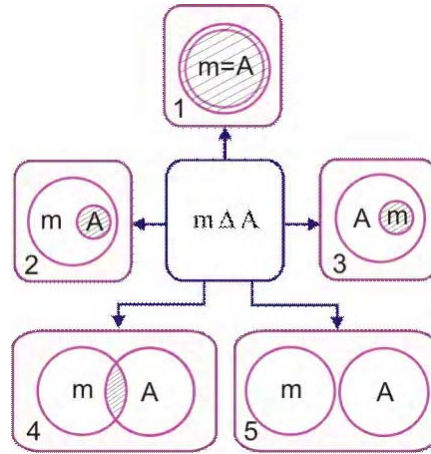


Рис. 3. Результаты пересечения двух векторов

Определение. Интегральная теоретико-множественная метрика для оценивания качества запроса есть функция взаимодействия многозначных векторов $m \cap A$, которая определяется средней суммой трех нормированных параметров: кодовое расстояние $d(m, A)$, функция принадлежности $\mu(m \in A)$ и функция принадлежности $\mu(A \in m)$:

$$\begin{aligned}
 Q &= \frac{1}{3} [d(m, A) + \mu(m \in A) + \mu(A \in m)], \\
 d(m, A) &= \frac{1}{n} [n - \text{card}(m_i \cap A_i = \emptyset)]; \\
 \mu(m \in A) &= 2^{\text{card}(m \cap A) - \text{card}(A)} \leftarrow \text{card}(m \cap A) = \\
 &= \text{card}(m_i \cap A_i = x) \& \text{card}(A) = \text{card}(\bigcup_{i=1}^n A_i = x); \\
 \mu(A \in m) &= 2^{\text{card}(m \cap A) - \text{card}(m)} \leftarrow \text{card}(m \cap A) = \\
 &= \text{card}(m_i \cap A_i = x) \& \text{card}(m) = \text{card}(\bigcup_{i=1}^n m_i = x).
 \end{aligned} \tag{1}$$

Пояснения. Нормирование параметров позволяет оценить уровень взаимодействия векторов в интервале $[0, 1]$. Если зафиксировано предельное максимальное значение каждого параметра, равное 1, то векторы равны между собой. Минимальная оценка, $Q = 0$, фиксируется в случае полного несовпадения векторов по всем n координатам. Если мощность пересечения $m \cap A = m$ равна половине пространства вектора A , то функции принадлежности и качества соответственно равны:

$$\mu(m \in A) = \frac{1}{2}; \mu(A \in m) = 1; d(m, A) = 1; Q(m, A) = \frac{5}{2 \times 3} = \frac{5}{6}.$$

Аналогичное значение будет иметь параметр, если мощность пересечения равна половине пространства вектора m . Если мощность пересечения равна половине мощностей пространств векторов m и A , то функции принадлежности равны:

$$\mu(m \in A) = \frac{1}{2}; \mu(A \in m) = \frac{1}{2}; d(m, A) = 1; Q(m, A) = \frac{4}{2 \times 3} = \frac{4}{6} = \frac{2}{3}.$$

Следует заметить, если пересечение двух векторов равно пустому множеству, то степень двойки от символа «пусто» принимается равным нулю: $2^{\text{card}(m \cap A) = \emptyset} = 2^{\emptyset} = 0$. Это действительно означает, что количество общих точек при пересечении двух пространств равно нулю.

Цель введения векторно-логического критерия качества решения заключается в существенном повышении быстродействия при подсчете качества Q взаимодействия компонентов m и A при анализе ассоциативных структур данных путем использования только векторных логических операций. Арифметический критерий (1) без усреднения функций принадлежности и кодового расстояния можно трансформировать к виду:

$$Q = d[m, A_{i(j)}] + \mu[m \in A_{i(j)}] + \mu[A_{i(j)} \in m],$$

$$d(m, A_{i(j)}) = \text{card}[m \oplus A_{i(j)} = 1];$$

$$\mu(m \in A_{i(j)}) = \text{card}[A_{i(j)} = 1] - \text{card}[m \wedge A_{i(j)} = 1];$$

$$\mu(A_{i(j)} \in m) = \text{card}[m = 1] - \text{card}[m \wedge A_{i(j)} = 1].$$
(2)

Первый компонент, составляющий критерий, формирует степень несовпадения n -мерных векторов – кодовое расстояние, путем выполнения операции хог, второй и третий определяют степень непринадлежности результата конъюнкции к числу единиц каждого из двух взаимодействующих векторов. Понятия принадлежности и непринадлежности являются взаимодополняющими, но в данном случае технологичнее вычислять непринадлежность. Таким образом, идеальный критерий качества равен нулю, когда два вектора равны между собой. Оценка качества взаимодействия двух двоичных векторов убывает по мере роста критерия от 0 к 1. Чтобы окончательно уйти от арифметических операций при подсчете уже векторного критерия качества, необходимо выражения (2) преобразовать к виду:

$$Q = d(m, A) \vee \mu(m \in A) \vee \mu(A \in m),$$

$$d(m, A) = m \oplus A;$$

$$\mu(m \in A) = A \wedge \overline{m \wedge A};$$

$$\mu(A \in m) = m \wedge \overline{m \wedge A}.$$
(3)

Здесь критерии представлены уже не числами, а векторами, которые оценивают взаимодействие компонентов m, A . При этом увеличение числа нулей в трех векторах качества повышает критерий, а наличие единиц индицирует ухудшение качества взаимодействия. Для сравнения оценок необходимо определять мощность единиц в каждом векторе без выполнения операций суммирования. Это можно сделать с помощью регистра (рис. 4), который позволяет за один такт выполнить сдвиг влево и уплотнить все единичные координаты n -разрядного двоичного вектора.

После процедуры сжатия номер правого единичного бита уплотненной серии единиц формирует индекс качества взаимодействия векторов. Для двоичных наборов $m = (110011001100), A = (000011110101)$ определение качества их взаимодействия по формулам (3) представлено в следующем виде (нулевые координаты отмечены точками):

m	1 1 . . 1 1 . . 1 1 . .
A	 1 1 1 1 . 1 . 1
$m \wedge A$	 1 1 . . . 1 . .
$m \wedge A$	1 1 1 1 . . 1 1 1 . 1 1
$d(m, A) = m \oplus A$	1 1 1 1 1 . . 1
$\mu(A \in m) = m \wedge \overline{m \wedge A}$	1 1 1 . . .
$\mu(m \in A) = A \wedge \overline{m \wedge A}$	 1 1 . . . 1
$Q = d(m, A) \vee \mu(m \in A) \vee \mu(A \in m)$	1 1 1 1 1 . . 1
$Q(m, A) = (6/12)$	1 1 1 1 1 1

Рис. 4. Регистр сдвига и уплотнения единиц

Здесь сформирована не только оценка взаимодействия векторов, равная $Q(m, A) = (6/12)$, но, что самое главное, единичные координаты строки $Q = d(m, A) \vee \mu(m \in A) \vee \mu(A \in m)$ идентифицируют все существенные переменные, по которым имеется некачественное взаимодействие векторов. Для сравнения двух решений, полученных в результате логического анализа, используются сжатые векторы качества Q , над которыми выполняется процедура, включающая следующие векторные операции:

$$Q(m, A) = \begin{cases} Q_1(m, A) \leftarrow \text{or}[Q_1(m, A) \wedge Q_2(m, A) \oplus Q_1(m, A)] = 0; \\ Q_2(m, A) \leftarrow \text{or}[Q_1(m, A) \wedge Q_2(m, A) \oplus Q_1(m, A)] = 1. \end{cases} \quad (4)$$

Вектор-бит ог -оператор девекторизации формирует двоичное битовое решение на основе применения логической операции ог к n разрядам вектора существенных переменных критерия качества. Схемотехническое решение процедуры выбора и аналитическая процесс-модель имеют три операции, которые представлены на рис. 5.

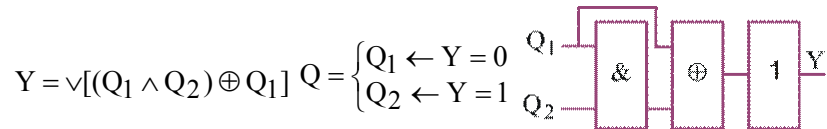


Рис. 5. Процесс-модель выбора решения

Для двоичных векторов, представляющих собой критерии качества, выполнена процедура выбора лучшего из них на основании выражения (рис. 6):

$Q_1(m, A) = (6, 12)$	1	1	1	1	1	1	.	.	.	.	.	.
$Q_2(m, A) = (8, 12)$	1	1	1	1	1	1	1	1	.	.	.	.
$Q_1(m, A) \wedge Q_2(m, A)$	1	1	1	1	1	1	.	.	.	.	.	.
$Q_1(m, A) \oplus Q_1(m, A) \wedge Q_2(m, A)$	.	.	.	.	.	.	.	.	.	.	.	.
$Q(m, A) = Q_1(m, A)$	1	1	1	1	1	1	.	.	.	.	.	.

Рис. 6. Выбор лучшего вектора качества

Векторные логические критерии качества взаимодействия ассоциативных наборов позволяют получать оценку поиска, распознавания и принятия решения с высоким быстродействием логических параллельных операций, что особенно существенно для критических систем реального времени.

3. Процесс-модель диагностирования СПАМА

Метрика качества, представленная в (3), дает возможность оценивать близость пространственных объектов друг к другу, а также взаимодействие векторных пространств. Практическим примером полезности интегрального критерия качества может служить стрельба по цели, которая иллюстрируется ранее приведенными диаграммами (см. рис. 3) взаимодействия векторов: 1) Снаряд попал точно в цель и поразил ее полностью. 2) Мишень поражена необоснованно большим калибром снаряда. 3) Калибр снаряда недостаточен для поражения крупной цели. 4) Неэффективный и неточный выстрел снарядом большого калибра. 5) Снаряд пролетел мимо мишени.

Процесс-модель взаимодействия $P(m, A)$ сопровождается интегральным критерием качества, который оценивает не только попадание или промах, но и эффективность использования калибра оружия. Аналитическая запись процесс-модели для выбора лучшего взаимодействия входного запроса m с системой логических ассоциативных отношений представлена в виде:

$$P(m, A) = \min_{i=1}^n Q_i(m \Delta A_i) = \vee[(Q_i \bigwedge_{j=1, n, j \neq i} Q_j) \oplus Q_i] = 0;$$

$$Q(m, A) = (Q_1, Q_2, \dots, Q_i, \dots, Q_n);$$

$$A = (A_1, A_2, \dots, A_i, \dots, A_n);$$

$$\Delta = \{\text{and, or, xor, not, slc, nop}\};$$

$$\begin{aligned}
A_i &= (A_{i1}, A_{i2}, \dots, A_{ij}, \dots, A_{is}); \\
A_{ij} &= (A_{ij1}, A_{ij2}, \dots, A_{ijr}, \dots, A_{msq}); \\
m &= (m_1, m_2, \dots, m_r, \dots, m_q). \\
Q_i &= d(m, A_i) \vee \mu(m \in A_i) \vee \mu(A_i \in m), \\
d(m, A_i) &= m \oplus A_i; \\
\mu(m \in A_i) &= A_i \wedge \overline{m \wedge A_i}; \\
\mu(A_i \in m) &= m \wedge \overline{m \wedge A_i}.
\end{aligned} \tag{5}$$

Комментарии: 1) Функциональность $P(m, A)$ задает аналитическую модель вычислительного процесса в виде высказывания, минимизирующего интегральный критерий качества. 2) Структуры данных представлены в виде вершин-таблиц графа $A = (A_1, A_2, \dots, A_i, \dots, A_m)$, логически взаимодействующих между собой. 3) Вершина графа задается упорядоченной совокупностью вектор-строк ассоциативной таблицы $A_i = (A_{i1}, A_{i2}, \dots, A_{ij}, \dots, A_{is})$ явных решений, где строка $A_{ij} = (A_{ij1}, A_{ij2}, \dots, A_{ijr}, \dots, A_{msq})$ представляет собой истинное высказывание.

Поскольку функционал, представленный в виде таблицы, не имеет постоянных во времени входных и выходных переменных, то данная структура отличается от последовательной машины фон Неймана, задаваемой конечными автоматами Мили и Мура. Равнозначность всех переменных в векторе $A_{ijr} = (A_{ij1}, A_{ij2}, \dots, A_{ijr}, \dots, A_{msq})$ создает одинаковые условия их существования, что означает инвариантность решения задач прямой и обратной импликации в пространстве $A_i \in A$. Ассоциативный вектор A_{ij} определяет собой явное решение, где каждая переменная задается в конечном, многозначном и дискретном алфавите $A_{ijr} \in \{\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_k\} = \beta$. Взаимодействие $P(m, A)$ входного вектора-запроса $m = (m_1, m_2, \dots, m_r, \dots, m_q)$ с графом $A = (A_1, A_2, \dots, A_i, \dots, A_m)$ формирует множество решений с выбором лучшего из них по минимальному критерию качества:

$$P(m, A) = \min Q_i [m \wedge (A_1 \vee A_2 \vee \dots \vee A_i \vee \dots \vee A_m)].$$

Конкретное взаимодействие вершин графа между собой создает функциональность $A = (A_1, A_2, \dots, A_i, \dots, A_m)$, которая может быть оформлена в следующие структуры: 1) Единственная ассоциативная таблица, содержащая все решения логической задачи в явном виде. Преимущество – максимальное быстроедействие параллельного ассоциативного поиска решения по таблице. Недостаток – максимально высокая аппаратная сложность размещения таблицы большой размерности. 2) Древовидная (графовая) структура бинарных отношений между функциональными примитивами, каждый из которых формирует таблицу истинности для незначительного количества переменных. Преимущество – максимально низкая аппаратная сложность решения задачи. Недостаток – минимальное быстроедействие последовательного ассоциативного поиска решения по дереву. 3) Компромиссная графовая структура логически понятных для пользователя отношений между примитивами, каждый из которых формирует таблицу истинности для логически сильно взаимосвязанных переменных.

В целях детализации структуры векторных вычислений ниже представлены аналитические и структурные процесс-модели, которые приводятся к анализу A -матрицы по столбцам или строкам. Первая из них изображена на рис. 7 и предназначена для определения множества допустимых решений относительно входного запроса m_b .

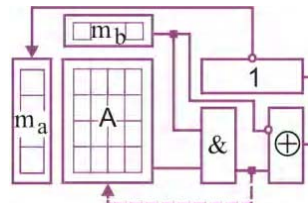


Рис. 7. Поиск всех допустимых решений

Вторая структура (рис. 8) осуществляет поиск оптимального решения на множестве найденных в первой процессной модели путем анализа строк. Кроме того, вторая модель имеет и самостоятельное применение, ориентированное на определение однозначного и многозначного решения, например, при поиске дефектов в цифровой системе на кристалле.

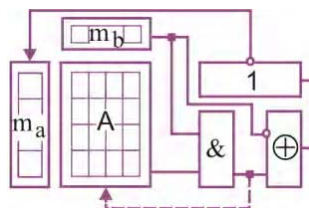


Рис. 8. Структура выбора оптимального решения

Предложенные процесс-модели анализа (графа) ассоциативных таблиц, идентифицирующих компоненты СПАМа, а также введенные критерии качества логических решений позволяют решать задачи квазиоптимального покрытия, диагностирования многообразия спамовых сообщений в индивидуальном кибернетическом пространстве (ИКП) пользователя. Модель векторных вычислений послужила основой для разработки специализированной мультипроцессорной архитектуры, ориентированной на поиск, распознавание и принятие решений на использование структур ассоциативных таблиц (рис. 9).

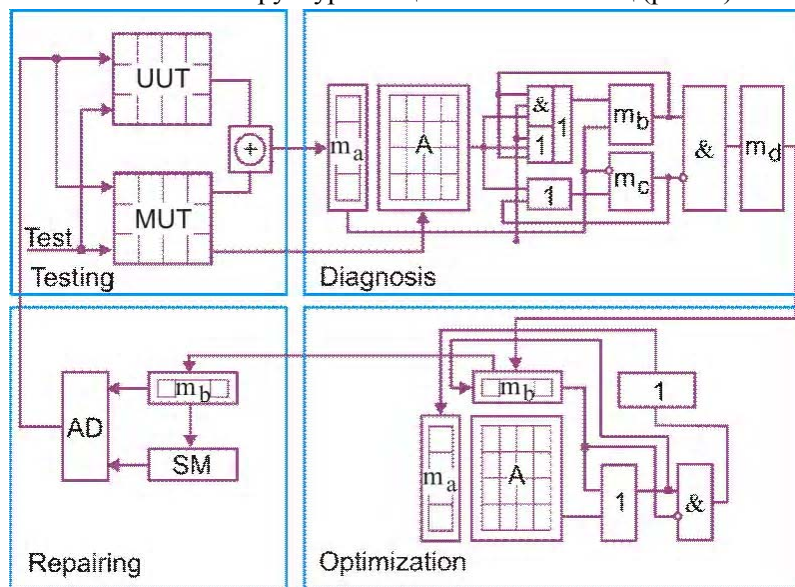


Рис. 9. Модель встроенного тестирования и восстановления ИКП

Оценка эффективности (рис. 10) проектного решения под эгидой специализации и стандартизации $S_p \cup S_t$ основывается на совместном использовании трех взаимно противоречивых параметров: качество Y , быстродействие T , программные затраты H :

$$E = F(Y, T, H),$$

$$Y = (1 - P)^{n(1-Q)};$$

$$T = \frac{1}{f} \times S \times d;$$

$$H = 2(H^S \times n).$$

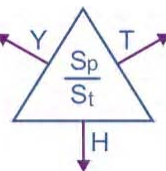


Рис. 10. Оценка эффективности процесс-модели

4. Практические результаты внедрения инфраструктуры

В качестве объекта исследования был выбран SquirrelMail — клиент электронной почты с веб-интерфейсом, написанным на PHP. Приложение может быть проинсталлировано практически на любом веб-сервере, на котором установлен PHP и имеется связь с

почтовым сервером по протоколам IMAP и SMTP. Интерфейсное окно представлено на рис. 11.

Данный клиент достаточно легко расширяем различными плагинами. Для проведения исследования был написан плагин, который реализовывал анализ определения полезности информации на основании предпочтений пользователей.



Рис. 11. Интерфейс SquirrelMail

Процесс-модель функционирования плагина представлена на рис. 12.

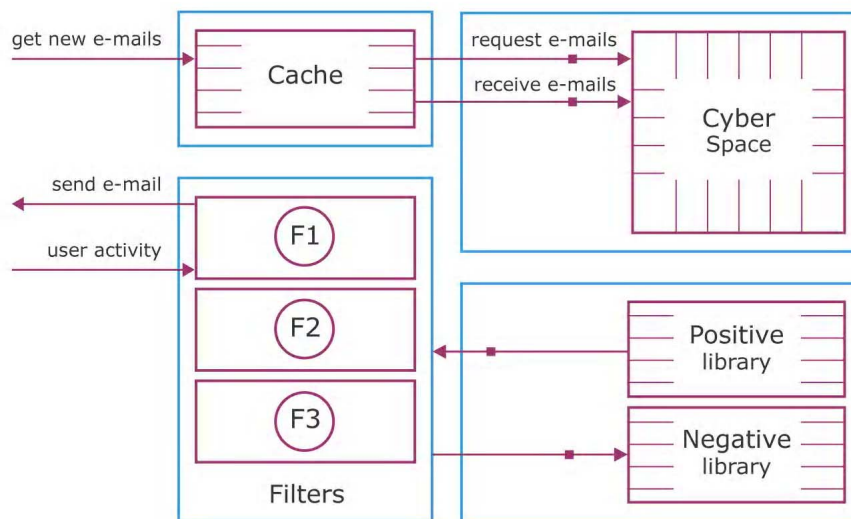


Рис. 12. Процесс-модель функционирования плагина под SquirrelMail

На основании активности пользователя и атрибутов, контентного анализа письма была выбрана и обучена система фильтров. При загрузке новых писем из индивидуального киберпространства (в данном случае киберпространство было представлено подмножеством электронных писем) информация фильтровалась не на основании «спам» или «не спам», а на основании персональных предпочтений пользователя (рис. 13).

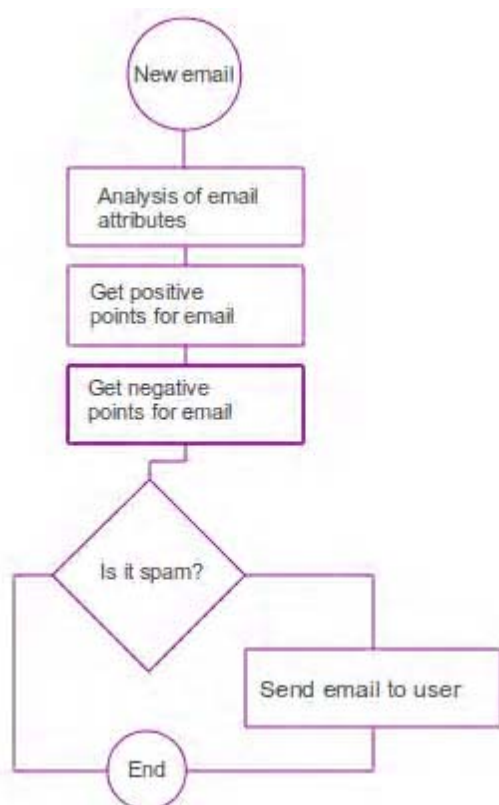


Рис. 13. Граф-схема анализа письма

Из всего множества писем, приходящих на почтовый ящик, «спам» (анонимные массовые рассылки) был частично идентифицирован уже не как спам, а как рассылка, которая может нести интересную информацию для пользователя.

На рис. 14 показана эффективность внедрения инфраструктуры сервисного обслуживания ИКП для одного пользователя, где TL – Total Letters, S – Spam, SAI – Spam after Infrastructure, UL – Useful Letters. Если предположить, что рыночная привлекательность инфраструктуры порядка 1 млрд пользователей, то экономия времени в целом на рынке пользователей кибернетического пространства составит (T_{Σ} – суммарная экономия времени за год; k – коэффициент уменьшения спама при внедрении инфраструктуры; L – количество писем в месяц; N – потенциальное количество пользователей в Украине; T – время анализа одного письма; M – количество месяцев в году; H_{Σ} – годовая экономия финансовых средств от внедрения инфраструктуры; C_h – стоимость одного часа рабочего времени пользователя в Украине):

$$T_{\Sigma} = k \times L \times N \times T \times M = 0,9 \times 800 \times 10000000 \times 1 \times 12 = 8640000000 \text{с} \approx 2740 \text{ years} \approx 24002400 \text{ hours};$$

$$H_{\Sigma} = T_{\Sigma} \times C_h = 24002400 \times \$5 = \$120012000.$$

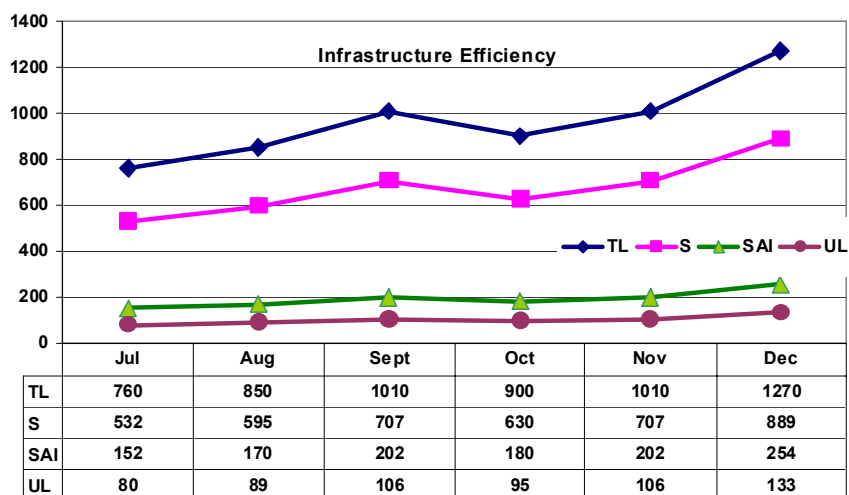


Рис. 14. Эффективность внедрения инфраструктуры ИКП

Заключение

1. *Научная новизна* результатов исследования заключается в том, что впервые предложена инфраструктура сервисного обслуживания индивидуального кибернетического пространства, которая характеризуется наличием встроенных средств тестирования, диагностирования и восстановления ИКП и двух пополняемых библиотек позитивных и негативных сообщений, что дает возможность существенно (в несколько раз) уменьшить время анализа получаемой информации.

2. *Практическая значимость результатов исследования:* инфраструктура сервисного обслуживания ИКП ориентирована на повышение качества жизни всех субъектов планеты, использующих почтовые сервисы для коммуникации с внешним миром. При этом ИКП – модель будущего общения человека с внешним миром, которая инвариантна по отношению к техническим средствам доступа в киберпространство планеты. Годовой экономический эффект от внедрения инфраструктуры ИКП для украинских пользователей может составить более 120 млн долларов.

3. *Направление будущих научных исследований.* Актуальной представляется проблема создания теории, методов и архитектуры параллельного анализа информации, представленной в виде аналитических, графовых и табличных форм ассоциативных отношений для поиска, распознавания, диагностирования деструктивных компонентов и принятия решений в n -мерном векторном дискретном пространстве. Здесь целесообразно использовать векторно-логические процесс-модели актуальных прикладных задач, в том числе – диагностирование вирусов и восстановление работоспособности программно-аппаратных компонентов компьютерных систем, качество решения которых оценивается неарифметической метрикой взаимодействия булевых векторов. Решение проблемы ориентировано на поиск, распознавание, диагностирование деструктивных компонентов аппаратно-программными методами в дискретном кибернетическом пространстве. Общность представленной теории синтеза и анализа кибернетического пространства основана на использовании равенства нулю триады равноценных компонентов, соединенных операциями $\text{hog } m \oplus A \oplus Q = 0$, формулирующей условия решения проблемы. Здесь первый компонент m – входной код программы, второй A – эталонные модели деструктивов, третий Q – результат взаимодействия первых двух, который может вырождаться в критерий качества отношения или принятия решения, оценку распознавания объектов или образов. Цель – существенное повышение качества программных продуктов и уменьшение стоимости эксплуатационных расходов за счет их вакцинации путем введения в код встроенной программной избыточности в виде инфраструктуры сервисного обслуживания, обеспечивающей тестирование, диагностирование и устранение вредоносных компонентов, классифицируемых в библиотеках. Объект исследования – кибернетическое пространство, представленное информацией, ее носителями и преобразователями, а также деструктивными компонентами, наносящими вред функциональностям, улучшающим качество жизни человека. Субъект исследования – инфраструктура сервисного обслуживания в виде встроенной программной избыточности, работающей в реальном масштабе времени, обеспечивающей тестирование, диагностирование и устранение вредоносных компонентов, описанных в соответствующих библиотеках.

Мотивация: 1) Отсутствие на рынке антивирусной защиты встроенных средств тестирования, диагностирования и удаления вредоносных компонентов, составляющих инфраструктуру сервисного обслуживания, подобной тому, как в цифровых системах на кристаллах существуют стандарты граничного сканирования, а в программных продуктах – ассерционная избыточность, ориентированные на встроенное тестирование дефектов и ошибок с последующим восстановлением работоспособности аппаратных или программных изделий. 2) Наличие теоретических разработок, связанных с технологией алгебрологического векторного анализа информационных данных, ориентированных на высокое быстродействие решения и оценивания задач распознавания образов, принятия решений и тестирования объектов. 3) Наличие образцовой производственной и маркетинговой инфраструктуры (Лаборатория Касперского), способной поддерживать проект создания технологии вакцинации программных продуктов и авторитетно предложить его рынку информационных технологий. 4) Миниатюризация цифровых и телекоммуникационных систем (телефоны смартфоны, IP-фоны, планшеты) требует постоянной защиты от несанкционированного доступа путем внедрения встроенных антивирусных средств, контролирующих информационный обмен. Задачи: 1) Разработка математического аппарата анализа кибернетического пространства, ориентированного на создание моделей и методов сервисного обслуживания программных продуктов для тестирования, диагностирования и устранения вредоносных компонентов. 2) Создание типовых процесс-моделей и критериев взаимодействия вредоносных компонентов с программными кодами полезных функциональностей. 3) Разработка технологии

анализа структуры программного кода для определения критических точек и установки в них ассерционных операторов наблюдения и управления в процессе его функционирования. 4) Создание инфраструктуры сервисного обслуживания функциональных программ для встроенного тестирования, диагностирования и устранения вредоносных компонентов из программного кода функциональности на основе использования библиотеки деструктивных элементов. 5) Тестирование и верификация встроенной инфраструктуры сервисного обслуживания функциональностей, защищающей программный код от вредоносных компонентов.

4. *Ожидаемые результаты и их рыночная привлекательность*: 1) Инфраструктура встроенной защиты программного кода от несанкционированной модификации, приводящей к изменению функциональности. 2) Избыточность инфраструктуры программного кода, которая автоматически синтезируется на стадии проектирования и верификации, составляет не более 5% от специфицированной функциональности. 3) Рыночная привлекательность инфраструктуры, определяемая многообразием программных продуктов, умноженной на уровень продаж каждого изделия, составляет в год порядка одного миллиарда экземпляров. 4) Стоимость создания инфраструктуры для программного продукта составляет 20% затрат от разработки функционального кода. Если уровень продаж – не менее 500 копий, то затраты на создание встроенного антивируса вполне окупаемы в течение года. 5) Внедрение запатентованной технологии вакцинации программных продуктов при их рождении может принести компании порядка 2-х миллиардов долларов в первые 3 года ее эксплуатации. 6) Маркетинговая проблема глобальной компании (Лаборатория Касперского) заключается в убеждении разработчиков программных продуктов имплементировать существующие внешние антивирусы вовнутрь кода полезной функциональности.

Список литературы: 1. *Инфраструктура мозгоподобных вычислительных процессов* / М.Ф. Бондаренко, О.А. Гузь, В.И. Хаханов, Ю.П. Шабанов-Кушнарченко. Харьков: Новое слово, 2010. 160 с. 2. *Проектирование и тестирование цифровых систем на кристаллах. Verilog & System Verilog* / В.И. Хаханов, Е.И. Литвинова, О.А. Гузь. Харьков: ХНУРЭ. 2009. 484с. 3. *Проектирование и верификация цифровых систем на кристаллах* / В.И. Хаханов, И.В. Хаханова, Е.И. Литвинова, О.А. Гузь. Харьков: Новое слово, 2010. 528с. 4. *Семенец В.В., Хаханова И.В., Хаханов В.И.* Проектирование цифровых систем с использованием языка VHDL. Харьков: ХНУРЭ. 2003. 492 с. 5. *Хаханов В.И., Хаханова И.В.* VHDL+Verilog = синтез за минуты. Харьков: ХНУРЭ. 2006. 264с. 6. *Хаханов В.И., Чумаченко С.В.* Модели пространств в научных исследованиях // Радиоэлектроника и информатика. 2002. №1. С. 124-132. 7. *Zorian Yervant.* Guest Editor's Introduction: Advances in Infrastructure IP // IEEE Design and Test of Computers. 2003. P.49-55. 8. *Bulent I. Dervisoglu.* A Unified DFT Architecture for Use with IEEE 1149.1 and VSIA/IEEE P1500 Compliant Test Access Controllers. Proceedings of the Design Automation Conference. 2001. P. 53-58. 9. *Bergeron J.* Writing Testbenches using SystemVerilog. Springer US. 2006. 414 p.

Поступила в редколлегию 11.11.2010

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. 70-21-326. E-mail: hahanov@kture.kharkov.ua.

Чумаченко Светлана Викторовна, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: математическое моделирование, методы дискретной оптимизации. Увлечения: спорт, путешествия, любительское фото. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326, e-mail: ri@kture.kharkov.ua.

Мищенко Александр Сергеевич, аспирант кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем, сетей, антивирусная защита, web-технологии. Увлечения: английский язык, web-дизайн. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326, e-mails: alex@simplesolutions.com.ua, alexander.mischenko@gmail.com, santific@gmail.com.

Зацарный Артем Вячеславович, студент гр. КИ-08-6 факультета КИУ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем, сетей, антивирусная защита, web-технологии. Увлечения: футбол. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.

Хаханова Юлия Владимировна, студентка гр. КИ-08-6 факультета КИУ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем, сетей, антивирусная защита, web-технологии. Увлечения: футбол, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.

АНАЛІТИЧНА МОДЕЛЬ ДИСКРЕТНО-БЕЗПЕРЕРВНОЇ СИСТЕМИ З ЗАСТОСУВАННЯМ АПАРАТА MAX-PLUS АЛГЕБРИ

Розглядається один з варіантів розв'язку задачі побудови моделей дискретно-безперервних систем з використанням апарата MAX-PLUS алгебри. Наводиться типова структура переходу графа синхронізації ДБС. Усі твердження розглядаються на абстрактному прикладі мережі Петрі.

Вступ

Теорія дискретно-безперервних систем (ДБС) – відносно новий напрямок застосування сучасної теорії систем для широкого кола задач управління, що мають місце в технологічних виробничих процесах, телекомунікаційних мережах, транспортних і логістичних системах. Характерною особливістю таких процесів є те, що їх динаміка залежить не тільки від часу, а й зумовлена внутрішніми дискретними подіями, які супроводжують розвиток процесу [1]. Деякі автори використовують акронім ДБДС – дискретно безперервні динамічні системи, тим самим підкреслюючи фактор розвитку дискретних станів, що характеризують систему у часі [3].

Для синтезу систем управління дискретно-безперервними процесами актуальною є задача розробки аналітичних моделей ДБС, які дозволяють застосовувати при синтезі підходи, характерні для сучасної теорії управління. В рамках цієї статті розглянуто один з можливих варіантів розв'язку задачі побудови моделей таких систем з використанням апарата одного з різновидів ідемпотентної алгебри – Max-Plus алгебри [2] для графа синхронізації ДБС.

Постановка задачі досліджень

Поняття графа синхронізації дискретно-безперервних систем стосовно управління процесами в таких системах вперше було введено в теорії моделювання динаміки ДБС за допомогою апарату мереж Петрі [1]. Але формальний опис дискретно-безперервних процесів за правилами мереж Петрі не є прийнятним в рамках підходів, які використовуються в сучасній теорії управління. При застосуванні апарата Max-Plus-алгебри можна використовувати векторно-матричні рівняння, які дозволяють формалізувати представлення ДБС у формі, подібній до моделей динаміки в просторі змінних стану [4].

Типова структура окремого переходу графа синхронізації ДБС в загальному випадку представлена на рис. 1.

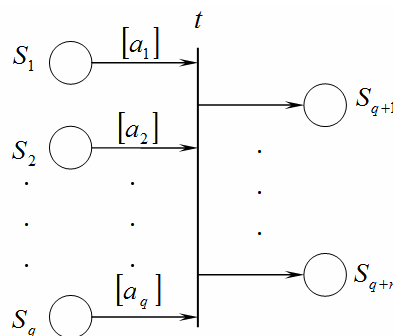


Рис. 1. Структура переходу графа синхронізації ДБС

В даному випадку розглядається перехід між станами, який має назву некерованого. Перехід в інший стан або спрацювання некерованого переключення відбувається лише тоді, коли всі стани від S_1 до S_q в передобласті переходу будуть зайняті умовним маркером, як це задається часовими оцінками сповільнення процесу a_i .

Якщо в графі розглядається керований ззовні перехід, то прийнято вважати, що він відбувається при додатковій логічній умові переходу, заданій апіорі. Часова відмітка, в якій логічна умова є дійсною, позначається u і тоді для маркованих часових точок стану ДБС в післяобласті керованого переходу маємо:

$$x_j = [\bigoplus_{i=1}^q a_i x_i] \oplus u, j = q+1, \dots, q+r. \quad (1)$$

Опис переміщень в графі синхронізації ДБС

Для кожного переміщення в графі синхронізації ДБС таким чином можна встановити часові точки переходу і марковані часові відмітки заняття стану в післяобласті. Розглянутий підхід для окремого переходу може бути узагальненим на всю послідовність переходів графа синхронізації ДБС. Нехай заданий деякий граф синхронізації системи з $|S|$ положеннями і P логічними зовнішніми умовами переключення переходів. Використовуючи підхід для окремого переходу (1) в цілому для графа синхронізації, можна отримати систему з n рівнянь:

$$\begin{aligned} x_1 &= a_{11}x_1 \oplus \dots \oplus a_{1n}x_n \oplus b_{11}u_1 \oplus \dots \oplus b_{1p}u_p \\ &\vdots \\ x_n &= a_{n1}x_1 \oplus \dots \oplus a_{nn}x_n \oplus b_{n1}u_1 \oplus \dots \oplus b_{np}u_p \end{aligned} \quad , \quad (2)$$

або у векторно-матричній формі:

$$\mathbf{x} = \mathbf{A} \mathbf{x} \oplus \mathbf{B} \mathbf{u}. \quad (3)$$

При цьому: $x = [x_1 \dots x_n]$ – вектор стану, кожний елемент x_i якого фіксує момент часу маркування (включення) переходу S_i . Вплив зовнішніх логічних умов маркування (переключення) задається за допомогою вектора керування: $u = [u_1 \dots u_n]$.

Кожна логічна умова маркування (переключення) переходу S_i стає дійною в деякій часовій точці розвитку процесу, яка також враховується в рівняннях (2). В цьому сенсі кожний окремий елемент $u_i, i = 1, \dots, p$ вектора управління розглядається як управляючий вплив на i -й перехід.

Значення кожного елемента a_{ij} матриці A рівняння (3) відповідає часовій оцінці передобласті переходу $T_i(S_i, t_k)$ з положення S_i в положення S_j , тобто характеризує часову інерційність переходу.

Якщо на переміщення t_k додатково впливає логічна умова переключення, наприклад u_r , то елемент матриці B дорівнює 0, в іншому випадку $b_{jr} = -\infty$.

Елементи матриці В таким чином вказують на ті переходи ДБС, які є керованими ззовні. В графах синхронізації переходи з одного положення в інше створюють замкнені цикли. Для того щоб відрізнити окремі цикли поведінки ДБС, всі змінні вектора стану x і вектора управління u мають індекс k , який показує, з якою частотою буде маркуватися відповідне положення в графі синхронізації, тобто $x_i(k)$ – це часова точка, в якій положення S_i займається (маркується) k -й раз.

Для встановлення початку нового переміщення по циклу має бути визначено положення S_i , зайняття (маркування) якого означає, що закінчено цикл k і починається $k+1$ цикл переміщення по графу синхронізації. Зазвичай для вибору границі початку нового циклу на графі задають початкову (стартову) позицію, відносно якої і ведуть відлік початку нового циклу. В цьому випадку в термінах змінних стану говорять про розрахунок стану $x(k+1)$ через $x_i(k)$. Якщо ж положення S_i не є стартовою позицією циклу, то $x_i(k+1)$ знаходиться через $x_j(k+1)$, тобто відносно часової відмітки в поточному циклі. Наведені ствердження для окремої стартової позиції графа синхронізації, що розглядається на рис. 1, матимуть таке представлення з урахуванням впливу зовнішнього логічного управління u :

$$x_j(k+1) = a_1 x_1(k) + \left(\bigoplus_{i=2}^{q-1} a_j x_j(k+1) \right) \oplus a_q x_q(k) \oplus u(k+1).$$

Індекс зовнішнього управління прийнятий таким самим, як індекс змінних стану відповідного циклу.

Як приклад розглядається граф синхронізації, що має структуру, представлену на рис. 2.

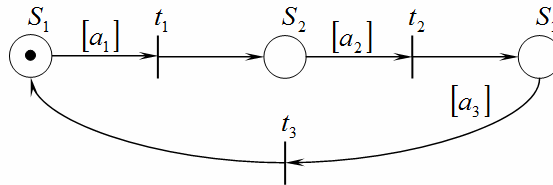


Рис. 2. Простий граф синхронізації

Як стартова позиція початку циклу умовно взята позиція S_2 . Тоді система рівнянь зі змінними стану в загальному випадку матиме вигляд:

$$\mathbf{x}(k+1) = \mathbf{A}_0 \mathbf{x}(k+1) \oplus \mathbf{A}_1 \mathbf{x}(k) \oplus \mathbf{B} \mathbf{u}(k+1). \quad (4)$$

Залежно від початкового маркування позиції S_i матриця A розбивається на дві A_0 і A_1 , причому:

$$\mathbf{A} = \mathbf{A}_0 \oplus \mathbf{A}_1. \quad (5)$$

Матриці A_0 і A_1 визначаються із матриці A за допомогою матриць трансформації T_{A_0} і T_{A_1} . Через матриці трансформації враховуються можливі різні початкові маркування позиції S_i так:

$$(T_{A_0})_{ij} = \begin{cases} \epsilon, & \text{якщо } i = j \text{ і положення } S_i \text{ не марковане,} \\ \epsilon - \text{ в усіх інших випадках;} \end{cases} \quad (6)$$

$$(T_{A_1})_{ij} = \begin{cases} e, & \text{якщо } i = j \text{ і положення } S_i \text{ марковане,} \\ \varepsilon - & \text{в усіх інших випадках.} \end{cases} \quad (7)$$

Тут $e=0, \varepsilon=-\infty$ – загальноприйняті поняття в MAX-PLUS алгебрі.

З визначення матриць (6) і (7) витікає, що тільки елементи головної діагоналі матриць трансформації можуть бути відмінними від e . Сума обох матриць завжди дає одиничну матрицю.

Система рівнянь (4) може бути розв'язана відомими методами:

[illegible]

Введемо позначення: $A_0^* = I \oplus A_0 \oplus A_0^2 \oplus \dots \oplus A_0^{n-1}$, $M = A_0^* A_1$; рівняння (8) спрощується:

$$\mathbf{x}(k+1) = \mathbf{M}\mathbf{x}(k) \oplus \mathbf{A}_0^* \mathbf{B}\mathbf{u}(k+1). \quad (9)$$

В цьому випадку матриця M може розглядатися як матриця динаміки системи без зовнішнього керування – динамічна характеристика вільної поведінки системи:

$$\mathbf{x}(k+1) = \mathbf{M} \mathbf{x}(k) .$$

Динамічні характеристики ДБС залежать від структури і властивостей матриці M . Початково марковані позиції графа синхронізації завжди розглядаються як вершини графа $G(M)$, які мають тільки вихідні ребра, і тому очевидно, що для кожного початково маркованого положення S_i графа синхронізації в матриці A_1 є хоча б один елемент в стовпці $(A_1)_{\bullet j}$, відмінний від ε , що говорить про те, що існує наступне положення S_k , в яке обов'язково переходить система з положення S_i . Якщо ж положення S_i не є початково маркованим, то j -й стовпець матриці $(A_1)_{\bullet j}$ має тільки елементи ε .

Поведінку в часі некерованого графа синхронізації можна дослідити за допомогою рівняння (9), якщо задані початкові умови $x(0)$. З урахуванням того, що циклічність критичного графа $G^C(M)$ дорівнює 1, існує число K , таке що:

$$\forall k \geq K : M^k = \lambda^k Q,$$

звідки випливає $x(k) = \lambda^k Qx(0) = \lambda^k v$.

На основі цього співвідношення можна стверджувати, що поведінка графа синхронізації для будь-яких початкових умов визначається сталим станом через деяку кількість пробігів по циклу залежно від власного вектора v матриці M .

В загальному випадку:

$$\forall k \geq K : M^{k-\rho} = \lambda^\rho M^k,$$

$$x(k + \rho) = M^{k+\rho} x(0) = \lambda^\rho x(k),$$

тобто залежно від циклічності ρ графа $G^C(M)$ заняття відповідних позицій графа синхронічно повторюється після λ^ρ одиниць часу.

Прийнято вважати, що для $k < K$ відповідні позиції займаються в нерегулярні часові проміжки, і цей період може розглядатись як перехідний процес в системі (рис.3):

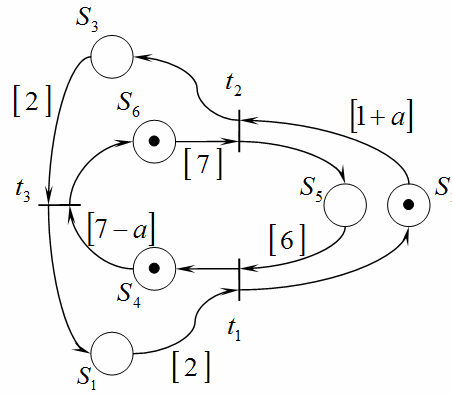


Рис. 3. Граф синхронізації мережі Петрі для абстрактного прикладу

Для ілюстрації наведених тверджень розглянемо абстрактний приклад. Нехай ДБС має шість операційних позицій $S_1, \dots, S_6$, три з яких S_2, S_4, S_6 – є початково маркованими.

Спочатку часовий параметр затримки a покладаємо рівним 0. В цьому випадку матриця динаміки A матиме вигляд:

$$A = \begin{bmatrix} \varepsilon & \varepsilon & 2 & 7 & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon & \varepsilon & 6 & \varepsilon \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 7 \\ 2 & \varepsilon & \varepsilon & \varepsilon & 6 & \varepsilon \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 7 \\ \varepsilon & \varepsilon & 2 & 7 & \varepsilon & \varepsilon \end{bmatrix}.$$

З урахуванням відсутності додаткових логічних умов переключення немає необхідності в матриці В. З урахуванням (9) знаходимо A_0^* і М:

$$A_0^* = \begin{bmatrix} 0 & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon \\ 2 & 0 & 4 & \varepsilon & 6 & \varepsilon \\ \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon \\ 2 & \varepsilon & 4 & 0 & 6 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon \\ \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 0 \end{bmatrix}, \quad M = \begin{bmatrix} \varepsilon & 3 & \varepsilon & 7 & \varepsilon & 9 \\ \varepsilon & 7 & \varepsilon & 9 & \varepsilon & 13 \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 7 \\ \varepsilon & 7 & \varepsilon & 9 & \varepsilon & 13 \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 7 \\ \varepsilon & 3 & \varepsilon & 7 & \varepsilon & 9 \end{bmatrix}.$$

Структура графа G(M) матиме вигляд (рис.4):

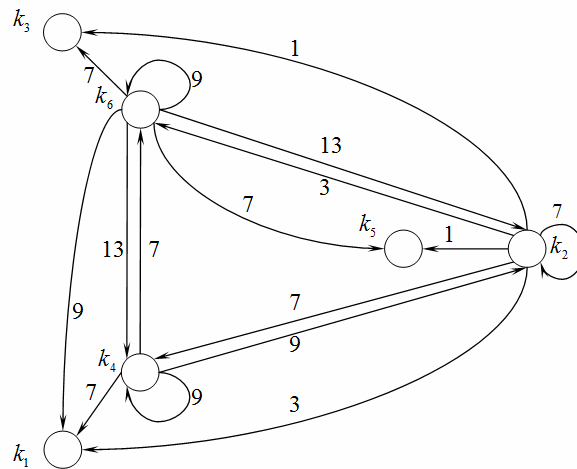


Рис. 4. Граф G(M)

Критичний граф $G^C(M)$ представлений на рис. 5.

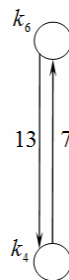


Рис. 5. Критичний граф $G^C(M)$

Середня вага критичного циклу є власне число матриці $M - \lambda = (13 + 7) / 2 = 10$ одиниць часу.

Можна перевірити достовірність цього результату і за допомогою будь-якого з алгоритмів формального знаходження λ для матриці М. Для власного числа матриці М існує власний вектор: $v = [3 \ 6 \ 0 \ 6 \ 0 \ 3]^T$.

При цьому перехідний процес закінчується для будь-яких початкових умов власним станом ДБС.

Поведінку графа синхронізації ДБС можна більш наочно пояснити за допомогою графіків, представлених на рис. 6.

З рис.6 видно, що проміжок часу між k -м і $(k+1)$ -м маркуваннями в період перехідного процесу може бути більшим, ніж власне число λ матриці M . Після закінчення перехідного режиму маркування позиції відбувається по чергові через 9 і 11 одиниць часу. Середнє значення цього часового проміжку точно дорівнює власному числу матриці M .

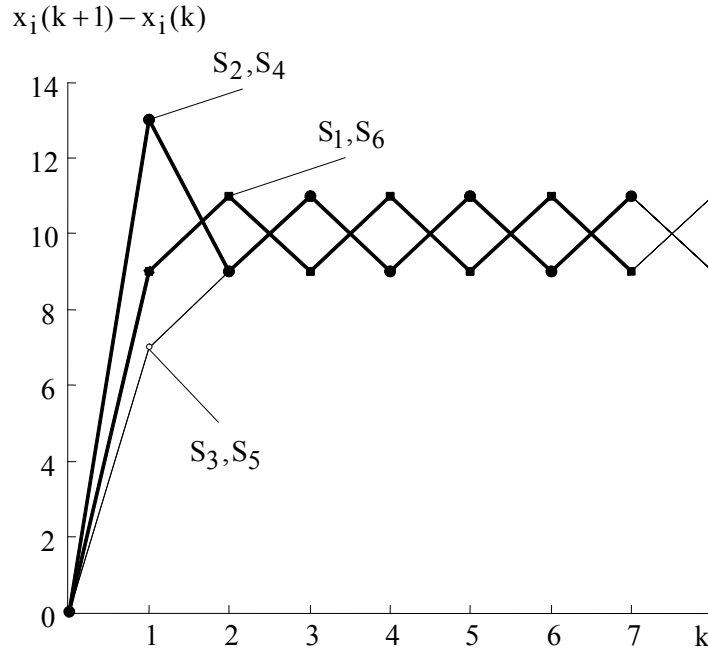


Рис. 6. Перехідний процес в ДБС при циклічності $\rho = 2$

Якщо змінити динаміку графа синхронізації, умовно прийнявши, що параметр $a = 2$ в графі на рис. 3, одержимо нову матрицю:

$$M' = \begin{bmatrix} \varepsilon & 5 & \varepsilon & 5 & \varepsilon & 9 \\ \varepsilon & 9 & \varepsilon & 7 & \varepsilon & 13 \\ \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & 7 \\ \varepsilon & 9 & \varepsilon & 7 & \varepsilon & 13 \\ \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & 7 \\ \varepsilon & 5 & \varepsilon & 5 & \varepsilon & 9 \end{bmatrix}$$

з власним числом $\lambda = 9$ і власним вектором $v = [2 \ 6 \ 0 \ 6 \ 0 \ 2]^T$, крім того, циклічність графа зміниться на $\rho = 1$.

Графічна інтерпретація поведінки ДБС в динаміці наочно проілюстрована на рис. 7.

Впродовж часу між k і $(k+1)$ маркуваннями часовий відрізок може змінюватись (коливатись). Ця поведінка може трактуватись як перехідний процес в ДБС.

Якщо має місце $k \geq K$, то позиції маркуються через регулярні проміжки часу. Виникає послідовність маркувань, що повторюється залежно від циклічності графа $G(M)$. Якщо для критичного графа $G^C(M)$ циклічність $\rho = 1$, то проміжок часу між k і $k+1$ маркуваннями складає точно λ одиниць часу. Поведінка ДБС на цьому етапі може розглядатись як власний усталений стан, або коротко – власний стан ДБС.

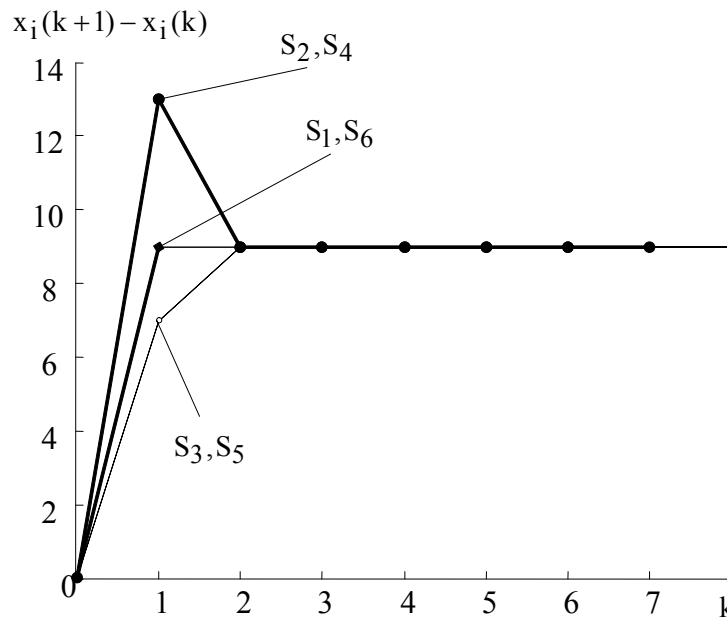


Рис. 7. Перехідний процес в ДБС при циклічності $\rho = 1$

Висновки

1. Аналітична модель представлення графа синхронізації ДБС дозволяє розглядати модель у формі, подібній до представлення за допомогою рівнянь стану процесу в сучасній теорії управління.
2. Динаміка ДБС залежить від властивостей матриці динаміки ДБС, зокрема перехідний процес визначається через власне число цієї матриці.
3. Встановлено зв'язок між показником циклічності критичного графа матриці динаміки ДБС і станом процесу, який може трактуватись як усталений стан системи.
4. Представлена модель динаміки ДБС дозволяє використовувати її для синтезу системи управління дискретно-безперервним процесом з заданими показниками якості, але в рамках цього дослідження задача синтезу не розглядається.

Список літератури: 1. Бесараб В.І. Використання апарата „MAX-PLUS” для моделювання динаміки в інформаційних мережах із простою топологією / В.І. Бесараб, Є.Г. Коваленко // Наукові праці Інституту проблем моделювання в енергетиці ім. Г.Є. Пухова. Вип. 44. Київ, 2007. С. 59-62. 2. David R. Petri Nets and Grafset: Tools for Modeling discrete event Systems/ R. David, H. Alla. London: Prentice Hall, 1992. ISBN: 0-13-327537-X. 3. Cassandras C. G. Introduction to Discrete event Systems/ C.G. Cassandras, S. Lafortune. – Kluwer academic Publishers, 1999. 4. Cunnindham-Green R. Minimax Algebra and applications/ R. Cunnindham-Green // advanced in Imaging and Electron Physics. Vol. 90. New York: Academic Press, 1995. P. 1-121.

Надійшла до редколегії 15.10.2010

Бесараб Володимир Іванович, канд. техн. наук, доцент, завідувач кафедри АТ ДонНТУ. Наукові інтереси: транспортні мережі SDH/PDH; методи аналізу динаміки телекомунікаційних мереж. Хобі: охота. Адреса: Україна, 83000, Донецьк, вул. Артема, буд. 58, тел. 062 332 55 37.

Воропасва Ганна Олександрівна, асистент кафедри АТ ДонНТУ. Наукові інтереси: методи оптимізації та управління розподіленими телекомунікаційними мережами. Хобі: аналіз динаміки функціонування об'єкта телекомунікацій. Адреса: Україна, 83000, Донецьк, вул. Артема, буд. 58, тел. 062 334 11 72.

ОБОБЩЕННАЯ МОДЕЛЬ СИСТЕМЫ СБАЛАНСИРОВАННЫХ ПОКАЗАТЕЛЕЙ

Разрабатывается общая модель системы сбалансированных показателей (ССП), обеспечивающая поддержку стратегического управления предприятием. Модель СПП позволяет формализовать процессы формирования и ведения сбалансированных показателей, выбора перспектив, целей и показателей, определяющих эффективность деятельности компании, а также обработку информации о предприятии в рамках функциональных задач подсистемы «Планирование, учет, контроль, и прогнозирование сбалансированных показателей» автоматизированной информационной системы (АИС) предприятия.

1. Введение

В целях эффективного стратегического управления предприятием целесообразно применение процессного подхода к организации и управлению его деятельностью. На данный момент большинство предприятий используют структурно-функциональный подход, в котором деятельность организации представляется в виде набора функций, закрепленных за структурными подразделениями. Главным недостатком такого подхода является то, что подразделения направляют свою деятельность на достижение локальных целей, между которыми могут быть объективные противоречия, что приводит к сложности внедрения корпоративных информационных систем. В то же время процессный подход ориентирован на бизнес-процессы, обеспечивает прозрачное стратегическое управление, гибко адаптируется под любые изменения.

Повышение эффективности такого управления возможно на основе системы сбалансированных показателей (ССП).

Назначение СПП заключается в формулировании перспектив стратегического развития предприятия, постановке стратегических целей для каждой перспективы и формировании списка показателей, оценивающих степень достижения данных целей. Недостаточно формализованный процесс выбора перспектив, целей и показателей, оценки важности показателей затрудняет применение информационных технологий для практической реализации СПП [1].

Процесс формирования СПП требует разработки модели, описывающей основные компоненты СПП и связи между ними, а также метода для формирования и ведения сбалансированных показателей. Данная модель и метод будут использоваться для определения комплекса функциональных задач подсистемы «Планирование, учет, контроль и прогнозирование сбалансированных показателей» (ПУКПСП) АИС предприятия, формирующей эффективные процессы управления предприятием, которые гарантируют высокое качество и аналитическое превосходство предлагаемых решений проблем развития бизнеса. Подсистема ПУКПСП должна обеспечивать реализацию функциональных задач по стратегическому управлению предприятием и оценки эффективности его деятельности с помощью учета влияния всевозможных факторов при выборе стратегических направлений и целей предприятия.

2. Описание модели СПП

Поддержка процесса формирования СПП может быть осуществлена подсистемой ПУКПСП в рамках следующих функциональных задач:

- планирование целевых показателей на год и среднесрочную перспективу;
- планирование и контроль рационального использования ресурсов предприятия;
- учет и контроль фактических значений показателей эффективности;
- планирование оперативных мероприятий;
- анализ и прогнозирование отклонений;
- управление разработкой стратегической карты предприятия;

– управление индивидуальными заданиями сотрудников по заполнению стратегической карты.

Реализация обмена данными в рамках функциональных задач ПУКПСР требует формального описания компонентов, позволяющих осуществлять управление предприятием с использованием ССП. Модель компонентов ССП представлена в следующем виде:

$$M_{ССП} = \langle P, G, I, A \rangle, \quad (1)$$

где P – множество перспектив деятельности предприятия (perspective); G – множество стратегических целей, устанавливаемых предприятием (goal); I – множество показателей эффективности достижения поставленной цели (indicator); A – множество стратегических мероприятий (activities).

Перспективы – это наиболее значимые сферы, в которых предприятие стремится достигнуть результатов.

Стратегическая цель – это глобальная цель, достижение которой наиболее важно для успеха и выживания организации. Она может быть представлена в виде:

$$G_i = \langle \lambda_i, \{G_{il}\}, \{I_{ij}\} \rangle, \quad (2)$$

здесь λ_i – весовой коэффициент цели (определяется методом экспертных оценок в пределах $[0;1]$; i – порядковый номер стратегической цели в выбранной перспективе; $\{G_{il}\}$ – множество стратегических целей уровня подразделений, где G_{il} – проекция цели G_i на l -е подразделение предприятия, содержащее данную цель; $\{I_{ij}\}$ – множество показателей, где I_{ij} – j -й показатель для i -й стратегической цели.

Показатель эффективности – это параметр, являющийся средством оценки продвижения к реализации стратегической цели.

В процессе исследований было выявлено, что для осуществления оценки деятельности объекта управления (ОУ) показателям эффективности необходимо определить следующие атрибуты:

$$I = \langle nI, \text{form}, f, p, v, pr, \{H\}, r, tgr, bgr, \delta \max, \delta \min, \tau \max, \tau \min \rangle, \quad (3)$$

где nI – наименование показателя эффективности достижения поставленной цели; form – формула расчета показателя; f – фактическое значение показателя эффективности за определенный период; p – плановое значение показателя эффективности; v – весовой коэффициент показателя эффективности (нормированная величина $[0;1]$, определяется методом экспертных оценок); pr – период измерения показателя; $\{H\}$ – множество предыдущих фактических значений показателя эффективности; r – данные о лице, ответственном за внесение значений показателя; tgr, bgr – верхняя и нижняя границы фактических значений показателя; $\delta \max, \delta \min$ – максимальная и минимальная границы допустимого отклонения фактического значения показателя от целевого значения; $\tau \max, \tau \min$ – верхняя и нижняя границы критического отклонения фактического значения показателя от целевого значения.

Стратегические мероприятия направлены на достижение поставленных целей в течение заданного периода времени и при установленном бюджете. Основные параметры, характеризующие мероприятие, следующие:

$$A = \langle nA, C_i, dsp, dep, dsf, def, \{I\}, e_{ji} \rangle, \quad (4)$$

где nA – название мероприятия; C_i – эффективность i -го стратегического мероприятия; dsp, dep – планируемые даты начала и завершения мероприятия; dsf, def – фактические даты начала и завершения мероприятия; $\{I\}$ – множество показателей эффективности, на достижение целевых значений которых направлено мероприятие; e_{ji} – количество ресурса j -го вида, требующегося для выполнения i -го стратегического мероприятия.

Модель ССП позволяет определить состав формируемой и хранимой информации при создании базы данных (БД) показателей, необходимой для формализации процессов выбора перспектив, целей и показателей эффективности деятельности компании, которые реализуются этапами метода формирования и ведения сбалансированных показателей в рамках функциональных задач подсистемы ПУКПСР

3. Метод формирования и ведения сбалансированных показателей

Метод формирования и ведения сбалансированных показателей включает следующие этапы:

1. Формирование стратегической карты предприятия.
2. Построение иерархии стратегических карт для подразделений.
3. Внесение выбранных показателей эффективности в БД.
4. Учет и контроль значений установленных показателей.

Стратегическая карта предприятия позволяет осуществить визуализированное представление стратегии в виде набора перспектив, стратегических целей, показателей и причинно-следственных связей между целями [2]. Поэтому на первом этапе для формирования стратегической карты предприятия изучаются цели, задачи и основные виды деятельности предприятия, на основании чего создаются согласованные модели ОУ. Построение моделей начинается с представления всей системы в виде контекстной диаграммы IDEF0, которая позволяет выявить внешние и внутренние тенденции в деятельности предприятия [3], что дает возможность определить перспективы деятельности предприятия P_m . Такая диаграмма позволяет выделить основные факторы для SWOT-анализа (Strengths, Weaknesses, Opportunities, Threats) на основе потоков информации и отнести к конкретной перспективе.

Использование SWOT-анализа для оценки экспертами частных факторов, представляющих собой сочетание сильных и слабых сторон деятельности предприятия с возможностями и угрозами, позволяет выбрать стратегические цели G_i для каждой перспективы P_m . При этом контекстная диаграмма декомпозируется, описывая взаимосвязи между бизнес-процессами, материальными и информационными потоками, параметрами, характеризующими бизнес-процесс, что делает возможным проведение анализа функционирования ОУ и позволяет выявить узкие и неоптимальные участки бизнес-процессов, препятствующие достижению выбранных стратегических целей.

Параметры бизнес-процесса, позволяющие оценить степень достижения выбранной стратегической цели, фиксируются на стратегической карте предприятия в качестве показателей эффективности цели I_{ij} . Описание их атрибутов позволяет контролировать выполнение стратегии предприятия, планировать целевые значения показателей на год и среднесрочную перспективу, а также ресурсы предприятия.

Таким образом, результатом этапа является стратегическая карта предприятия и диаграмма IDEF0, отражающая бизнес-процессы верхнего уровня, необходимые для реализации стратегических целей ОУ.

При решении задачи взаимосвязи стратегического и оперативного управления используется структурный подход для построения иерархии стратегических карт подразделений, что реализуется вторым этапом метода формирования и ведения сбалансированных показателей. На этом этапе осуществляется управление индивидуальными заданиями сотрудников по заполнению стратегической карты, которое включает: декомпозицию стратегической карты предприятия на все уровни организационной структуры, установки целевых значений показателей эффективности по отдельным отрезкам периода, формирование заданий сотрудникам по заполнению стратегических карт подразделений.

Построение стратегических карт подразделений предусматривает построение моделей бизнес-процессов с выделением функций подразделений, информационных потоков, бизнес-правил, временных характеристик и описанием регламента работы персонала, что позволяет выбрать цели стратегической карты предприятия, являющиеся целями данного подразделения. Помимо этого, такие модели дают возможность формировать взаимосвязанные инструкции сотрудникам по заполнению стратегической карты, реализованных в виде динамических веб-страниц. Создание моделей бизнес-процессов предприятия в целом и его подразделений предусматривает наличие технологических средств их документирования, репозитория унифицированных сервисов, технологий их реализации и выполнения.

Результатом этого этапа являются иерархические стратегические карты подразделений, модели бизнес-процессов и полная функциональная библиотека показателей подразделения и обобщенных оценок показателей по уровням подразделений.

Реализация 1-го и 2-го этапов метода предусматривает формирование БД. При этом модель ССП определяет структуру БД: атрибуты и связи сущностей в виде концептуальной модели БД.

Управление стратегическими картами предприятия и подразделений требует создания таблиц БД «Стратегическая карта предприятия», «Стратегическая карта подразделений», «Перспектива», «Цели», «Показатели».

На третьем этапе метода осуществляется попарное сравнение выбранных показателей и уже существующих на предприятии, корректировка при необходимости их атрибутов, внесение показателей в БД для набора существующих показателей до стратегической карты. Управление заданиями сотрудников по заполнению стратегической карты требует создания сущностей «Подразделение», «Индивидуальные задания» и «Сотрудники». Соответствие целей стратегической карты предприятия при декомпозиции до уровня подразделений реализуется сущностями «Связь» и «Родитель». Учет и контроль значений показателей эффективности, планирование оперативных мероприятий, бюджетов и ресурсов ОУ предусматривает внесение информации в таблицы БД «Реестр рисков», «Отклонения», «Мероприятия», «Планируемые мероприятия», «Ресурсы».

Логическая модель БД разрабатывается на основе концептуальной с учетом выбранной реляционной модели организации данных (рисунок). Формализация логической модели БД дает возможность построить физическую модель, реализуемую в виде объектов БД, которые применяются в пользовательских формах и приложениях [4].

Реализация третьего этапа позволяет создать единое информационное пространство управленческих решений и управляющих воздействий, что исключает дублирование потоков данных и функций управления, а также интегрировать подсистему ПУКПСР с подсистемами планирования, отчетности и бюджетирования. Решение проблемы интеграции данных требует установления всех источников данных, используемых для работы с показателями эффективности [5], для чего могут быть использованы модели бизнес-процессов подразделений предыдущего этапа. Использование модели ССП решает проблемы согласования формата внутренней отчетности и обеспечения достоверности данных различных источников, что требуется для следующего этапа метода.

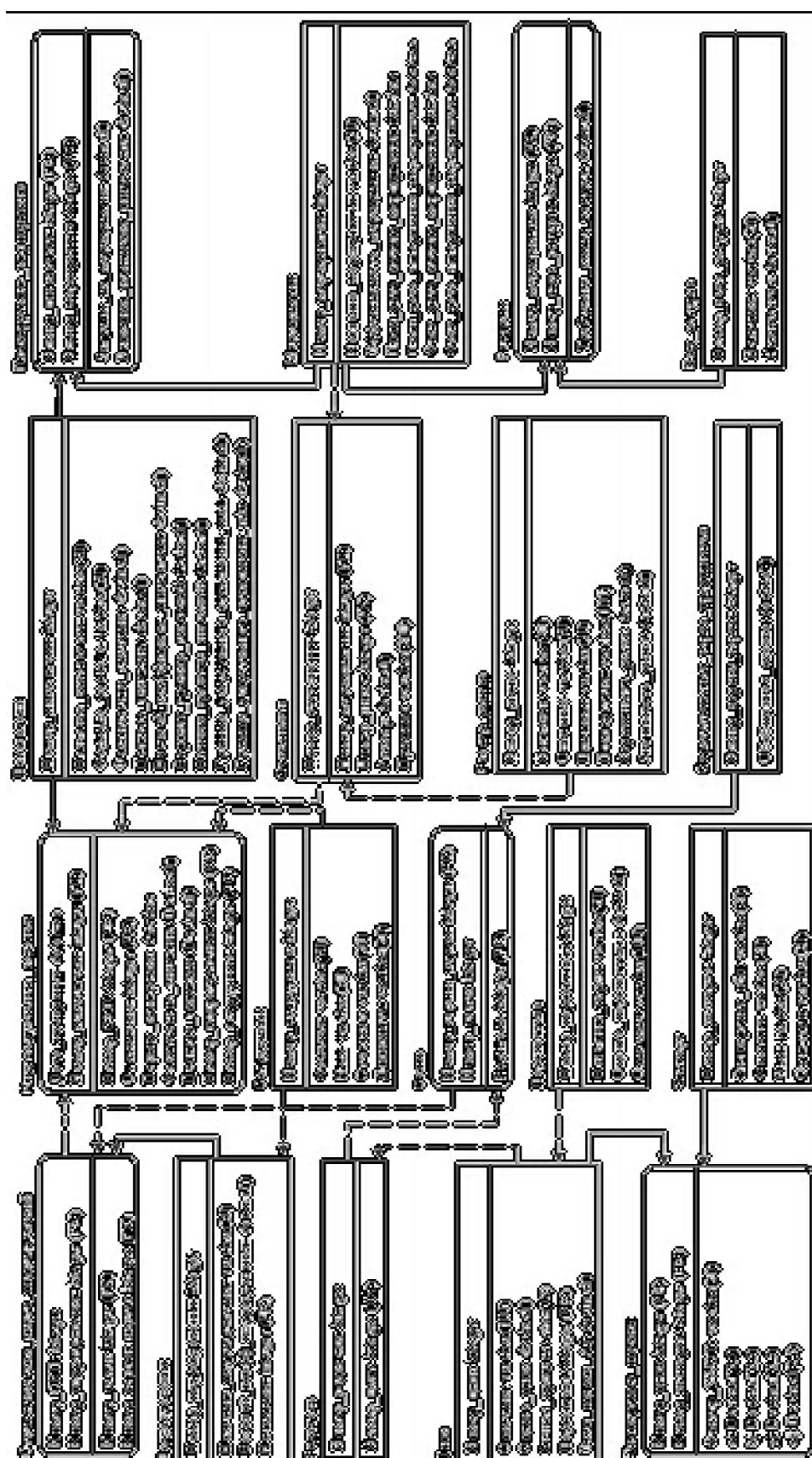
Четвертый этап метода осуществляется путем учета фактических значений показателей эффективности согласно установленным периодам измерения в подсистеме ПУКПСР, что требует описания процессов сбора информации по состоянию целей и показателей.

Для оперативного контроля реализации стратегии деятельности предприятия необходимо обеспечить хранение как информации для стратегического управления, так и детальных данных, на основе которых эта информация была получена, чтобы обеспечить возможность перехода от стратегических целей к фактическим данным, которые определяют эти цели. Реализация функциональных задач подсистемы ПУКПСР требует разработки хранилища данных, которое загружает, согласовывает и обобщает данные из различных источников и поддерживает аналитические возможности. Это позволяет осуществить планирование оперативных мероприятий при возникновении отклонения фактического значения показателя от целевого, проводить их анализ и прогнозирование, управлять модификацией стратегической карты предприятия.

4. Выводы

Научная новизна. Разработанная общая модель ССП описывает компоненты, необходимые для стратегического управления предприятием на основе сбалансированных показателей и формализации процессов выбора перспектив, целей и показателей эффективности деятельности компании, которые реализуются этапами метода формирования и ведения сбалансированных показателей.

Практическая значимость. Данный метод описывает процессы создания и модификации стратегических карт, согласования плановых целевых и фактических показателей эффективности, контроля деятельности предприятия на основе анализа достижения целей. Применение метода при реализации функциональных задач подсистемы ПУКПСР позволит осуществлять учет и контроль показателей эффективности, повышать степень адекватности принимаемых управленческих решений, а также интегрировать подсистемы планирования, отчетности, бюджетирования и мотивации персонала в единую информационную систему стратегического управления предприятием.



Обобщенная логическая модель базы данных подсистемы ПУКПС

Список литературы: 1. *Ольве, Нильс-Горан, Рой, Жан, Веттер, Магнус.* Оценка эффективности деятельности компании по ССП. Практическое руководство по использованию по ССП: Пер. с англ. М.: Издательский дом «Вильямс», 2006. 304 с. 2. *Каплан Р.* Сбалансированная система показателей: от стратегии к действию / Р. Каплан, Д. Нортон. М.: ЗАО «Олимп-Бизнес», 2003. 214 с. 3. *Черемных С.В.* Структурный анализ систем / Черемных С.В. М.: ФИНАНСЫ И СТАТИСТИКА, 2001. 208с. 4. *Конноли Т.* Базы данных: проектирование, реализация и сопровождение. Теория и практика: учеб./ Т.Конноли, К. Бегг. М.: «Вильямс», 2001. 1120с. 4. *Системы стратегического управления для бизнеса: сегодня и завтра* [Электронный ресурс] / *А. Кочнев.* Режим доступа: http://www.iteam.ru/publications/it/section_92/article_3005/ – 05.02.2010.

Поступила в редколлегию 27.11.2010

Левыкин Виктор Макарович, д-р техн. наук, проф., зав.каф. ИУС ХНУРЭ. Научные интересы: методологии, методы и информационные технологии разработки интегрированных и Web-базируемых информационных систем. Адрес: Украина, 61000, Харьков, ул. Чичибабина, 2, кв. 83, e-mail: levykinviktor@gmail.com.

Хворостинина Ольга Сергеевна, аспирант кафедры ИУС ХНУРЭ. Научные интересы: методы, модели и информационные технологии разработки информационных систем. Адрес: Украина, 61121, Харьков, ул. Гарибальди, 7а, кв. 50, тел.: 063-68-98-061, e-mail: guenya@gmail.com.

УДК 681.325.53: 37: 004.5

*Н.Я.КАКУРИН, Ю.В.ЛОПУХИН, В.В. ВАРЕЦА, В.В. КАТАСОНОВ,
А.Н. МАКАРЕНКО*

АНАЛИЗ ПРЕОБРАЗОВАТЕЛЕЙ ДВОИЧНО-К-ИЧНОГО КОДА В ДВОИЧНЫЙ КОД ПО МЕТОДУ ДОСЧЕТА

Рассматривается структура и функционирование преобразователей двоично-К-ичного кода в двоичный код. Выполняется анализ их основных характеристик.

1. Постановка задачи

В современных универсальных ЭВМ преобразование чисел выполняется обычно программным способом, что приводит к уменьшению производительности центрального процессора. При использовании специальных преобразователей кодов (ПК) в качестве предпроцессоров и постпроцессоров этого не происходит [1].

В результате увеличивается производительность ЭВМ, а в случае применения преобразователей кодов в интерфейсах ввода-вывода информации улучшаются их основные характеристики.

Преимуществом преобразователей кодов по методу досчета (ПК ДСЧ) являются малые аппаратурные затраты, схемная простота и линейная зависимость аппаратурных затрат от разрядности входного кода и невысокая стоимость.

Следует отметить, что стоимость ПК ДСЧ является по отношению к аппаратурным затратам вторичной и определяет качество схемы в зависимости от типов применяемых элементов.

Целью настоящей работы является:

- анализ основных характеристик ПК ДСЧ;
- рассмотрение методики нахождения ПК ДСЧ с максимальным быстродействием;
- сравнительная характеристика основных параметров структур ПК ДСЧ, а также анализ программного средства для расчета основных характеристик ПК этого типа.

2. Структура и функционирование многосекционного ПК ДСЧ

В простейшем случае преобразователем ПК ДСЧ из К-ичной системы в двоичную является односекционная схема, содержащая один входной вычитающий счетчик (в К-ичной системе) и один выходной суммирующий в двоичной системе счета.

Недостатком односекционного ПК ДСЧ является значительное число тактов на преобразование максимального числа.

Пусть система счисления на входе – К; система счисления на выходе – р; число входных разрядов – n_z ; длительность периода импульсов генератора – T_{Γ} .

Тогда при $K=10$; $p=2$; $nz=6$ время преобразования определяется формулой:

$$t_{\text{пр}} = T_{\Gamma} \cdot N_{\text{макс}} = T_{\Gamma} \cdot (K^{nz} - 1) \quad (1)$$

при $T_{\Gamma} = 1 \cdot 10^{-6} \text{ с.}$; $t_{\text{пр}} = 1 \cdot 10^{-6} (10^6 - 1) = 1 \cdot 10^{-6} \cdot 999999 = 0,999999 \text{ с} \approx 1 \text{ с.}$

В целях ускорения процесса преобразования чисел можно использовать различные приемы, в частности разбиение входных и выходных счетчиков на две [2,3] или на большее число секций [4].

Преобразование чисел в многосекционном ПК ДСЧ (рис.1) происходит путем последовательно-возвратного обнуления каждого входного счетчика. При этом способе обнуление входного счетчика второй секции начинается только после полного обнуления входного счетчика первой секции, обнуление третьей секции возможно после полного обнуления предыдущих входных секций, т.е. второй и первой.

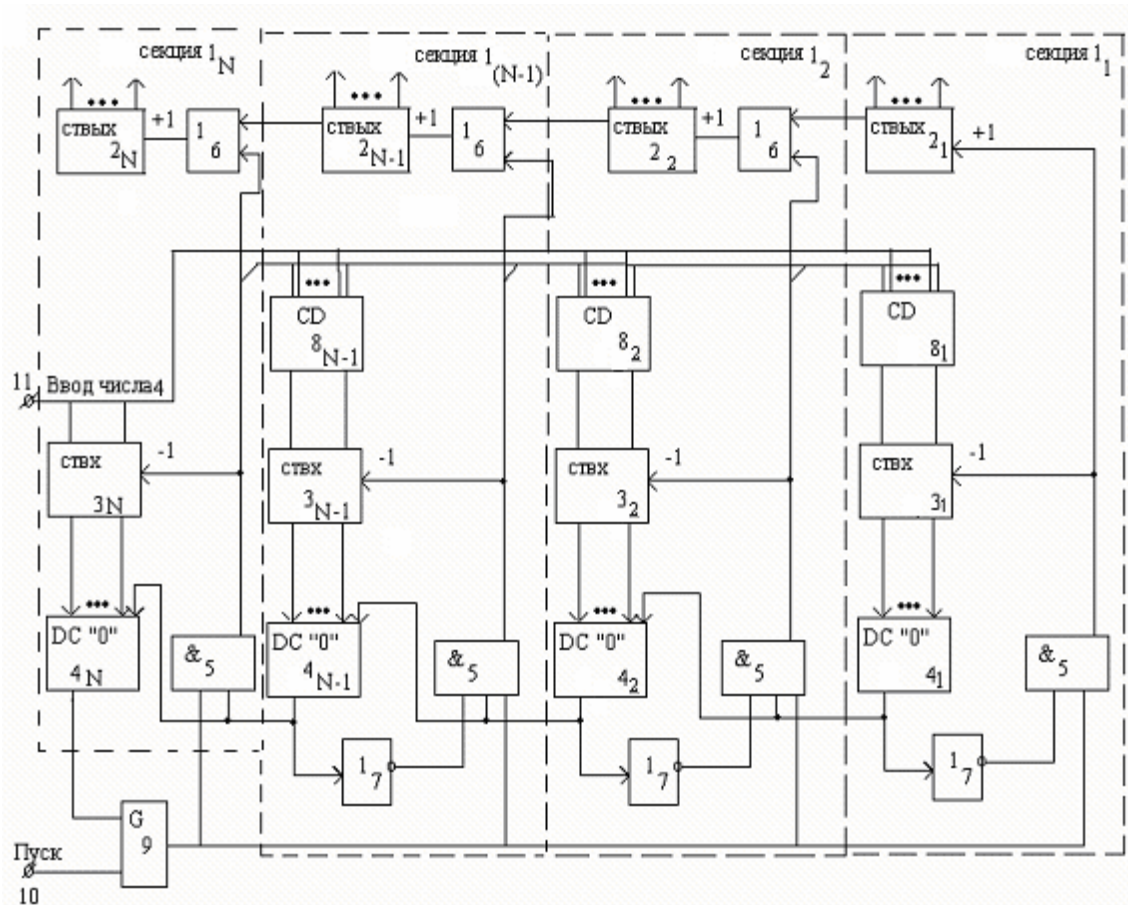


Рис. 1. Структура многосекционного ПК ДСЧ

Преобразователь содержит nz преобразующих секций I ; каждая преобразующая секция I – декаду (при $K=10$) двоично-десятичного счетчика 3, двоичный ($p=2$) счетчик 2, дешифратор нуля 4, элемент ИЛИ 5. Все секции I , кроме младшей, содержат элемент ИЛИ 6; все секции, кроме старшей, – элемент НЕ 7. Все преобразующие секции, кроме старшей, содержат шифраторы 8.

Преобразователь содержит генератор импульсов 9 и вход пуска 10. Обозначим коэффициент пересчета двоичных счетчиков 3_i через M_i ($i=1, nz$). Устройство работает следующим образом. В исходном состоянии в K -ичных счетчиках 3_i ($i=1, nz$) записаны числа $a_1, a_2, \dots, a_n$ преобразуемого K -ичного кода, а двоичные счетчики 2_i ($i=1, nz$) установлены в нуль. При этом на выходе дешифратора нуля 4_i появляется запрещающий потенциал, если в двоично-десятичном счетчике 3_i записано какое-либо число, отличное от нуля. По

сигналу “Пуск” импульсы с выхода генератора 9 поступают через элемент И 5₁ на счетный вход вычитания счетчика 3₁ и на счетный вход сложения счетчика 2₁ и производят вычитание единиц из счетчика 3₁ и прибавление единиц в счетчик 2₁ (по единице на каждый импульс) до тех пор, пока в счетчике 3₁ не установится нуль. Таким образом, число a_1 будет перенесено в двоичный счетчик 2₁. Если a_1 больше или равно максимальному двоичному числу 2^3 , которое может быть записано в счетчик 2₁, то в процессе прибавления единиц в счетчик 2₁ с его выхода возникнет единица переноса, которая через элемент ИЛИ 6₂ поступает на счетный вход счетчика 2₂, а в счетчике 2₁ останется число $m_1 = a_1 - 2^3$.

Если $a_1 < 2^3$, то в счетчике 2₁ остается число $m_1 = a_1$.

На выходе дешифратора нуля 4₁ появляется разрешающий потенциал, а на выходе элемента НЕ 7₁ – запрещающий. Импульс с выхода генератора 9 проходит через элемент И 5₂ и производит вычитание единицы из счетчика 3₂, прибавление единицы в счетчик 2₂ и через шифратор 8 – установку числа $R_2 = R_{21} = 10 - M_1$ в счетчик 3₁. На выходе дешифратора 4₁ снова появляется запрещающий потенциал, а на выходе элемента НЕ 7₁ – разрешающий. Следующая серия импульсов производит вычитание числа R_{21} из счетчика 3₁ и прибавление этого числа в счетчик 2₁. Если при этом возникает переполнение счетчика 2₁, то единица переноса через элемент ИЛИ 6₂ поступает на счетный вход счетчика 2₂. Такая последовательность действий циклически продолжается до тех пор, пока число из счетчика 3₂ не будет полностью перенесено в счетчик 2₂. После этого открывается элемент И 5 следующей секции I и начинается вычитание единиц из счетчика 3 и прибавление их в счетчик 2 этой секции.

Процесс разбиения выходных счетчиков на секции будет детально рассмотрен в пункте 3. Сейчас примем, что входные счетчики 3_i ($i = \overline{1, nz}$) содержат по одному K-ичному разряду, а выходные счетчики 2_i содержат соответственно (3,3,4,3,3,3) разрядов (число справа соответствует числу двоичных разрядов первой секции; число слева – шестой секции).

Модель процесса преобразования трехсекционного ПК ДСЧ с представлением для удобства результатов в десятичной системе представлена в табл.1.

Таблица 1

Такт	0	1	8	9	10	12	13	15
СТВЫХ	000	001	010	011	021	023	033	035
СТВХ	X99	X98	X91	X90	X82	X80	X72	X70
Такт	16	18	19	20	21	22	24	25
СТВЫХ	045	047	057	060	061	071	073	103
СТВХ	X62	X60	X52	X51	X50	X42	X40	X32
Такт	27	28	30	31	32	33	34	36
СТВЫХ	105	115	117	127	130	131	141	143
СТВХ	X30	X22	X20	X12	X11	X10	X02	X00

В табл.1 символ X означает, что входной счетчик 3-й секции может принимать значения от 0 до K-1. Проведем интерпретацию результата моделирования (табл. 1).

После 36-го такта в трех младших двоичных счетчиках будут зафиксированы следующие значения бит:

	СТВЫХ ₃	СТВЫХ ₂	СТВЫХ ₁
Биты	987	654	321
Значения бит	001	100	011
Степень бита	876	543	210

Окончательный результат двоичного числа на выходе ПК ДСЧ равен $99_{10} = 2^6 + 2^5 + 2^1 + 2^0 = 64 + 32 + 2 + 1 = 1100011_2$.

3. Методика расчета основных характеристик ПК ДСЧ К-2

Структуру ПК ДСЧ, состоящую из ряда n_z секций, будем отображать числами K -ичных входных разрядов $n_i (i = \overline{1, n_z})$ и числами p -ичных выходных разрядов $m_j (j = \overline{1, n_z})$ в виде (примем, что $n_z = N$):

$$\begin{aligned} m_N, m_{N-1}, m_{N-2}, \dots, m_2, m_1, \\ n_N, n_{N-1}, n_{N-2}, \dots, n_2, n_1 \end{aligned} \quad (1)$$

Наибольшим быстродействием обладает структура ПК, называемая фундаментальным разбиением (ФР) и содержащая по одному K -ичному разряду на входе.

ФР может быть представлено следующей многосекционной структурой ПК:

$$\begin{aligned} m_N, m_{N-1}, m_{N-2}, \dots, m_2, m_1, \\ 1, 1, 1, \dots, 1, 1 \end{aligned} \quad (2)$$

Число выходных двоичных разрядов m_j секции j определяют из неравенства:

$$\frac{\sum_{j=1}^p m_j}{p} \leq K^{\frac{\sum_{i=1}^p n_i}{p}}, \quad (3)$$

как наибольшую целую степень p , удовлетворяющую неравенству (3).

Рассмотрим ряд примеров нахождения структуры секции.

Дано $p=2$; $K=10$; $n_z=6$. Для первой секции $2^{m_1} \leq 10^1$; $m_1 = 3$; для второй секции $2^{m_1+m_2} \leq 10^2$; $2^{3+3} \leq 10^2$; $2^6 \leq 100$; $m_2 = 3$; аналогично для третьей и четвертой секций получим:

$$2^{m_1+m_2+m_3} \leq 10^3; 2^9 \leq 10^3; 512 \leq 1000; m_3 = 3,$$

$$2^{m_1+m_2+m_3+m_4} \leq 10^4; 2^{13} \leq 10^4; 8192 \leq 10000; m_4 = 4;$$

для пятой и шестой секций получим соответственно

$$2^{m_1+m_2+m_3+m_4+m_5} \leq 10^5; 2^{16} \leq 10^5; 65536 \leq 100000; m_5 = 3;$$

$$2^{m_1+m_2+m_3+m_4+m_5+m_6} \leq 10^6; 2^{19} \leq 10^6; 524288 \leq 1000000; m_6 = 3.$$

Величины корректирующих поправок $R_i (i = \overline{2, N})$ рассчитывают по формуле

$$R_i = K^{\frac{\sum_{i=1}^{i-1} n_i}{i-1}} - p^{\frac{\sum_{i=1}^{i-1} m_i}{i-1}}. \quad (4)$$

Тогда для предыдущего примера получим:

$$R_2 = 10^1 - 2^3 = 2_{10}; R_3 = 10^2 - 2^{3+3} = 100 - 64 = 36_{10};$$

$$R_4 = 10^3 - 2^{3+3+3} = 10^3 - 2^9 = 1000 - 512 = 488_{10};$$

$$R_5 = 10^4 - 2^{3+3+3+4} = 10^4 - 2^{13} = 10000 - 8192 = 1808_{10};$$

$$R_6 = 10^5 - 2^{3+3+3+4+3} = 10^5 - 2^{16} = 100000 - 65536 = 34464_{10}.$$

Общую суммарную поправку R_i необходимо разделить на секции по соответствующему числу разрядов в виде корректирующих поправок в секции.

В итоге получим: $R_2 = R_{21} = 2$; $R_3 = R_{32}; R_{31}; R_{32} = 3; R_{31} = 6$;

$$R_4 = R_{43}; R_{42}; R_{41}; R_{43} = 4; R_{42} = R_{41} = 8;$$

$$R_5 = R_{54}; R_{53}; R_{52}; R_{51}; R_{54} = 1; R_{53} = 8; R_{52} = 0; R_{51} = 8;$$

$$R_6 = R_{65}; R_{64}; R_{63}; R_{62}; R_{61}; R_{65} = 3; R_{64} = R_{63} = R_{61} = 4; R_{62} = 6;$$

Матрицей корректирующих поправок для рассматриваемого примера является:

$$R_6^{10} = [R_{ij}] = \begin{matrix} & \begin{matrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} \\ \begin{matrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{bmatrix} & & & & 2 \\ & & & 3 & 6 \\ & & 4 & 8 & 8 \\ & 1 & 8 & 0 & 8 \\ 3 & 4 & 4 & 6 & 4 \end{bmatrix} \end{matrix} \quad (5)$$

Преобразование чисел в многосекционной схеме ПК ДСЧ аналогично подсчету числа импульсов в неоднородной позиционной системе счисления с весами Q_i ($i = \overline{1, N}$).

Весовые коэффициенты Q_i определяют по формуле:

$$Q_i = 1 + \sum_{j=1}^{i-1} R_{ij} Q_j, \quad (6)$$

где R_{ij} – корректирующие поправки из секции i в секцию j .

Максимальное число тактов преобразователей T_N многосекционного ПК ДСЧ находят из выражения:

$$T_N = \sum_{i=1}^N (K^{n_i} - 1) Q_i. \quad (7)$$

В соответствии с формулами (6) и (7) для $K=10$; $p=2$; $N=6$ находим:

$$Q_1 = 1; Q_2 = 1 + R_{21} \cdot Q_1 = 1 + 2 \cdot 1 = 3; Q_3 = 1 + R_{32} \cdot Q_2 + R_{31} \cdot Q_1 = 1 + 3 \cdot 3 + 6 \cdot 1 = 16;$$

$$Q_4 = 1 + R_{43} \cdot Q_3 + R_{42} \cdot Q_2 + R_{41} \cdot Q_1 = 1 + 4 \cdot 16 + 8 \cdot 3 + 8 \cdot 1 = 97;$$

$$Q_5 = 1 + R_{54} \cdot Q_4 + R_{53} \cdot Q_3 + R_{52} \cdot Q_2 + R_{51} \cdot Q_1 = 1 + 1 \cdot 97 + 8 \cdot 16 + 0 \cdot 3 + 8 \cdot 1 = 234;$$

$$Q_6 = 1 + R_{65} \cdot Q_5 + R_{64} \cdot Q_4 + R_{63} \cdot Q_3 + R_{62} \cdot Q_2 + R_{61} \cdot Q_1 = \\ = 1 + 3 \cdot 234 + 4 \cdot 97 + 4 \cdot 16 + 36 + 41 = 1177;$$

$$T_6 = 13752 \text{ такта.}$$

С помощью рассмотренного выше ПК ДСЧ для ряда значений $K=3,5,7,9,10,12$; $p=2$ и $N=9$ было выполнено разбиение ПК на секции. Результаты приведены в табл. 2.

Таблица 2

K	$\sum m_i$	9	8	7	6	5	4	3	2	1
3	13	2	1	2	2	1	1	1	2	1
5	19	2	2	3	2	2	3	2	2	2
7	25	3	3	3	2	3	3	3	3	2
9	28	3	3	3	4	3	3	3	3	3
10	29	3	3	4	3	3	4	3	3	3
12	32	4	3	4	4	3	4	3	4	3

Структурная схема алгоритма расчета основных параметров ПК ДСЧ из K-ичной системы счисления в двоичную приведена на рис.2.

4. Программное средство для анализа ПК ДСЧ «PREOBRAZOVATEL K-2»

В качестве инструмента для разработки программного пакета «Preobrazovatel K-2» была использована среда программирования Microsoft Visual Studio 2005. Microsoft Visual Studio 2005 – средство быстрой разработки приложений, позволяющее создавать программы на языке C#. Данная среда программирования частично использует принципы RAD. RAD означает быструю разработку приложений. Это концепция позволяет создавать программные продукты, причем особое внимание уделяется скорости и удобству программирования, созданию технологического процесса, позволяющего программисту максимально быстро писать компьютерные программы.

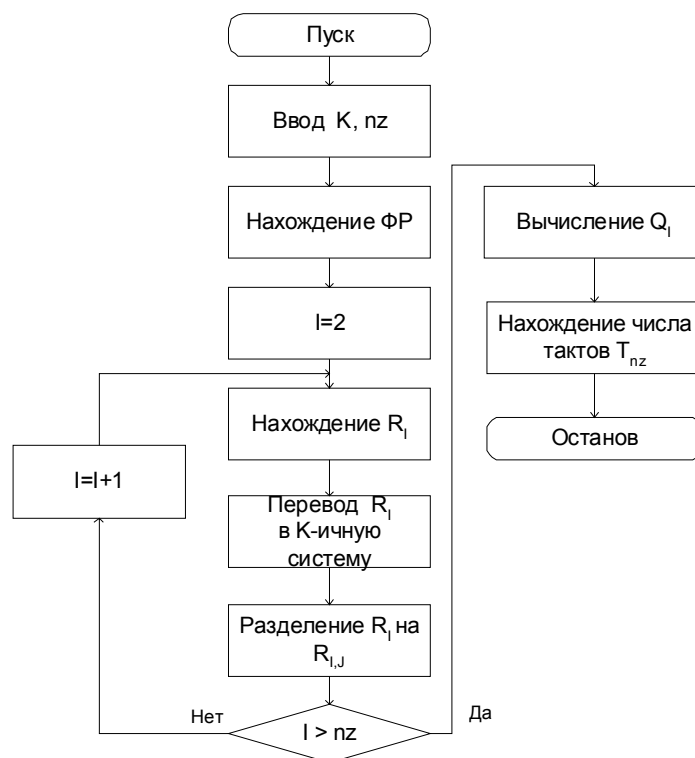


Рис. 2. Структурная схема алгоритма расчета параметров ПК ДСЧ

При разработке программного средства использовались следующие структуры данных.

Класс Convertor : K – входная система счисления; matrix – двумерный массив, в котором формируется результат вычислений; N – определяет результирующий размер матрицы, а также списков n_list, z_list, R_list, Q_list; n_list – список коэффициентов n; nz – входной параметр, который определяет число разрядов в результате вычислений и условие окончания преобразования, где p – система счисления; Q_list – список коэффициентов Q; R_list – список коэффициентов R; T – время работы алгоритма преобразования в тактах; z_list – список коэффициентов z.

Методы, используемые в программе: Calculate – функция для преобразования; Convertor – конструктор класса, который выполняет начальную инициализацию переменных в классе; GenerateList – выполняет инициализацию списков n и z; GenerateMatrix – функция, которая рассчитывает результирующую матрицу; GenerateQList – функция для расчета значений Q; GenerateR – функция для расчета значений R; GenerateT – функция для расчета времени выполнения алгоритма; GetBoolArray – функция для преобразования входящего числа R в K-ичную систему счисления; GetInt – функция для преобразования входного числа с K-ичной системы счисления в десятичную; GetN – функция для определения параметра N; IsEndOfListGeneration – определяет условие окончания цикла для преобразования.

После запуска программы появятся три текстовых поля для ввода данных. С помощью этих элементов управления необходимо задать исходные данные для режима преобразования чисел (рис.3).

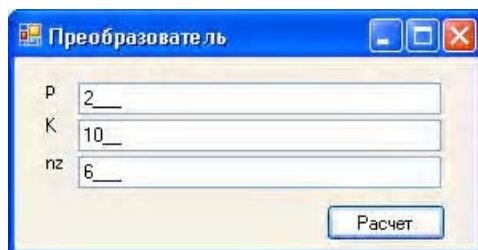


Рис. 3. Интерфейс для ввода исходных данных программы «Preobrazovatel K-2»

Результат работы программного средства «Preobrazovatel K-2» для p=2; K=10; nZ=20 приведен на рис.4.

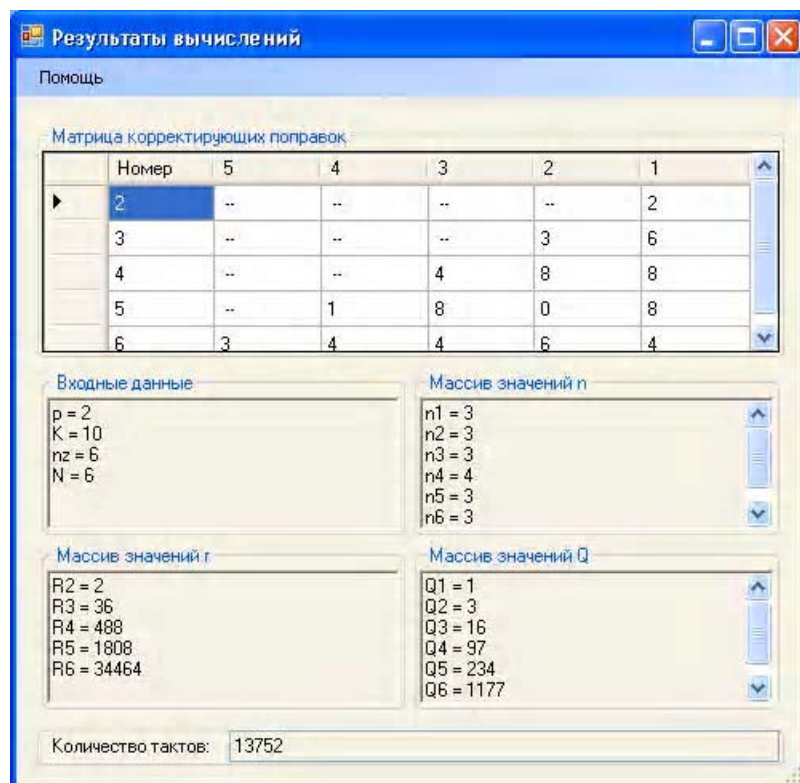


Рис. 4. Результат работы программы «Преобразователь К-2»

Программное средство работает при $p=2$; $K=3-60$. Для $K=3$ число входных разрядов $nZ=20$; для $K=10$ ($nZ=12$); для $K=60$ ($nZ=5$). С помощью данной программы получен и проанализирован ряд основных характеристик многосекционных преобразователей двоично-К-ичного кода в двоичный код по методу досчета для $p=2$ (табл. 3,4).

Таблица 3

K	T_9	T_8	T_7	T_6	T_5	T_4	T_3	T_2	T_1
3	294	184	138	78	38	20	10	6	2
5	11608	4288	2248	744	264	120	40	12	4
7	39342	12558	3618	990	582	318	102	30	6
9	98168	22792	8680	6720	1648	400	112	24	8
10	825324	206073	76293	13752	3159	1053	180	36	9
12	1972828	368313	132682	28919	4312	1265	176	66	11

В табл.3 приведено число тактов преобразования T_i для числа разрядов $i=1-9$ и значений $K = 3,5,7,9,10,12$. В табл. 4 приведены значения коэффициента выигрыша числа тактов преобразования многосекционной схемы по сравнению с односекционной T_1 / T_i ($i = \overline{2,9}$).

Таблица 4

K	9	8	7	6	5	4	3	2	1
3	66,9	35,6	15,8	9,3	6,4	4	2,6	2	1
5	168,3	91	34,7	21	11,8	5,2	3,1	2	1
7	1024,9	459	227,6	118,8	28,9	7,5	3,35	1,6	1
9	3945,5	1888,8	551	79,1	35,8	16,4	6,5	3,3	1
10	1211,6	485,3	131	72,7	31,6	9,5	5,55	2,75	1
12	2616,2	1167,4	270	103,3	57,7	16,4	9,8	2,17	1

Выводы

1. Проанализирована структура и функционирование многосекционного ПК двоично-К-ичного кода в двоичный код по методу досчета.

2. Рассмотрен метод расчета основных характеристик ПК ДСЧ: нахождение структуры секций, матрицы корректирующих поправок, весовых коэффициентов секций и числа тактов преобразования фундаментального разбиения.

3. Предложено программное средство «Преobrazovatel K-2», позволяющее автоматизировать этапы системного проектирования и нахождения основных системных характеристик ПК ДСЧ.

4. Выполнен сравнительный анализ структуры секций ФР для различных оснований $K=3-12$ и показано, что по сравнению с односекционной схемой многосекционный ПК ДСЧ имеет выигрыш в 4-5 порядков, который растет с увеличением основания K и числа входных разрядов nZ .

Научная новизна состоит в разработке метода расчета основных характеристик ПК ДСЧ из К-ичной системы счета в двоичную на основе ФР и реализации этого метода на языке программирования C#.

Практическая значимость результатов исследования заключается в автоматизированном нахождении основных параметров ПК ДСЧ, что позволяет существенно ускорить этап системного проектирования преобразователей этого типа.

Список литературы: 1. Сухомлинов М.М., Выхованец В.Н. Преобразователи кодов чисел. Киев: Техника, 1965. 136 с. 2. А.С. 468236 5G06F 5/02. Устройство для преобразования кодов /В.М.Гусятин, Н.В.Алипов, А.П.Руденко //Открытия, изобретения. 1975. №15. С.108. 3. Макаренко А.Н. Алгоритмизация разбиений преобразователей кодов // АСУ и приборы автоматики. 1990. Вып. 94. С.103-109. 4. А.С. 1153323 5G06F 5/00. Преобразователь двоичного кода в двоично-К-ичный код / Н.Я.Какурин, Ю.К.Кирияков, В.М.Гусятин // Открытия, изобретения. 1985. №16. С.167.

Поступила в редколлегию 02.10.2010

Какурин Николай Яковлевич, канд. техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: прикладная теория цифровых автоматов, 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, автоматизация проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

Вареца Виталий Викторович, аспирант кафедры автоматизации проектирования вычислительной техники ХНУРЭ. Научные интересы: проектирование программного обеспечения, автоматизация проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

Катасонов Валерий Вячеславович, студент группы КИ-07-6 ХНУРЭ. Научные интересы: автоматизация проектирования цифровых устройств, проектирование программного обеспечения. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.

Макаренко Анна Николаевна, канд. техн. наук, доцент кафедры информационных технологий Харьковского института банковского дела. Научные интересы: информационные технологии, анализ и синтез преобразователей кодов. Адрес: Украина, 61074, Харьков, пр. Победы, 55, тел. 336-05-64.

КРИТЕРИИ ВЫБОРА МЕТОДА БИНАРИЗАЦИИ ПРИ ОБРАБОТКЕ ИЗОБРАЖЕНИЙ ЛАБОРАТОРНЫХ АНАЛИЗОВ

Приводятся результаты исследования основных методов бинаризации изображений применительно к различным изображениям лабораторных медицинских анализов. На основании этого предлагаются критерии выбора оптимального метода бинаризации.

Введение

Одной из важных задач при исследовании фотоснимков лабораторных медицинских анализов является выделение объектов интереса (ОИ), например, для подсчета их количества, анализа формы. Во многих случаях первой процедурой обработки изображения является бинаризация, после которой пиксели изображения могут принимать одно из двух значений: 0 – пиксел черного цвета или 1 – пиксел белого цвета.

Цель данного исследования – разработка правила выбора оптимального метода бинаризации для изображений, содержащих объекты интереса различного вида.

Для исследований были выбраны цветные изображения формата bmp разрешением 300 dpi, которые преобразуются в полутоновые изображения и помещаются в память как матрица размерностью $M \times N$, соответствующая размерности исходного изображения. Каждый элемент матрицы содержит информацию о яркости соответствующего пикселя. Далее к полутоновым изображениям применяется процедура бинаризации.

Исследования проводились по следующим методам бинаризации.

Бинаризация отсечением по порогу яркости

Для исследований была выбрана бинаризация с нижним порогом, реализованная согласно правилу:

$$f'(m, n) = \begin{cases} 0, & f(m, n) \leq t; \\ 1, & f(m, n) > t, \end{cases}$$

где $f(m, n)$ – яркость пикселя на исходном изображении; $f(m, n) \in [0.2^k - 1]$, $f'(m, n)$ – значение пикселя результирующего изображения; $f'(m, n) \in [0, 1]$, t – порог бинаризации.

Любая точка изображения, для которой выполняется условие $f(m, n) > t$, называется точкой объекта, а в противном случае – точкой фона [1].

Метод Отса

В этом методе вычисляется порог t , минимизирующий среднюю ошибку от принятия решения о принадлежности пикселей изображения объекту или фону. Значения яркостей пикселей изображения рассматриваются как случайные величины, а их гистограмма – как оценка плотности распределения вероятностей. Если плотности распределения вероятностей известны, то можно определить оптимальный порог для сегментации изображения на два класса c_0 и c_1 (объекты и фон) [2].

Исследования основывались на следующем: изображение представляется с помощью L уровней яркости; h_i – число элементов изображения, имеющих яркость i , $i = 0, 1, \dots, L-1$; N – общее число пикселей на изображении; гистограмма изображения является нормализованной и ее можно рассматривать как распределение вероятностей:

$$p_i = \frac{h_i}{N}, i = 0, 1, \dots, L-1; \sum_{i=0}^{L-1} p_i = 1.$$

Элементы изображения делятся на два класса c_0 и c_1 с помощью порогового значения t , где класс c_0 содержит пиксели с яркостями из множества $(0, 1, \dots, t)$, а класс c_1 – пиксели с яркостями из множества $(t, t+1, \dots, L-1)$. Вероятности каждого из этих двух классов и средние значения их яркости описываются выражениями:

$$P_0 = \sum_{i=0}^t p_i = P_t, \quad P_1 = \sum_{i=t+1}^{L-1} p_i = 1 - P_t, \quad \mu_0 = \sum i p_i / P_0 = \mu_t / P_t, \quad \mu_1 = \sum_{i=t+1}^{L-1} i p_i / P_1 = (\mu_T - \mu_t) / (1 - P_t),$$

где $\mu_T = \sum_{i=0}^{L-1} i p_i$ означает среднюю яркость всего изображения.

Метод Бернсена

Все изображение делится на квадраты $r \times r$ (r – нечетное) с центром в точке (m, n) . Для каждого пикселя изображения в пределах квадрата используется порог, имеющий значение

$$t(m, n) = \frac{j_{\text{high}} + j_{\text{low}}}{2},$$

где j_{high} и j_{low} являются соответственно наименьшим и наибольшим уровнем яркости в квадрате. Если в принятой области используемая мера контраста удовлетворяет условию

$$G(m, n) = (j_{\text{high}} - j_{\text{low}}) \leq \varepsilon,$$

где ε – заданная пороговая величина, то исследуемый квадрат содержит объекты только одного класса: объектов или фона [3].

Наилучшие результаты были получены при $\varepsilon = 15$ и $r = 15$.

Метод Эйквеля

В этом методе используются два окна r и R , из которых большее по размерам (окно R) служит для вычисления значения порога, в то время как меньшее (окно r) определяет область изображения, в которой будет использоваться полученный порог. Оба окна перемещаются параллельно по изображению с шагом, равным размеру меньшего окна r , и каждый раз для всех элементов окна R вычисляется оптимальный порог по методу Отса. Если вычисленные средние значения μ_0 и μ_1 различаются сильно и для них выполняется условие $|\mu_0 - \mu_1| \geq \varepsilon$, где ε – заданный параметр, то пиксели внутри окна r подвергаются бинаризации в соответствии с вычисленным порогом t . Если же $|\mu_0 - \mu_1| < \varepsilon$, то все пиксели внутри окна r относятся к классу с ближайшим средним значением [4]. Были использованы следующие значения параметров: $r = 3 \times 3$, $R = 15 \times 15$ и $\varepsilon = 15$.

Метод Ниблэка

В данном методе для каждого пикселя изображения используется свое значение порога. Величина порога определяется на основе вычисления локального среднего и локального среднеквадратического отклонения. Значение порога в точке с координатами (m, n) вычисляется в соответствии с формулой:

$$t(m, n) = \mu(m, n) + k \cdot \sigma(m, n),$$

где $\mu(m, n)$ – среднее, а $\sigma(m, n)$ – среднеквадратичное отклонение в локальной окрестности точки изображения (m, n) [5].

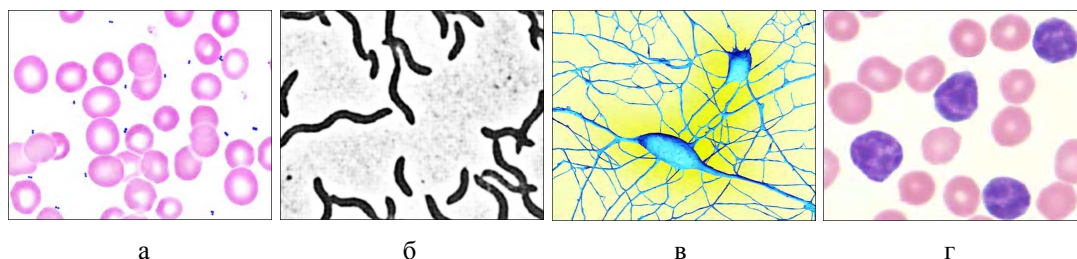
Размер окрестности пикселя был выбран $r = 15 \times 15$, а $k = -0.2$.

Для реализации рассмотренных методов была написана программа на C++.

Результаты исследований

Исследования проводились со следующими классами изображений (рис.1).

Изображения, содержащие ОИ в виде объектов округленной формы однородного цвета, расположенных на однотонном фоне (рисунок, поз.а); изображения, на которых ОИ представлены в виде объектов, отличающихся от остальных элементов изображения размером, цветом и контрастностью (рисунок, поз.б, в); изображения, на которых ОИ представлен объектом, распределенным по всему изображению в виде сети (рис.1,г).



Шпорообразные спирали (а); бактерии стрептококка пневмонии – точечные объекты повышенной контрастности (б); увеличенные тромбоциты при острой лимфоцитарной лейкемии (в); нейроны Пуркинью (г)

Применение для изображений на рисунке, поз.а,б,в метода отсечения по порогу яркости со значениями порога 0,3, 0,5, 0,7, 0,9 показало, что наилучший результат достигается при пороге яркости 0,5 – фон становится белым, шум на изображении практически отсутствует, а объекты интереса становятся более контрастными. При других значениях порога яркости на полученном изображении остаются элементы фона, заметны другие элементы исходного изображения, иногда наблюдаются искажения формы ОИ. Для изображения на рисунке, поз.г наилучшие результаты достигнуты при значении порога яркости 0,7 – фон стал белым, и все фрагменты ОИ остались на изображении без искажений.

Применение метода Отса дало хорошие результаты для изображения на рисунке, поз.а – цвет фона стал белым, а ОИ стали равномерно окрашенными. Для изображений на рисунке, поз.б,в,г этот метод применять нельзя, поскольку на результирующем изображении остаются и ОИ, и элементы фона, а также другие объекты исходного изображения.

Метод Бернсена не показал положительных результатов ни с одним из изображений, представленных на рисунке. На изображении остались фрагменты фона, ОИ получились сильно искаженными.

Метод Эйквеля показал хорошие результаты только с изображением на рисунке, поз.г. Все фрагменты ОИ сохранились без искажений, фон стал белым.

Метод Ниблэка не дал положительных результатов ни с одним из использованных изображений. ОИ получились искаженными, фон был сильно зашумлен.

Эффективность использованных методов бинаризации предлагается оценивать следующим образом: «хорошо» – ОИ без искажений, фон – белого цвета, без шума; «удовлетворительно» – ОИ без искажений, но фон зашумлен; «плохо» – ОИ с искажениями, фон зашумлен. Результаты исследований отражены в таблице.

Результаты исследований методов бинаризации

Изображение на рисунке	Методы бинаризации и их результаты							
	Отсечением по порогу яркости				Метод Отса	Метод Бернсена	Метод Эйквеля	Метод Ниблэка
	0,3	0,5	0,7	0,9				
Поз., а	плохо	хорошо	удовл.	удовл.	хорошо	плохо	удовл.	плохо
Поз., б	плохо	хорошо	плохо	плохо	плохо	плохо	плохо	плохо
Поз., в	плохо	хорошо	плохо	плохо	плохо	плохо	удовл.	плохо
Поз., г	плохо	плохо	хорошо	плохо	плохо	плохо	хорошо	плохо

Выводы

Практическая значимость проведенных исследований заключается в разработке рекомендаций по выбору метода бинаризации, оптимального для используемых изображений. Так, для бинаризации изображений, на которых объекты интереса имеют гладкую форму однородного цвета, целесообразно выбирать или метод Отса, или

метод отсечения по порогу яркости, значение которого рекомендуется взять 0,5. Для изображений, на которых объекты интереса отличаются повышенной контрастностью по сравнению с остальными объектами изображения и фона, рекомендуется применять бинаризацию отсечением по порогу яркости со значением 0,5 в случае, если ОИ имеют небольшие размеры по сравнению с другими объектами на изображении. В том случае, когда размеры ОИ больше, чем у остальных элементов изображения, рекомендуется воспользоваться методом Отса. Бинаризацию изображений с ОИ в виде распределенной структуры целесообразно проводить по методу Эйквеля или по методу отсечения по порогу яркости, значение которого рекомендуется взять 0,7.

Дальнейшим направлением работ может быть исследование эффективности различных методов фильтрации применительно к рассмотренным изображениям.

Литература: 1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2005. 1072 с. 2. Otsu N. A threshold selection method from gray-level histograms IEEE Trans. Systems, Man, and Cybernetics. 1979. Vol. 9, no. 1. P. 62–66. 3. Bernsen J. Dynamic thresholding of grey-level images Proc. Eighth Int'l Conf Pattern Recognition, 1986. P. 1, 251–1, 255, Paris. 4. Eikvil L., Taxt T., and Moen K. A fast adaptive method for binarization of document images Proc. First Int'l Conf Document Analysis and Recognition, 1991. P. 435–443, Saint-Malo, France. 5. Niblack W. An Introduction to Digital Image Processing, 1986. P. 115–116. Englewood Cliffs, N.J.: Prentice Hall.

Поступила в редколлегию 28.11.2010

Янковский Александр Аркадиевич, канд. техн. наук, доцент кафедры ЭВМ ХНУРЭ. Адрес: Украина, 61276, Харьков, пр. Ленина, 14, тел. 70-21-354.

Бугрий Андрей Николаевич, канд. техн. наук, старший преподаватель кафедры ЭВМ ХНУРЭ. Адрес: Украина, 61276, Харьков, пр. Ленина, 14, тел. 70-21-354.

ДИСПЕРСИОННЫЙ АНАЛИЗ В ПРОЦЕССАХ ПАССИВНОГО МОНИТОРИНГА И ДИАГНОСТИРОВАНИЯ КОРПОРАТИВНЫХ КОМПЬЮТЕРНЫХ СЕТЕЙ

Рассматриваются аспекты применения дисперсионного анализа в процессах пассивного мониторинга и диагностирования корпоративных компьютерных сетей (ККС). Целью применения дисперсионного анализа является выявление фактора, оказывающего наибольшее влияние на критерий качества работы компьютерной сети, что входит в задачи диагностирования ККС. Описывается модель ККС как объекта мониторинга. Формулируются правила выполнения дисперсионного анализа в рамках модели ККС как объекта мониторинга. Выполняется практическая реализация и описываются экспериментальные результаты применения дисперсионного анализа в процессах пассивного мониторинга и диагностирования ККС.

Введение

Согласно [1], внедрение статистических методов анализа и мониторинга гетерогенных телекоммуникационных сетей, к каковым относятся корпоративные компьютерные сети (ККС), на основе перехвата и декодирования трафика с использованием доступного инструментария (анализаторы протоколов, встроенные средства операционных систем для мониторинга системы и сети и др.) является задачей, в настоящее время требующей решения. Это обусловлено необходимостью эксплуатации и надежного функционирования достаточно большого набора инфокоммуникационных сервисов, обеспечивающих эффективную работу пользователя с разнородной информацией в ККС. При росте числа упомянутых сервисов возрастает сложность обслуживания и обеспечения должного уровня качества работы компьютерной сети. Как следствие, возрастает трудоемкость процессов мониторинга и диагностирования ККС, и необходимость формализации упомянутых про-

цессов и задействования математических методов в ходе постановки диагноза представляется очевидной.

Исходя из [2], под мониторингом ККС будем понимать комплекс мер и процедур, включающий в себя сбор информации о работе основных компонентов и систем сети, анализ собранной информации и, как следствие, выполненного анализа – определение нормального профиля сети и выявление отклонений от него. Установление причин появления этих отклонений относится к задачам диагностирования ККС и непосредственно базируется на результатах работы процессов мониторинга.

Особенностью пассивного мониторинга является отсутствие активных воздействий на объект наблюдения со стороны наблюдателя, что дает возможность поставить в соответствие пассивному мониторингу функциональное диагностирование в терминах [3]. С одной стороны, такой режим диагностирования не требует прекращения штатной работы диагностируемого устройства, что является существенным преимуществом с точки зрения пользователей ККС. С другой стороны, для постановки достоверного диагноза по результатам пассивного мониторинга требуется гораздо больший объем собранных данных в сравнении с активным мониторингом, что ведет к нелинейному возрастанию трудоемкости и временных затрат на их анализ. В совокупности со сказанным это подтверждает актуальность решения задачи разработки формализованных методов и алгоритмов статистической обработки данных, собранных в ходе пассивного мониторинга, в целях снижения трудоемкости и повышения эффективности процессов мониторинга и диагностирования ККС.

Модель ККС как объекта мониторинга

ККС как объект мониторинга (ОМ) определяется соотношением (1), связывающим критерий качества работы ККС y с факторами, оказывающими на него влияние:

$$y = \varphi(S_p\{x_1, x_2 \dots x_k\}, S_a\{x_1, x_2 \dots x_l\}, S_{sys}\{x_1, x_2 \dots x_m\}, S_{nos}\{x_1, x_2 \dots x_n\}), \quad (1)$$

где y – время реакции прикладного ПО сервера на запрос клиента; факторы, характеризующие компоненты ККС как ОМ и влияющие на значение критерия качества работы КС, представлены следующими множествами: $S_p\{x_1, x_2 \dots x_k\}$ – совокупность характеристик кабельной системы и другого пассивного оборудования; $S_a\{x_1, x_2 \dots x_l\}$ – совокупность характеристик активного сетевого оборудования (сетевые платы, концентраторы, коммутаторы, маршрутизаторы); $S_{sys}\{x_1, x_2 \dots x_m\}$ – совокупность характеристик системных ресурсов сервера и рабочих станций; $S_{nos}\{x_1, x_2 \dots x_n\}$ – совокупность конфигурационных и сетевых настроек сетевой операционной системы.

Особенностью ККС как ОМ является то, что при исправном техническом состоянии каждого из сетевых компонентов в отдельности может иметь место ситуация, когда работа сети в целом не отвечает требуемому уровню качества и, следовательно, ККС с точки зрения пользователей и выполняемых в ней сетевых задач исправной не является. Это объясняется наличием ошибок на этапе проектирования и развертывания сети, несбалансированностью нагрузки и сетевых компонентов, использованием ПО с неэффективными алгоритмами реализации. Таким образом, становится актуальной задача определения степени влияния каждого из факторов на критерий качества работы ККС, что, при неудовлетворительных значениях критерия, позволит определить, какой из факторов с наибольшей вероятностью является потенциальным носителем причины их появления.

С другой стороны, модель ККС как ОМ можно представить в виде ациклической древовидной структуры $H = \langle V, W \rangle$ [4], где $V = D \cup E \cup SS \cup F$ – множество компонентов компьютерной сети; $W \subseteq D \times E \times SS \times F$ – иерархические взаимосвязи между ними. Здесь элемент d подмножества D – компонент как устройство, принадлежащее подмножеству высшего уровня; $e \in E$ – компонент как элемент устройства; $ss \in SS$ – компонент как подсистема элемента устройства; $f \in F$ – фактор, принадлежащий подмножеству низшего уровня ККС как ОМ.

Взаимосвязи между указанными подмножествами и их элементами организуются следующим образом (рис. 1):

1) для каждого $ss \in SS$ существует группа элементов $f_H(ss) \subseteq F$, для которых ss является общей вершиной;

- 2) для каждого $e \in E$ существует группа элементов $ss_H(e) \subseteq SS$, для которых e является общей вершиной;
- 3) для каждого $d \in D$ существует группа элементов $e_H(d) \subseteq E$, для которых d является общей вершиной.



Рис.1. Иерархическая структура ККС как ОМ

Чем глубже вложенность компонентов, тем выше эффективность данных, собранных на начальном этапе мониторинга ККС, с точки зрения определения нормального профиля ККС и отклонений от него. Рис. 2 иллюстрирует пример построения иерархической структуры ККС как ОМ (рис.2, б) для тракта передачи данных из рис.2, а.

Дисперсионный анализ в процессах мониторинга и диагностирования

Дисперсионный анализ (ДА) – статистический метод, позволяющий анализировать влияние различных факторов на исследуемую переменную путем проверки значимости различия между средними с помощью сравнения дисперсий [5]. Таким образом, ДА является инструментом, с помощью которого непосредственно решается задача определения степени влияния факторов, описывающих модель ККС как ОМ в соотношении (1), на критерий качества работы сети y , о важности которой уже было сказано выше.

Для обеспечения возможности применения ДА к статистическим данным, собранным в процессе пассивного мониторинга, необходимо, чтобы в ходе сбора обеспечивалось варьирование значений факторов. Поскольку при пассивном мониторинге исследователь не оказывает никаких активных воздействий на характеристики сети, что гарантированно обеспечило бы требуемое варьирование по уровням, то необходимо проводить наблюдения за ККС как ОМ таким образом, чтобы варьирование обеспечивалось типовыми задачами, решаемыми в ККС. Для этого необходимо знать характер и периодичность интенсивности бизнес- и иных процессов, выполняемых на предприятии с применением ККС.

В процессах мониторинга и диагностирования ККС будет применяться как однофакторный, так и многофакторный ДА в соответствии со следующими правилами, базирующимися на иерархической структуре ККС как ОМ:

- 1) Если факторы принадлежат одной вершине $ss \subseteq SS$, – выполняется однофакторный ДА, при этом $p=n$, где p – число, определяющее, сколько раз будет выполняться однофакторный ДА, n – число факторов, принадлежащих заданной вершине ss . Таким образом, выполняется анализ каждого фактора на предмет обнаружения оказывающего наибольшее влияние на критерий качества работы сети. Данную задачу целесообразно решать в рамках оптимизации (или «тонкой» настройки) ККС.

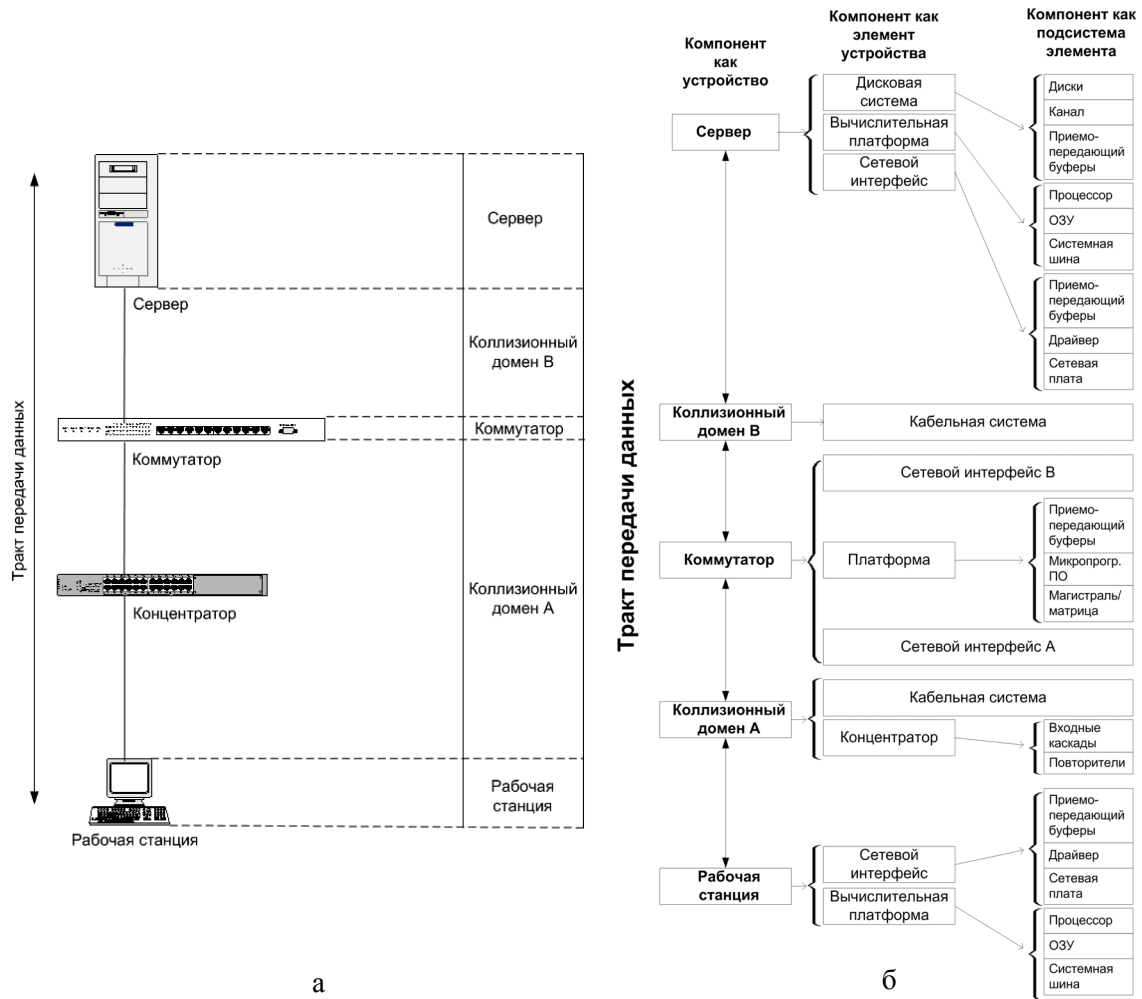


Рис.2. Построение иерархической структуры ККС как ОМ: а – тракт передачи данных (ТПД) ККС; б – иерархическая структура ТПД

2) Если факторы принадлежат разным вершинам $ss \subseteq SS$, но при этом каждая из этих вершин принадлежит такой группе элементов $ss_H(e) \subseteq SS$, для которой имеется общая вершина $e \in E$, – выполняется n -факторный ДА, где n – количество элементов $ss_H(e) \subseteq SS$ с общей вершиной e . Таким образом, решается задача обнаружения потенциального носителя причины отклонения от нормального профиля ККС на уровне компонента как подсистемы элемента. Данная задача решается в рамках диагностирования ККС (обнаружение неисправности и постановка диагноза).

3) N -факторный ДА применяется также и тогда, когда факторы принадлежат разным вершинам $ss \subseteq SS$ и для этих ss нет общей вершины e , но есть общая вершина $d \subseteq D$. Здесь n – количеству элементов группы $e_H(d) \subseteq E$ с общей вершиной d . Решается задача обнаружения потенциального носителя причины отклонения от нормального профиля ККС на уровне компонента как элемента устройства. Данная задача так же, как и в п.2, решается в рамках диагностирования ККС, но с меньшей детализацией постановки диагноза.

4) Если факторы принадлежат разным вершинам $ss \subseteq SS$ и у них нет общих вершин ни в подмножестве E , ни в подмножестве D , – рекомендуется выполнять двухфакторный ДА, при этом выполнять его p раз, $1 \leq p \leq C_d^2$, где C_d^2 – количество сочетаний по 2 из всех

$$d \subseteq D : C_d^2 = \frac{d!}{2!(d-2)!}.$$

Здесь решается задача обнаружения потенциального носителя причины отклонения от нормального профиля ККС на уровне компонента как устройства, что является начальным этапом диагностирования ККС.

Практическая реализация и экспериментальные данные

Рассмотрим пример выявления компонента как устройства, оказывающего наибольшее влияние на критерий качества работы сети. Выполним дисперсионный анализ для статистических данных по факторам «утилизация процессора сервера, %» (фактор А) и «загрузка магистрального коммутатора, %» (фактор Б), где наблюдаемой переменной является критерий качества работы сети y «время реакции прикладного ПО сервера на запрос клиента, ms». В соответствии с приведенными выше правилами выбора типа ДА, а именно с п.4, здесь следует применить двухфакторный ДА. Согласно [5], двухфакторная дисперсионная модель имеет вид:

$$x_{ijk} = \mu + F_i + G_j + I_{ij} + \varepsilon_{ijk}, \quad (2)$$

где x_{ijk} – значение наблюдения в ячейке ij с номером k ; μ – общая средняя; F_i – эффект, обусловленный влиянием i -го уровня фактора А, где $i = \overline{1, m}$; G_j – эффект, обусловленный влиянием j -го уровня фактора Б, где $j = \overline{1, l}$; I_{ij} – эффект, обусловленный взаимодействием двух факторов, т.е. отклонение от средней по наблюдениям в ячейке ij от суммы первых трех слагаемых в модели (2); ε_{ijk} – возмущение, обусловленное вариацией переменной внутри отдельной ячейки ij .

Предполагается, что ε_{ijk} имеет нормальный закон распределения $N(0; \sigma^2)$, а все математические ожидания F, G, I_{ij} равны нулю. Групповые средние вычисляются следующим образом:

в ячейке: $\bar{x}_{ij} = \frac{\sum_{k=1}^n x_{ijk}}{n}$, по строке: $\bar{x}_i = \frac{\sum_{j=1}^l \bar{x}_{ij}}{l}$, по столбцу: $\bar{x}_j = \frac{\sum_{i=1}^m \bar{x}_{ij}}{m}$, общая средняя:

$$\bar{x} = \frac{\sum_{i=1}^m \sum_{j=1}^l \bar{x}_{ij}}{ml}, \text{ где } n - \text{ количество наблюдений в ячейке } ij.$$

В табл. 1 представлен общий вид вычисления значений с помощью дисперсионного анализа.

Таблица 1. Базовая таблица двухфакторного дисперсионного анализа

Компоненты дисперсии	Сумма квадратов	Число степеней свободы	Средние квадраты
Межгрупповая (фактор А)	$Q_1 = l \cdot n \sum_{i=1}^m (\bar{x}_i - \bar{x})^2$	$m-1$	$S_1^2 = \frac{Q_1}{m-1}$
Межгрупповая (фактор Б)	$Q_2 = m \cdot n \sum_{j=1}^l (\bar{x}_j - \bar{x})^2$	$l-1$	$S_2^2 = \frac{Q_2}{l-1}$
Взаимодействие	$Q_3 = n \sum_{i=1}^m \sum_{j=1}^l (\bar{x}_{ij} - \bar{x}_i - \bar{x}_j + \bar{x})^2$	$(m-1)(l-1)$	$S_3^2 = \frac{Q_3}{(m-1)(l-1)}$
Остаточная	$Q_4 = \sum_{i=1}^m \sum_{j=1}^l \sum_{k=1}^n (x_{ijk} - \bar{x}_{ij})^2$	$mln - ml$	$S_4^2 = \frac{Q_4}{ml(n-1)}$
Общая	$Q = \sum_{i=1}^m \sum_{j=1}^l \sum_{k=1}^n (x_{ijk} - \bar{x})^2$	$mln - 1$	

Проверка нулевых гипотез H_A, H_B, H_{AB} об отсутствии влияния на рассматриваемую переменную факторов А, Б и их взаимодействия АВ осуществляется сравнением отноше-

ний $\frac{S_1^2}{S_4^2}, \frac{S_2^2}{S_4^2}, \frac{S_3^2}{S_4^2}$ с соответствующими табличными значениями F–критерия Фишера. Если $F_{\text{эксп.}} < F_{\text{табл.}}$ – гипотеза об отсутствии влияния принимается, в противном случае отвергается.

Сбор и анализ статистических данных по заданным выше факторам выполнялся при следующих условиях: $m=2$, $A_1=20\%$, $A_2=75\%$ при пороговом значении фактора 90%; $l=3$, $B_1=10\%$, $B_2=15\%$, $B_3=25\%$ при пороговом значении фактора 35%; $n=4$.

Таблица 2. Двухфакторный ДА для выявления компонента как устройства

	Б ₁	Б ₂	Б ₃	средние в ячейке (ij):		ср. по стр. (i):	
А ₁	10,00	8,00	13,00	x ₁₁	8,50	x ₁	9,50
	8,00	9,00	15,00	x ₁₂	8,25	x ₂	50,42
	7,00	7,00	10,00	x ₁₃	11,75		
	9,00	9,00	9,00			ср. по столб. (j):	
А ₂				x ₂₁	47,50	x ₁	28,00
	50,00	55,00	55,00	x ₂₂	51,25	x ₂	29,75
	45,00	50,00	60,00	x ₂₃	52,50	x ₃	32,13
	40,00	55,00	45,00				
	55,00	45,00	50,00	общая средняя:			
				x	29,96		
суммы квадратов:			средние квадраты:		критерий Фишера (эксп.):		
Q ₁	837,09	10045,04	S ² ₁	10045,04	ф. А		517,87
Q ₂	8,57	68,58	S ² ₂	34,29	ф. Б		1,77
Q ₃	4,02	16,06	S ² ₃	8,03	ф. АБ		0,41
Q ₄		349,23	S ² ₄	19,40			
					критерий Фишера (табл.):		
					ф. А(0,05;1;18)		4,41
					ф. Б(0,05;2;18)		3,55
					ф. АБ(0,05;2;18)		3,55

При указанных условиях для наблюдаемой переменной у были получены результаты, приведенные в табл.2, выполнен двухфакторный ДА, ход которого также отображен в табл. 2, получены экспериментальные значения критерия Фишера по каждому из факторов и их взаимодействию. Из табл. 2 видно, что нулевая гипотеза Н_д отвергается, нулевые гипотезы Н_б и Н_{аб} - принимаются. Более того, по существенному разрыву между экспериментальным и табличным значениям критерия Фишера можно однозначно утверждать, что в данном случае потенциальным критическим компонентом компьютерной сети является устройство, характеризуемое фактором А. Дальнейшие исследования показали, что сервер, после подключения его к гигабитному порту коммутатора, не справляется с экспоненциально возросшим уровнем входящего трафика в периоды осуществления массовых операций с прикладным ПО. Рекомендуется модернизация вычислительной платформы сервера.

Выводы

Рассмотрены модель и метод пассивного мониторинга и диагностирования ККС на базе дисперсионного анализа. Их научная новизна заключается в применении к задачам, обозначенным в статье и традиционно решаемым эмпирическим путем, аппарата математической статистики. Практическая значимость состоит в формализации процесса обработки статистических данных, собранных в ходе пассивного мониторинга, что приводит к снижению трудоемкости и повышению эффективности процессов мониторинга и диагностирования ККС. В перспективах исследования – дальнейшее развитие применения методов статистической обработки данных к процессам активного мониторинга и диагностирования ККС.

Список литературы: 1. *Скуратов А.К.* Использование временных рядов для целей статистического анализа телекоммуникационных сетей на основе исследования информационных потоков // Вестник Новгородского Государственного университета. 2005. №34. С. 112-117. 2. *Киселев Г.Д., Шпакаускас М.С.* Мониторинг мультисервисных компьютерных сетей средствами системы Nagios // Электроника и связь. Тематический выпуск «Электроника и нанотехнологии». 2010. №3. С. 178-181. 3. *ГОСТ 20911-75.* Техническая диагностика. Основные термины и определения. 4. *Губко М.В.* Математические модели оптимизации иерархических структур. М.: ЛЕНАНД, 2006. 264 с. 5. *Кремер Н.Ш.* Теория вероятности и математическая статистика. М.: Юнити – Дана, 2002. 343 с.

Поступила в редколлегию 04.11.2010

Бабич Анна Витальевна, канд. техн. наук, доцент кафедры АПВТ ХНУРЭ. Научные интересы: сетевые технологии, технологии дистанционного образования. Увлечения: активный отдых, путешествия, иностранные языки. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: babich@kture.kharkov.ua

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.

ЭКСПЕРТНАЯ СИСТЕМА ВЫБОРА МОДИФИКАЦИИ МЕТОДА АНАЛИЗА ИЕРАРХИЙ

Предлагается экспертная система, которая позволяет охватить существующие модификации метода анализа иерархий. Реализация экспертной системы способствует повышению эффективности работы пользователей в интерактивной среде коллективного взаимодействия и обуславливает перспективность использования таких систем при решении практических задач.

1. Введение

В современных условиях большинство важных социально-экономических решений принимаются в процессе коллективного обсуждения. Групповое принятие решений предпочтительней индивидуального тем, что с увеличением сложности и объема задач один человек не может быть компетентным во всех вопросах и выполнить всей работы по выработке и реализации решения. Кроме того, отсутствие полной и точной информации, необходимой для принятия решения, приводит к тому, что выбор оптимального исхода происходит именно в результате группового обсуждения руководителями, специалистами, экспертами и консультантами.

Актуальность разработки моделей систем поддержки принятия групповых решений обусловлена потребностью в повышении эффективности коллективного взаимодействия на этапе принятия групповых решений за счет разработки новых, усовершенствовании и адаптации существующих методов принятия групповых решений.

Из множества известных методов и подходов к принятию решений наибольший интерес представляют те, которые дают возможность учитывать многокритериальность и неопределенность, а также позволяют осуществлять выбор решений из множеств альтернатив различного типа при наличии критериев, имеющих разные типы шкал измерения [1]. Среди таких методов, которые учитывают специфику процесса обсуждения проблемной ситуации, является классический метод анализа иерархий (МАИ), предложенный Т. Л. Саати [2].

Целью работы является разработка экспертной системы выбора модификации метода анализа иерархий на основе продукционной модели для интеллектуальных систем поддержки принятия групповых решений.

Задачи, которые необходимо решить для достижения поставленной цели:

– выполнить анализ существующих модификаций МАИ и провести их классификацию;

– разработать экспертную систему, которая бы обеспечила рациональный выбор методов принятия решений.

2. Постановка задачи

Для повышения степени объективности и качества процедуры принятия решений целесообразно учитывать мнение коллектива экспертов. МАИ [2] предоставляет возможность группе людей взаимодействовать по интересующей их проблеме, модифицировать свои суждения и в результате объединять их в соответствии с основным критерием: при проведении попарных сравнений объектов по отношению к некоторой характеристике, или характеристик относительно главной цели. Обратные связи обеспечивают ключ к объединению групповых суждений рациональным образом.

Данный метод позволяет выполнить исследование сложных систем взаимосвязанных компонент (ресурсы, желаемые исходы или цели) путем декомпозиции их на отдельные составляющие. Для оценки компонентов систем используется метод последовательного попарного сравнения объектов.

Можно выделить следующие этапы МАИ: построение иерархии, формирование матрицы попарных сравнений (МПС), получение вектора приоритетов, оценка степени согласованности МПС, получение наилучшей альтернативы.

С момента появления МАИ он был глубоко исследован, подробно изучены его преимущества и потенциальные недостатки, что привело к разработке множества модификаций метода, касающихся, в основном, изменения внутренних процедур (улучшения, усовершенствования определенного этапа) и направленных на исправление присущих классическому методу недостатков.

Для повышения эффективности систем поддержки принятия групповых решений необходимо выполнить разработку экспертной системы выбора модификации МАИ на основе продукционной модели.

3. Анализ модификаций классического метода анализа иерархий

В настоящее время существуют множество модификаций МАИ [3-19], которые можно классифицировать относительно следующих критериев:

- по типу системы предпочтения экспертов;
- по типу иерархий;
- по методу формирования матриц попарных сравнений;
- по способу измерения предпочтений экспертов (использование различных шкал измерения);
- по способу оценки и коррекции экспертных суждений;
- по методу синтеза итогового решения;
- в зависимости от структуры метода.

В табл. 1 приведено описание существующих модификаций классического МАИ.

4. Разработка экспертной системы выбора модификаций метода анализа иерархий

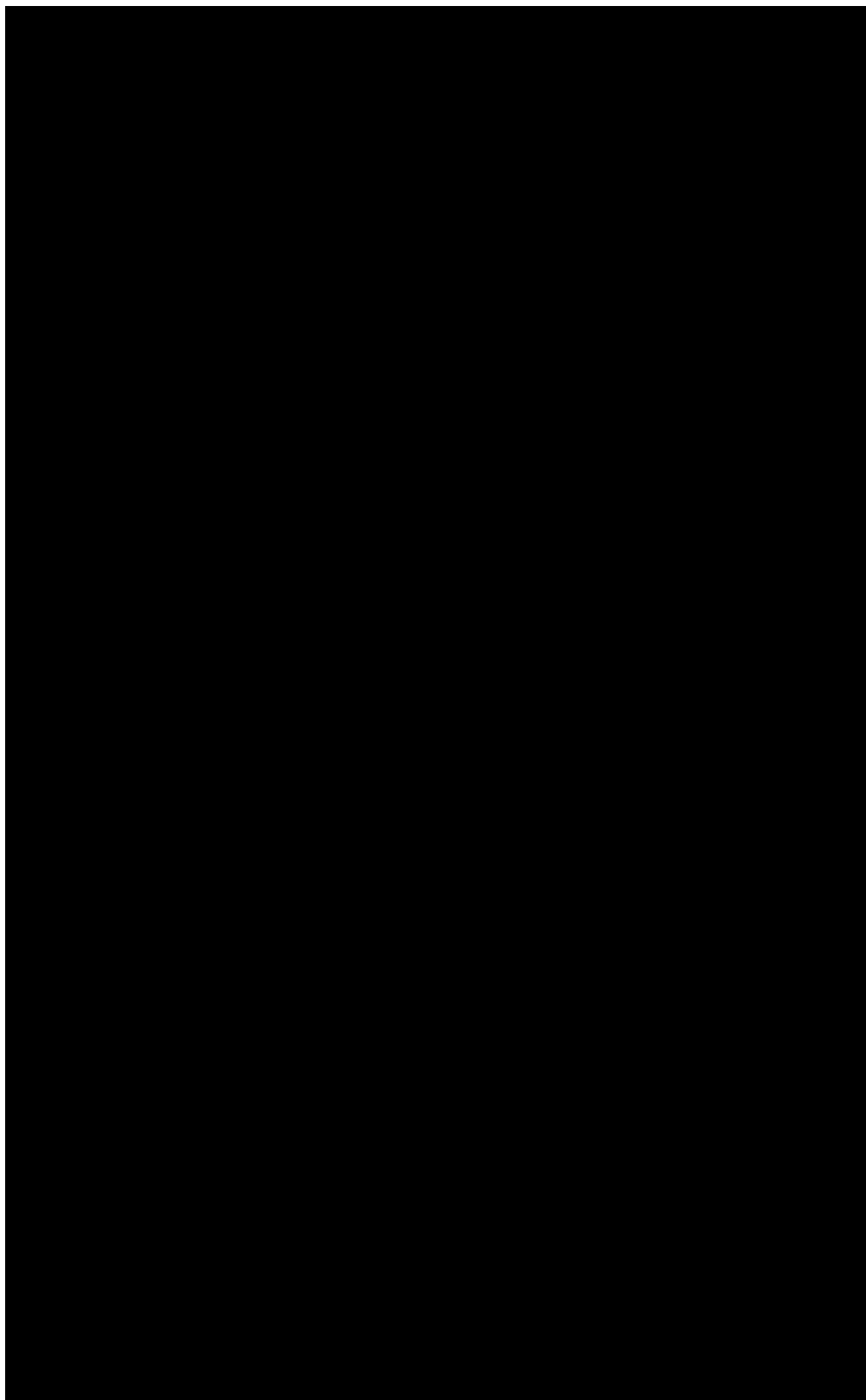
Задача анализа и выбора моделей принятия решений для автоматизированной системы поддержки группового обсуждения в основном соответствует постановке задачи определения предпочтительного варианта. В настоящее время проектирование систем компьютерной поддержки коллективного принятия решений связано с привлечением интеллектуальных средств, таких как модель представления, способы обработки, хранения и преобразования знаний. Наличие перечисленных элементов сочетается в интеллектуальных системах поддержки принятия решений (ИСППР). ИСППР относятся к классу интегрированных интеллектуальных систем, сочетающих строгие математические методы и модели поиска решения с нестрогими, эвристическими моделями и методами, базирующимися на знаниях специалистов, моделях человеческих рассуждений и накопленном опыте [20].

Ядром ИСППР является экспертная система (ЭС), в архитектуре которой выделяют следующие основные компоненты [21]:

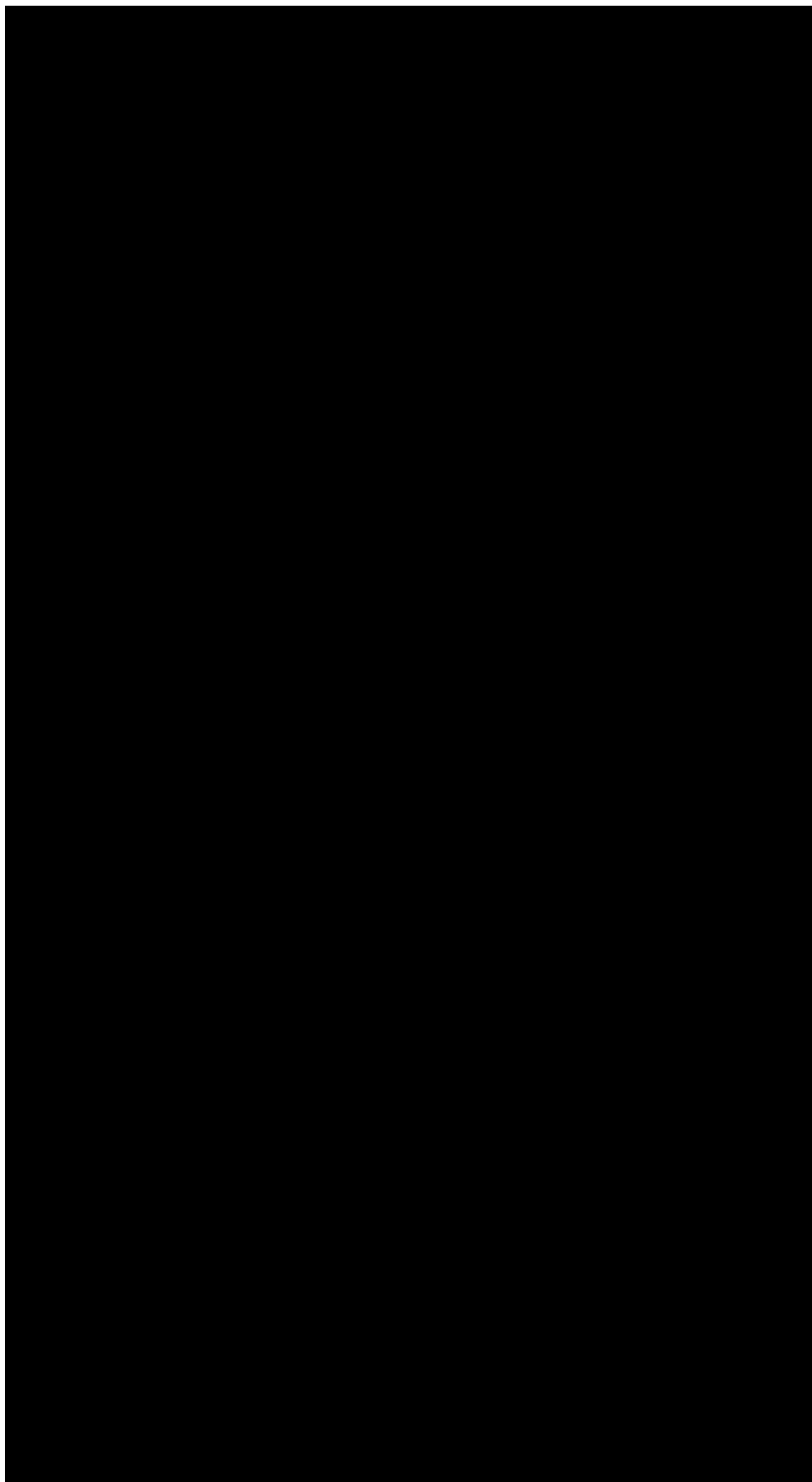
- базу знаний (факты, правила и эвристики, представляющие экспертные знания о предметной области);

– машину логического вывода (содержит стратегии и управляющие структуры, используемые для применения знаний, содержащихся в базе знаний для решения поставленной проблемы);

Таблица 1



Продолжение табл. 1



1. *Journal of the American Medical Association*, 1997; 277: 1039-1043.

- подсистему приобретения знаний (автоматизирует процесс заполнения экспертной системы знаниями);
- подсистему объяснения решений (позволяет контролировать ход суждений экспертной системы и объяснять ее решения или их отсутствие);
- рабочую память (база данных);
- пользовательский интерфейс, который управляет взаимодействием с пользователем (управление экраном, организация диалога, объяснительные способности системы).

В качестве модели представления знаний в проектируемой ЭС используется продукционная модель, в которой знания представлены совокупностью так называемых продукционных правил вида «если – то», дополняемых логическими операторами.

Разработка ЭС выбора модификации МАИ включает в себя формирование базы знаний «Модификации метода анализа иерархий». Опишем множество возможных фактов (базу фактов) $F = (f_1, f_2, \dots, f_{nf})$, формируемое при разработке данной информационной системы (табл. 2).

Заполнение базы правил (множество правил $P = (p_1, p_2, \dots, p_{np})$) выполняется в соответствии с синтаксисом продукционной модели. Приведем пример правила:

p_1 : ЕСЛИ *тип иерархии* = f_{11} ,

И *способ измерения предпочтений экспертов* = f_{21} ,

И *метод парных сравнений* = f_{31} ,

И *метод получения вектора приоритета* = f_{41} ,

И *способ оценки и коррекции экспертных суждений* = f_{51} ,

И *метод синтеза итогового решения* = f_{611} ,

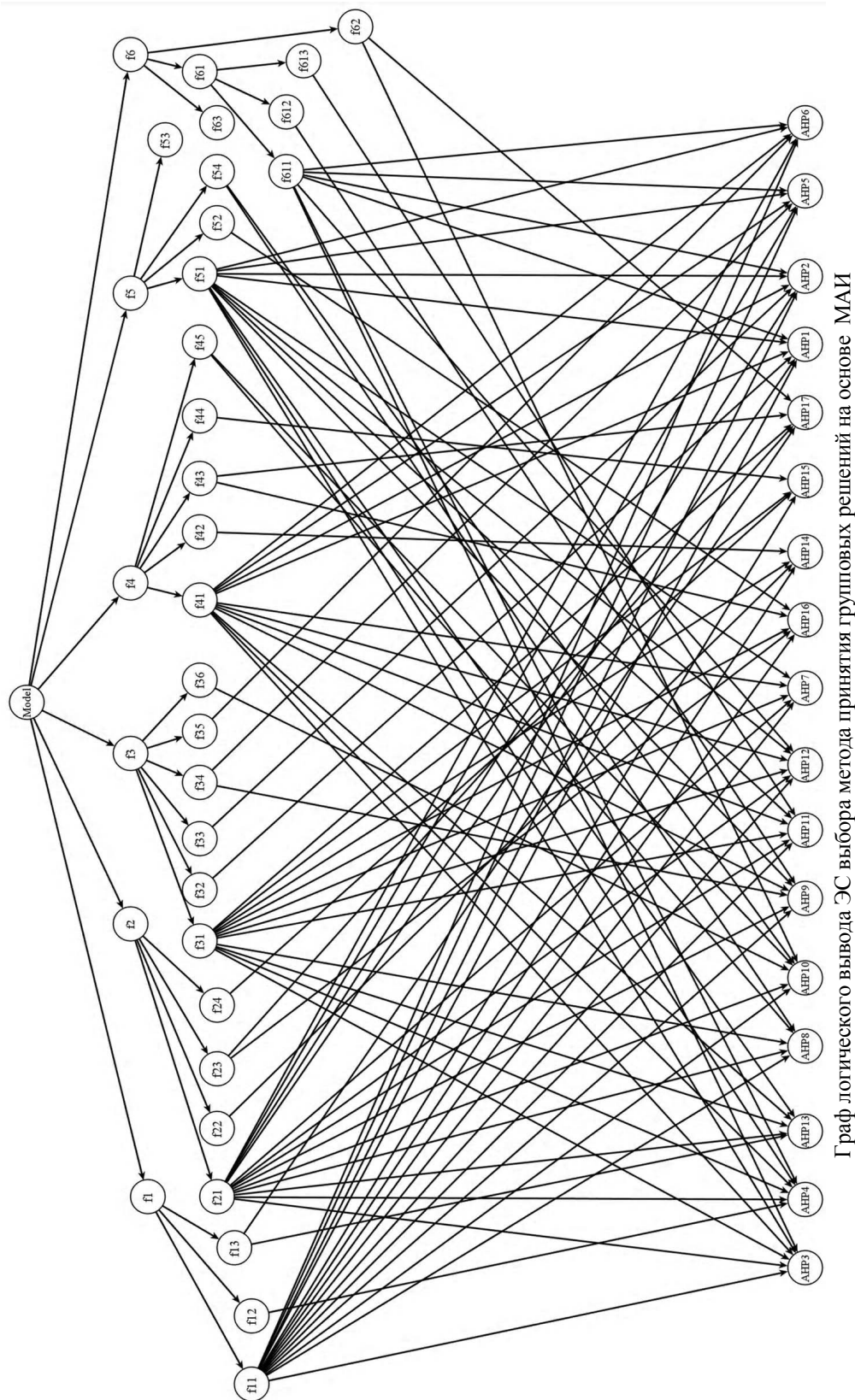
ТО *рекомендуемый метод* = *классический метод анализа иерархий*.

Сформируем граф логического вывода для разрабатываемой системы (рисунок). Вершины графа – факты базы знаний, дуги характеризуют механизм рассуждения ЭС, т.е. правила базы знаний.

Таблица 2

Название объекта	Определение сущности	Название атрибута (свойства) объекта
1	2	3
$f_1 = \text{тип иерархии}$	Характер связей между критериями и альтернативами	f_{11} = первый, f_{12} = второй, f_{13} = третий
$f_2 = \text{способ измерения предпочтений экспертов}$	Используемая шкала измерения суждений экспертов	f_{21} = точечная, f_{22} = стохастическая, f_{23} = нечеткая, f_{24} = интервальная
$f_3 = \text{метод парных сравнений}$	Способ формирования МПС	f_{31} = Саати, f_{32} = относительно стандартов, f_{33} = копированием, f_{34} = сокращенная процедура формирования МПС, f_{35} = модификация для случаев с большим количеством оцениваемых параметров, f_{36} = не требует формирования МПС
$f_4 = \text{метод получения вектора приоритета}$	Метод обработки МПС	f_{41} = для точечных МПС, f_{42} = для стохастических МПС, f_{43} = для нечетких МПС, f_{44} = для интервальных МПС, f_{45} = не требуется
$f_5 = \text{способ оценки и коррекции экспертных суждений}$	Способ оценки, улучшения согласованности и коррекции суждений	f_{51} = в зависимости от структуры метода, f_{52} = по типу оценок, f_{53} = по способу пересмотра, f_{54} = не требуется
$f_6 = \text{метод синтеза итогового решения}$	Метод получения наилучшего решения	f_{61} = способ свертки критериев, f_{611} = линейная, f_{612} = нелинейная, f_{613} = мультипликативная, f_{62} = тип модели, f_{63} = тип оценок

В результате работы создано программное обеспечение, реализующее предложенную ЭС выбора модификации МАИ, в состав которой входят следующие программные средства: пользовательский интерфейс, база знаний, редактор базы знаний, машина логического вывода, база моделей методов анализа иерархий. Разработанная ЭС обеспечивает рациональный выбор методов принятия групповых решений.



Граф логического вывода ЭС выбора метода принятия групповых решений на основе МАИ

5. Выводы

Выполнена разработка экспертной системы выбора модификации метода анализа иерархий на основе продукционной модели.

Научная новизна исследования состоит в том, что выполнена классификация существующих модификаций метода анализа иерархий по следующим критериям: по типу системы предпочтения экспертов; по типу иерархий; по методу формирования матриц попарных сравнений; по способу измерения предпочтений экспертов (использование различных шкал измерения); по способу оценки и коррекции экспертных суждений; по методу синтеза итогового решения; в зависимости от структуры метода, что позволяет осуществить рациональный выбор метода при принятии групповых решений, описать онтологию применения данного метода анализа иерархий в системах поддержки принятия групповых решений.

Практическая ценность исследования заключается в том, что разработана экспертная система, которая позволяет охватить существующие модификации метода анализа иерархий.

Использование реализованной экспертной системы в качестве надстройки для систем поддержки принятия коллективных решений приводит к дальнейшему развитию интеллектуальных систем поддержки принятия групповых решений и обуславливает перспективность применения данных программных средств при решении практических задач.

Список литературы: 1. Андрейчиков А. В. Анализ, синтез, планирование решений в экономике [Текст] / А. В. Андрейчиков, О.Н. Андрейчикова. М.: Финансы и статистика, 2002. 368 с. 2. Саати Т. Л. Принятие решений. Метод анализа иерархий [Текст] / Т. Л. Саати. М.: Радио и связь, 1993. 320 с. 3. Миронова Н.А. Обзор методов аналитической иерархии [Текст] / Н.А. Миронова, В.И. Дубровин // Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта: Материалы международной научной конференции. Том 1. Херсон: ХНТУ, 2009. С. 192-194. 4. Трофимец В.Я. К вопросу разработки вспомогательных вычислительных процедур МАИ: [Электрон. ресурс]. Режим доступа: <http://zhurnal.ape.relarn.ru/articles/2004/079.pdf> 5. Rotshtein A.P. Modification of the Saaty Method for the Construction of Fuzzy Set Membership Functions [Текст] / A.P. Rotshtein // Proc. of the International Conference on Fuzzy Logic and its Applications, Zichron, Israel, 1997. P. 125-130. 6. Сайко В. В. Модификация метода парных сравнений для случаев с большим количеством оцениваемых параметров [Текст] / В. В. Сайко // Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта: Материалы международной научной конференции. Том 1. Херсон: ХНТУ, 2009. С. 210-214. 7. Панкратова Н.Д. Комплексне оцінювання чутливості рішення на основі методу аналізу ієрархій [Текст] / Н.Д. Панкратова, Н.І. Недашківська // Системні дослідження та інформаційні технології. 2006. № 3. С. 7–25. 8. Ногин В. Д. Упрощенный вариант МАИ на основе нелинейной свертки критериев [Текст] / В. Д. Ногин // Журнал вычислительной математики и математической физики. 2004. Т. 44. № 7. С. 1259-1268. 9. Ротштейн А.П., Козачко А.Н. Нечеткий многокритериальный выбор альтернатив: метод наилучшего случая [Текст] / А.П. Ротштейн, А.Н. Козачко // Автоматика-2008: доклады XV международной конференции по автоматическому управлению, 23-26 сентября 2008 г. Одесса: ОНМА. С. 488-491. 10. Triantaphyllou E. Two new cases of rank reversals when AHP and some of its additive variants are used that do not occur with multiplicative AHP [Текст] / E. Triantaphyllou // Journal of multicriteria decision analysis. 2001. № 10. P. 11-25. 11. Недашківська Н.І. Оцінювання реверсу рангів у методі аналізу ієрархій [Текст] / Н.І. Недашківська // Системні дослідження та інформаційні технології. 2005. № 4. С. 120–130. 12. Saaty T.L. Decision making with Dependence and Feedback. The Analytic Network Process. [Текст] / T.L. Saaty; PWS Publications. Pittsburgh: 2001. 386 p. 13. Hahn E. D. Decision making with uncertain judgments: a stochastic formulation of the analytic hierarchy process [Текст] / E. D. Hahn // Decision Sciences. 2003. № 34(3). P. 443-466. 14. Kazutomi Sugihara. Interval priorities in AHP by interval regression analysis [Текст] / Kazutomi Sugihara, Hiroaki Ishii and Hideo Tanaka // European Journal of Operational Research. 2004. № 158(3). P. 745-754. 15. Lipovetsky S. Interval estimation of priorities in the AHP [Текст] / Stan Lipovetsky, Asher Tishler // European Journal of Operational Research. 1999. № 114(1). P. 153-164. 16. Панкратова Н.Д. Методология обработки нечеткой экспертной информации в задачах предвидения. Ч. 1 [Текст] / Н.Д. Панкратова, Н.И. Недашковская // Проблемы управления и информатики. 2007. № 2. С. 40-55. 17. Панкратова Н.Д. Методология обработки нечеткой экспертной информации в задачах предвидения. Ч. 2 [Текст] / Н.Д. Панкратова, Н.И. Недашковская // Проблемы управления и информатики. 2007. № 3. С. 49-63. 18. Ахрамейко А. А. Обобщение метода анализа иерархий Саати для использования нечетко-интервальных экспертных данных: [Электрон. ресурс]. Режим доступа: http://sedok.narod.ru/_s_files/belorusia_2002.htm 19. Prabjot K. A fuzzy ANP-based approach for selection ERP vendors [Текст] / K. Prabjot, N.C. Mahanti // International Journal of Soft Computing. 2008. № 3(1). P. 24-32.

20. Орлов С.П., Чуваков А.В., Нечаев Д.А. Разработка экспертной системы в составе интеллектуальной системы поддержки принятия решений в области водоподготовки и водоочистки природных вод // Современные наукоемкие технологии. 2010. № 5. С. 44–51. **21.** Субботін С.О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: Навчальний посібник. Запоріжжя: ЗНТУ, 2008. 341 с.

Поступила в редколлегию 03.12.2010

Миронова Наталья Алексеевна, ассистент кафедры программных средств ЗНТУ. Научные интересы: метод анализа иерархий, принятие групповых решений, информационные технологии. Адрес: Украина, 69063, Запорожье, ул. Жуковского, 64, тел. (061) 769-82-67.

Табунщик Галина Владимировна, канд. техн. наук, доцент кафедры программных средств ЗНТУ. Научные интересы: управление проектами, проектирование программного обеспечения, нечеткая логика. Адрес: Украина, 69063, Запорожье, ул. Жуковского, 64, тел. (061) 769-82-67.

МЕТОД ПРЕОБРАЗОВАНИЯ СТАТИЧЕСКИХ SPT АЛГОРИТМОВ В ДИНАМИЧЕСКИЕ

Предлагается метод трансформации статических алгоритмов поиска кратчайших путей на графах в динамические. Отличительной особенностью данного метода является поддержка специальных структур данных для коррекции текущих значений длин маршрутов. Приводятся зависимости числа операций сравнения от количества вершин графа для статических и динамических алгоритмов. Проводится краткий анализ полученных результатов.

1. Введение

В современной телекоммуникационной индустрии множество транспортных проблем, в частности проблем маршрутизации, нуждаются в очень гибких и эффективных решениях задач, связанных с поиском кратчайших путей на графах. При этом ключевым показателем эффективности решения задачи является, как правило, значение вычислительной сложности, с точки зрения продолжительности работы алгоритмов, количества операций сравнения, а также системных требований к оперативной памяти.

Непрерывный исследовательский интерес к проблемам маршрутизации в сетях передачи данных и работе алгоритмов SPT, в частности, сохраняется в течение последних десятилетий [1-3], и что частично мотивируется проблемой неустойчивости маршрутизации в сети Интернет [5].

Основной недостаток используемых в процессе маршрутизации статических алгоритмов заключается в необходимости полного пересчета дерева кратчайших путей (SPT) в случае изменения топологии сети. Фактически после некоторых изменений в состояниях связей топология нового SPT отличается от старого не значительно. Поэтому статические алгоритмы, которые повторно вычисляют SPT, явно неэффективны.

Цель исследования – преобразование статических SPT алгоритмов с организацией очередей в виде связанных списков в динамические для снижения алгоритмической сложности путем использования доступной информации SPT.

2. Метод преобразования алгоритмов

Пусть $G = (V, E)$ обозначает ориентированный граф, где V – набор узлов, а E – набор ребер в графе. Пусть N – общее количество узлов в V и E – общее количество ребер в E . Граф G состоит из корневого узла, обозначенного n_0 , и всех других узлов $n_1, \dots, n_{N-1}$, доступных из n_0 по ориентированным дугам в G . Каждое ребро e_i ($i = 0, \dots, E-1$) имеет положительный вес ω_i .

Для каждой дуги $e \in E$, пусть $W(e)$ обозначает вес (расстояние), ассоциированный с e , $S(e)$ и $E(e)$ обозначают соответственно начальный и конечный узел ребра e . Длина или

расстояние направленного пути – сумма весов ребер в пути. Дан ряд узлов $N \subseteq V$, который ассоциируется с двумя наборами ребер графа: $I(N) = \{e \in E \mid E(e) \in N\}$ (набор ребер, направленных в узлы N) и $O(N) = \{e \in E \mid S(e) \in N\}$ (набор ребер, направленных из узлов N). Пусть n_0 определяет корневой узел SPT.

Пусть древовидная структура данных T поддерживается алгоритмом для того, чтобы следить за существующим деревом кратчайших путей. Пусть каждый узел графа G , наряду с его родительским атрибутом $P(n, T)$, имеет список потомков $C(n, T)$, а также его атрибут расстояния $D(n, T)$. Структура данных T изменяется прогрессивно в течение вычисления, и когда работа алгоритма завершена, представляет собой SPT.

В дополнение к структуре данных T алгоритм также поддерживает связанный список Q , который временно содержит подмножество узлов вместе с двумя признаками. Каждый элемент в Q имеет форму $\{n, (x, D)\}$, где x определяет потенциального родителя для узла n , а D определяет потенциальное расстояние к узлу n .

Инструкция $ENQUEUE(Q\{n, (x, D)\})$ добавляет еще один элемент к Q . Если узел уже присутствует в очереди, то новые атрибуты заменяют старые. В любой момент только один набор атрибутов доступен для каждого узла в Q . Когда выполняется инструкция $EXTRACT(Q)$, выбирается первый (верхний) элемент и удаляется из очереди.

Алгоритм также использует вспомогательную процедуру. Процедура $B_{child}(n, T)$ находит множество узлов (включая n), которые являются потомками n . Это множество можно легко вычислить, спускаясь непосредственно вниз по T . SPT строится в T , изменяя атрибуты узлов в течение каждой итерации алгоритма.

Этот метод может быть использован в нескольких известных статических алгоритмах, например, таких как D'Esopo-Pape, Беллмана-Форда или Дейкстры. Основное различие в работе данных алгоритмов заключается в том, каким способом происходит организация временного связанного списка Q .

3. Спецификация алгоритма

Шаг 1: Инициализация

(A) Статическая версия

$\forall (n \in V) \neq \text{root}(G)$

$P(n, T) \leftarrow (O)$

$D(n, T) \leftarrow \text{MaxInt}$

$C(n, T) \leftarrow (O)$

$ENQUEUE(Q\{\text{root}(G), (O, D)\})$

Go To Step 2

(B) Динамическая версия (при наличии старого SPT)

Случай 1 (Ребро $e_i \in E^+$ увеличивает свой вес на Δ_i):

$\hat{N} \leftarrow \emptyset$

$\forall e_i \in E^+$

$W(e_i) \leftarrow W(e_i) + \Delta_i$

if $S(e_i) = P(E(e_i), T)$

$N \leftarrow B_{child}(E(e_i), T)$

$\forall n \in N$

$D(n, T) = D(S(e), T) + W(e)$

$\hat{N} \leftarrow \hat{N} \cup N$

$\forall e \in I(\hat{N})$
 if $D(E(e), T) > D(S(e), T) + W(e)$
 $D(E(e), T) = D(S(e), T) + W(e)$
 $C(S(e), T) \leftarrow C(S(e), T) + \{E(e)\}$
 $C(P(E(e), T), T) \leftarrow C(P(E(e), T), T) - \{E(e)\}$
 $P(E(e), T) \leftarrow S(e)$
 $ENQUEUE(Q\{E(e), (S(e), D)\})$
 Go To Step 2

Случай 2 (Ребро $e_i \in \mathcal{E}^-$ уменьшает свой вес на Δ_i):

$\forall e_i \in \mathcal{E}^-$
 $W(e_i) \leftarrow W(e_i) - \Delta_i$
 if $D(E(e_i), T) > D(S(e_i), T) + W(e_i)$
 $C(S(e_i), T) \leftarrow C(S(e_i), T) + \{E(e_i)\}$
 $C(P(E(e_i), T), T) \leftarrow C(P(E(e_i), T), T) - \{E(e_i)\}$
 $P(E(e_i), T) \leftarrow S(e_i)$
 $N \leftarrow B_{child}(E(e_i), T)$
 $\forall n \in N$
 $D(n, T) = D(S(e), T) + W(e)$
 $ENQUEUE(Q\{E(e), (S(e), D)\})$
 Go To Step 2

Шаг 2: Извлечение вершины

if $Q = \emptyset$
 Terminate
 else
 $\{y, (x, D)\} \leftarrow EXTRACT(Q)$

Шаг 3: Обновление расстояний

$\forall e \in O(y)$
 if $D(E(e), T) > D(y, T) + W(e)$
 $D(E(e), T) = D(y, T) + W(e)$
 $C(y, T) \leftarrow C(y, T) + \{E(e)\}$
 $C(P(E(e), T), T) \leftarrow C(P(E(e), T), T) - \{E(e)\}$
 $P(E(e), T) \leftarrow y$
 $ENQUEUE(Q\{E(e), y, D\})$
 Go To Step 2

Во время выполнения фазы инициализации производятся начальные установки для статической версии алгоритма. В дальнейшем также корректируются значения атрибутов, затронутых изменениями весов ребер графа узлов для динамической версии. Предполагается, что в существующем (устаревшем) SPT все узлы имеют родительский атрибут $P(n, T)$, имеют список потомков $C(n, T)$, а также атрибут расстояния $D(n, T)$. Алгоритм сначала обновляет SPT со всеми ребрами, веса которых увеличиваются (случай 1), а затем обновляет SPT со всеми ребрами, веса которых уменьшаются (случай 2).

В случае 1, после обновления каждого ребра e_i с новым увеличенным весом, проверяется, находится ли ребро в существующем SPT. Если условие выполняется, то выбирается узел $E(e_i)$ и все его потомки в существующем SPT. Все такие узлы включаются процеду-

рой $V_{child}(n, T)$ в множество N для дальнейшего обновления их расстояний. Для каждого ребра e , направленного в узел n , принадлежащий множеству N , производится сравнение текущего атрибута расстояния $D(n, T)$ с потенциально возможным расстоянием через $S(e_i)$. В случае, если потенциально возможное расстояние имеет меньшее значение, чем $D(n, T)$, происходит коррекция атрибута $D(n, T)$, узел n удаляется из списка потомков $P(n, T)$ и добавляется к списку потомков $S(e)$. Также изменяется родительский атрибут $P(n, T)$ на $S(e)$ и выполняется инструкция ENQUEUE.

Случай 2 на шаге инициализации более прост. После обновления каждого ребра e_i с новым уменьшенным весом, подобно случаю 1, производится сравнение текущего атрибута расстояния $D(n, T)$ с потенциально возможным, и если выполняется условие неравенства, с помощью процедуры $V_{child}(n, T)$ формируется множество N . Для всех узлов из N выполняется коррекция атрибутов $D(n, T)$, а также инструкция ENQUEUE.

После шага инициализации на шаге 2 из списка Q извлекается первый элемент. Как было замечено ранее, формирование связанного списка Q происходит в соответствии с используемым статическим алгоритмом.

На шаге 3 рассматривается каждый узел $E(e_i)$, который присоединен к ребру e_i , направленному из узла u . Если потенциально новое расстояние $E(e_i)$ (которое является равным $W(e_i)$ плюс расстояние $D(u, T)$) является меньшим, чем его старое значение, то происходит соответствующее корректирование атрибутов и $E(e_i)$ ставится в очередь в список Q . После того, как этот шаг закончен, выполнение шагов 2 и 3 повторяется до тех пор, пока список Q не будет пуст.

4. Методика моделирования и анализ полученных результатов

Было проведено экспериментальное моделирование в целях подтверждения теоретических выкладок, изложенных выше, проверки эффективности предложенного метода преобразования статических алгоритмов поиска кратчайших путей на графах в динамические, а также сравнительный анализ работы алгоритмов с точки зрения количества операций сравнения.

В ходе проведенного моделирования производилась генерация графов со случайной топологией в диапазоне числа вершин от 100 до 2000 с шагом в 100 вершин по методике, указанной в [4]. Средняя степень вершин составляла 8. Веса ребер графа устанавливались равными евклидовому расстоянию между вершинами:

$$L_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2},$$

где x_i, x_j, y_i, y_j – соответствующие координаты вершин i и j .

Для каждого значения числа вершин графа генерировалось 50 различных топологий. Для статических алгоритмов на графе 300 раз случайным образом выбиралась начальная вершина. Для динамических версий алгоритмов по каждой топологии графа 300 раз случайным образом менялись наборы весов ребер (число ребер, одновременно меняющих свой вес, находилось в интервале 5 – 30).

На рис. 1 показаны полученные зависимости числа операций сравнения от количества вершин графа для статических алгоритмов.

Из графиков, приведенных на рис.1, следует, что алгоритмы Беллмана-Форда и D'Esopo-Раре, имеющие большую асимптотическую сложность, на практике работают намного эффективнее алгоритма Дейкстры, что связано с проведением моделирования на разреженных графах. Полученные результаты указывают на то, что D'Esopo-Раре имеет явные преимущества перед алгоритмами Беллмана-Форда и Дейкстры.

В случае изменения состояния связей (весов ребер графа), как и ожидалось, за счет поддержки древовидной структуры T происходит локализация областей графа, для которых требуется повторное построение SPT. Стоит обратить внимание на то, что данный подход позволяет добиться значительного снижения алгоритмической сложности (рис. 2). Кроме того, в отличие от статических алгоритмов предложенный метод приводит к динамическим алгоритмам с сублинейной сложностью относительно размера сети.

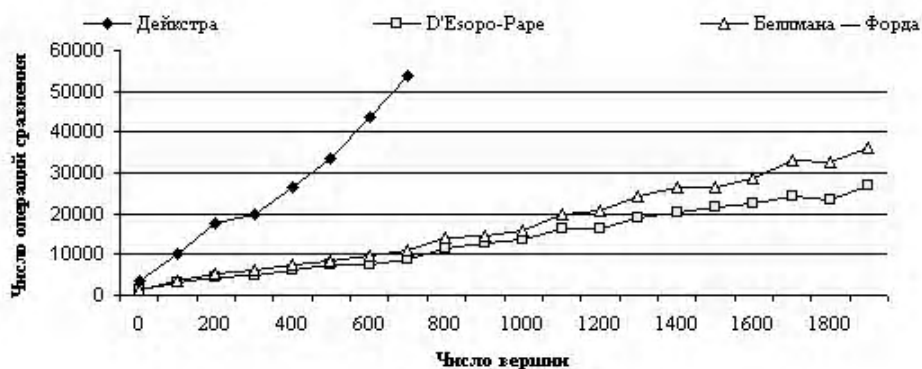


Рис. 1. Трудоемкость статических алгоритмов

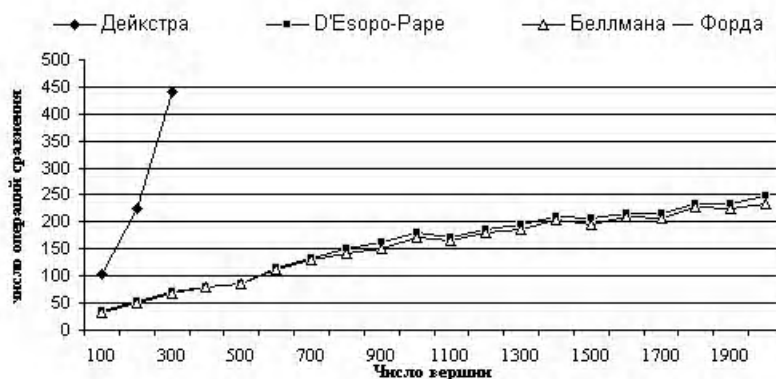


Рис. 2. Трудоемкость динамических алгоритмов при изменяющейся топологии графа

5. Заключение

Научная новизна. Для преобразования статических версий SPT алгоритмов в динамические предложен метод, позволяющий добиться значительного снижения алгоритмической сложности. Приведенные иллюстрации показывают существенное сокращение вычислительных затрат при использовании динамических версий алгоритмов.

Практическое значение. Показано, что использование динамических алгоритмов исключает необходимость полного пересчета дерева кратчайших путей в случае изменения топологии сети за счет использования доступной информации из устаревшего SPT. Таким образом, существует возможность значительного снижения вычислительной нагрузки на процессор маршрутизатора.

Список литературы: 1. Narvaez P., Kai-Yeung Siu, Hong-Yi Tzeng. New Dynamic SPT Algorithm Based on a Ball-and-String Model. IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 9, DECEMBER 2001. N 6. P. 706-718. 2. Breslau L., Estrin D. Design of inter-administrative domain routing protocols // Proc. SIGCOMM, Sept. 1990. P. 231-241. 3. Perlman R., Varghese G. Pitfalls in the design of distributed routing algorithms, in Proc. SIGCOMM, vol. Aug., 1988. P. 43-54. 4. Агеев Д. В. Модернизированная методика синтеза начальной структуры транспортной сети передачи данных. Одесса: Труды УНДИРТ, 2001, No 2(26). С. 42-47. 5. Dilworth S., Kutzarova D., Shuman K., Temlyakov V.N., Wojtaszchuk P. Weak Convergence of Greedy Algorithms in Banach spaces // Journal of Fourier Analysis and Applications. 2008. Vol. 14. P. 608-629.

Поступила в редколлегию 04.12.2010

Завизиступ Юрий Юрьевич, канд. техн. наук, проф. кафедры ЭВМ ХНУРЭ. Научные интересы: компьютерные сети, надежность и живучесть компьютерных систем. Увлечения: история науки и техники, литература. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 80577021354. E-mail: zaviz@kture.kharkov.ua

Партыка Станислав Александрович, ассистент кафедры ЭВМ ХНУРЭ. Научные интересы: компьютерные сети, математическое моделирование. Увлечения: путешествия, туризм. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 80577021354. E-mail: stas_partyka@mail.ru

ІМОВІРНІСНІ ТА СТАТИСТИЧНІ ХАРАКТЕРИСТИКИ МОДЕЛІ ВИДІЛЕННЯ НАЙБІЛЬШ СТАБІЛЬНИХ ОБ'ЄКТІВ КЛАСІВ

Розглядається консенсусний підхід до прийняття рішень. Консенсус будується на основі двох алгоритмів, що є максимально несхожі за числом об'єктів, на яких не досягається консенсус. Підхід дає можливість виділяти найбільш стабільні об'єкти, на яких досягається консенсус максимально несхожих алгоритмів. Одночасно підхід дає можливість будувати оцінки ймовірностей попадання кожного об'єкта до однієї із трьох груп об'єктів: групи об'єктів, на яких досягається правильний консенсус, групи об'єктів, на яких консенсусні алгоритми одночасно помиляються, та групи об'єктів, на яких не досягається консенсус.

1. Вступ

Алгоритми прийняття рішень використовуються в таких задачах розпізнавання образів: розпізнавання з вчителем та розпізнавання без вчителя. Задачі розпізнавання без вчителя носять назву задач кластеризації і стосуються теорії кластерного аналізу. Задачі розпізнавання, де передбачено втручання оператора в процес розпізнавання, стосуються теорії навчання, зокрема машинного. Великий напрямок в теорії машинного навчання носить назву статистичного машинного навчання, започаткований в роботах В. Вапніка та Я. Червоненкіса в 60-70-х рр. минулого століття і продовжений у 90-х рр. того ж століття [1]. Задача кластерного аналізу формулюється так. Потрібно розбити вхідні дані на зони (кластери) згідно з прийнятим критерієм. Критерієм може виступати, наприклад, розкид даних відносно деякого центру, що називається центром кластера. В теорії кластеризації найбільш відомі два алгоритми: алгоритм внутрішньогрупових середніх (k-means algorithm) та алгоритм максимізації математичного очікування (Expectation maximization (EM) algorithm) [2]. Останній є найбільш популярним алгоритмом в останній час, однак має той недолік, що часто не знаходиться глобальний максимум. В задачах кластеризації часто вважається, що немає жодної апріорної інформації про належність даних до відповідних кластерів. Із задачами кластерного аналізу та із застосуванням різних алгоритмів і підходів можна ознайомитись в [3, 4].

Потрібно відзначити, що алгоритми побудови класифікаторів на основі навчаючих вибірок є нестабільними, оскільки нерегулярною є сама навчаюча вибірка. Внаслідок цього і виникла ідея у розробці інших алгоритмів, які частково використовують статистичне машинне навчання, однак мають значно меншу чутливість до неоднорідності навчаючих вибірок.

2. Формулювання задачі

В даній роботі увага приділяється задачам, що частково використовують навчання, тобто задачам машинного навчання. Згідно із загальною концепцією машинного навчання генеральна вибірка розбивається на навчаючу та тестову, або контрольну підвибірки. Для навчаючої підвибірки вважається, що належність об'єкта до свого класу є відомою. На контролі перевіряється надійність роботи алгоритму. Надійність роботи алгоритмів перевіряється методами ковзаючого контролю, яких є достатньо велика кількість різновидів [5,6].

Залежно від складності класифікації всі об'єкти можна розбити на три групи: об'єкти, які є стабільними і класифікуються з великою надійністю, об'єкти, що належать пограничній зоні між класами, та об'єкти, що належать одному класу та занурені глибоко в середину іншого класу [7]. Серед тих об'єктів, які можуть спричинити помилку, найбільшу частину складають пограничні об'єкти. Тому важливим є розробити алгоритм, що дозволяє виділяти найбільшу кількість пограничних об'єктів. Оскільки класифікація пограничних об'єктів є ненадійною, то потрібно застосувати спеціальні верифікаційні алгоритми щодо визначення класів, до яких дійсно належать пограничні об'єкти. Однак дана робота присвячена першо-

му етапу ієрархічного алгоритму класифікації – розробці консенсусного алгоритму для детектування пограничних об'єктів. Також задача полягає у визначенні ймовірностей попадання об'єкта у кожну із згаданих вище груп об'єктів.

Оскільки математичний апарат, що використовується в даній роботі, відноситься до побудови комбінованих або ієрархічних алгоритмів, то важливо присвятити декілька розділів розгляду задачі класифікації саме з цієї точки зору.

3. Комбінування результатів класифікації

Розглянемо задачу класифікації об'єктів на n класів. Вважаємо, що класифікація здійснюється ансамблем або композицією класифікаторів. Нехай I – число об'єктів, а p – число алгоритмів класифікації. Кожний алгоритм (позначається індексом p , $p = 1, \dots, p$) асоціює кожний об'єкт з одним і лише одним класом. Можна зобразити результат роботи p -го алгоритму за допомогою бінарної прямокутної матриці B^p з I рядками і J_p стовпцями, де J_p – це число класів:

$$B_i^k = \begin{cases} 1, & i \in k; \\ 0, & i \notin k, \end{cases}$$

тут $i = 1, \dots, I$, $k = 1, \dots, J_p$, а B^p називається блочною матрицею. Перелічимо властивості матриці B^p :

- 1) Всі колонки матриці B ортогональні.
- 2) $\sum_{k=1}^{J_p} B_{ik}^p = 1$.
- 3) Якщо об'єкти i та l з одного класу (мають однакову мітку), то $\sum_{k=1}^{J_p} B_{ik}^p B_{lk}^p = 1$, інакше $\sum_{k=1}^{J_p} B_{ik}^p B_{lk}^p = 0$.

Представимо матрицю B у вигляді об'єднання матриць B^p так, що

$$B = [B^1, \dots, B^p, \dots, B^P].$$

Тут відстань між об'єктами i та l може бути представлена у вигляді добутку B_i та B_l . Якщо об'єкти i та l завжди в тому самому класі для різних алгоритмів P , то скалярний добуток векторів $B_i B_l'$ рівний P , в іншому випадку $B_i B_l' = 0$. Скалярний добуток

$$B_i B_l' = \sum_{k=1}^P \sum_{j=1}^{J_p} B_{ik} B_{lk} = 0$$

має дві важливі властивості:

- 1) обмеження знизу значенням нуля;
- 2) обмеження значенням P зверху.

Після нормалізації на P отримаємо $0 \leq B_i B_l' \leq 1$.

Відстань між об'єктами i та l може бути представлена у вигляді:

$$d(B_i B_l') = 1 - \sum_{k=1}^P \sum_{j=1}^{J_p} B_{ik} B_{lk}.$$

4. Блочні алгоритми

Визначимо евклідову відстань $d_E(B_i B_l')$ між точками B_i та B_l :

$$d_E(B_i, B_l) = (B_i - B_l)^2 = \sum_{k=1}^P \sum_{j=1}^{J_p} (B_{ik} - B_{lk})^2 = 2P - 2B_i B_l'.$$

Хемінгова відстань між об'єктами i та l представляється у вигляді:

$$d_H(B_i, B_l) = \frac{1}{2} d_E(B_i, B_l) = P - B_i B_l'.$$

Хемінгова відстань показує число біт, на які відрізняються дві двійкові послідовності.

5. Ймовірнісна модель кластеризації на основі розподілу Бернуллі

Оскільки матриця B є бінарною, то вважаємо, що всі її колонки мають розподіл Бернуллі з параметром μ_j :

$$P(B|\mu) = \prod_{j=1}^J \mu_j^{B_j} (1 - \mu_j)^{1-B_j},$$

де $\mu = \{\mu_1, \dots, \mu_j, \dots, \mu_J\}$. Середнє значення розподілу рівне $E[B] = \mu$, а коваріаційна матриця дорівнює $\text{cov}[B] = \text{diag}\{\mu_j(1 - \mu_j)\}$.

Суміш з K компонент розподілу Бернуллі для об'єкта B_i має вигляд

$$P(B_i | \mu, \alpha) = \sum_{k=1}^K \alpha_k P_k(B_i | \mu_k),$$

де $\mu = \{\mu_{1j}, \dots, \mu_{kj}, \dots, \mu_{Kj}\}$, а ймовірність для B_i даного μ_k є

$$P_k(B_i | \mu_k) = \prod_{j=1}^J \mu_{kj}^{B_{ij}} (1 - \mu_{kj})^{1-B_{ij}}.$$

Знайдемо правдоподібність для об'єкта B_i , даних μ і α , якщо дані у B незалежні та ідентично розподілені:

$$\log P(B | \mu, \alpha) = \log \prod_{i=1}^I P(B_i | \mu, \alpha) = \sum_{i=1}^I \log \left\{ \sum_{k=1}^K \alpha_k P_k(B_i | \mu_k) \right\}.$$

Використовуючи теорему Баєса, можна отримати ваги w_{ik} або умовну ймовірність того, що значення B_i належить класу k :

$$w_{ik} = \frac{\alpha_k P_k(B_i | \mu_k)}{\sum_{l=1}^K \alpha_l P_l(B_i | \mu_l)}.$$

Для визначення параметрів розподілу Бернуллі пропонується ЕМ – алгоритм. Параметри μ та α обчислюються на кроці М, а ваги обчислюються на кроці Е. Очікувана кількість

об'єктів N_k компонента k є: $N_k = \sum_{i=1}^I w_{ik}$.

Середнє значення μ_k компонента дорівнює

$$\mu_{kj} = \frac{1}{N_k} \sum_{i=1}^I w_{ik} B_{ij} \quad \text{і} \quad \alpha_k = \frac{N_k}{N}.$$

Відзначимо, що мають виконуватись такі нерівності:

$$\sum_{k=1}^K \alpha_k = 1;$$

$$\sum_{j=1}^J \mu_{kj} = 1;$$

$$0 \leq \alpha_k \leq 1;$$

$$0 \leq \mu_k \leq 1.$$

На рис.1 наведені псевдокоди ЕМ – алгоритму для розподілу Бернуллі.

- 1: Initialise K means μ_k and α_k .
- 2: **E-step**
 - 2.1: Calculate probabilities $P_k(B_i | \mu_k)$

$$P_k(B_i | \mu_k) = \prod_{j=1}^J \mu_{kj}^{B_{ij}} (1 - \mu_{kj})^{(1-B_{ij})}$$
 - 2.2: Calculate weights w_{ik}

$$w_{ik} = \frac{\alpha_k P_k(B_i | \mu_k)}{\sum_{l=1}^K \alpha_l P_l(B_i | \mu_l)}$$
- 4: **M-step** Re-estimate parameters N_k
 - 4.1: $N_k = \sum_{i=1}^I w_{ik}$.
 - 4.2: $\mu_{kj} = \frac{1}{N_k} \sum_{i=1}^I w_{ik} B_{ij}$.
 - 4.3: $\alpha_k = \frac{N_k}{N}$.
- 5: Evaluate the log-likelihood function:

$$\log P(B | \mu, \alpha) = \sum_{i=1}^I \log \left\{ \sum_{k=1}^K \alpha_k P_k(B_i | \mu_k) \right\}$$
- 6: If log-likelihood is converged,
then stop,
else go to Step 2.

Рис.1. Псевдокод ЕМ – алгоритму для розподілу Бернуллі

6. Поліноміальна модель

Поліноміальний розподіл застосовується тоді, коли елементи бінарної матриці є взаємовиключними. Якщо p – число різних кластеризацій, кожна з яких має K^p кластерів, $p = 1, \dots, P$, $j_p = 1, \dots, K^p$ – індекс кластеру j_p в кластеризації P . Ймовірність того, що $B_{ij_p} = 1$, рівна μ_{ij_p} . Тоді

$$P(B_{ij_p}) = \prod_{j_p=1}^{J_p} \mu_{ij_p},$$

тут $\mu_{ij_p} > 0$ і $\sum_{j_p=1}^{J_p} \mu_{ij_p} = 1, \forall i, p$. Поліноміальний розподіл при цьому запишеться у вигляді:

$$P(B | \mu) = \prod_{p=1}^P \prod_{j_p=1}^{J_p} \mu_{ij_p}^{B_{ij_p}},$$

де $\mu = \{\mu_1, \dots, \mu_{j_p}, \dots, \mu_{J_p}\}$. Суміш з k компонент для поліноміального розподілу запишеться у вигляді:

$$P(B_i | \mu, \alpha) = \sum_{k=1}^K \alpha_k P_k(B_i | \mu_k).$$

Функція відношення правдоподібностей та умовна ймовірність того, що $B_i \in k$, представляється аналогічно біноміальному випадку.

Очікувана кількість елементів N_k k -го компонента та інші елементи, що входять у модель суміші, представляється так:

$$N_k = \sum_{i=1}^I w_{ik};$$

$$\alpha_k = \frac{N_k}{N};$$

$$\mu_{kjp} = \frac{1}{N_k} \sum_{i=1}^I B_{ijp}.$$

Для ініціалізації коефіцієнтів α_k часто приймають, що $\alpha_k = 1 / K$.

На рис.2 наведені псевдокоди ЕМ – алгоритму для поліноміального розподілу.

1: Initialise K means μ_k and α_k .

2: E-step

2.1: Calculate probabilities $P_k(B_i | \mu_k)$

$$P_k(B_i | \mu_k) = \prod_{p=1}^P \prod_{j_p=1}^{J_p} \mu_{kj_p}^{B_{ij_p}}$$

2.2: Calculate weights w_{ik}

$$w_{ik} = \frac{\alpha_k P_k(B_i | \mu_k)}{\sum_{l=1}^K \alpha_l P_l(B_i | \mu_l)}$$

4: M-step Re-estimate parameters N_k

4.1:
$$N_k = \sum_{i=1}^I w_{ik}.$$

4.2:
$$\mu_{kj_p} = \frac{1}{N_k} \sum_{i=1}^I B_{ij_p}.$$

4.3:
$$\alpha_k = \frac{N_k}{N}.$$

5: Evaluate the log-likelihood function:

$$\log P(B | \mu, \alpha) = \sum_{i=1}^I \log \left\{ \sum_{k=1}^K \alpha_k P_k(B_i | \mu_k) \right\}$$

6: If log-likelihood is converged,

then stop,

else go to Step 2.

Рис.2. Псевдокод ЕМ – алгоритму для поліноміального розподілу

7. Модель кластеризації на основі визначення найбільш стабільних об'єктів

Ідея побудови даної моделі полягає в тому, що загальна сукупність об'єктів, які підлягають класифікації розбивається на три функціональні групи. До першої групи належать об'єкти, що мають високу надійність класифікації. Висока надійність класифікації означає, що об'єкти класифікуються коректно при сильних (максимальних) відхиленнях параметрів від оптимальних. З точки зору складності класифікації ці об'єкти відносяться до групи легких об'єктів. До другої групи належать об'єкти, на яких немає консенсусу. Якщо вибрати два алгоритми у композиції алгоритмів, то вони повинні бути максимально несхожі [8] і на них не повинно бути консенсусу. Якщо використовувати більшу кількість алгоритмів, то об'єкт належатиме до другої групи, якщо немає консенсусу на всіх алгоритмах. Якщо для побудови консенсусу використовувати проміжні алгоритми, параметри яких знаходяться в межах інтервалів між параметрами двох найбільш неподібних алгоритмів, то це не дає можливості виділяти більшу кількість об'єктів, на яких немає консенсусу. Неподібність між алгоритмами визначається на основі хемінгової відстані між результатами роботи двох алгоритмів, заданих у вигляді двійкових послідовностей. На практиці це означає також, що в цілому не будуть виділятися і нові за складом об'єкти, якщо використовувати консенсус із

більшого числа алгоритмів. Третя група складається з тих об'єктів, на яких помиляються обидва алгоритми і при цьому вони знаходяться у консенсусі. Помилка, що обумовлюється цими об'єктами, не може бути зменшена вже за жодних умов. Таким чином, помилка не може бути меншою за значення, що обумовлюється відносною долею об'єктів із третьої групи. Наступним кроком буде перекласифікація об'єктів із другої групи, тобто визначення, до якого саме класу належить той чи інший об'єкт.

Дослідження, що проводяться в цій роботі, стосуються аналізу статистичних характеристик результатів консенсусу, побудованих на основі двох алгоритмів. Задачею аналізу є визначення регулярності статистичних характеристик на різних підвбірках, взятих шляхом розбиття генеральної вибірки на блоки різних розмірів. Розподіли ймовірностей по консенсусу для трьох груп об'єктів здійснювались непараметричним оцінюванням за допомогою вікна Парзена з використанням гаусівських ядер.

На рис. 3-6 наведені графічні залежності результатів консенсусу для задач, взятих із репозиторію UCI. Цей репозиторій сформований у Каліфорнійському університеті. Структура даних в задачах із цього репозиторію є такою. Кожна задача записана у вигляді текстового файлу, де стовпцями є ознаки того чи іншого об'єкта, а рядки складаються із сукупності ознак для того чи іншого об'єкта, тобто кількість рядків відповідає кількості об'єктів, а кількості стовпців – число ознак для кожного об'єкта. Окремий стовпець складається із міток класів, якими позначений кожний об'єкт. Дуже багато задач із цього репозиторію стосуються таких галузей, як біологія та медицина. На рис. 3-6 суцільною лінією позначено розподіл результатів, отриманих за допомогою алгоритмів крос-валідації на підвбірках з мінімальним розміром $Q=200$, а штрихпунктирною – на підвбірках із мінімальним розміром $Q=30$.

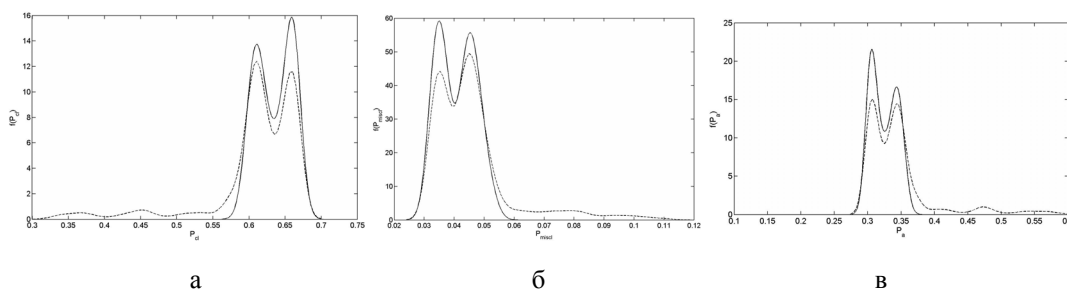


Рис. 3. Задача “prima-indians-diabetes” із репозиторію UCI: а – непараметрично оцінена густина розподілу ймовірностей правильного консенсусу із двох алгоритмів; б – непараметрично оцінена густина розподілу ймовірностей неправильного консенсусу із двох алгоритмів; в – непараметрично оцінена густина розподілу ймовірностей неконсensusу із двох алгоритмів

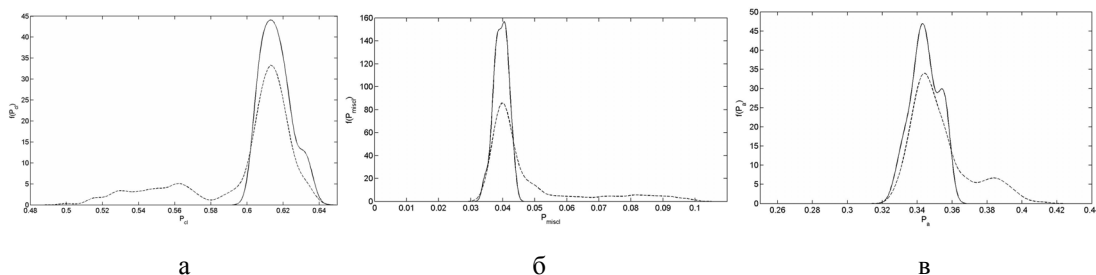


Рис. 4. Задача “vira” із репозиторію UCI: а – непараметрично оцінена густина розподілу ймовірностей правильного консенсусу із двох алгоритмів; б – непараметрично оцінена густина розподілу ймовірностей неправильного консенсусу із двох алгоритмів; в – непараметрично оцінена густина розподілу ймовірностей неконсensusу із двох алгоритмів

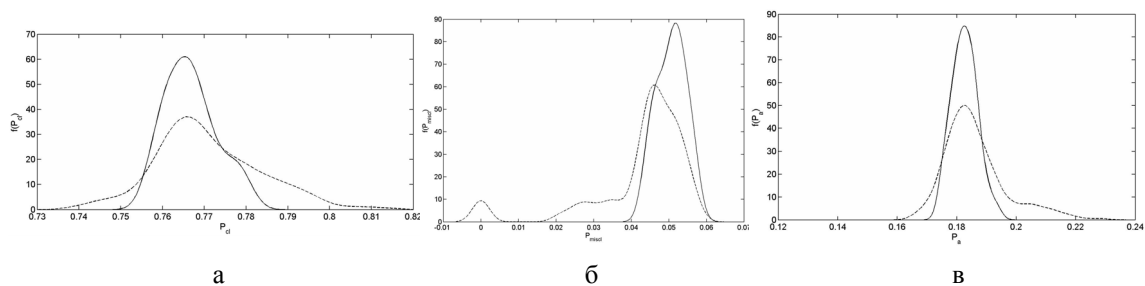


Рис. 5. Задача “haberman” із репозиторію UCI: а – непараметрично оцінена густина розподілу ймовірностей правильного консенсусу із двох алгоритмів; б – непараметрично оцінена густина розподілу ймовірностей неправильного консенсусу із двох алгоритмів; в – непараметрично оцінена густина розподілу ймовірностей неконсенсусу із двох алгоритмів

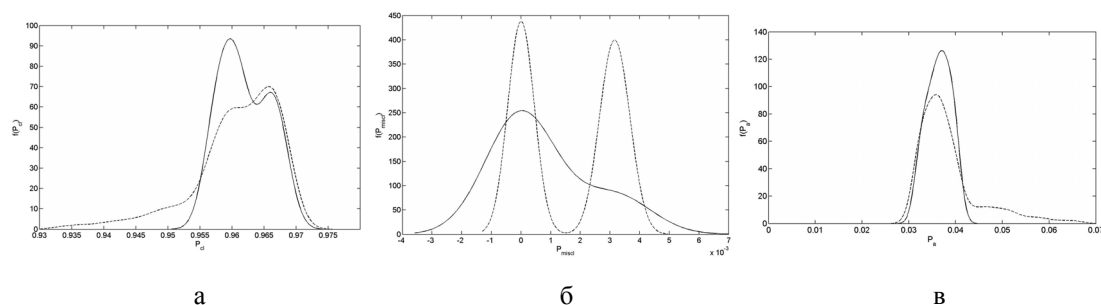


Рис. 6. Задача “dermatology” із репозиторію UCI: а – непараметрично оцінена густина розподілу ймовірностей правильного консенсусу із двох алгоритмів; б – непараметрично оцінена густина розподілу ймовірностей неправильного консенсусу із двох алгоритмів; в – непараметрично оцінена густина розподілу ймовірностей неконсенсусу із двох алгоритмів

У табл. 1-4 наведені оцінки ймовірностей попадання кожного об’єкта із задач репозиторію UCI у кожну із трьох груп об’єктів. У даному випадку об’єкти, на яких існує консенсус найбільш неподібних алгоритмів, відносяться до “простих” об’єктів, а об’єкти, на яких обидва алгоритми, що утворюють помилковий консенсус, відносяться до групи об’єктів, що спричиняють некоректовану помилку, яку не можна зменшити в принципі; об’єкти, на яких немає консенсусу, відносяться до групи пограничних об’єктів. У табл.1-4 також наведені дисперсії відповідних ймовірностей. Мінімальний розмір блоків, на основі яких побудовані оцінки, в одному випадку складає 30 об’єктів, а в іншому – 200.

Таблиця 1. Задача “prima” із репозиторію UCI

	Q=200		Q=30	
	μ	σ	μ	σ
Ймовірність консенсусу	0.635	0.024	0.611	0.064
Помилка консенсусу	0.041	0.006	0.046	0.013
Ймовірність неконсенсусу	0.324	0.019	0.344	0.052

Таблиця 2. Задача “bupa” із репозиторію UCI

	Q=200		Q=30	
	μ	σ	μ	σ
Ймовірність консенсусу	0.616	0.008	0.599	0.030
Помилка консенсусу	0.040	0.002	0.048	0.016
Ймовірність неконсенсусу	0.344	0.008	0.353	0.017

Таблиця 3. Задача “haberman” із репозиторію UCI

	Q=200		Q=30	
	μ	σ	μ	σ
Ймовірність консенсусу	0.767	0.006	0.771	0.013
Помилка консенсусу	0.051	0.004	0.043	0.013
Ймовірність неконсенсусу	0.183	0.004	0.186	0.011

Таблиця 4. Задача “dermatology” із репозиторію UCI

	Q=200		Q=30	
	μ	σ	μ	σ
Ймовірність консенсусу	0.962	0.004	0.961	0.007
Помилка консенсусу	0.002	0.002	0.001	0.001
Ймовірність неконсенсусу	0.036	0.003	0.038	0.007

У табл. 5 наведена доля помилок на тестових даних при тестуванні різних алгоритмів класифікації для двох задач із репозиторію UCI. Для запропонованого методу наведені мінімальні та максимальні помилки, що можна отримати на приведених тестових задачах.

Таблиця 5. Порівняння результатів класифікації різними методами

Метод	Задача	bupa	prima
Monotone (SVM)		0.313	0.236
Monotone (Parzen)		0.327	0.302
AdaBoost (SVM)		0.307	0.227
AdaBoost (Parzen)		0.33	0.290
SVM		0.422	0.230
Parzen		0.338	0.307
RVM		0.333	-
Запропонований метод (min/max)		0.040/0.212	0.041/0.203

В табл. 5 для запропонованого алгоритму значення мінімальної помилки дорівнює помилці консенсусу, а максимальної – сумі мінімальної та половині помилки неконсенсусу. Як видно із табл. 5, значення максимальної помилки значно менше від найменшого значення помилки всіх наведених алгоритмів для обох задач із репозиторію UCI. В порівнянні із деякими алгоритмами значення мінімальної помилки для запропонованого алгоритму менше на порядок. Запропонований метод характеризується значно більшою стабільністю помилок класифікації за інші методи.

8. Аналіз отриманих статистичних характеристик

На рис. 3-6 наведені параметрично оцінені густини розподілів ймовірностей для ймовірності правильного консенсусу, ймовірності неправильного консенсусу та ймовірності неконсенсусу. Як можна побачити з цих рисунків, дані розподіли можуть бути представлені за допомогою однокомпонентної, двоконпонентної або багатоконпонентної моделі розподілів. Багатоконпонентна модель задається сумішшю гаусіан, які входять із своїми коефіцієнтами впливу. Параметри розподілів та коефіцієнтів участі у моделі оцінюються за допомогою ЕМ – алгоритму. Оцінювання значень відповідних ймовірностей здійснювалось блоками з мінімальним розміром у $Q = 30$ та $Q = 200$ елементів. Дані розміри блоків обумовлюються розмірами малих вибірок, які за різними критеріями коливаються в межах від 30 до 200 елементів [9]. Згідно із стандартним означенням малою вибіркою називається вибірка, що характеризується нерегулярними статистичними характеристиками. Як видно із всіх рисунків, оцінки по блоках із мінімальним розміром у 30 елементів є нерегулярними, що вказує на те, що для даних задач підвибірки розміром у 30 елементів і дещо більшим є малими. На це вказують довгі хвости у відповідних розподілах ймовірностей. Максимум у нульовій точці для двоконпонентних моделей характеризується великою кількістю нульових ймовір-

ностей. Це може бути, коли немає помилок у роботі консенсусу із двох алгоритмів. Оцінки відповідних ймовірностей на основі середніх значень та за максимумом відповідних розподілів ймовірностей (оцінка за максимумом правдоподібності) не сильно відрізняються, що дає додаткову гарантію на правдоподібність відповідних оцінок. Значення отриманих оцінок ймовірностей правильного консенсусу, неправильного консенсусу та ймовірності неконсенсусу дає можливість оцінити складність задач з точки зору класифікації. Задачі та алгоритми оцінки складності задач розпізнавання розглянуті у [10]. Так, задачі “prima” і “bura” є приблизно однакові, оскільки значення зазначених трьох ймовірностей є приблизно однаковими. Задача “haberman” є менш складною, а задача “dermatology” є найпростішою із всіх чотирьох задач. Математичний аналіз побудови композицій алгоритмів детально розглянутий у [11].

9. Висновки

В роботі визначені ймовірності попадання в кожен із трьох груп об’єктів: група простих об’єктів, на яких досягається правильний консенсус із двох алгоритмів, група об’єктів, на яких досягнутий неправильний консенсус із двох алгоритмів, та група об’єктів на яких не досягається консенсус. Аналіз показує, що існують розподіли даних ймовірностей, які можна представляти у вигляді багатокомпонентних моделей, зокрема багатокомпонентної моделі суміші гаусіан. Все це дає можливість аналізувати запропоновані алгоритми методами математичної статистики та теорії ймовірностей. З наведених рисунків і таблиць видно, що оцінки ймовірностей при використанні методів ковзаючого контролю з усередненням по блоках з мінімальним розміром у 30 та 200 елементів мало відрізняються між собою, що дає можливість зробити висновок про те, що такий метод побудови консенсусу, де у консенсусі беруть участь найбільш несхожі алгоритми, є достатньо регулярним і не має такої чутливості до вибірки, як інші алгоритми, що використовують навчання. Як видно із відповідних таблиць, мінімальна помилка класифікації є практично на порядок меншою від найкращих існуючих алгоритмів, а максимальна помилка – меншою від 1,5 до 2-х разів. Також відповідні помилки є значно стабільніші як відносно задачі, на якій тестується метод, так і відносно серії наведених алгоритмів, де значення помилки має достатньо велику дисперсію. Більше того, оскільки значення мінімальної помилки є достатньо малим і стабільним, то це гарантує стабільність отримання коректних результатів класифікації на об’єктах, на яких досягається консенсус максимально несхожих алгоритмів. Відносно інших алгоритмів такої впевненості не буде. Дійсно, значення помилки на рівні 30-40% (у порівнянні із 4%) не дає жодної впевненості у результатах класифікації.

Список літератури: 1. *Vapnik V.* The nature of statistical learning theory. New York: Springer-Verlag, 2 edn, 2000. 2. *Bishop C.* Pattern recognition and machine learning. New York: Springer, 2006. 3. *Kyrgyzov I.* Recherche dans les bases de donnees satellitaires despaysages et application au milieu urban :clustering, consensus et catgorisation. Ph.D. thesis. Paris: L’cole Nationale Supriere des Tlcommunications, 2008. 4. *Taylor, J., Cristianini, N.:* Kernel methods for pattern analysis. New York: Cambridge University Press, 2004. 5. *Kohavi, R.* A study of cross-validation and bootstrap for accuracy estimation and model selection // 14th International Joint Conference on Artificial Intelligence, Palais de Congres, Montreal, Quebec, Canada, 1995. P. 1137–1145. 6. *Mullin, M., Sukthankar, R.* Complete cross-validation for nearest neighbor classifiers // Proceedings of International Conference on Machine Learning, 2000. P. 639–646. 7. *Vorontsov K.* Combinatorial approach to quality estimation of learning algorithms // Mathematical questions of cybernetic, 13, 2004. P. 5–36. 8. *Vorontsov K.* On the influence of similarity of classifiers on the probability of overfitting pattern recognition and image analysis: new information technologies // Pattern Recognition and Image Analysis: new information technologies (PRIA-9), Volume 2, Nizhni Novgorod, Russian Federation, 2008. P. 303–306. 9. *Гуров С.И.* Оценка надёжности классифицирующих алгоритмов. М.: Издательский отдел ф-та ВМиК МГУ, 2003. 45 с. 10. *Basu M.* Data complexity in pattern recognition. London: Springer, 2006. 11. *Zhuravlev, J.* About the algebraic approach to recognition or classification tasks solution // Problems of cybernetics, 33, 1978. P. 5–68.

Тянов Віталій Анатолійович, канд. техн. наук, старший викладач ЛДІНТУ ім. В’ячеслава Чорновола. Наукові інтереси: математичні методи розпізнавання образів. Адреса: Україна, 79057, Львів, вул. генерала Чупринки, 130, тел. 237-80-73, e-mail: vtayanov@yahoo.com.