

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 150

**Харьков
2010**

МОДЕЛЬ ЛОГИЧЕСКОГО ВЫЧИСЛИТЕЛЯ ДЛЯ АНАЛИЗА АССОЦИАТИВНЫХ ТАБЛИЦ

Предлагаются быстродействующие технологии описания и решения логических задач на основе использования графов ассоциативных таблиц путем аппаратно-ориентированных параллельных средств анализа графовых и табличных структур ассоциативных отношений для получения детерминированного многозначного вывода в n -мерном дискретном пространстве. Рассматриваются проблемы создания теоретических основ логических вычислений в форме: метрики ассоциативных отношений, модели описания ассоциаций, оптимизации графовых структур данных, моделей вычислительных процессов системного уровня.

1. Актуальность создания логических вычислителей

Мотивация – создание инфраструктуры для описания, анализа и синтеза логических ассоциативных отношений, моделирующих мозгоподобные (Brain-Like – BL) вычисления. К ним можно отнести: 1. Анализ и синтез синтаксических и семантических языковых конструкций для решения многочисленных естественных языковых задач (реферирование, исправление ошибок, анализ качества текстов). 2. Определение степени принадлежности объекта существующим ассоциативным компонентам BL-системы на основе введенных критериев качества выбора. 3. Распознавание видео- и аудио-образов путем их представления вектором существенных параметров. 4. Оперативное диагностирование неисправностей, технического состояния объекта и ремонт в процессе функционирования изделия. 5. Тестирование знаний и экспертное обслуживание объектов или субъектов для определения их валидности. Например, анализ кредитной истории для выдачи банком кредита. 6. Идентификация объекта или процесса для принятия решений в условиях неопределенности.

Существует также многообразие практически интересных для рынка задач, которые можно свести к выбору оптимального решения на основе использования логической структуры ассоциаций. 1. 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. 2. Коррекция текста в процессе его набора, когда автоматически исправляются только тривиальные ошибки, типа повторения буквы в слове, но можно корректировать более сложные ошибки, связанные с неверным окончанием, а также предлагать более приемлемые варианты порядка слов в предложении или в его части. Данная задача актуальна для 100% пользователей компьютеров. 3. Более серьезная проблема относится к критическим технологиям. Наведение на цель (выбор цели), будь то целеуказатель в истребителе, или в автоматической системе посадки лайнера. Здесь очень существенно функционирование системы наведения в реальном масштабе времени в микросекундном диапазоне измерения. 4. Обратной задачей выбора цели в критических технологиях является разведение объектов во времени и в пространстве, например, в диспетчерской службе аэропорта или оптимальной организации городского транспорта для исключения коллизий в воздушном пространстве и пробок на дорогах. Практически все упомянутые задачи должны решаться в реальном времени, они сходны по логической структуре ассоциативных многозначных векторов, которые составляют модель процесса или объекта. Они нуждаются в быстродействующей и специальной аппаратной платформе, которой может выступать логический ассоциативный мультипроцессор, ориентированный на параллельное выполнение поиска в ассоциативных структурах памяти и принятия (выбора) решения на основе использования интегрального критерия качества.

В плане предлагаемых исследований интересным представляется опубликованный в EE Times (12.2009) десяток перспективных технологий от Gartner Research Group (Gary Smith) на ближайшие годы. В нем нашли отражение только технологии, связанные с разработкой

специализированных цифровых изделий на кристаллах, хотя развитие программных технологий также будет оказывать сильное влияние на состояние рынка электроники. Глобально важными остаются технологии снижения потребляемой мощности и решения, направленные на уменьшение содержания ценных материалов в продукте. Данные технологии выступают двигателями многих направлений развития электроники, перечисленных ниже: 1. Биологическая обратная связь или электроника, управляемая мыслью. 2. Печатная электроника на основе использования органических материалов. 3. Пластиковая память на основе полимеров, проявляющих ферроэлектрические свойства. 4. Безмасочная литография на основе использовании электронного луча для создания топологии схемы. 5. Параллельная обработка данных для многоядерных гетерогенных графических процессоров. 6. Сбор энергии от механических и электрических процессов в окружающей среде. 7. Биоэлектроника и *wetware*, сочетающие биологические объекты и электронику для медицины. 8. Резистивное ОЗУ или мемристор с эффектом памяти как четвертый пассивный элемент электронной схемы, дополняющий резистор, конденсатор и индуктивность. 9. Переходные отверстия в кремнии (Through-Silicon-Via – TSV) для создания реальных 3D-SiP и кристаллов. 10. Различные технологии батарей на основе сочетания никеля и лития.

Цель – существенное повышение быстродействия анализа логических ассоциативных отношений путем аппаратно-ориентированной реализации параллельных средств обработки аналитических, графовых и табличных структур данных для детерминированного многозначного вывода в n -мерном дискретном пространстве.

Задачи: 1) Актуальность создания мозгоподобных вычислителей. 2) Метрика ассоциативных отношений. 3) Архитектуры ассоциаций: таблицы, графы, уравнения. 4) Оптимизация логических структур данных. 5) Модель вычислительного процесса системного уровня. 6) Практические примеры применения логического анализа ассоциативных таблиц.

Сущность – экспертное обслуживание запросов по анализу логических ассоциативных структур, ориентированных на предоставление пользователю детерминированного и многозначного вывода о принадлежности запроса к векторам n -мерного ассоциативного пространства.

Нейро-фаззи системы, изначально ориентированные на решение задач о принадлежности объекта или процесса существующим аналогам в реальном масштабе времени, остаются игрушками, если они не имплементируются в программируемый логический кристалл на основе использования структуры ассоциативных памяти. Кроме того, определенный интерес со стороны рынка электронных технологий вызывают работы, связанные с расширением функциональных возможностей структур ассоциативных памяти, использующих уже достаточно наработанную нейро-фаззи теорию. Взаимопроникновение практически ориентированных и теоретизированных подходов, положенных на современную технологическую платформу цифровых систем на кристаллах, может предоставить пользователю мощные средства для эффективного, в смысле быстродействия, решения задач искусственного интеллекта, информационно-логического синтеза и анализа словоформ и более сложных конструкций естественных языков.

Объект исследования – аппаратно-ориентированная математическая инфраструктура экспертного обслуживания задач информационно-логического синтеза и анализа в дискретном булевом пространстве на основе использования интегрального критерия качества и иерархических структур ассоциативных памяти.

Предмет исследования – логическая ассоциативная сеть экспертного обслуживания задач информационно-логического анализа и выбора решения путем использования интегрального критерия качества в дискретном булевом пространстве, ориентированная на имплементацию в иерархическую структуру ассоциативной памяти.

Источники. 1. Нейро-фаззи структуры, ориентированные на использование ассоциативной памяти [1-7]. 2. Структурные организации ассоциативных памяти для решения информационно-логических задач [8-14]. 3. Аппаратно-ориентированные методы и средства для решения информационно-логических задач [15-18].

Специализация и универсализация – две фазы спирального развития технологий компьютерной индустрии. Универсализация имеет стремление на создание еще более мощных изделий, или на предоставление пользователю новых функциональностей. Специализация

тяготеет к разработке более быстродействующих и дешевых устройств, ориентированных на эффективное обслуживание задач определенного класса. Затем разработчик начинает увеличивать функциональные мощности специального изделия, что означает новый виток на пути к универсализации.

Для решения 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 компилятора, 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.

2. Интегральная метрика определения расстояния и принадлежности

Речь идет о качестве взаимодействия запроса (входного многозначного, в частности, троичного вектора m) с системой ассоциативных векторов (ассоциаторов), результатом 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 $m = (m_1, m_2, \dots, m_i, \dots, m_n)$, $m_i \in \{0, 1, x\}$ и матрица ассоциаторов

$$S = (A_1, A_2, \dots, A_i, \dots, A_k),$$

$$A_i = (a_{i1}, a_{i2}, \dots, a_{ij}, \dots, a_{in}),$$

$$A_i \in S, a_i \in \{0, 1, x\}$$

должны иметь одинаковую размерность, равную n . Далее, для удобства изложения материала степень принадлежности m -вектора к одному ассоциатору или A -вектору будет обозначаться в виде $\mu(m \in A)$, $A \in S$.

Существует всего 5 видов или результатов взаимодействия (пересечения) двух векторов $m \cap A$, определенных на рис. 1.

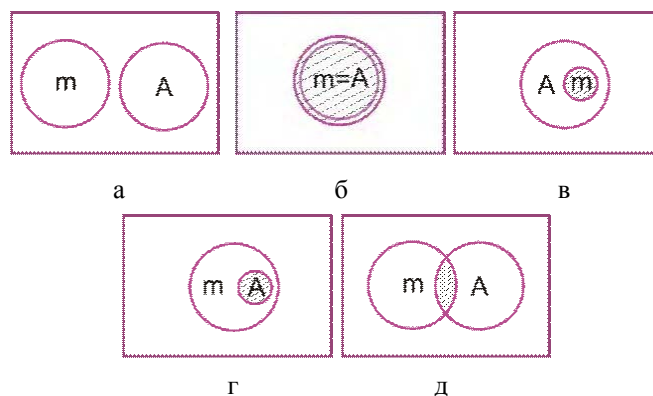


Рис. 1. Результаты пересечения двух векторов

Степень близости запроса – входного m -вектора к ассоциатору A – определяется кодовым расстоянием – мощностью множества пустых пересечений соответствующих координат взаимодействующих векторов (см. рис. 1, случай а):

$$d^*(m, A) = \text{card}(m_i \bigcap_{i=1}^n A_i = \emptyset).$$

При кодовом расстоянии, равном 0, степень (функция) принадлежности равна 1 только в случае полного совпадения векторов (см. рис. 1, случай б). Иначе, функция принадлежности определяет часть A-векторного пространства, которая принадлежит m-вектору (см. рис. 1, случай в). Степень принадлежности здесь есть степень участия m-вектора в формировании множества состояний A-пространства при кодовом расстоянии, равном нулю. Естественно, если кодовое расстояние отлично от 0, то степень принадлежности $\mu(m \in A) = 0 \leftarrow (d^*(m, A) > 0)$. Два вектора, один из них входной, другой – ассоциатор или вектор-строка, определенная в троичном или многозначном алфавите, в результате пересечения образуют непустой вектор, имеющий символы $X = \{0, 1\}$. В данном случае функция принадлежности определяется как общее пространство, отнесенное к пространству ассоциатора, при условии, что кодовое расстояние между векторами равно 0 – пересечение двух векторов не равно пустому множеству. Для троичных векторов, имеющих символы X, формула подсчета функции принадлежности имеет вид:

$$\mu(m \in A) = \frac{2^{\text{card}(m \cap A)}}{2^{\text{card}(A)}} = 2^{\text{card}(m \cap A) - \text{card}(A)} \leftarrow$$

$$\leftarrow \text{card}(m \cap A) = \text{card}(m_i \bigcap_{i=1}^n A_i = x) \ \& \ \text{card}(A) = \text{card}(\bigcup_{i=1}^n A_i = x).$$

Примеры вычисления функций принадлежности для различных вариантов взаимодействия двух векторов представлены ниже:

$$\mu(m \in A) = \left[\frac{m = (011xxx111)}{A = (01xxxxx1x)} \right] = \frac{2^3}{2^6} = \frac{1}{8} = 0,125;$$

$$\mu(m \in A) = \left[\frac{m = (011000111)}{A = (011000x1x)} \right] = \frac{2^0}{2^2} = \frac{1}{4} = 0,25;$$

$$\mu(m \in A) = \left[\frac{m = (xxx000111)}{A = (111000xxx)} \right] = \frac{2^0}{2^3} = \frac{1}{8} = 0,125;$$

$$\mu(m \in A) = \left[\frac{m = (xx100011x)}{A = (xx100011x)} \right] = \frac{2^3}{2^3} = \frac{8}{8} = 1;$$

$$\mu(m \in A) = \left[\frac{m = (xxx000111)}{A = (111000111)} \right] = \frac{2^0}{2^0} = \frac{1}{1} = 1.$$

Здесь фактически рассматривается принадлежность $\mu(m \cap A \in A)$ непустого пересечения между векторами. Но в последнем примере существует неучтенная часть пространства входного m-вектора, которая не вошла в A-ассоциатор, что также иллюстрируется рассмотренным случаем - (см. рис. 1). Оценка $\mu(m \in A) = 1$, полученная в результате взаимодействия векторов, не является корректной, поскольку она не учитывает мощность векторов, участвующих в пересечении. Улучшить оценку можно путем учета введения двух функций принадлежности, а также нормированного кодового расстояния $\mu(m \in A), \mu(A \in m), d(m, A)$. Итак, кодовое расстояние – грубая оценка взаимодействия векторов. Если $d^*(m, A) = 0$, то вектор m принадлежит вектору A:

$$\begin{cases} m \in A \leftarrow d^*(m, A) = 0 \leftarrow m \cap A \neq \emptyset; \\ m \notin A \leftarrow d^*(m, A) \neq 0 \leftarrow m \cap A = \emptyset. \end{cases}$$

При выполнении первого условия необходимо вычислять степень или функцию принадлежности, учитывая, что векторы могут создавать пять видов взаимодействия (см. рис. 1):

- 1) $[m \notin A \leftarrow m \cap A = \emptyset] \rightarrow d(m, A) = \frac{1}{n}(n - d^*)$;
- 2) $[(m \in A) \wedge (A \in m) \leftarrow m \cap A = (m \vee A) \leftarrow m = A] \rightarrow$
 $\rightarrow \mu(m \in A) = 2^{\text{card}(m \cap A) - \text{card}(A)}$;
- 3) $[m \in A \leftarrow m \cap A = m] \rightarrow \mu(m \in A) = 2^{\text{card}(m \cap A) - \text{card}(A)}$;
- 4) $[A \in m \leftarrow m \cap A = A] \rightarrow \mu(A \in m) = 2^{\text{card}(m \cap A) - \text{card}(m)}$;
- 5) $[(m \in A) \wedge (A \in m) \leftarrow m \cap A \neq (\emptyset \vee m \vee A)] \rightarrow$
 $\rightarrow \mu(m \in A) = 2^{\text{card}(m \cap A) - \text{card}(A) - \text{card}(m)}$.

Здесь относительно первой формулы следует подчеркнуть необходимость приведения кодового расстояния к нормированной оценке, определенной в интервале $[0,1]$. Данное обстоятельство вызвано тем фактом, что абсолютная целочисленная оценка не несет информации об относительном количестве несовпадений или пустых пересечений в масштабе количества (n) разрядов всего вектора. Например, два несовпадения для вектора, содержащего 20 переменных, и для вектора, имеющего 2 разряда, есть две большие разницы, которые следует учитывать при формировании функции принадлежности. Чтобы учесть все виды взаимодействия пары векторов (см. рис. 1), необходим общий и интегральный критерий качества. Все пять формул, представленных выше, необходимо объединить в интегральный критерий качества взаимодействия векторов.

Определение. Интегральная метрика для оценивания качества запроса есть функция качества (выбора) взаимодействия 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 $\mu(A \in m)$:

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

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:

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

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, если результат пересечения двух векторов равен пустому множеству, то степень двойки от символа «пусто» принимается равной 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.

Ниже приведены примеры расчета интегральной функции качества для следующих 4 векторов: (00001111, 0011xx11, 0000xxxx, xxxxxxxx), которые образуют пересечения «каждый с другими» и «каждый сам с собой». Функции принадлежности и качества для всех пар представлены в таблице.

Функции качества

| | $\mu(m \in A)$ | $\mu(A \in m)$ | $d(m, A)$ | Q |
|----------------------------------|------------------------------------|---------------------------------|------------------------|------|
| $m=(00001111)$ $A=(0011xx11)$ | $\frac{2^{\emptyset}}{2^2} = 0$ | $\frac{2^{\emptyset}}{2^0} = 0$ | $\frac{8-2}{8} = 0,75$ | 0,25 |
| $m=(00001111)$ $A=(0000xxxx)$ | $\frac{2^0}{2^4} = \frac{1}{16}$ | $\frac{2^0}{2^0} = 1$ | $\frac{8-0}{8} = 1$ | 0,69 |
| $m=(00001111)$ $A=(xxxxxxx)$ | $\frac{2^0}{2^8} = \frac{1}{256}$ | $\frac{2^0}{2^0} = 1$ | $\frac{8-0}{8} = 1$ | 0,67 |
| $m=(00001111)$ $A=(00001111)$ | $\frac{2^0}{2^0} = 1$ | $\frac{2^0}{2^0} = 1$ | $\frac{8-0}{8} = 1$ | 1 |
| $m=(0011xx11)$ $A=(0011xx11)$ | $\frac{2^2}{2^2} = 1$ | $\frac{2^2}{2^2} = 1$ | $\frac{8-0}{8} = 1$ | 1 |
| $m=(0011xx11)$ $A=(0000xxxx)$ | $\frac{2^{\emptyset}}{2^4} = 0$ | $\frac{2^{\emptyset}}{2^2} = 0$ | $\frac{8-2}{8} = 0,75$ | 0,25 |
| $m=(0011xx11)$ $A=(xxxxxxx)$ | $\frac{2^2}{2^8} = \frac{4}{256}$ | $\frac{2^2}{2^2} = 1$ | $\frac{8-0}{8} = 1$ | 0,67 |
| $m=(0000xxxx)$ $A=(0000xxxx)$ | $\frac{2^4}{2^4} = 1$ | $\frac{2^4}{2^4} = 1$ | $\frac{8-0}{8} = 1$ | 1 |
| $m=(0000xxxx)$ $A=(xxxxxxx)$ | $\frac{2^4}{2^8} = \frac{16}{256}$ | $\frac{2^4}{2^4} = 1$ | $\frac{8-0}{8} = 1$ | 0,69 |
| $m=(xxxxxxx)$ $A=(xxxxxxx)$ | $\frac{2^8}{2^8} = 1$ | $\frac{2^8}{2^8} = 1$ | $\frac{8-0}{8} = 1$ | 1 |

3. Инфраструктура решения задач анализа и синтеза на логических ассоциативных графах

Инфраструктура – совокупность моделей, методов и средств представления, анализа и синтеза структур данных для решения функциональных задач.

Итак, если говорить о функционировании инфраструктуры (системы) LAMP (Logical Associative Multiprocessor), то при подаче на ее вход некоторого вектора m , маскированного двоичным вектором X , она должна сформировать на выходе вектор A , максимально непротиворечивый для запроса m , а также оценку качества полученного решения в виде функции Q :

$$\begin{cases} Q = f(m, X, A); \\ A = g(m, X, S). \end{cases}$$

Структура интерфейса системы, соответствующая данным выражениям, представлена на рис. 2.

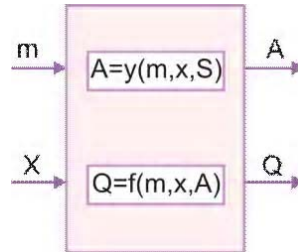


Рис. 2. Обобщенный интерфейс системы

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 пролетела мимо мишени; 2) пуля попала точно в цель и поразила ее полностью; 3) калибра пули недостаточно для поражения крупной цели; 4) мишень поражена необоснованно большим калибром пули (снаряда); 5) неэффективный и неточный выстрел снарядом большого калибра. Для решения данной практической задачи интегральный критерий качества дает точную оценку попадания (промаха), а также эффективность использования калибра оружия.

Поиск информации в Internet. Не обеспечивается глубина поиска информации по ассоциации признаков. Выдаются данные по запросу, которые чаще всего объединены функцией ИЛИ. Нет ассоциаций по минимальному кодовому расстоянию между запросом и информацией. Окончания слов оказывают большое влияние на информационный объем результирующей корзины. Здесь необходимы «умные» операции и структуры данных, а также эффективные критерии поиска информации.

Грамматическая (синтаксическая и семантическая) поддержка текстовых редакторов. Полезная функциональность – оперативная (в реальном времени) коррекция текста при его наборе в редакторе Microsoft Word. Слово с различными окончаниями (суффиксами, префиксами) воспринимается редактором как два различных слова. Решение – в плоскости анализа словоформы, когда ассоциация корня, суффикса и окончаний дают возможность исправлять ошибки при наборе текста. При этом должно учитываться синтаксическое и семантическое влияние окружающих слов. Задачи, подлежащие решению: 1) Определение оптимальной структуры логического ассоциативного графа, как формы, ориентированной на параллельное выполнение процедур анализа каждого компонента естественного языка. Структура также должна быть понятна и прозрачна для разработчика модуля IP-core, который, представляя законченную функциональность, должен компонентом входить в совокупную систему LAMP. Количество графов должно соответствовать частям речи, что представляет собой языковую структурно-логическую основу нижнего уровня. Далее над графами, как компонентами, создается интеллектуальная надстройка, определяющая практически существующие в языке ассоциативные бинарные (n-арные) логические отношения между частями речи в предложении, мощностью $\text{card} = 2, 3, 4, \dots, n$. Структура нижнего уровня предназначена для обслуживания иерархического словаря всех частей речи. Каждая из них также имеет свою иерархию в виде многозначного дерева, например, разделение всех прилагательных по первой букве. Древовидная структура обеспечивает быстрый поиск требуемого слова за счет избыточности надстройки выбора по первой букве над информативными массивами всех прилагательных. Для создания иерархического словаря LAMP, содержащего основу, окончания и признаки, необходимо иметь два компонента: электронный словарь естественного языка, к которому есть доверие со стороны разработчика, а также генератор моделей частей речи, представляющий собой анализатор (семантики и синтаксиса) естественно-языкового словаря для построения соответствующих графов (деревьев). Генератор выполняет функцию создания

полных моделей частей речи, которую можно определить как процесс интеллектуального обучения логических ассоциативных структур второго уровня. Третий уровень есть ассоциативные взаимодействия слов в предложении, где также следует разрабатывать генераторы, создающие надстройку над ассоциативными и словарными графами. Входной информацией для обучения данного уровня ассоциаций является доверительный контент (Пушкин А.С., Толстой Л.Н.), имеющий идеальные построения предложений естественного языка, которые можно принять за эталон в данном случае русского языка. Следующим может быть уровень, обслуживающий качество структуризации контента для легкости семантического понимания или восприятия совокупного сочинения. Все компоненты LAMP любого уровня иерархии в аппаратном исполнении представляют собой модули IP-cores, из которых собирается мозаика совокупной системы. Каждый модуль есть логический ассоциативный граф по форме, имплементируемый в структуру ассоциативных памяти и имеющий локальную систему управления параллельным вычислительным процессом. Для создания системы IP-cores необходимо иметь такое же количество генераторов моделей (IP-cores), которые синтезируют функциональности, как примитивные (0 и 1 уровней), так и более интеллектуальные, для верхних уровней (2-4) иерархии графа (дерева) естественного языка.

4. Модель анализа графа ассоциативных таблиц

Аналитическая модель описания и решения логических ассоциативных отношений представлена трехуровневой системой предикатов:

$$P(m, A) = Q(m, A) = [0, 0; 1, 0];$$

$$Q(m, A) = \frac{1}{3}[d(m, A) + \mu(m \in A) + \mu(A \in m)];$$

$$P(m, A) = 1 \leftarrow (m = A);$$

$$P(m, A) = \max_i Q_i(m, A_i);$$

$$Q(m, A_i) = \{Q_1, Q_2, \dots, Q_i, \dots, Q_n\};$$

$$P(m, A) = m \Delta (A_1, A_2, \dots, A_i, \dots, A_k);$$

$$\Delta = \{\neg, \wedge, \vee\};$$

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

$$A_{ij} = (A_{ij1}, A_{ij2}, \dots, A_{ijr}, \dots, A_{k sq}); \quad m = (m_1, m_2, \dots, m_r, \dots, m_q); \quad (2)$$

$$P(m, A) = m \wedge \{A_1, A_2, \dots, A_i, \dots, A_k\} = p; \quad p = \{p_1, p_2, \dots, p_r, \dots, p_n\};$$

$$m \wedge (A_1 \vee A_2 \vee \dots \vee A_i \vee \dots \vee A_k) = \max_i Q_i(m, A_i).$$

$$p_r (\in p) = m \bigwedge_{i=1, k}^{j=1, s} A_{ij} \leftarrow \begin{cases} \forall i \exists j (m \wedge A_{ij} \neq \emptyset); \\ \exists i \forall j (m \wedge A_{ij} = \emptyset) \wedge d(m, A_{ij}) = \max; \end{cases}$$

$$m \wedge A_{ij} = m_r \bigwedge_{r=1}^q A_{ijr} = \begin{cases} m \leftarrow m_r \bigwedge_{r=1}^q A_{ijr} = m_r; \\ A_{ij} \leftarrow m_r \bigwedge_{r=1}^q A_{ijr} = A_{ijr}; \\ m \wedge A_{ij} \leftarrow m_r \bigwedge_{r=1}^q A_{ijr} = m_r \vee A_{ijr}; \\ \emptyset \leftarrow \exists (m_r \bigwedge_{r=1}^q A_{ijr} = \emptyset); \end{cases}$$

$$P(m, A) = p_r (\in p) \leftarrow Q_i(m, p_r) = \max, \quad r = \overline{1, n}.$$

Здесь определены предикаты трех уровней иерархии: 1) Системный уровень функциональности $P(m, A) = Q(m, A)$ задает не структуры данных, а аналитическую модель вычис-

лительного процесса в виде предиката, определенного интегральным критерием принадлежности в интервале $Q(m, A) = [0, 0; 1, 0]$, на множестве введенных операций: объединение, пересечение и дополнение. 2) Система предикатов среднего уровня представляется в виде граф-упорядоченной совокупности $A = (A_1, A_2, \dots, A_i, \dots, A_k)$ взаимодействующих таблиц предикатов, принимающих только истинное значение. 3) Предикат нижнего уровня задает (упорядоченную) совокупность вектор-строк ассоциативной таблицы $A_i = (A_{i1}, A_{i2}, \dots, A_{ij}, \dots, A_{is})$, каждая из которых $A_{ij} = (A_{ij1}, A_{ij2}, \dots, A_{ijr}, \dots, A_{ksq})$ принимает только истинное значение. 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. Иначе, ассоциативность или равнозначность всех переменных в векторе $A_{ij} = (A_{ij1}, A_{ij2}, \dots, A_{ijr}, \dots, A_{ksq})$ создает для них равные условия существования, что означает инвариантность решения задач прямой и обратной импликаций в пространстве $A_i \in A$. Ассоциативный вектор $A_{ij} = (A_{ij1}, A_{ij2}, \dots, A_{ijr}, \dots, A_{ksq})$ определяет собой явное решение, где каждая переменная задается в конечном, многозначном и дискретном алфавите $A_{ijr} \in \{\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_l\} = \beta$. Взаимодействие $P(m, A)$ входного вектора-запроса $m = (m_1, m_2, \dots, m_r, \dots, m_q)$ с графом ассоциативных таблиц $A = (A_1, A_2, \dots, A_i, \dots, A_k)$ формирует множество решений $p = (p_1, p_2, \dots, p_r, \dots, p_n)$, каждое из которых имеет интегральную оценку качества $Q(m, A) = (Q_1, Q_2, \dots, Q_i, \dots, Q_n)$. Выбор лучшего из них осуществляется на основе критерия:

$$P(m, A) = p_r (\in p) \leftarrow Q_i(m, p_r) = \max, r = \overline{1, n}.$$

Следует заметить, что алгебраическая (аналитическая) форма представления графа – АФПГ (Graph Representation Algebraic Form – GRAF) [16]

$$m \wedge (A_1 \vee A_2 \vee \dots \vee A_i \vee \dots \vee A_k) = \max_i Q_i(m, A)$$

формирует точную модель взаимодействия во времени ассоциативных таблиц, ориентированных на параллельное выполнение вычислительного процесса. В данном случае она иллюстрирует факт равнозначности выполнения операции AND по отношению ко всем операндам, заключенным в скобки. Иначе, если необходима последовательность операций, представленная на рис. 3, то АФПГ будет иметь следующий вид:

$$p = m A_1 \vee (m A_4 \wedge m A_2) \vee m A_3 \approx (A_1 \vee A_4 A_2) A_3.$$

Здесь в структуре графа можно не учитывать входную переменную m , поскольку она связана со всеми вершинами. Фактически $p = (A_1 \vee A_4 A_2) A_3$ граф определяет последовательность параллельных (конвейерных) действий, необходимых для вычисления результата – синтеза словоформы прилагательного: 1) Параллельная обработка взаимодействия входного вектора с вершинами $A_1 \vee A_4$. 2) Анализ ассоциативной таблицы A_2 . 3) Обработка вершины A_3 завершает процесс формирования анализа графа ассоциативных таблиц. Второй граф, представленный на рис 3, задается предикатом $p = (A_4 A_2 A_3 A_1)$, который назначает процессору 4 такта для последовательной обработки вершин в целях анализа словоформы имени прилагательного.

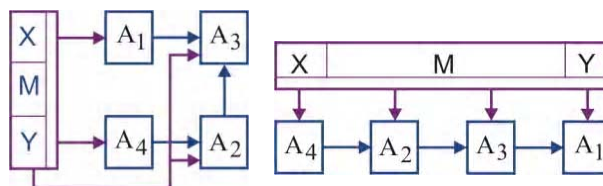


Рис. 3. Графы синтеза и анализа

Предикатная структура формирует дискретное пространство переменных или признаков, предельная мощность которого равна декартову произведению мощностей предикатов: $Card(A) = cardA_1 \times cardA_2 \times \dots \times cardA_3 \times \dots \times cardA_k$.

Конкретное взаимодействие предикатов между собой формирует функциональность $A = (A_1, A_2, \dots, A_i, \dots, A_k)$, 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) Древоподобная (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 за счет 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. Учитывая, что ассоциативную матрицу можно представить квадратом со стороной n , суммарные аппаратные затраты для реализации памяти системы имеют явно выраженную функциональную зависимость от числа разбиений, которая определяется следующим выражением:

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

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:

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

где t_{clk} – период синхронизации.

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. 4.

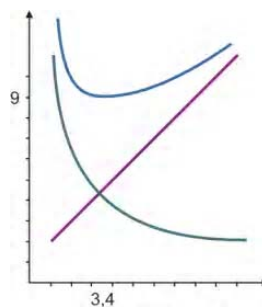


Рис. 4. Функции аппаратуры и времени от числа разбиений

Построение обобщенной функции эффективности графовой структуры от числа вершин

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

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

5. Диагностирование SoC на основе технологии LAMP

Логический граф ассоциативных таблиц совместно с метрикой оценивания решения предлагает интересную формулировку задачи технического диагностирования цифровых систем на кристаллах. Нельзя сказать, что данное направление является белым пятном в технической диагностике. Метод приближения [18] использует минимальное кодовое расстояние для поиска дефектов в цифровых структурах. Его недостатки связаны с ориентацией на одиночные константные неисправности булевых переменных, что порождает неприемлемо большой размер таблиц неисправностей для изделий, содержащих миллионы эквивалентных вентилей. Предлагаемый далее метод использует иерархическую инфраструктуру диагностирования, состоящую из системы таблиц неисправностей, оформленной в граф, но уже не элементарных дефектов, а неисправных функциональных компонентов (подсхем), взаимодействующих с вектором экспериментальной проверки m . Существенное дополнение метода заключается в использовании интегрального критерия качества для формирования более точного решения. Аналитическая модель процесса диагностирования представлена в виде целевой функции Z , обращающей критерий Q в минимум. Модель содержит параметры: число вершин графа G , соединенных дугами V , где для каждой вершины графа существует собственный входной вектор экспериментальной проверки из множества M :

$$\begin{aligned} Z &= f(G, V, M) = \min Q, \\ G &= (G_1, G_2, \dots, G_i, \dots, G_p); \\ V &= \{V_{ij} \leftarrow (G_i G_j)\}; \\ M &= (M_1, M_2, \dots, M_i, \dots, M_p); \\ A &= G_i; \quad m = M_i; \\ A &= [A] \vee [B] = [A_i] \vee [B_j], \\ A &= (A_1, A_2, \dots, A_i, \dots, A_n); \\ B &= (B_1, B_2, \dots, B_j, \dots, B_k). \\ S &= \{\cup, \cap, \neg, \oplus\}, \\ \{A_{ij}, B_{ij}, M_{ij}\} &\in \{0, 1, x\}. \end{aligned}$$

В зависимости от анализа строк или столбцов матрица (таблица) $A = [A_{ij}]$ будет записываться двумя способами – по строкам или столбцам:

$$\begin{aligned} A &= [A] \vee [B] = [A_i] \vee [B_j], \quad A = (A_1, A_2, \dots, A_i, \dots, A_n), \\ B &= (B_1, B_2, \dots, B_j, \dots, B_k). \end{aligned}$$

Две формы записи служат для сокращения сложности представления формул, задающих процесс обработки матрицы, а также для улучшения понимания аналитически записанных вычислительных процессов. Входной вектор m также будет иметь различные идентификаторы $m = m^a \vee m^b$, соответствующие анализу строк или столбцов таблицы. Четыре векторные операции (and, or, not, xor) позволяют решать интересные практические задачи логического анализа на графе ассоциативных таблиц, определенных в троичном алфавите.

6. Логический метод поиска дефектов по таблице неисправностей

Рассматривается на примере транзакционного графа одного из программных модулей Row_buffer, используемых при создании IP-core вейвлет-преобразования для стандарта JPEG 2000 (рис. 5). Вершины графа представлены входными шинами и переменными, регистрами и выходными шинами. Дуги между ними означают существование транзакций между вершинами при выполнении операторов HDL-кода.

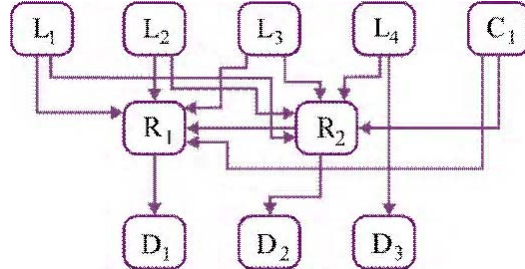


Рис. 5. Row_buffer транзакционный граф

Для данного Row_buffer графа построена таблица, задающая поведение неисправных блоков В на сгенерированном тесте А путем использования системы моделирования и генерации тестов SIGETEST [17]:

| Test | B ₁ | B ₂ | B ₃ | B ₄ | B ₅ | B ₆ | B ₇ | B ₈ | B ₉ | B ₁₀ | m ₁ ^b | m ₂ ^b |
|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-----------------|-----------------------------|-----------------------------|
| | L ₁ | L ₂ | L ₃ | L ₄ | C ₁ | R ₁ | R ₂ | D ₁ | D ₂ | D ₃ | | |
| A ₁ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| A ₂ | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| A ₃ | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| A ₄ | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| A ₅ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| A ₆ | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| A ₇ | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| A ₈ | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| A ₉ | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| A ₁₀ | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| A ₁₁ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| A ₁₂ | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| A ₁₃ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

6.1. Логический метод анализа столбцов таблицы неисправностей (ТН)

Он основан на применении операции логического умножения или конъюнкции вектора экспериментальной проверки, формально рассматриваемого в качестве входного вектор-столбца или маски m , на столбцы таблицы неисправностей $m^b \wedge (B_1 \vee B_2 \vee \dots \vee B_j \vee \dots \vee B_m)$ и подсчете качества взаимодействия векторов $Q_j(m^b \wedge B_j)$ в целях выбора лучшего из них при наличии максимальной оценки. При этом столбец $B_j \in A$ фактически идентифицирует метрику поведения неисправности или дефектного блока на тестовых наборах. Предикатная запись процесса получения решения в виде совокупности ошибок, присутствующих в HDL-коде, представлена в следующем виде:

$$\begin{aligned}
 P^s &= P^s \bigvee_{i=1}^n [B_j \leftarrow \max(Q_j(m^b \bigwedge_{j=1}^k B_j))]; \\
 P^k &= P^k \bigvee_{i=1}^n [B_j \leftarrow (m^b \wedge B_j = B_j)] \leftarrow \\
 &\leftarrow Q[m^b \wedge (\bigvee_{j=1}^k B_j \in P^k)] = 1 \vee \max;
 \end{aligned}
 \tag{3}$$

Здесь вектор экспериментальной проверки

$$m^b = f(A, B) \oplus f^*(A, B, L) \quad (3)$$

есть результат проведения тестового эксперимента – сравнение функционалов (состояний выходов) эталонного $f(A, B)$ и реального $f^*(A, B, L)$ устройства с дефектами L на тестовых наборах A .

Достоинство метода – выбор всегда лучшего решения из всех возможных как для одиночных, так и для кратных дефектов. В последнем случае, если одиночный дефект не идентифицируется оценкой, равной 1, выполняется дизъюнкция таких вектор-строк (главное отличие метода от существующих технологий), которые формируют оценку качества, равную 1 или максимально близкую к единице $Q[m^b \wedge (\bigvee_{j=1}^k B_j \in P^k)] = 1 \vee \max$. По существу, в список кратных дефектов включаются такие одиночные неисправности, которые при логическом умножении на вектор экспериментальной проверки дают результат в виде соответствующего вектор-столбца. Дизъюнкция всех столбцов, составляющих решение, равна

$$\bigvee_{j=1}^k (B_j \in P^k) = m^b.$$

Используя таблицу и процедуры диагностирования (2), определяем дефектные компоненты программного кода модуля Row_buffer методом логического умножения вектор-столбцов таблицы истинности на вектор экспериментальной проверки. Здесь векторы m_1^b, m_2^b формируют результаты диагностического эксперимента, выполненные по технологии (3). Результат диагностирования одиночных и кратных дефектов имеет следующий вид:

$$\begin{aligned} P^S(m_1^b) &= m_1^b \wedge (\bigvee_{j=1}^{10} B_j) = B_9 \rightarrow D_2; \\ P^k(m_2^b) &= m_2^b \wedge (\bigvee_{j=1}^{10} B_j) = B_1 \vee B_2 \rightarrow L_1 \vee L_2; \\ Q(m_1^b, D_2) &= 1; \\ Q(m_2^b, (L_1 \vee L_2)) &= \frac{1}{3} \left(\frac{4}{13} + \frac{1}{4} + 1 \right) = 0,52. \end{aligned} \quad (4)$$

В первом случае диагноз определен в виде одного дефектного блока D_2 , присутствующего в транзакционном графе, качество решения равно 1. Во втором случае процедура диагностирования выявила наличие двух дефектных модулей $L_1 \vee L_2$, которые не смогли сформировать идеальную оценку качества. Тем не менее, решение является лучшим среди всех возможных, которое максимально приближено к вектору экспериментальной проверки по критерию принадлежности $Q[m_2^b, (L_1 \vee L_2)]$.

Вычислительная сложность метода анализа столбцов определяется следующей зависимостью:

$$Z^c = 3n^2 + n^2 = 4n^2; Z^r = 3n + n = 4n.$$

Здесь первая оценка учитывает выполнение координатных операций над матрицей, размерностью $n \times n$. Вторая оценка определяет вычислительную сложность регистровых параллельных операций для подсчета критериев качества и обработки матрицы соответственно.

6.2. Логический метод анализа строк таблицы неисправностей

Стратегия определения ошибок программного кода по таблице неисправностей связана с анализом ее строк, состоящим из двух процедур: 1) Вычисление логического произведения конъюнкции строк, отмеченных единичными значениями вектора $A_i(m_i^b = 1)$, на отрицание дизъюнкции нулевых строк $A_i(m_i^b = 0)$ для одиночных дефектных блоков. 2) Вычисление логического произведения дизъюнкции единичных строк на отрицание дизъюнкции нулевых строк для кратных дефектных блоков:

$$P^s = \left(\bigwedge_{\forall m_i^b=1} A_i \right) \wedge \left(\overline{\bigvee_{\forall m_i^b=0} A_i} \right);$$

$$P^m = \left(\bigvee_{\forall m_i^b=1} A_i \right) \wedge \left(\overline{\bigvee_{\forall m_i^b=0} A_i} \right); \quad (5)$$

Формулы интересны тем, что они не привязаны к критериям качества диагностирования, а оперируют лишь двумя компонентами, таблицей неисправностей и вектором экспериментальной проверки. Выполнение процедуры диагностирования по формулам (5) для вектора экспериментальной проверки $m^b = (0101010010010)$, заданного в последней таблице неисправностей, дает результат: $P^s(m_1^b, A) = D_2$, который не хуже, чем ранее полученный методом анализа столбцов. Для вектора экспериментальной проверки $m^b = (1110011100000)$ результат диагностирования имеет вид: $P^k(m_2^b, A) = L_1 \vee L_2$. Вычислительная сложность метода анализа строк определяется следующей зависимостью: $Z^c = n^2$; $Z^r = n$. Первая оценка предназначена для подсчета числа координатных операций, вторая определяет вычислительную сложность процесса обработки на основе регистровых параллельных операций.

7. Выводы

1. *Научная новизна* представлена алгеброй логического анализа векторных и табличных форм задания информации для решения задач поиска, диагностирования, распознавания образов и принятия решений в векторном дискретном булевом пространстве. Предложены быстродействующие модели и методы параллельного векторного логического анализа информации, в пределе полностью исключают использование арифметических операций, в том числе и для подсчета критерия качества решения, где применяются только логические операции. В качестве примеров описаны новые методы для решения задач диагностирования и нахождения квазиоптимального покрытия, использующие векторные операции, ориентированные на распараллеливание вычислительных процессов.

2. *Практическая значимость* заключается в ориентации алгебры ассоциативных таблиц и методов их анализа на создание логического мультипроцессора с ограниченной системой команд, ориентированной на высокое быстродействие параллельной обработки больших массивов информации, представленных графовыми структурами ассоциативных матриц. Дальнейшие исследования будут направлены на разработку прототипа мультипроцессора и исследование новых практических задач с помощью предложенной алгебры.

Список литературы: 1. Zorian Yervant. Test Strategies for System-in-Package // The Plenary Paper of IEEE East-West Design & Test Symposium (EWDTS'08). Lvov, Ukraine. October 9-12. 2008. 2. Smith L. 3D Packaging Applications, Requirements, Infrastructure and Technologies // 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. Какурин Н.Я., Хаханов В.И., Лобода В.Г., Какурина А.Н. / Регистр сдвига. А.С. №1439682. 22.07.88. 4с. 5. Бондаренко М.Ф., Дударь З.В., Ефимова И.А., Лецинский В.А., Шабанов-Кушнаренко С.Ю. О мозгоподобных ЭВМ // Радиоэлектроника и информатика. Харьков: ХНУРЭ. 2004. № 2. С. 89–105. 6. Бондаренко М.Ф., Шабанов-Кушнаренко Ю.П. Об алгебре предикатов // Бионика интеллекта. Харьков: ХНУРЭ. 2004. № 1. С. 15–26. 7. Бондаренко М.Ф., Шабанов-Кушнаренко Ю.П. Теория интеллекта. Учебник. Харьков: СМИТ, 2006. 592 с. 8. Бондаренко М.Ф., Шабанов-Кушнаренко Ю.П. Модели языка // Бионика интеллекта. Харьков: ХНУРЭ. 2004. № 1. С. 27–37. 9. Акритас А. Основы компьютерной алгебры с приложениями: Пер. с англ. М.: Мир, 1994. 544 с. 10. Гилл Ф. Мюррей У., Райт М. Практическая оптимизация. Москва. Мир. 1985. 509с. 11. Амтетков А.В., Галкин С.В., Зарубин В.С. Методы оптимизации. М.: Издательство МГТУ им. Н.Э. Баумана. 2003. 440 с. 12. Дегтярев Ю.И. Методы оптимизации: Учебное пособие для вузов. М.: Сов. радио, 1980. 270с. 13. Bergeron J. Writing Testbenches Using SystemVerilog / J. Bergeron // Springer Science and Business Media, Inc. 2006. 414 p. 14. Abramovici M., Breuer M.A. and Friedman A.D. Digital System Testing and Testable Design. Comp. Sc. Press. 1998. 652 p. 15. Densmore Douglas 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., Литвинова Е.И., Гузь О.А. Проектирование и тестирование цифровых систем на кристаллах. Харьков: ХНУРЭ. 2009. 484с. 17. Hahanov V.I., Gorbunov D.M., Miroshnichenko Y.V., Melnikova

O.V., Obrizan V.I., Kamenuka E.A. SIGETEST – Test generation and fault simulation for digital design // Сб. материалов научно-практической конференции “Современные технологии проектирования систем на микросхемах программируемой логики”. Харьков. 2003. С. 50-53. 18. Автоматизация диагностирования электронных устройств/ Ю.В.Малышенко и др. / Под ред. В.П. Чипулиса. М.: Энергоатомиздат, 1986. 216с.

Поступила в редколлегию 11.01.2010

Бондаренко Михаил Федорович, ректор ХНУРЭ, д-р техн. наук, профессор кафедры ПОАС ХНУРЭ. Научные интересы: техническая диагностика цифровых систем, сетей и программных продуктов. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua.

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.

УДК 519.63:532.5

А.П. БЛИШУН, М.В. СИДОРОВ, И.Г. ЯЛОВЕГА

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ СТАЦИОНАРНЫХ ФИЛЬТРАЦИОННЫХ ТЕЧЕНИЙ СО СВОБОДНОЙ ГРАНИЦЕЙ МЕТОДОМ R-ФУНКЦИЙ

Рассматривается задача расчета фильтрационного течения жидкости со свободной границей. Предлагается численный метод её решения, использующий конструктивную теорию R-функций для построения структуры решения краевой задачи, и метод наименьших квадратов для итерационного уточнения свободного участка границы области фильтрации.

Введение

Актуальность исследования. В последние десятилетия наблюдается катастрофическое снижение естественной дренажности и устойчивый подъем уровня грунтовых вод, что приводит к подтапливанию аграрных, городских и других ландшафтов. Поэтому проблема математического моделирования движения грунтовых вод (т.н. фильтрационных течений) в пористых средах является актуальной. Методы моделирования физических процессов в пористых средах можно найти в [1 – 5]. Математическое моделирование фильтрационных течений использует методы теории функций комплексного переменного. Однако они часто труднореализуемы в областях сложной геометрии. Еще больше вычислительных сложностей возникает, если область фильтрации содержит участки свободной поверхности. Например, в [6] предлагается делать в таких задачах невырожденную замену переменной, которая преобразует область со свободной границей в область с границей фиксированной, однако делает уравнение задачи нелинейным. В связи с этим разработка новых численных методов решения задач этого класса является актуальной.

Цели и задачи исследования. Целью настоящего исследования является разработка нового метода решения краевых задач теории фильтрации со свободной границей, который бы точно учитывал всю имеющуюся аналитическую и геометрическую информацию. Для этого необходимо решить следующие задачи:

- построить полную структуру решения краевой задачи, т.е. пучок функций, точно удовлетворяющий всем краевым условиям задачи;
- описать алгоритм итерационного уточнения уравнения свободной границы.

Для решения поставленных задач будут использованы методы конструктивной теории R-функций и методы аппроксимации функций в гильбертовых пространствах.

Настоящая работа распространяет результаты, полученные в [7, 8], на случай наличия в области свободной границы. В работе не рассматриваются вопросы существования и

единственности поставленных краевых задач. Предполагается, что все задачи поставлены корректно в некоторых функциональных пространствах для заданных входных данных и все математические модели рассматриваются с точки зрения их алгоритмизации для дальнейшего решения на ЭВМ.

1. Постановка задачи

Рассмотрим случай двумерного несжимаемого установившегося потока через ортотропный грунт в области Ω , для которого закон Дарси может быть записан следующим образом: $v_1 = -k_{11} \frac{\partial u}{\partial x_1}$, $v_2 = -k_{22} \frac{\partial u}{\partial x_2}$. Здесь u – напор, k_{11} , k_{22} – коэффициенты фильтрации, v_1 , v_2 – компоненты скорости потока. Уравнение неразрывности для такого потока имеет вид:

$$\frac{\partial}{\partial x_1} \left(k_{11} \frac{\partial u}{\partial x_1} \right) + \frac{\partial}{\partial x_2} \left(k_{22} \frac{\partial u}{\partial x_2} \right) = 0 \text{ в } \Omega. \quad (1)$$

Граничные условия задачи таковы:

$$u = \bar{u} \text{ на } \partial\Omega_1; \quad v_n = - \left(k_{11} \frac{\partial u}{\partial x_1} \alpha_{n1} + k_{22} \frac{\partial u}{\partial x_2} \alpha_{n2} \right) = \bar{v}_n \text{ на } \partial\Omega_2, \quad (2)$$

где \bar{v}_n – скорость, направленная по нормали к границе; α_{n1} и α_{n2} – направляющие косинусы нормали с осями Ox_1 и Ox_2 ; $\partial\Omega_1$ – непроницаемая часть границы; $\partial\Omega_2$ – проницаемая часть границы.

Пусть в пористом грунте образуется свободная поверхность фильтрующейся жидкости, называемая обычно поверхность депрессии. Точное положение этой поверхности неизвестно и её определение составляет одну из частей общего решения. Для этой цели используется простое условие: в любой точке свободной поверхности полный потенциальный напор u равен напору H , соответствующему геометрическому возвышению свободной поверхности над плоскостью сравнения (атмосферное давление принимается равным нулю). Отметим, что наличие свободного участка границы области фильтрации делает краевую задачу (1), (2) нелинейной.

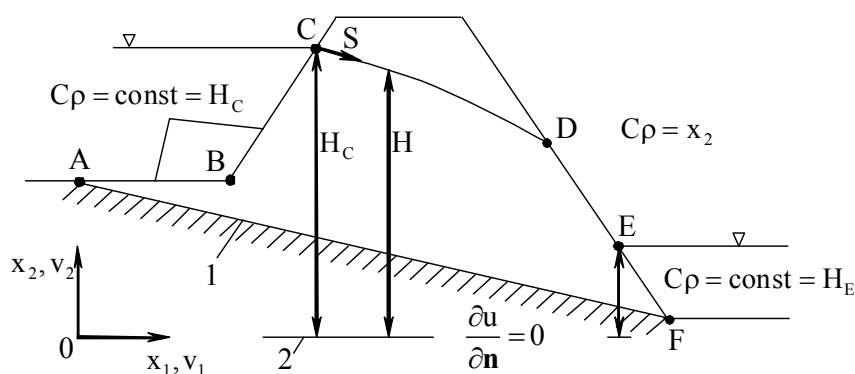


Рис. 1

Рассмотрим общий случай фильтрации, показанный на рис. 1, с четырьмя различного типа границами.

1) Непроницаемая граница (линия AF) – поверхность слоя почвы и твердые породы.

Условие на таких границах есть $\frac{\partial u}{\partial n} = 0$ на $\partial\Omega_2$, следовательно, они являются линиями тока.

2) Границу грунта с жидкостью составляют поверхности пористой плотины, ограничивающие район фильтрации вверх и вниз по потоку (линии ABC и EF). На данные поверхности

действует гидростатическое давление; полный потенциальный напор вдоль них может быть принят постоянным и равным возвышению жидкой поверхности. Эти границы являются эквипотенциальными линиями.

3) Линия фильтрации (или кривая депрессии) CD есть самая верхняя линия тока в области течения. В каждой точке вдоль этой линии давление в порах равно атмосферному и поэтому полный напор равен геометрическому напору, т.е. $u = H = x_2$ в произвольной точке CD. Кроме того, эта кривая является линией тока.

4) Поверхность высачивания – это поверхность, где вода просачивается сквозь грунт в воздух. Так как давление на такой границе постоянно и равно атмосферному, полный напор равен возвышению ($u = x_2$). Эта граница не является ни эквипотенциальной линией, ни линией тока.

Рассмотрим двумерный случай: $x_1 = x$, $x_2 = y$, $x_3 = 0$ и, кроме того, $k_{ij} = 0$ при $i \neq j$, $i, j = 1, 2$, т.е. оси координат являются главным направлением тензора коэффициента фильтрации. Будем решать смешанную краевую задачу, связанную с определением подземной части гидросооружения:

$$-\frac{\partial}{\partial x} \left(k_{11} \frac{\partial u}{\partial x} \right) - \frac{\partial}{\partial y} \left(k_{22} \frac{\partial u}{\partial y} \right) = 0 \text{ в } \Omega, \quad (3)$$

$$u|_{AE} = u_1, \quad u|_{FC} = u_3, \quad (4)$$

$$\frac{\partial u}{\partial N} + \sigma_0 u \Big|_{\text{ABC} \cup \text{A'B'C'}} = \gamma_0(x, y), \quad (5)$$

где $\frac{\partial u}{\partial N} = k_{11} \frac{\partial u}{\partial x} \cos(\mathbf{n}, x) + k_{22} \frac{\partial u}{\partial y} \cos(\mathbf{n}, y)$ – производная по конормали; \mathbf{n} – единичный вектор внешней нормали; u_1, u_3 – постоянные величины. Часть границы области A'B'C' (рис. 2) – неизвестная линия, которую обозначим через $\partial\Omega_*$. В частном случае, когда ABC и A'B'C' – непроницаемые границы, имеем $\sigma_0 = 0$ и $\gamma_0 = 0$.

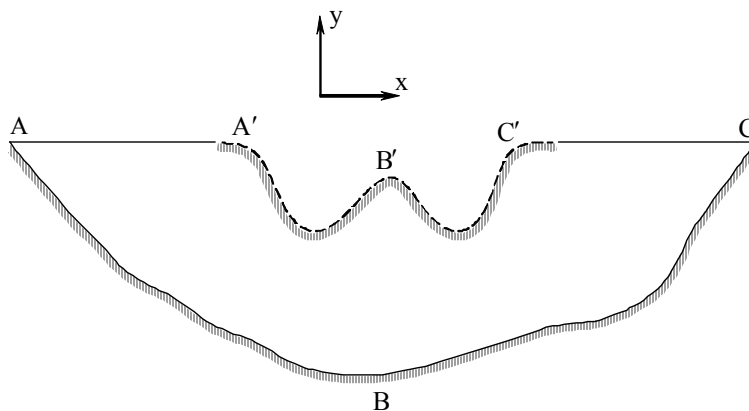


Рис. 2

Предполагаем, что точки A' и C' неизвестны. Так как у нас $\partial\Omega_*$ означает неизвестную границу, то согласно теории на ней задается дополнительное, так называемое «геометрическое условие» (или «кинематическое условие»)

$$u = \xi(s) \text{ на } \partial\Omega_*, \quad (6)$$

где ξ означает заданный напор на подземном контуре гидросооружения.

2. Общие положения теории R-функций

Наиболее общим методом учета априорной информации аналитического и геометрического характера, содержащейся в краевой задаче, является метод R-функций, разработанный школой академика НАН Украины В.Л. Рвачёва. Рассмотрим общие положения теории этого метода [9]. Этот раздел носит вспомогательный характер и призван объяснить используемые в разделе 3 понятия и преобразования.

Сформулируем обратную задачу аналитической геометрии. Пусть в R^2 задан геометрический объект Ω с кусочно-гладкой границей $\partial\Omega$ и требуется построить функцию $\omega(x, y)$, положительную внутри Ω , отрицательную вне Ω и равную нулю на $\partial\Omega$. Уравнение $\omega(x, y) = 0$ будет в неявной форме определять геометрическое место точек, представляющих границу $\partial\Omega$ области Ω .

Определение 1. R-функцией (функцией В.Л. Рвачева), соответствующей разбиению числовой оси на интервалы $(-\infty, 0)$ и $[0, +\infty)$, называется такая функция, знак которой вполне определяется знаками ее аргументов, т.е. функция $z = f(x, y)$ называется R-функцией, если существует такая булева функция F , что $S[z(x, y)] = F[S(x), S(y)]$, где двужначный предикат $S(x) = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0. \end{cases}$

Наиболее часто используется система R-функций \mathfrak{R}_α :

$$x \wedge_\alpha y \equiv \frac{1}{1+\alpha}(x+y-\sqrt{x^2+y^2-2\alpha xy}), \quad x \vee_\alpha y \equiv \frac{1}{1+\alpha}(x+y+\sqrt{x^2+y^2-2\alpha xy}), \quad \bar{x} \equiv -x.$$

Здесь $-1 < \alpha(x, y) \leq 1$, $\alpha(x, y) \equiv \alpha(y, x) \equiv \alpha(-x, y) \equiv \alpha(x, -y)$.

Пусть область Ω может быть сконструирована из более простых (опорных) областей $\Omega_1 = \{\omega_1(x, y) \geq 0\}, \dots, \Omega_m = \{\omega_m(x, y) \geq 0\}$ с помощью теоретико-множественных операций объединения, пересечения и дополнения. Тогда области Ω можно поставить в соответствие предикат

$$\Omega = F(\Omega_1, \Omega_2, \dots, \Omega_m), \quad (7)$$

который принимает значение 1, если $(x, y) \in \bar{\Omega}$, и значение 0, если $(x, y) \notin \bar{\Omega}$.

Переход от предикатной формы задания области (7) к обычному, принятому в аналитической геометрии, уравнению для границы области осуществляется с помощью формальной замены Ω на $\omega(x, y)$, Ω_i на $\omega_i(x, y)$ ($i=1, 2, \dots, m$), а символы $\{\cap, \cup, \neg\}$ – на символы R-операций $\{\wedge_\alpha, \vee_\alpha, \neg\}$ соответственно. Получим в итоге аналитическое выражение $\omega(x, y)$, определяющее в элементарных функциях требуемое уравнение $\omega(x, y) = 0$ границы $\partial\Omega$. При этом для внутренних точек области $\omega(x, y) > 0$, а для внешних $\omega(x, y) < 0$.

Определение 2. Уравнение $\omega(x, y) = 0$ границы $\partial\Omega$ области $\Omega \subset R^2$ называется нормализованным на границе $\partial\Omega$ до n -го порядка, если функция $\omega(x, y)$ удовлетворяет

условиям $\omega|_{\partial\Omega} = 0$, $\frac{\partial\omega}{\partial\mathbf{n}}|_{\partial\Omega} = -1$, $\frac{\partial^l\omega}{\partial\mathbf{n}^l}|_{\partial\Omega} = 0$ ($l=2, 3, \dots, n$), где \mathbf{n} – вектор внешней нормали

к $\partial\Omega$, определенный в ее регулярных точках.

Нормализованное до первого порядка уравнение $\omega(x, y) = 0$ может быть получено из уравнения $\omega_1(x, y) = 0$ следующим образом.

Теорема. Если $\omega(x, y) \in C^m(R^2)$ удовлетворяет условиям $\omega|_{\partial\Omega} = 0$ и $\frac{\partial\omega}{\partial\mathbf{n}}|_{\partial\Omega} > 0$, то

функция $\omega_1 \equiv \frac{\omega}{\sqrt{\omega^2 + |\nabla\omega|^2}} \in C^{m-1}(R^2)$, где $|\nabla\omega| \equiv \sqrt{\left(\frac{\partial\omega}{\partial x}\right)^2 + \left(\frac{\partial\omega}{\partial y}\right)^2}$, удовлетворяет усло-

виям $\omega_1|_{\partial\Omega} = 0$, $\frac{\partial\omega_1}{\partial\mathbf{n}}|_{\partial\Omega} = -1$ во всех регулярных точках границы $\partial\Omega$.

Если $|\nabla\omega_1| \neq 0$ в $\bar{\Omega} = \Omega \cup \partial\Omega$, то для построения нормализованного до первого порядка уравнения можно воспользоваться более простой формулой $\omega \equiv \frac{\omega_1}{|\nabla\omega_1|}$.

Рассмотрим схему применения аппарата теории R-функций для решения краевых задач. Задача расчета физического поля сводится к отысканию в некоторой области Ω решения u уравнения $Au = f$ при следующих условиях на границе $\partial\Omega$ области Ω : $L_i u = \varphi_i$, $i = 1, \dots, m$, где A и L_i – заданные дифференциальные операторы (L_i в частном случае могут представлять собой тождественный оператор); f и φ_i – функции, определенные соответственно внутри области Ω и на участках $\partial\Omega_i$ ее границы. Участки $\partial\Omega_i$ не обязательно все различные и могут совпадать со всей границей $\partial\Omega$. Приведенные в постановках краевых задач функции u , f , φ_i и операторы A и L_i называются аналитическими компонентами краевой задачи; область Ω , ее граница $\partial\Omega$, участки границы $\partial\Omega_i$ – геометрическими компонентами. Существование двух разнородных видов информации (аналитической и геометрической) является серьезным препятствием при отыскании решения краевой задачи. При исследовании и решении краевых задач необходимо не только учитывать вид формул, входящих в постановку задачи, но и приводить геометрическую информацию к аналитическому виду, позволяющему включать ее в разрешающий алгоритм. Осуществить эту процедуру позволяет метод R-функций.

С помощью нормализованного уравнения можно строить пучки функций, нормальная производная которых, либо произвольная линейная комбинация нормальной производной и самой функции на границе области принимает заданные значения. Для этого введем

сначала следующий оператор: $D_1 \equiv \frac{\partial\omega}{\partial x} \frac{\partial}{\partial x} + \frac{\partial\omega}{\partial y} \frac{\partial}{\partial y}$, где $\omega(x, y)$ – нормализованное уравнение

границы области. При этом для произвольной достаточно гладкой функции f на границе области $\partial\Omega$ будет иметь место $D_1 f|_{\partial\Omega} = -\frac{\partial f}{\partial \mathbf{n}}|_{\partial\Omega}$, где \mathbf{n} – вектор внешней нормали к границе. Аналог оператора D_1 , соответствующий участкам $\partial\Omega_i$ границы $\partial\Omega$, будем

обозначать через $D_1^{(i)} \equiv \frac{\partial\omega_i}{\partial x} \frac{\partial}{\partial x} + \frac{\partial\omega_i}{\partial y} \frac{\partial}{\partial y}$, где $\omega_i(x, y)$ – нормализованные уравнения участков

границы $\partial\Omega_i$. Можно доказать, что $D_1\omega = 1 + O(\omega)$, $D_1(\omega\Phi) = (D_1\omega)\Phi + \omega D_1\Phi = \Phi + O(\omega)$, где $\omega(x, y)$ – нормализованное уравнение границы области.

Определение 3. Общей структурой решения краевой задачи называется выражение $u = V(\Phi, \omega, \{\omega_i\}_{i=1}^m, \{\varphi_j\}_{j=1}^m)$, которое при любом выборе неопределенной компоненты Φ точно удовлетворяет всем краевым условиям задачи. Здесь V – оператор, зависящий от геометрии области Ω и участков $\partial\Omega_i$ ее границы, а также операторов краевых условий, но не зависящий от вида оператора A и функции f .

Соответственно, частной структурой решения будем называть выражение вида $u = V_i(\Phi, \omega, \omega_i, \varphi_j)$, при любом выборе неопределенной компоненты точно удовлетворяющее лишь граничному условию на i -м участке границы $\partial\Omega$.

Таким образом, структура решения осуществляет продолжение граничных условий внутрь области.

Задача построения уравнения сложного геометрического объекта является частным случаем более общей проблемы, когда искомая функция φ принимает на различных участках $\partial\Omega_i$ границы области Ω заданные значения φ_i , т.е.

$$\varphi = \varphi_i \text{ на } \partial\Omega_i, \quad i = 1, \dots, m. \quad (8)$$

Для простоты будем считать, что φ_i – элементарные функции, определенные везде в области $\Omega \cup \partial\Omega$. С применением методики, описанной выше, конструируются функции ω_i^0 , равные нулю везде на $\partial\Omega$, за исключением участка $\partial\Omega_i$. Тогда функция

$$\varphi = \left(\sum_{i=1}^m \varphi_i \omega_i^0 \right) \left(\sum_{j=1}^m \omega_j^0 \right)^{-1} \quad (9)$$

удовлетворяет условиям (8) и определена везде в области, за исключением точек, общих для различных участков. Вместо выражения (9) можно также применять формулу

$$\varphi = \left(\sum_{i=1}^m \varphi_i \omega_i^{-1} \right) \left(\sum_{j=1}^m \omega_j^{-1} \right)^{-1}, \quad (10)$$

где $\omega_i = 0$ – уравнение участка $\partial\Omega_i$ границы $\partial\Omega$, причем $\omega_i > 0$ вне $\partial\Omega_i$. При приближении к участку $\partial\Omega_i$ функция $\omega_i \rightarrow 0$ и предельные значения функции φ совпадают со значениями соответствующей функции φ_i .

Оператор “склеивания” граничных значений, определяемый какой-либо из приведенных формул (9), (10), будем обозначать в дальнейшем ЕС ($ЕС\varphi_i = \varphi$).

Практически все приближенные методы решения краевых задач для уравнений в частных производных основаны на сведении бесконечномерной задачи к конечномерной. В методе R-функций это достигается представлением неопределенной компоненты Φ структуры решения в виде суммы $\Phi(x, y) \approx \Phi_n(x, y) = \sum_{k=1}^n c_k \varphi_k(x, y)$, где $\varphi_k(x, y)$ – известные

элементы полной функциональной последовательности, а c_k ($k = 1, 2, \dots, n$) – неизвестные коэффициенты разложения. Неопределенные функции, входящие в структурные формулы, должны выбираться так, чтобы наилучшим образом (в том или ином смысле) удовлетворить основному дифференциальному уравнению задачи. Методы поиска приближений к неопределенным функциям могут быть самыми различными, например, можно воспользоваться вариационными (Ритца, наименьших квадратов и т.д.), проекционными (Галёркина, коллокаций и т.д.), сеточными и другими методами.

3. Построение структуры решения краевой задачи

Будем предполагать, что уравнение (4) эллиптическое невырождающееся, т.е. при всех $t_1, t_2 \in \mathbb{R} \setminus \{0\}$ в Ω выполнено неравенство $k_{11}t_1^2 + k_{22}t_2^2 \geq \mu_0(t_1^2 + t_2^2)$, $\mu_0 > 0$.

3. Построение структуры решения краевой задачи

Будем предполагать, что уравнение (4) эллиптическое невырождающееся, т.е. при всех $t_1, t_2 \in \mathbb{R} \setminus \{0\}$ в Ω выполнено неравенство $k_{11}t_1^2 + k_{22}t_2^2 \geq \mu_0(t_1^2 + t_2^2)$, $\mu_0 > 0$.

Введем следующие обозначения:

$$Au \equiv -\frac{\partial}{\partial x} \left(k_{11} \frac{\partial u}{\partial x} \right) - \frac{\partial}{\partial y} \left(k_{22} \frac{\partial u}{\partial y} \right), \quad Nu \equiv k_{11} \frac{\partial u}{\partial x} \cos(\mathbf{n}, x) + k_{11} \frac{\partial u}{\partial y} \cos(\mathbf{n}, y),$$

$\partial\Omega_* = A'B'C'$ – свободная граница, $\partial\Omega_1 = AA' \cup C'C$, $\partial\Omega_2 = ABC$.

Пусть функция $\omega'(x, y)$ обладает следующими свойствами:

$$1) \omega'(x, y) = 0 \text{ на } AA'; \quad 2) \omega'(x, y) > 0 \text{ в } \Omega \cup A'B'C' \cup C'C \cup ABC; \quad 3) \left. \frac{\partial \omega'}{\partial \mathbf{n}} \right|_{AA'} = -1,$$

а функция $\omega''(x, y)$ – следующими:

$$1) \omega''(x, y) = 0 \text{ на } C'C; \quad 2) \omega''(x, y) > 0 \text{ в } \Omega \cup A'B'C' \cup AA' \cup ABC; \quad 3) \left. \frac{\partial \omega''}{\partial \mathbf{n}} \right|_{C'C} = -1.$$

Такие функции всегда могут быть построены по описанной в п. 2 методике.

Продолжим краевые условия (4) по формуле (10) внутрь области. Функция

$$\varphi(x, y) = \frac{u_3 \omega'(x, y) + u_1 \omega''(x, y)}{\omega'(x, y) + \omega''(x, y)}$$

определена всюду в $\bar{\Omega}$ и $\varphi|_{\partial\Omega_1} = \begin{cases} u_1 & \text{на } AA', \\ u_3 & \text{на } C'C. \end{cases}$ Пусть $\gamma(x, y) = ЕС\gamma_0$, $\sigma(x, y) = ЕС\sigma_0$ – про-

должение функций γ и σ внутрь области. Тогда задача (3) – (6) запишется в виде:

$$Au = 0 \text{ в } \Omega, \quad (11)$$

$$u|_{\partial\Omega_1} = \varphi|_{\partial\Omega_1}, \quad (12)$$

$$Nu + \sigma u|_{\partial\Omega_2 \cup \partial\Omega_*} = \gamma|_{\partial\Omega_2 \cup \partial\Omega_*}, \quad (13)$$

$$u = \xi(x, y) \text{ на } \partial\Omega_*. \quad (14)$$

Если $\partial\Omega_*$ была бы заданной границей, то краевая задача (11) – (13) могла бы быть сведена к нахождению минимума соответствующего квадратичного функционала.

В данном случае мы сразу не можем находить минимум полученного функционала, так как область Ω , по которой ведется интегрирование, и часть её границы $\partial\Omega_*$ неизвестны. Поэтому для полного решения задачи необходимо использовать и условие (14).

Для решения поставленной задачи сформируем итерационный процесс следующим образом. Пусть получено k -е приближение к свободной границе $\partial\Omega_*^{(k)}$. Тогда получим конкретную область $\Omega^{(k)}$ с известной границей.

Определим функцию $u^{(k)}(x, y)$ как решение задачи

$$Au^{(k)} = 0 \text{ в } \Omega^{(k)}, \quad (15)$$

$$u^{(k)}|_{\partial\Omega_1} = \varphi|_{\partial\Omega_1}, \quad (16)$$

$$Nu^{(k)} + \sigma u^{(k)}|_{\partial\Omega_2 \cup \partial\Omega_*^{(k)}} = \gamma|_{\partial\Omega_2 \cup \partial\Omega_*^{(k)}}. \quad (17)$$

Для решения задачи (15) – (17) воспользуемся методом R-функций.

Пусть уравнение $\partial\Omega_*^{(k)}$ представлено в виде $y = h(x)$, причем $y - h(x) > 0$ в

$\Omega^{(k)} \cup \partial\Omega_1 \cup \partial\Omega_2$, тогда $\omega_*^{(k)}(x, y) = \frac{y - h(x)}{\sqrt{1 + (h'(x))^2}}$ обладает следующими свойствами:

$$1) \omega_*^{(k)}(x, y) = 0 \text{ на } \partial\Omega_*^{(k)}; \quad 2) \omega_*^{(k)}(x, y) > 0 \text{ в } \Omega^{(k)} \cup \partial\Omega_1 \cup \partial\Omega_2; \quad 3) \left. \frac{\partial\omega_*^{(k)}}{\partial\mathbf{n}} \right|_{\partial\Omega_*^{(k)}} = -1, \quad \mathbf{n} -$$

внешняя нормаль.

Пусть $\omega_1(x, y)$ и $\omega_2(x, y)$ обладают свойствами:

- 1) $\omega_1(x, y) = 0$ на $\partial\Omega_1$; $\omega_2(x, y) = 0$ на $\partial\Omega_2$;
- 2) $\omega_1(x, y) > 0$ в $\Omega \cup \partial\Omega_2 \cup \partial\Omega_*^{(k)}$; $\omega_2(x, y) > 0$ в $\Omega \cup \partial\Omega_1 \cup \partial\Omega_*^{(k)}$;
- 3) $\left. \frac{\partial\omega_1}{\partial\mathbf{n}} \right|_{\partial\Omega_1} = -1$; $\left. \frac{\partial\omega_2}{\partial\mathbf{n}} \right|_{\partial\Omega_2} = -1$.

Далее строим

$$\tilde{\omega}(x, y) = \omega_2(x, y) \wedge_{\alpha} \omega_*^{(k)}(x, y),$$

$$\bar{\Gamma} = (l_1, l_2) = \frac{1}{\sqrt{\left(k_{11} \frac{\partial\tilde{\omega}}{\partial x}\right)^2 + \left(k_{22} \frac{\partial\tilde{\omega}}{\partial y}\right)^2 + \tilde{\omega}^2}} \left(k_{11} \frac{\partial\tilde{\omega}}{\partial x}, k_{22} \frac{\partial\tilde{\omega}}{\partial y} \right).$$

Вектор $\bar{\Gamma}$ определен всюду в $\Omega^{(k)}$ и на границе совпадает с вектором единичной внутренней конормали. Тогда дифференциальный оператор $D_1^{\bar{\Gamma}}u = \frac{\partial u}{\partial x} l_1 + \frac{\partial u}{\partial y} l_2$ обладает

свойством: $-\sqrt{\left(k_{11} \frac{\partial\tilde{\omega}}{\partial x}\right)^2 + \left(k_{22} \frac{\partial\tilde{\omega}}{\partial y}\right)^2 + \tilde{\omega}^2} D_1^{\bar{\Gamma}}u \Big|_{\partial\Omega_2 \cup \partial\Omega_*^{(k)}} = Nu|_{\partial\Omega_2 \cup \partial\Omega_*^{(k)}}$. Значит, краевое условие (17) можно записать в виде

$$SD_1^{\bar{1}}u^{(k)} - \sigma u^{(k)} \Big|_{\partial\Omega_2 \cup \partial\Omega^{(k)}} = -\gamma \Big|_{\partial\Omega_2 \cup \partial\Omega^{(k)}}, \quad (18)$$

где $S = \sqrt{\left(k_{11} \frac{\partial \tilde{\omega}}{\partial x}\right)^2 + \left(k_{22} \frac{\partial \tilde{\omega}}{\partial y}\right)^2} + \tilde{\omega}^2$. Продолжив краевое условие (16) внутрь области $\Omega^{(k)}$,

получим частичную структуру решения краевой задачи (15) – (17), удовлетворяющую краевому условию (16), в виде

$$u^{(k)} = \varphi + \omega_1 \Phi, \quad (19)$$

где Φ – неопределенная функция.

Будем предполагать, что $(\bar{1}, \nabla \tilde{\omega}) \geq 0$. Используя оператор $D_1^{\bar{1}}$, краевое условие (18) заменим равенством

$$SD_1^{\bar{1}}u^{(k)} - \sigma u^{(k)} = -\gamma + \tilde{\omega} \Psi, \quad (20)$$

где Ψ – неопределенная функция. Неопределенную функцию Φ , входящую в формулу (19), представим в виде $\Phi = \Phi_0 + \tilde{\omega} \Phi_1$ и подставим (19) в (20):

$$SD_1^{\bar{1}}(\varphi + \omega_1 \Phi_0 + \omega_1 \tilde{\omega} \Phi_1) - \sigma(\varphi + \omega_1 \Phi_0 + \omega_1 \tilde{\omega} \Phi_1) = -\gamma + \tilde{\omega} \Psi, \\ SD_1^{\bar{1}}\varphi + SD_1^{\bar{1}}(\omega_1 \Phi_0) + S\omega_1 \Phi_1 D_1^{\bar{1}}\tilde{\omega} + S\tilde{\omega} D_1^{\bar{1}}(\omega_1 \Phi_1) - \sigma\varphi - \sigma\omega_1 \Phi_0 - \sigma\omega_1 \tilde{\omega} \Phi_1 = -\gamma + \tilde{\omega} \Psi. \quad (21)$$

Слагаемые, содержащие $\tilde{\omega}$, могут быть поглощены членом $\tilde{\omega} \Psi$ за счет произвольности функции Ψ . Это дает право переписать (21) в виде

$$SD_1^{\bar{1}}\varphi + SD_1^{\bar{1}}(\omega_1 \Phi_0) + S\omega_1 D_1^{\bar{1}}\tilde{\omega} \Phi_1 - \sigma\varphi - \sigma\omega_1 \Phi_0 = -\gamma + \tilde{\omega} \Psi_1,$$

где Ψ_1 – неопределенная функция. К левой и правой частям последней формулы прибавим функцию $\tilde{\omega}(S + \delta\omega_1)\Phi_1$, где $\delta = 1 - (\bar{1}, \nabla \tilde{\omega}) > 0$:

$$SD_1^{\bar{1}}\varphi + SD_1^{\bar{1}}(\omega_1 \Phi_0) + (S\omega_1 D_1^{\bar{1}}\tilde{\omega} + S\tilde{\omega} + \delta\tilde{\omega}\omega_1)\Phi_1 - \sigma\varphi - \sigma\omega_1 \Phi_0 = -\gamma + \tilde{\omega} \Psi_2,$$

где Ψ_2 – новая неопределенная функция. Находим отсюда

$$\Phi_1 = \frac{-SD_1^{\bar{1}}\varphi - SD_1^{\bar{1}}(\omega_1 \Phi_0) + \sigma\varphi + \sigma\omega_1 \Phi_0 - \gamma + \tilde{\omega} \Psi_2}{\omega_1(SD_1^{\bar{1}}\tilde{\omega} + \delta\tilde{\omega}) + S\tilde{\omega}}.$$

Подставив Φ_1 в $u^{(k)} = \varphi + \omega_1 \Phi_0 + \omega_1 \tilde{\omega} \Phi_1$, получим общую структуру решения краевой задачи (15) – (17):

$$u^{(k)} = \varphi + \omega_1 \Phi_0 + \frac{\omega_1 \tilde{\omega}}{\omega_1(SD_1^{\bar{1}}\tilde{\omega} + \delta\tilde{\omega}) + S\tilde{\omega}} [-SD_1^{\bar{1}}\varphi - SD_1^{\bar{1}}(\omega_1 \Phi_0) + \sigma\varphi + \sigma\omega_1 \Phi_0 - \gamma + \tilde{\omega} \Psi_2]. \quad (22)$$

Неопределенные компоненты Φ_0 и Ψ_2 структуры (22) можно аппроксимировать, например, каким-либо вариационным методом. При этом получим функцию $u_m^{(k)}$ – приближенное решение задачи (15) – (17). Ясно, что условию на свободной границе оно не удовлетворит.

4. Алгоритм уточнения уравнения свободной границы

Для итерационного уточнения уравнения свободной границы построим алгоритм, основанный на использовании точечного метода наименьших квадратов и условия (14).

Уравнение свободной границы $\partial\Omega_*^{(k+1)}$ для следующей $(k+1)$ -й итерации ищем в виде:

$$y^{(k+1)} = \sum_{r=0}^n c_r^{(k+1)} \psi_r(x), \quad (23)$$

где n – натуральное число, $c_r^{(k+1)}$ ($r=0, 1, 2, \dots, n$) – неизвестные коэффициенты, $\psi_0(x), \dots, \psi_n(x)$ – базисные функции (степенные или тригонометрические полиномы, сплайны и т.д.). Пусть $a^{(k)}$ и $c^{(k)}$ – абсциссы точек $A^{(k)}$ и $C^{(k)}$. Положим $d^{(k)} = |c^{(k)} - a^{(k)}|$. Выберем натуральное число N такое, что $N > n$, и зададим систему точек (узлов)

$$x_l = a^{(k)} + \frac{N-1}{N-1} d^{(k)}, \quad l=1, 2, \dots, N. \text{ Обозначим } y_l^{(k+1)} = \sum_{r=0}^n c_r^{(k+1)} \psi_r(x_l).$$

Кроме единичной нормали $\mathbf{n} = (n_1, n_2)$ введем в рассмотрение конормаль $\mathbf{v} = (v_1, v_2)$ согласно правилу: $v_1 = \frac{k_{11}n_1}{S}$, $v_2 = \frac{k_{22}n_2}{S}$, где $S = [(k_{11}, n_1)^2 + (k_{22}, n_2)^2]^{\frac{1}{2}}$, $n_1 = \cos(\mathbf{n}, \mathbf{x})$, $n_2 = \cos(\mathbf{n}, \mathbf{y})$.

Из условий (13) на $\partial\Omega_*^{(k)}$ имеем $S \frac{\partial u^{(k)}(P_1)}{\partial v} = \gamma(P_1) - \sigma u^{(k)}(P_1)$, откуда, приближенно заменяя $u^{(k)}(P_1) \approx u_m^{(k)}(P_1)$, получим $\frac{\partial u^{(k)}(P_1)}{\partial v} \approx \frac{1}{S} [\gamma(P_1) - \sigma u_m^{(k)}(P_1)] = Q(P_1)$, где $u_m^{(k)}(P_1)$ – значение приближенного решения исходной задачи для области $\Omega^{(k)}$ с границей $\partial\Omega_*^{(k)}$ в точке P_1 , точка $P_1 = (x_1, y_1)$ лежит на $\partial\Omega_*^{(k)}$.

Кинематическое условие в точках, лежащих на $\partial\Omega_*^{(k)}$, очевидно, не выполняется. Пусть $\tilde{P}_1 = (\tilde{x}_1, \tilde{y}_1)$ – точка нормали (рис. 3) к $\partial\Omega_*^{(k)}$, проходящей через P_1 и лежащей на $\partial\Omega_*^{(k+1)}$, в которой (14) выполняется приближенно, т.е. $u^{(k)}(\tilde{P}_1) - \xi(\tilde{P}_1) \approx 0$, откуда, разлагая в окрестности точки P_1 в ряд Тейлора и ограничиваясь линейным приближением, имеем

$$\begin{aligned} \xi(\tilde{P}_1) - u^{(k)}(\tilde{P}_1) &= \xi(P_1) - u^{(k)}(P_1) + |\tilde{P}_1 - P_1| \left[\frac{\partial \xi(P_1)}{\partial v} - \frac{\partial u^{(k)}(P_1)}{\partial v} \right] \approx \\ &\approx \xi(P_1) - u_m^{(k)}(P_1) + |\tilde{P}_1 - P_1| \left[\frac{\partial \xi(P_1)}{\partial v} - Q(P_1) \right]. \end{aligned}$$

Из $\Delta P P_1 \tilde{P}_1$ получим

$$\tilde{x}_1 - x_1 = |\tilde{P}_1 - P_1| \cos(v, x), \quad \tilde{y}_1 - y_1 = |\tilde{P}_1 - P_1| \cos(v, y). \quad (24)$$

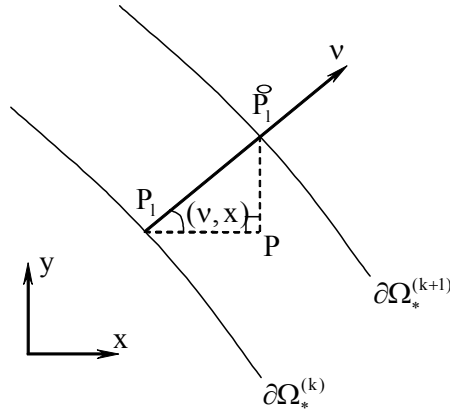


Рис. 3

Из (24) вытекает, что

$$|\tilde{P}_1 - P_1| = \left| \frac{u_m^{(k)}(P_1) - \xi(P_1)}{\frac{\partial \xi(P_1)}{\partial v} - Q(P_1)} \right|. \quad (25)$$

Согласно (25) из (24) имеем соотношение для вычисления координат точек \tilde{P}_1 , лежащих на $\partial\Omega_*^{(k+1)}$:

$$\tilde{x}_1 = x_1 + \cos(v, x) \left| \frac{u_m^{(k)}(P_1) - \xi(P_1)}{\frac{\partial \xi(P_1)}{\partial v} - Q(P_1)} \right|, \quad \tilde{y}_1 = y_1 + \cos(v, y) \left| \frac{u_m^{(k)}(P_1) - \xi(P_1)}{\frac{\partial \xi(P_1)}{\partial v} - Q(P_1)} \right|. \quad (26)$$

Для нахождения коэффициентов в уравнении кривой $\partial\Omega_*^{(k+1)}$ потребуем, чтобы кривая каким-то образом определялась через множество точек $\tilde{P}_l = (\tilde{x}_l, \tilde{y}_l)$, $l=1, 2, \dots, N$. Для этого воспользуемся точечным методом наименьших квадратов: потребуем, чтобы выражение

$$R = \sum_{l=1}^N \left[\tilde{y}_l - \sum_{r=0}^n c_r^{(k+1)} \psi_r(\tilde{x}_l) \right]^2$$

имело минимальное значение, где значения \tilde{x}_l и \tilde{y}_l

берутся из (26). Необходимое условие минимума $\frac{\partial R}{\partial c_r^{(k+1)}} = 0$, $r=0,1,2,\dots,n$, дает систему

из $n+1$ линейных алгебраических уравнений, что позволяет определить $n+1$ параметров $c_0^{(k+1)}, c_1^{(k+1)}, \dots, c_n^{(k+1)}$. После определения всех параметров из (34) кривая $y^{(k+1)}$ будет известной, значит, имеем известную область $\Omega^{(k+1)}$. На следующей $(k+2)$ -й итерации строится приближенное решение $u_m^{(k+2)}$ для новой области $\Omega^{(k+1)}$. Итерация проводится до тех пор, пока “расстояние” между $\partial\Omega_*^{(k)}$ и $\partial\Omega_*^{(k+1)}$ не станет достаточно малым:

$\max_{1 \leq l \leq N} \sqrt{(x_l^{(k+1)} - x_l^{(k)})^2 - (y_l^{(k+1)} - y_l^{(k)})^2} < \varepsilon$, где $\varepsilon > 0$ – наперед заданное достаточно малое число.

Выводы

Для краевой задачи теории фильтрации со свободной границей в работе впервые получена структура решения и предложен алгоритм уточнения уравнения свободной границы. Предложенные методы можно использовать при проектировании различных гидротехнических сооружений, в частности, для предотвращения последствий паводков. Этим и определяется *научная новизна и практическая значимость* полученных результатов.

Список литературы: 1. Бурак Я.Й., Чапля С.Я., Чернуха О.Ю. Континуально-термодинамічні моделі механіки твердих розчинів. К.: Наук. думка, 2006. 272 с. 2. Бомба А.Я., Присяжнюк І.М. Задачі типу “фільтрація-конвекція” у трьохзв’язних областях з умовами усереднення // Мат. методи та фіз.-мех. поля, 2005. Т. 48, № 2. С. 53 – 58. 3. Ляшко И.И., Сергиенко Н.В., Мистецкий Г.Е., Скопецкий В.В. Вопросы автоматизации решения задач фильтрации на ЭВМ. К.: Наук. думка, 1977. 288 с. 4. Ляшко И.И., Великоиваненко Н.М. Численно-аналитическое решение краевых задач теории фильтрации. К.: Наук. думка, 1973. 264 с. 5. Шаманский В.Е. Численное решение задач фильтрации грунтовых вод на ЭЦВМ. К.: Наук. думка, 1969. 376 с. 6. Вабищевич П.Н. Численные методы решения задач со свободной границей. М., Изд-во Моск. ун-та, 1987. 164 с. 7. Сидоров М.В., Стороженко А.В. Математическое и компьютерное моделирование некоторых фильтрационных течений // Радиоэлектроника и информатика. 2004. №4. С. 58 – 61. 8. Блишун А.П., Сидоров М.В., Яловега И.Г. Математическое моделирование и численный анализ фильтрационных течений под гидротехническими сооружениями с помощью метода R-функций // Радиоэлектроника и информатика. 2010. № 2. С. 24-30. 9. Рвачев В.Л. Теория R-функций и некоторые её приложения. К.: Наук. думка, 1982. 552 с.

Поступила в редколлегию 16.02.2010

Блишун Александр Павлович, ст. гр. ПМм-09-1 факультета прикладной математики и менеджмента ХНУРЭ. Научные интересы: математическое моделирование, численные методы математической физики. Увлечения и хобби: покер, футбол. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (057) 7021436.

Сидоров Максим Викторович, канд. физ.-мат. наук, доцент кафедры прикладной математики ХНУРЭ. Научные интересы: математическое моделирование, численные методы, математическая физика, теория R-функций и её приложения, стохастический анализ и его приложения. Увлечения и хобби: всемирная история, история искусств. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (057) 7021436.

Яловега Ирина Георгиевна, канд. техн. наук, ст. преп. кафедры прикладной математики ХНУРЭ. Научные интересы: математическое моделирование, анализ динамических систем, стохастический анализ и его приложения. Увлечения и хобби: оригами. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (057) 7021436.

*И.М. ПРОХОРЕЦ, С.И. ПРОХОРЕЦ, А.И. СКРЫПНИК, В.П. ЛУКЬЯНОВА,
М.А. ХАЖМУРАДОВ*

ИССЛЕДОВАНИЕ ДЕТЕКТОРА ТЕПЛОВЫХ НЕЙТРОНОВ ДЛЯ МНОГОШАРОВОГО СПЕКТРОМЕТРА

Рассматривается возможность регистрации тепловых нейтронов на фоне гамма-квантов ^{239}Pu - Be источника. Экспериментально показывается, что путем дискриминации сигналов от гамма-квантов низких энергий возможно их отделение от сигналов, вызванных нейтронами в сцинтилляторе $\text{ZnS}(\text{Ag})$. Регистрация сигнала с $\text{ZnS}(\text{Ag})$ осуществлялась фотоэлектрическим умножителем. Амплитудное разрешение системы регистрации, в которой использовались предварительный усилитель и основной усилитель с R-C-формирователя, достигало 15%. Показывается изменение спектра нейтронов в эксперименте. Для этого использовалась программа моделирования прохождения нейтронов через полиэтиленовый замедлитель на основе кода MCNP.

1. Введение

Дозиметрические измерения вокруг источников нейтронного излучения связаны с определенными трудностями, так как в прямой видимости этих объектов или за их защитой наблюдается широкий энергетический спектр нейтронов от 10^{-2} до 10^7 эВ [1]. В технике радиационного контроля мощность дозы нейтронов вместо прямого измерения часто определяют по потоку нейтронов, используя сведения об энергетическом спектре нейтронов. Тепловые ($\sim 0,025$ эВ) и промежуточные нейтроны измеряют с помощью реакций (n, α) на элементах ^{10}B и ^6Li . Измерение быстрых нейтронов с энергией от нескольких десятых до нескольких десятков МэВ основано на регистрации протонов отдачи, образующихся при упругом рассеянии быстрых нейтронов в водосодержащих материалах. Измерение потока нейтронов, не зависящее от энергии, в диапазоне от 10^{-2} до 10^9 эВ осуществляется с помощью многошарового метода [2, 3].

Спектрометр нейтронов, работающий по этому принципу, состоит из датчика тепловых нейтронов, расположенного в центре шаров различного диаметра, каждый из которых выполняет роль замедлителя нейтронов. Чувствительность каждого шара имеет максимум при определенной энергии нейтронов, положение которого зависит от диаметра шара. Информация о спектре нейтронов содержится в измерениях, проведенных с использованием шаров различного диаметра. Получение такой информации является непростой задачей, так как она требует решения, основанного на некорректных методах [4, 5].

Чувствительность многошарового спектрометра зависит от диаметра шара и датчика тепловых нейтронов. Она характеризуется функциями отклика, для получения которых используются экспериментальные и расчетные методы.

Экспериментальный метод определения функций отклика использовался в работе Боннера с сотрудниками [2]. Для этих целей понадобились моноэнергетические нейтроны в диапазоне энергий от 50 кэВ до 15 МэВ и нейтроны с тепловыми энергиями, а в качестве детектора замедленных в полиэтиленовых сферах нейтронов – сцинтиллятор $\text{Li}^6\text{I}(\text{Eu})$. Сцинтилляции, возникающие в результате экзотермической реакции $\text{Li}^6(n, \alpha)\text{H}^3$ ($Q=4,8$ МэВ), регистрировались фотоумножителем.

Такой метод получения функций отклика обладает существенными недостатками, обусловленными трудностями получения и регистрации нейтронов с энергиями меньше нескольких килоэлектронвольт и больше 20 МэВ. По этой причине для получения функций отклика шарового спектрометра начали применять математическое моделирование процессов прохождения нейтронов через замедляющие шары и регистрации их детектором. Распространению этого метода способствовало увеличение быстродействия компьютеров, наличие проверенных библиотек сечений процессов взаимодействия нейтронов с различными химическими элементами и распространением появившихся на их основе программных кодов типа ANISN [3] и MCNP [6, 7]. Используя этот метод, были определе-

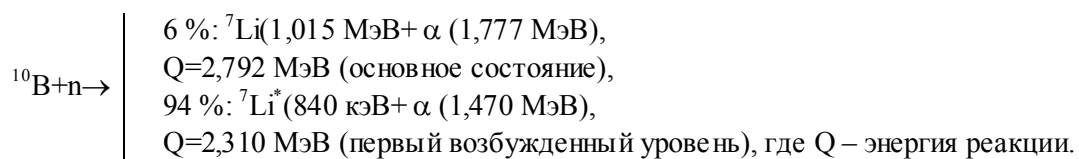
ны функции отклика для шаровых спектрометров с детекторами тепловых нейтронов на основе сцинтилляторов ${}^6\text{LiI}$, сферических газоразрядных пропорциональных счетчиков [8] и активационных фольг из индия и золота [3]. Результаты расчетов с использованием, в основном, кода ANISN сравнивались с экспериментальными данными о функциях отклика, полученными за последние годы на моноэнергетических и тепловых нейтронах и нейтронах с изотопных источников. Было получено хорошее согласие экспериментальных и расчетных данных при высоких энергиях нейтронов, но обнаружено расхождение при тепловых энергиях.

Регистрация нейтронов в различных средах человеческой деятельности сцинтиллятором ZnS , обогащенным бором или другими элементами с большими сечениями взаимодействия медленных нейтронов, используется продолжительное время [1, 9, 10, 11]. Такая «продолжительность жизни» метода регистрации нейтронов, прежде всего, обусловлена большим сечением взаимодействия нейтронов теплового диапазона энергий с изотопом B^{10} , который в природной смеси химического элемента составляет 19,8% [12]. В работе [1] в примечании переводчика с немецкого упоминается о преимуществе сцинтиллятора $\text{ZnS}(\text{Ag})$ по сравнению с другими сцинтиллирующими материалами при регистрации нейтронов при наличии гамма-фона, а З.Я. Соколова и В.Б. Черняев [9] провели исследования дискриминации сигналов при регистрации нейтронов и гамма-излучения и показали, что эффективность регистрации нейтронов и чувствительность детектора зависит от уровня дискриминации регистрируемых сигналов, который можно выбирать из условий допустимого гамма-фона в эксперименте. В работе [11] было показано, что новые сцинтилляторные материалы $\text{ZnS}:\text{Ag}/{}^6\text{LiF}$, $\text{ZnS}:\text{Ag}/{}^{10}\text{B}_2\text{O}_3$ обеспечивают лучшую эффективность регистрации тепловых нейтронов по сравнению с коммерческими сцинтилляторами BC-704 фирмы Bicron. В.Д. Рытиков с сотрудниками [10] показали, что тепловые нейтроны от источника Pu-Be и гамма-кванты от источника ${}^{241}\text{Am}$ с энергией 26,3 и 59,5 кэВ регистрируются фотодиодом типа Si-PIN-PD S-3590 при поглощении сцинтиллятором типа $\text{ZnS}(\text{Ag})+{}^{10}\text{B}_2\text{O}_3$.

Учитывая многолетний интерес к нейтронной тематике, приведенные результаты исследований сцинтиллятора $\text{ZnS}(\text{Ag})$ с конвертером ${}^{10}\text{B}$, представленные в этой работе, являются актуальными и полезными для развития дозиметрии нейтронов на основе многошарового спектрометра.

2. Методика исследований и результаты

Детектирование тепловых нейтронов сцинтиллятором $\text{ZnS}(\text{Ag})$ с добавкой ${}^{10}\text{B}$ осуществляется путем реакции ${}^{10}\text{B}(n, \alpha){}^7\text{Li}$, в результате которой образуются заряженные частицы по двум каналам распада составного ядра с вероятностью 94 и 6 %:



Из зависимости приведенного процесса от энергии нейтронов (рис. 1) следует, что поперечное сечение поглощения тепловых нейтронов (энергия 0,0259 эВ) составляет 3840 барн, оно уменьшается с увеличением энергии нейтронов по закону $1/\vartheta$, где ϑ – скорость нейтрона. После поглощения нейтрона в 94% реакции образуется ион ${}^7\text{Li}$ в первом возбужденном состоянии, которое в результате испускания гамма-кванта с энергией 480 кэВ возвращается в основное состояние за время около 10^{-13} секунд [13].

Детектор тепловых нейтронов представляет собой цилиндр из оргстекла диаметром 10 мм и высотой 40 мм, покрытый слоем $\text{ZnS}(\text{Ag})+{}^{10}\text{B}$ толщиной 100 мкм, за исключением основания цилиндра, соединенного с фотоумножителем ФЭУ-87 диаметром 40 мм путем оптического контакта. Детектор, за исключением поверхности, примыкающей к ФЭУ, упакован в алюминиевый корпус. Фотоумножитель с детектором размещаются в светонепроницаемом корпусе. В этом же корпусе находятся делитель ФЭУ и предварительный усилитель сигнала (рис. 2). Для подачи напряжения на усилитель и съема с него сигнала используется один и тот же высокочастотный кабель.

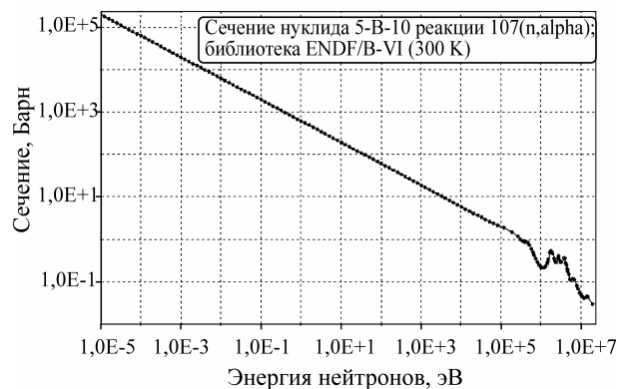


Рис. 1. Сечение взаимодействия нейтронов

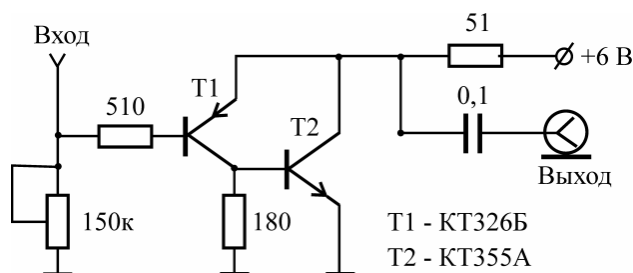


Рис. 2. Схема предварительного усилителя детектора

В экспериментальном варианте шарового спектрометра используются полиэтиленовые шары диаметром 90, 120, 152, 200 и 245 мм. Для регистрации только тепловых нейтронов спектрометр используется без замедляющих шаров световодом, покрытым слоем $ZnS(Ag)+^{10}B$. Для использования замедляющих сфер диаметром больше 245 мм в конструкцию спектрометра легко добавляется более длинный световод между фотоумножителем и детектором тепловых нейтронов.

Такая конструкция спектрометра не обладает сферической симметрией, поэтому для детального моделирования влияния геометрических параметров детектора тепловых нейтронов на функцию отклика необходимо использовать современные программные коды, например, MCNP [6, 7].

Экспериментальная установка для исследования возможности разделения сигналов, вызванных нейтронами и фотонами от $Pu-Be$ источника, была создана на спектрометре без замедляющего шара. Сигнал с фотоумножителя усиливался предварительным усилителем, расположенным в корпусе ФЭУ, дополнительно усиливался и формировался спектрометрическим усилителем типа 1101 фирмы Polon и подавался на амплитудный анализатор, подключенный к персональному компьютеру.

В эксперименте в качестве излучателя использовался $^{239}Pu-Be$ источник нейтронов. В этом источнике нейтроны образуются в результате экзотермической реакции $\alpha + {}^9_4Be \rightarrow n + {}^{12}_6C + 5,71 \text{ МэВ}$, которую инициируют α -частицы, вылетающие при распаде ядра ^{239}Pu .

Активность такого источника составляла $1,13 \cdot 10^5$ нейтронов в секунду в телесный угол 4π , а его спектр представлен на рис. 3. Спектр нейтронов жесткий и имеет два максимума, зависящих от возбужденного состояния, в котором остается ядро ^{12}C [14, 15].

Для проведения эксперимента между детектором и $^{239}Pu-Be$ источником располагался замедлитель нейтронов в виде полиэтиленовой пластины толщиной 5 или 10 см. Спектры нейтронов в диапазоне до 1 МэВ, полученные в результате моделирования по программе на основе кода MCNP-4C [6] прохождения нейтронов спектра источника $^{239}Pu-Be$ через пластины полиэтилена толщиной 5 и 10 см, показаны на рис. 4.

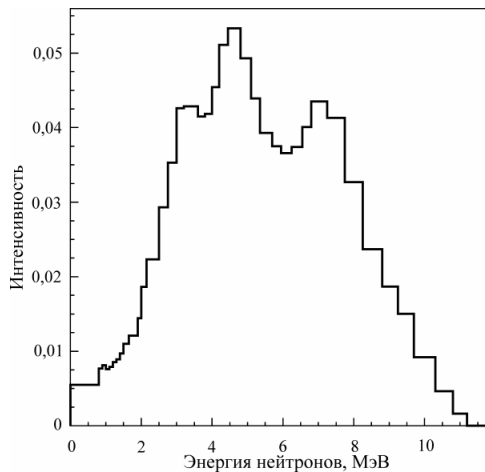


Рис. 3. Спектр ^{239}Pu -Be источника нейтронов

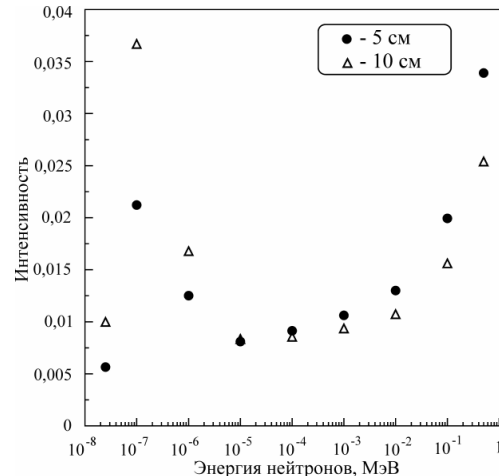


Рис. 4. Спектр нейтронов после прохождения полиэтиленовой пластины толщиной 5 и 10 см

Кроме нейтронов, источник ^{239}Pu -Be испускает гамма-кванты, которые также регистрируются детектором тепловых нейтронов на основе $\text{ZnS}(\text{Ag})$ с добавкой ^{10}B . Фотоны с энергией 4,44 МэВ возникают в результате перехода возбужденного состояния ядра $^{12}\text{C}^*$ в основное. После испускания α -частицы ядром ^{239}Pu возникает ядро ^{235}U , переходы которого в основное состояние дают гамма-кванты различных энергий: 12,9; 51,6; 38,6 кэВ с вероятностью на один нейтрон 2096, 1066 и 404 соответственно [14].

Выводы

Результаты измерения амплитудных спектров для определения рабочего напряжения фотоумножителя в целях выбора оптимального режима, при котором дискриминируются сигналы от низкоэнергетических фотонов источника излучения и шумы фотоумножителя, представлены на рис. 5 и 6,а.

Результаты измерения амплитудного спектра нейтронов и фотонов источника ^{239}Pu -Be, измеренные с использованием сцинтиллятора на основе $\text{ZnS}(\text{Ag})$ с добавкой ^{10}B , представлены на рис. 6,б.

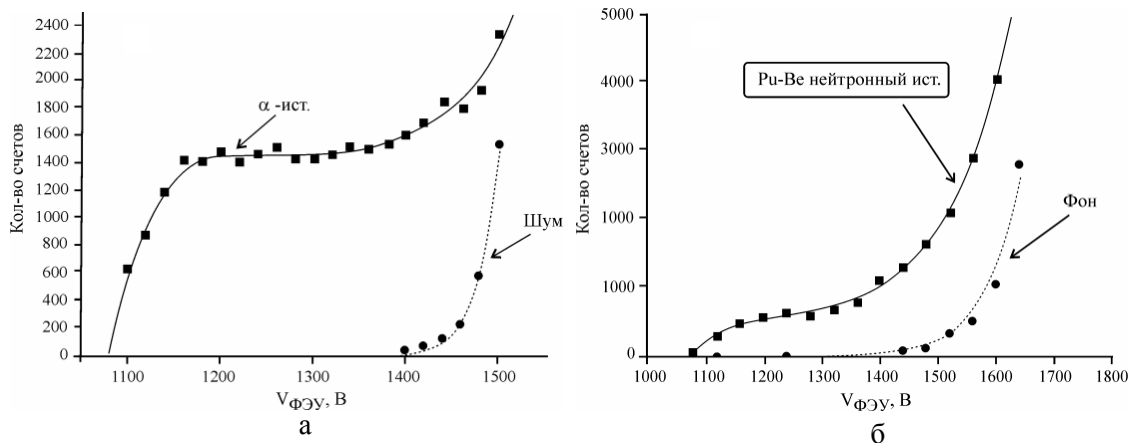


Рис. 5. Спектры сигналов: а – от α -источника и шума ФЭУ; б – от Pu -Be источника нейтронов и фона

Анализ полученных результатов позволяет сделать вывод о возможности разделения сигналов рассмотренным выше спектрометром от гамма-квантов и тепловых нейтронов и использования сцинтиллятора на основе $\text{ZnS}(\text{Ag})+^{10}\text{B}$ совместно с фотоумножителем в качестве детектора тепловых нейтронов в многошаровом спектрометре.

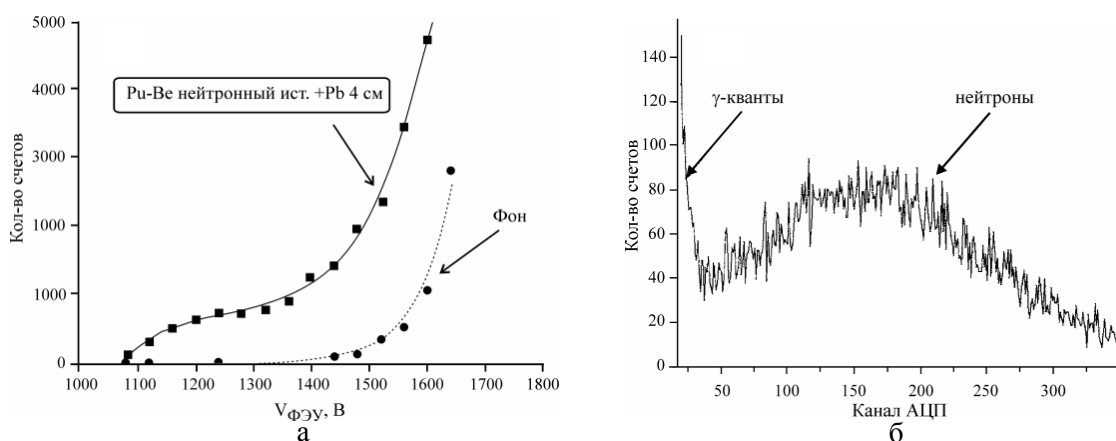


Рис. 6. Результаты измерения: а – спектра Pu-Be нейтронного источника, закрытого свинцом толщиной 4 см; б – амплитудного спектра нейтронов и фотонов Pu-Be источника при $V_{ФЭУ} = 1340$ В

Для продолжения исследований по многосферному спектрометру необходимо промоделировать его функции отклика на нейтроны различного диапазона энергий и рассмотреть возможность использования методики наведенной активности взамен прямой регистрации нейтронов.

Список литературы: 1. Франк М., Штольц В. Твердотельная дозиметрия ионизирующего излучения / Под ред. И.Б. Керим-Маркуса. М.: Атомиздат, 1973. 247с. 2. Bramblett R.L., Eving R.J., Bonner T.W. A new type of neutron spectrometer // Nucl. Instrum. Meth. 1960. Vol.9. P.1-12. 3. Thomas D.J., Alevra A.V. Bonner sphere spectrometers – a critical review // Nucl. Instrum. Meth. 2002. A476. P.12-20. 4. Тихонов А.Н., Арсенин В.Я. Методы решения некорректных задач. М.: Наука, 1974. 142с. 5. Reginatto M. Bayesian approach for quantifying the uncertainty of neutron doses derived from spectrometric measurements // Radiat. Prot. Dosim. 2006. Vol.121, No.1. P.64-69. 6. MCNP – A General Monte Carlo N-Particle Transport Code. Version 4C / Ed. J.F. Briesmeister. 2000. 7. MCNPX 2.4.0. RSICC Computer Code Collection. Monte Carlo N-Particle Transport Code System for Multiparticle and High Energy Applications. CCC-715. 2002. 8. Mares V., Schraube H. Improved response matrices of Bonner sphere spectrometers with ^6LiI scintillation detector and ^3He proportional counter between 15 and 100 MeV // Nucl. Instrum. Meth. 1995. A366. P.203-206. 9. Соколова З.Я., Черняев В.Б. Параметры скintилляторов на основе ZnS(Ag) для регистрации нейтронов // АЭ. 1970. Т.78. Вып.2. С.162-164. 10. Ryzhikov V., Chernikov V., Nagornaya L. et al. Oxide and Semiconductor Scintillators in Scintillometric Detectors for detection of Neutrons. <http://www.ndt.net/article>. 11. Kojona T., Katagiri M., Tsutsui N. et al. Neutron scintillators with high detection efficiency // Nucl. Instrum. Meth. 2004. A259. P.325-328. 12. Физические величины. Справочник / Под ред. И.С. Григорьева, Е.М. Мейлихова. М.: Энергоатомиздат, 1991. 1232с. 13. McGregor D.S., Vernon S.M., Gersch H.K. et al. Self-Biased Boron-10 Coated High-Purity Epitaxial GaAs Thermal Neutron Detectors // IEEE Trans. Nucl. Sci. 2000. NS-47. P.1364-1370. 14. Sagatova-Perdochova A., Dubecky F., Zatko B. et al. Detector of fast neutron based on semi-insulating GaAs with neutron converter layers // Nucl. Instrum. Meth. 2007. A576. P.56-59. 15. http://www.ornl.gov/sci/scal/papers/RPSD2002_Origen_sources.pdf.

Поступила в редколлегию 13.03.2010

Прохорец Иван Михайлович, канд. физ.-мат. наук, старший научный сотрудник Национального Научного Центра Харьковский Физико-технический институт. Научные интересы: математическое моделирование физических процессов и систем. Адрес: Украина, 61108, Харьков, ул. Академическая, 1, тел. (057)335-68-46. e-mail: khazhm@kipt.kharkov.ua

Прохорец Светлана Ивановна, канд. техн. наук, научный сотрудник Национального Научного Центра Харьковский Физико-технический институт. Научные интересы: математическое моделирование, программирование. Адрес: Украина, 61108, Харьков, ул. Академическая, 1, тел. (057)335-68-46. e-mail: khazhm@kipt.kharkov.ua

Скрыпник Анна Ивановна, студентка Харьковского университета. Научные интересы: математическое моделирование. Адрес: Украина, 61108, Харьков, ул. Академическая, 1, тел. (057)335-68-46. e-mail: khazhm@kipt.kharkov.ua

Лукьянова Валентина Петровна, ведущий инженер-программист Национального Научного Центра Харьковский Физико-технический институт. Научные интересы: математическое моделирование, программирование. Адрес: Украина, 61108, Харьков, ул. Академическая, 1, тел. (057)335-68-46. e-mail: khazhm@kipt.kharkov.ua

Хажмурад Манап Ахмадович, д-р техн. наук, профессор, начальник отдела Национального Научного Центра Харьковский Физико-технический институт. Научные интересы: математическое моделирование физических процессов и систем, автоматизация проектирования, программирование. Адрес: Украина, 61108, Харьков, ул. Академическая, 1, тел. (057)335-68-46, e-mail: khazhm@kipt.kharkov.ua

DRTLQ-МОДЕЛЬ ДЛЯ ФУНКЦИОНАЛЬНОЙ ВЕРИФИКАЦИИ ЦИФРОВЫХ СИСТЕМ НА ОСНОВЕ ЛИНЕЙНОЙ ТЕМПОРАЛЬНОЙ ЛОГИКИ

Предлагается аналитическая модель верификации, которая характеризуется использованием динамических регистровых очередей для анализа темпоральных ассерций в процессе моделирования тестов для цифровых систем на кристаллах, что обеспечивает высокое быстродействие моделирования и заданную глубину диагностирования ошибок кода.

1. Введение

Актуальность исследования. Специализированные и стандартизированные системы, согласно топ-десятке от Gartner Research Group, есть одно из главных направлений развития рынка электронных технологий. Оно определяется существенными инвестициями со стороны ведущих компаний планеты, формирующих индекс NASDAQ, включая Intel, Xilinx, Apple, которые предлагают интересные решения в виде планшетов, телекоммуникационных и DSP изделий. Данные компании ориентируются на дальнейшее развитие сегмента рынка, основанного на предоставлении новых сервисов со стороны компактных и энерго-сберегающих цифровых систем на кристаллах с годовым выпуском более 2,5 миллиардов.

Вентильная мощность силиконового кристалла, составляющая до 1 млрд транзисторов, заставляет производителей искать новые эффективные пути системного и RTL- проектирования, тестовой и временной верификации, а также синтеза функциональных модулей на основе языков System Verilog и Verilog с использованием IEEE стандартов тестопригодного проектирования и новейших ESL Design технологий. Что касается проектирования цифровых проектов, то здесь следует выделить наиболее сложный и дорогостоящий этап современного цикла проектирования – функциональная верификация, как процесс поиска, обнаружения и устранения ошибок системной модели относительно спецификации, который составляет до 70% общего времени создания проекта. Для достижения качества проекта, удовлетворяющего требованиям рынка электронных технологий, ведущие компании координируют свои действия по созданию и внедрению современных технологий тестирования и верификации, которые формируют инфраструктуры тестопригодного проектирования моделей системного регистрового и вентильного уровней описания проекта. Такие действия приводят к повышению эффективности проектирования под эгидой специализации и стандартизации решений, определяемых совместным использованием трех взаимно противоречивых параметров: качество, быстродействие, аппаратурные затраты.

Таким образом, исследование направлено как на создание новых моделей и методов верификации, так и на эффективное устранение ошибок, связанных с: 1) ошибками, допускаемыми инженерами в системной модели, тестах и спецификации в процессе проектирования; 2) несовершенством средств диагностирования в системах автоматизации, затрудняющих локализацию и устранение причины возникновения ошибки; 3) недостаточной производительностью и точностью программных систем автоматизации, качество которых растет существенно медленнее увеличения сложности обрабатываемых моделей. Комплексное решение проблемы верификации системных моделей позволит в значительной степени снизить затраты на проектирование цифровых систем на кристаллах. Согласно исследованиям ведущих мировых компаний в области EDA (Cadence Design Systems, Synopsys Inc., Mentor Graphics Corporation, Magma, IBM, Intel, Sun Microsystems, Cisco Systems Inc., Atrenta, Aldec Inc.) усилия ученых должны быть сосредоточены на создании эффективных методов верификации, способных: 1) в десятки раз снизить вероятность возникновения ошибок за счет уменьшения участия человека в процессе проектирования; 2) обеспечить обнаружение и диагностирование допущенных неточностей на ранней стадии проектирования в целях сокращения времени и стоимости устранения несоответствий спецификации; 3) на порядок повысить производительность и надежность систем верифи-

кации за счет повышения уровня абстракции, как самих моделей, так и тестовых воздействий.

Цель исследования – разработка моделей и методов функциональной верификации цифровых систем на кристаллах на основе использования темпоральных ассерций для тестового диагностирования ошибок в процессе программно-аппаратного моделирования, обеспечивающего существенное повышение качества цифрового изделия, а также уменьшение временных и материальных затрат проектирования.

Для достижения цели необходимо решить следующие *задачи*: 1. Разработать математическую аналитическую модель верификации цифровых систем на основе использования динамических регистровых очередей для анализа ассерций линейной темпоральной логики. 2. Спроектировать эффективные структуры данных и разработать модель процесса верификации, осуществляющих подготовительную обработку языковых описаний ассерций и их проверку в рамках цикла имитационного моделирования системы на кристалле в HDL-симуляторе.

Объект исследования – процесс проектирования и верификации цифровых систем на кристаллах с помощью языков описания аппаратуры высокого уровня и HDL-симулятора.

Предмет исследования – системные модели и методы проектирования, верификации цифровых систем на кристаллах на основе использования линейной темпоральной логики и языков описания ассерций

Методы исследования – булева алгебра, линейная темпоральная логика, теория множеств, теория графов, теория цифровых автоматов – для построения математической модели верификации; объектно-ориентированный анализ, теория алгоритмов, методы проектирования программных систем, теория формальных языков, методы динамического моделирования цифровых систем – для построения структур данных и разработки программной системы верификации; экспериментальные исследования производительности верификации моделей цифровых систем – для достижения быстродействия предложенных методов; методы логического синтеза, методы верификации пересечений тактовых доменов, методы анализа функционального покрытия, методы генерации ограниченных псевдослучайных тестовых воздействий – для выработки маршрутов практического применения разработанной системы.

Источники: алгоритмы перехода от традиционных моделей конечных автоматов к структурам Крипке и алгоритмы обратного перехода [1, 2], основы темпоральной логики [3], автоматы Buchi [4-7], алгоритмы и модели минимизация автоматов [8-10], темпоральные ограничения при имитационном моделировании системы (симуляции) [11], методы формальной верификации [12, 13].

2. Модель динамических регистровых очередей

2.1 Основные элементы модели динамических регистровых очередей. Типичная математическая база анализа ассерций во время симуляции, состоящая в трансформации операторов LTL-логики к детерминированным конечным автоматам:

$$A = \{\Sigma, Q, q_0, F, \delta\},$$

(где Σ – алфавит системы, Q – множество всех состояний автомата, $q_0 \in Q$ – единственное начальное состояние, $F \subseteq Q$ – множество конечных состояний, $\delta: Q \times \Sigma$ – функция переходов), лишь частично способна удовлетворить основные требования к верификации на основе ассерций. Основной проблемой классической модели является экспоненциальный рост количества состояний в зависимости от структурной сложности реализуемой темпоральной формулы. Кроме того, накладывается существенное ограничение на языки описания ассерций в виде простого подмножества, поскольку классическая модель неспособна осуществить проверку ряда темпоральных конструкций с приемлемой для практического применения вычислительной сложностью. Поддержка введенного в [14] режима глобального времени практически невозможна на основе модели конечных автоматов.

Очевидно, для обеспечения существенно лучшей производительности анализа ассерций и поддержки темпоральных операторов, определяемых для бесконечных вычислительных путей, необходима принципиально иная модель. Такими свойствами обладает предлагаемая в данном исследовании модель динамических регистровых очередей (DRTLQ –

Dynamic Register-Transfer-Level Queues). Модель ориентирована на высокопроизводительный анализ элементов линейной темпоральной логики и эффективность обнаружения и локализации нарушений. Ключевые понятия, вводимые в модели, представляют собой обобщенную надстройку над общеизвестными терминами LTL и языков описания ассерций.

Модель DRTLQ можно логически разделить на четыре взаимодействующих уровня, соответствующих трем уровням в языках описания ассерций, определенным в [14] (рис. 1).



Рис. 1. Уровни модели DRTLQ в контексте уровней семантики ассерций

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

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

Ассерционный монитор M в модели DRTLQ – вершина темпорального утверждения в ассерционном процессе, осуществляющая верификацию некоторого одного темпорального свойства и управление обработкой результата этой верификации в рамках родительского тест-бенча. Главной задачей ассерционного монитора является обеспечение задаваемой тест-бенчем реакции на результат верификации темпорального утверждения.

В простейших случаях такой реакцией может быть некоторое текстовое сообщение в консоли системы моделирования, визуальная демонстрация результата анализа ассерции на временных диаграммах и других графических инструментов отладки. Возможны и другие виды реакции на результат верификации утверждения, зависящие от конкретного используемого верификационного языка. В обычном режиме интерпретации уровень последовательностных функций непосредственно взаимодействует с монитором.

Атомарным объектом анализа ассерций в модели DRTLQ является верификационная переменная $b \in B$ – любой объект верифицируемой системы, текущее значение или состояние которого используется как операнд темпоральных соотношений в рассматриваемом ассерционном процессе и тем или иным преобразованием может быть приведено к булевому типу $\{0, 1\}$.

В качестве верификационной переменной может выступать как элементарный объект верифицируемой модели – константа (параметр), скалярный или векторный сигнал, переменная встроенного типа, так и сложные объекты, такие как выражения, функции, объекты классов.

Модель DRTLQ полностью абстрагируется от особенностей конкретного типа верифицируемых данных, языка описания или среды проектирования. Обеспечение преобразования текущего значения верификационной переменной к булевому типу, а также уведомление об изменении текущего значения относятся к обязанностям системы моделирования, которая взаимодействует с верификационной системой. Такое отстранение от деталей верифицируемой модели имеет ряд немаловажных преимуществ:

– абстрагирование от типов данных способствует ускорению анализа ассерций, поскольку только система моделирования, непосредственно задающая формат и способы преобразования данных, способна произвести необходимое приведение с оптимальной производительностью;

– значительное упрощение и ускорение анализа ассерций достигается за счет отказа от рассмотрения небулевых значений, таких как X и Z – любое верификационное утверждение, требующее учета 4-значности, может быть сведено к выражению с булевым результатом;

– абстрагирование от типов данных делает модель DRTLQ универсальной относительно предметной области данных – этот факт эффективно применяется при верификации высокоуровневых моделей систем на кристалле, в частности, на основе технологии SystemC, где в качестве переменной может быть использовано состояние объекта-транзактора.

Также верификационные переменные могут представлять другие элементы языков описания ассерций, не имеющие эквивалента в языках моделирования верифицируемых систем, в частности:

– абстрактное понятие перманентной истины и лжи (булевы константы, используемые для реализации семантики более сложных операторов);

– признак окончания анализа заданной последовательности событий (понятие endpoint в языке PSL, встроенные функции ended/matched языков SystemVerilog и OVA);

– локальные переменные ассерционного процесса (SystemVerilog).

Вычислительные процедуры, осуществляющие анализ темпоральных операторов в ассерционном процессе, выполняются при изменении значений верификационных переменных. Тактовой верификационной переменной называется верификационная переменная, момент изменения значения которой инициирует обработку группы элементов ассерционного процесса.

В общем случае, ассерционный процесс может иметь несколько тактовых переменных. Если таковых не определено, вычисления инициируются изменением значения любой из участвующих в соотношении верификационных переменных.

2.2. Потоки активации и транспортирование событий. В моменты изменения верификационные переменные создают события:

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

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

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

Событие может быть связано с другими событиями в двух различных видах цепочек. Правая цепочка r_{\rightarrow} , формируемая двунаправленными событийными связями $e_{\rightarrow}, e_{\downarrow}$, соответствует событиям, связанным одним кольцом активации. Основная цель связывания событий в правые цепочки состоит в возможности каскадных операций над всем потоком (например, уничтожение при завершении потока). Левая однонаправленная цепочка r_{\leftarrow} , формируемая событийными связями e_{\leftarrow} , соответствует событиям различных колец активации, которые одновременно транспортируются через выбранный элемент модели. Правая цепочка неупорядочена, она формируется свободно, здесь порядок не играет никакой роли, а левая цепочка упорядочена по времени создания события, что принципиально важно для конвейерной обработки нескольких потоков активации. Наглядно цепочки событий можно изобразить на рис. 2.

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

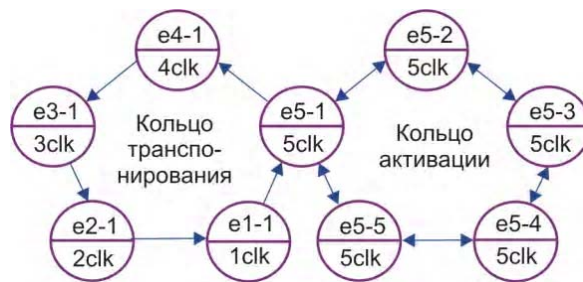


Рис. 2. Цепочки событий в модели DRTLQ

Таблица 1. Компоненты характеристического вектора события

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

В ассерционном процессе всегда существует не более одного потока активации, соответствующего некоторому определенному начальному моменту времени. Одновременно может существовать несколько потоков активации, однако все они будут относиться к вычислительным путям, соответствующим различным начальным моментам. Событие может принадлежать не более чем одному потоку активации. Активированное событие всегда относится к единственному потоку активации. Если событие не относится ни к одному потоку активации, оно не является активированным. Моменты t_b и t_a в общем случае могут не совпадать (в частности, при анализе операторов последовательностной импликации), но для большинства простых темпоральных формул $t_b = t_a$. В ряде случаев событие может не быть активировано в течение своего существования, если путь его обработки носит вспомогательный характер для анализа определенного потока активации. Активация события осуществляется первым элементом на пути обработки события, удовлетворение критерия работы которого позволяет производить дальнейший анализ вычислительного пути.

Поскольку все события в рамках потока активации связаны, удовлетворение потока активации моментально уничтожает всю цепочку связанных с ним событий. Условие удовлетворения потока активации определяется целью верификации. Если целью является опровержение возможности некоторого поведения в системе, то первое же событие с текущим значением $\rho^{VAL} = 1$, достигающее некоторой контрольной точки в модели, опровергает утверждение в целом. Лишь в случае, если все события потока активации на пути обработки при достижении контрольной точки имели значение $\rho^{VAL} = 0$, опровергающее утверждение удовлетворяется успешно. Если целью верификации является, наоборот, доказательство определенного поведения системы, то критерии удовлетворения потока активации диаметрально противоположны: первое событие, достигающее контрольной точки с текущим значением $\rho^{VAL} = 0$, опровергает утверждение, и лишь успешное завершение всех альтернативных событий потока активации со значением $\rho^{VAL} = 1$ говорит о достоверности утверждения.

Транспортирование событий осуществляется путем перегруппировки левых цепочек. Левые цепочки ассоциируются с элементами модели более высокого уровня, формирую-

щими пути транспортирования событий. Пусть $r_{\leftarrow}^1 : \{e_1^1 \cdots e_{N_1}^1\}$ представляет собой левую цепочку, из которой транспортируются события, а $r_{\leftarrow}^2 : \{e_1^2 \cdots e_{N_2}^2\}$ – левую цепочку, к которой они направляются. Суть задачи простого $r_{\leftarrow}^2 : r_{\leftarrow}^2 \cup r_{\leftarrow}^1$ и копирующего транспортирования $r_{\leftarrow}^2 : \kappa(r_{\leftarrow}^2) \cup \kappa(r_{\leftarrow}^1)$ состоит в корректном расширении цепочки r_{\leftarrow}^2 событиями из цепочки r_{\leftarrow}^1 с сохранением относительного порядка событий в результирующей цепочке r_{\leftarrow}^2 :

$$\begin{cases} e_{\leftarrow}(e_0^2) = \varepsilon; \\ \forall k, 1 \leq k < |r_{\leftarrow}^2| - 1, e_{\leftarrow}^2(e_{k-1}^2) = e_{\leftarrow}(e_k^2) \Rightarrow t_b(e_{k-1}^2) < t_b(e_k^2). \end{cases} \quad (3)$$

Слияние цепочек событий происходит начиная с правого края с одновременным итерированием в единственно допустимом направлении, при этом не теряется позиция вставки, что обеспечивает линейную сложность алгоритма относительно суммы $|r_{\leftarrow}^1| + |r_{\leftarrow}^2|$. Вставка предполагает изменение связей e_{\leftarrow} непосредственно вносимого в цепочку события, а также события, следующего за ним. При копирующем транспортировании оригинальные цепочки не модифицируются, в то время как простое транспортирование полностью очищает r_{\leftarrow}^1 и перезаписывает результатом r_{\leftarrow}^2 цепочку r_{\leftarrow}^2 .

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

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

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

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

2.3. Последовательностные функции в DRTLQ-модели. Последовательностной функцией $f \in F$ в модели DRTLQ называется некоторая вычислительная процедура, характеризующаяся множествами входных, внутренних и выходных событий, осуществляющая их пошаговое преобразование на каждом тактовом цикле верификации. Результатом последовательностной функции в некоторый момент времени (номер тактового цикла с начала моделирования), является содержимое множества выходных событий в этом моменте времени. Активирующей последовательностной функцией является такая последовательностная функция, выходные события которой, во-первых, инициируют некоторый поток активации, соответствующий текущему тактовому циклу, если таковой еще не был инициирован; во-вторых, назначают связи между событиями по правой цепочке, в соответствии с принадлежностью к инициированному потоку; в-третьих, устанавливают флаг $\rho^{\text{ACTIVE}} = 1$.

С точки зрения обработки одного события последовательностные функции можно интерпретировать как конечные автоматы. Событие, поступив на вход последовательност-

ной функции, попадает в некоторое начальное состояние. Из начального состояния в зависимости от особенностей вычислительной процедуры и внешних условий событие попадает в одно из внутренних состояний. Наконец, достигнув конечного состояния, событие передается на выход функции и используется следующими элементами модели.

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

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

$$f_{gen(b \in B)}(t) = [e_0 : \{e_{\leftarrow} = e_0, \begin{cases} e_{\uparrow \rightarrow} = \varepsilon, \rho^{VAL} = b(t) \end{cases}\}]. \quad (4)$$

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

$$f_{conj}(t, f_1, f_{gen}) : a = \rho^{VAL}(e^0 = \text{first}(f_{gen}(t))), \begin{cases} a = 1 \Rightarrow f_{conj}(t, f_1, f_{gen}) = f_1(t); \\ a = 0 \Rightarrow f_{conj}(t, f_1, f_{gen}) = \emptyset; \end{cases} \quad (5)$$

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

- функции-очереди, моделирующие временные интервалы;
- функции-репетиции, реализующие циклический анализ булевого выражения или другой последовательностной функции;
- функции, реализующие логические последовательностные операторы, такие как OR, AND, INTERSECT, WITHIN, а также импликации $|\rightarrow$ и $|\Rightarrow$.

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

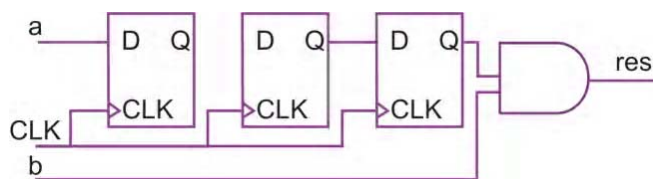


Рис. 3. Наглядная иллюстрация понятия DRTLQ-очереди

В данном случае функция-очередь $f_{\#}$ является операндом функции конъюнктивного конкатенирования f_{conj} с $f_{gen}(b)$, и имеет функцию-генератор $f_{gen}(a)$ на входе $f(t, \{a;[*2];b\}) = f_{conj}(t, f_{\#[3:3]}(t, f_{gen}(a)), f_{gen}(b))$.

В общем виде, функция-очередь, имеющая идентичные конечные интервалы, функционирует путем хранения и продвижения к выходу списка цепочек событий, считанных с входа в различные моменты моделирования, и упорядоченного по времени считывания. Цепочка событий, достигнувшая последней в очереди ячейки, является фактическим результатом функции на следующем такте обработки:

$$f_{\#[N:N]}(t, f_1(t)), N > 0 : \langle r_0(t), \dots, r_{N-1}(t) \rangle; r_0(t) = f_1(t);$$

$$\forall i, 0 < i < N, r_i(t) = r_{i-1}(t-1); f_{\#[N:N]}(t, f_1(t)) = r_{N-1}(t-1). \quad (6)$$

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

$$f_{\#[N:M]}(t, f_1(t)), N > 0, N < M : \langle r_0(t), \dots, r_{M-1}(t) \rangle; r_0(t) = f_1(t); \\ \forall i, 0 < i < M, r_i(t) = r_{i-1}(t-1); f_{\#[N:M]}(t, f_1(t)) = r_{M-1}(t-1) \cup \bigcup_{i=N-1}^{M-2} \kappa(r_i(t-1)). \quad (7)$$

Если $M \rightarrow \infty$, суть вычислительной процедуры (6) не изменяется, однако структура внутренних цепочек событий $\langle r_0(t), \dots, r_{M-1}(t) \rangle$ преобразуется из статической формы в динамическую. Очередь с бесконечным интервалом будет генерировать и продвигать к выходу копии считанного события до тех пор, пока не будет остановлен породивший событие поток активации.

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

$$f_{\#[0:M]}(t, f_1(t)), M > 0 = f_{\#[1:M]}(t, f_1(t)) \cup \kappa(f_1(t)). \quad (8)$$

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

$$f_{[*N]}(t, f_{\text{gen}(b \in B)}(t)), N > 0 : \langle v(t), r_0(t), \dots, r_{N-1}(t) \rangle; \\ v(t) = \rho^{\text{VAL}}(\text{first}(f_{\text{gen}(b)}(t))); \\ \left[\begin{array}{l} v(t) = 1 \Rightarrow \left\{ \begin{array}{l} r_0(t) = f_{\text{gen}(b)}(t); \\ \forall i, 0 < i < N, r_i(t) = r_{i-1}(t-1); \\ f_{[*N]}(t, f_{\text{gen}(b)}(t)) = r_{N-1}(t); \end{array} \right. \\ v(t) = 0 \Rightarrow \left\{ \begin{array}{l} \forall i, k, 0 \leq i < N, 0 \leq k < |r_i(t-1)|, \rho^{\text{VAL}}(e_k \in r_i(t-1)) \leftarrow 0; \\ f_{[*N]}(t, f_{\text{gen}(b)}(t)) = \bigcup_{i=0}^{N-1} r_i(t-1); \\ \forall i, 0 \leq i < N, r_i(t) = \emptyset. \end{array} \right. \end{array} \right. \quad (9)$$

Аналогично очереди (7), интервальная репетиция требует создания и продвижения к выходу копий цепочек событий, выдержанных внутри репетиции минимальное количество итераций, но не достигших максимума:

$$v(t) = 1 \Rightarrow f_{[*N:M]}(t, f_{\text{gen}(b \in B)}), N > 0, N < M = \bigcup_{i=N}^{M-1} \kappa(r_i(t-1)). \quad (10)$$

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

Реализация галопирующих репетиций модифицирует базовую репетиционную функцию (6) добавлением элемента $v(t)$, который должен обеспечивать буферизацию последнего успешного входного события до момента появления следующего. Соответственно, галопирующая репетиция никогда не выполняет полную очистку, как обычная репетиция:

$$\begin{aligned}
& f_{[\rightarrow N]}(t, f_{\text{gen}(b \in B)}(t)), N > 0 : \langle v(t), (t), r_0(t), \dots, r_{N-1}(t) \rangle; \\
v(t) = \rho^{\text{VAL}}(\text{first}(f_{\text{gen}(b)}(t))); & \begin{cases} v(t) = 1 \Rightarrow (t) = f_{\text{gen}(b)}(t); \\ v(t) = 0 \Rightarrow (t) = \kappa((t-1)); \end{cases} \\
& \begin{cases} r_0(t) = (t); \\ \forall i, 0 < i < N, r_i(t) = r_{i-1}(t-1); \\ f_{[\rightarrow N]}(t, f_{\text{gen}(b)}) = r_{N-1}(t). \end{cases} \quad (11)
\end{aligned}$$

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

$$f_{[=N]}(t, f_{\text{gen}(b)}(t)) = \kappa(f_{[\rightarrow N]}(t-1, f_{\text{gen}(b)}(t-1))) \cup f_{[\rightarrow N]}(t, f_{\text{gen}(b)}). \quad (12)$$

Наиболее сложными являются функции, реализующие логические операторы над последовательностями. Пусть имеется SERE $s = \{ \{a; b\} \parallel \{c; d\} \} \&\& \{ !e[*2] \}$. На рис. 4 в графическом виде представлена DRTLQ-реализация данного SERE, а формула (11) демонстрирует то же SERE в аналитическом виде:

$$\begin{aligned}
f_s(t) &= f_{\supset}^1(t); f_{\supset}^1(t) = \text{join}_{2-\&\&}(t, f_{\supset}^2(t), f_{[*2:2]}(t, f_{\text{gen}(-e)}(t))); \\
f_{\supset}^2(t) &= \text{join}_{2-\parallel} \left(\begin{array}{l} t, \\ f_{\text{conj}}(t, f_{\#[1:1]}(t, f_{\text{conj}}(t, f_{\subset}^2(t), a)), b), \\ f_{\text{conj}}(t, f_{\#[1:1]}(t, f_{\text{conj}}(t, f_{\subset}^2(t), c)), d); \end{array} \right); \\
f_{\subset}^2(t) &= \text{split}_{2-\parallel}(t, f_{\subset}^1(t)); f_{\subset}^1(t) = \text{split}_{2-\&\&}(t, f_{\text{gen}(l)}(t)). \quad (13)
\end{aligned}$$

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

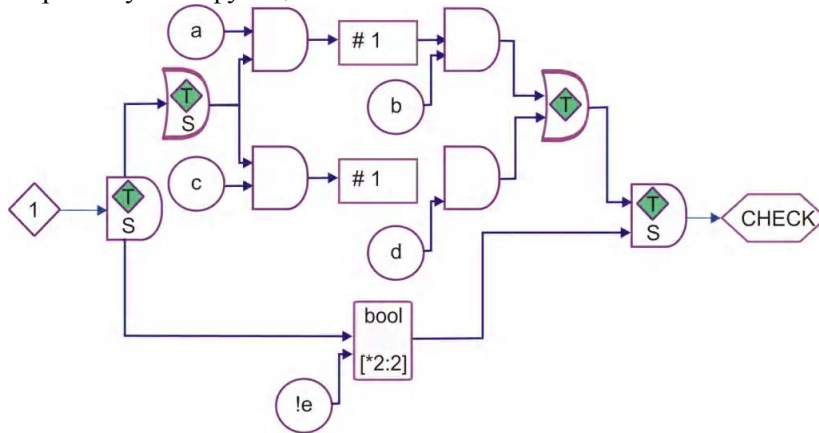


Рис. 4. DRTLQ-реализация последовательности с логическими операторами

Аналогичным образом реализуются репетиции с операндами-последовательностями: вводятся функция-разделитель и функция-соединитель, причем разделитель, определяющий направление транспортирования событий, находится выше по иерархии, чем соединитель.

тель, помещаемый на нижнем уровне. В зависимости от количества итераций, пройденных событием в рамках репетиции, событие либо направляется разделителем на выход, либо возвращается к соединителю на очередную итерацию.

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

Итерации анализа делятся на типы в зависимости от момента запуска:

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

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

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

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

Регулярная итерация анализа ассерций включает в себя следующие шаги:

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

Очередь обработки последовательностных функций упорядочивает запуск всех элементов для осуществления полной обработки модели за один проход. Первыми выполняются функции-генераторы, порождающие события от ассерционных переменных. Далее выполняются все последовательностные функции, в соответствии с иерархией.

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

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

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

$$\forall e \in r_{\leftarrow}, \begin{cases} \rho^{\text{VAL}}(e) = 1 \Rightarrow \begin{cases} \text{report}(\text{failed}); \\ \text{recursively} \\ \forall e' \in \{e \uparrow_{\rightarrow}, e \downarrow_{\rightarrow}\}(e), \rho^{\text{RDEAD}}(e') \leftarrow 1; \end{cases} \\ \rho^{\text{VAL}}(e) = 0 \Rightarrow \begin{cases} \rho^{\text{LDEAD}}(e) \leftarrow \rho^{\text{RDEAD}}(e) \leftarrow 1; \\ e \uparrow_{\rightarrow}(e) = e \downarrow_{\rightarrow}(e) = e \Rightarrow \text{report}(\text{passed}). \end{cases} \end{cases} \quad (14)$$

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

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

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

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

$$\forall e \in r_{\leftarrow}, \begin{cases} \rho^{\text{VAL}}(e) = 1 \Rightarrow \begin{cases} \text{report}(\text{passed}); \\ \text{recursively} \\ \forall e' \in \{e_{\uparrow \rightarrow}, e_{\downarrow \rightarrow}\}(e), \rho^{\text{RDEAD}}(e') \leftarrow 1; \end{cases} \\ \rho^{\text{VAL}}(e) = 0 \Rightarrow \rho^{\text{LDEAD}}(e) \leftarrow \rho^{\text{RDEAD}}(e) \leftarrow 1. \end{cases} \quad (15)$$

В случае прихода события с положительным значением $\rho^{\text{VAL}}(e) = 1$ поток активации удовлетворяется, и монитор сигнализирует об успешном статусе анализа. Отрицательный статус $\rho^{\text{VAL}}(e) = 0$ лишь уничтожает поступившее событие. Никакие действия с потоком активации не происходят, поскольку, в соответствии с [14, формула (26)], в режиме локального времени свойство `eventually!` не может дать отрицательного результата, за исключением конца моделирования.

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

При наличии в модели уровня темпоральных свойств задачи проверки критериев удовлетворения потоков активации (14), (15) выполняются именно этими элементами. Вспомогательные последовательностные функции f_{ACTION} доставляют выходные цепочки событий на первый уровень иерархии темпоральных свойств. С ассерционным монитором взаимодействует только вершина иерархии свойств. Роль самого монитора упрощается и сводится к обработке статуса единственного за весь сеанс моделирования события.

Каждое свойство состоит из следующих структурных элементов:

$$p \in P : \{p_{\uparrow}, P_{\downarrow} \in P\} \times \{\text{INIT}, \text{ABORT}, \text{EVAL}, \text{FINAL}\}, \quad (16)$$

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

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

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

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

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

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

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

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

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

$$p_{\text{BUF}} : \{p_{\uparrow}, \left[\begin{array}{l} P_{\downarrow} = \emptyset, f_{\text{ACTION}} \in F; \\ P_{\downarrow} : \{p_1 \in P\}; \end{array} \right], \Omega_{\text{BUF}} \in \Omega\} \times \{\text{INIT}_{\text{BUF}}, \text{ABORT}_{\text{BUF}}, \text{EVAL}_{\text{BUF}}, \text{FINAL}_{\text{BUF}}\}, \quad (17)$$

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

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

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

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

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

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

$$P_{NOT} : \left\{ p \uparrow, \left[P \downarrow = \emptyset, f_{ACTION} \in F; \right. \right. \\ \left. \left. P \downarrow : \{p_1 \in P\}; \right. \right. \Omega_{NOT} \in \Omega \times \left. \left. \begin{array}{l} INIT_{NOT} = INIT_{BUF}; \\ ABORT_{NOT} = ABORT_{BUF}; \\ EVAL_{NOT}; \\ FINAL_{NOT}. \end{array} \right. \right\}. \quad (18)$$

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

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

$$P_G : \left\{ p \uparrow, \left[P \downarrow = \emptyset, f_{ACTION} \in F; \right. \right. \\ \left. \left. P \downarrow : \{p_1 \in P\}; \right. \right. \Omega_G \in \Omega \times \left. \left. \begin{array}{l} INIT_G; \\ ABORT_G = ABORT_{BUF}; \\ EVAL_G; \\ FINAL_G; \end{array} \right. \right\}. \quad (19)$$

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

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

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

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

Коммутативные свойства, такие как AND , имеют следующую структуру:

$$p_{\wedge} : \left\{ p \uparrow, P \downarrow : \{p_1, \dots, p_N\}, \mu(t) : \langle k \in [0; N], v \in [0; 1] \rangle \times \right. \\ \left. \begin{array}{l} INIT_{\wedge} : \forall i, 0 \leq i < N, INIT(p_i); \\ ABORT_{\wedge} : \forall i, 0 \leq i < N, ABORT(p_i); \\ EVAL_{\wedge}; \\ FINAL_{\wedge}; \end{array} \right\}, \quad (20)$$

где N – число операндов; p_1, \dots, p_N – свойства-операнды, а $\mu(t)$ – функция истории выполнения потока активации, каждый элемент которой соотносится с некоторым такто-

вым циклом и включает информацию о количестве завершенных на текущий момент операндов, а также общий статус завершения.

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

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

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

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

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

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

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

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

$$p_U : \{p_{\uparrow}, p_{\downarrow} : \{p_1, p_2\}, v(t) : \{0, 1, X\}, \{s_1, s_2, s'_1, s'_2\}(t) : \{0, 1, X\}\} \times \{INIT_U, ABORT_U = \{ABORT(p_1); ABORT(p_2)\}, EVAL_U, FINAL_U\}, \quad (21)$$

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

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

Работа процедуры $EVAL_U$ состоит из следующих шагов:

1. Обновление состояния операнда, с которого пришло событие: $s_{\left[\begin{smallmatrix} 1 \\ 2 \end{smallmatrix} \right]}(t) \leftarrow \rho^{VAL}(e)$.

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

$$\exists k, 0 \leq k < (t_{now} - t_A(e)), \quad s'_2(t_A(e) + k) = \begin{cases} 0 \Rightarrow \begin{cases} s'_1(t_A(e) + k) = 0 \Rightarrow v(t_A(e) + k); \\ v(t_A(e) + k) \leftarrow X; \end{cases} \\ 1 \Rightarrow \begin{cases} k > 0 \Rightarrow v(t_A(e) + k) \leftarrow s'_1(t_A(e) + k - 1); \\ k = 0 \Rightarrow v(t_A(e) + k) \leftarrow 1; \\ X \Rightarrow v(t_A(e) + k) \leftarrow X; \end{cases} \end{cases} \quad (22)$$

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

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

Таблица 2. Определения бинарных темпоральных операторов

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

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

Другие свойства определяются в модели DRTLQ аналогичным образом в соответствии с семантикой LTL -операторов.

3. Выводы

1/ Аналитическая модель верификации HDL-кода с использованием механизма темпоральных ассерций ориентирована на достижение заданной глубины диагностирования и представлена в следующем виде:

$$M = f(F, L, T, C, A, t), C = \{C_1, C_2, \dots, C_i, \dots, C_m\}; L = \{L_1, L_2, \dots, L_i, \dots, L_n\};$$

$$A(t) = \{A_1, A_2, \dots, A_i, \dots, A_k\}; A \subseteq L; F = L \times C; k \leq n; T = \{T_1, T_2, \dots, T_i, \dots, T_p\}.$$

Здесь C_i – группа операторов кода, нагруженная на вершину L_i (переменная, регистр, счетчик, память) и формирующая ее состояние; F – функциональность, представленная транзакционным графом $F = L \times C$ в виде декартова произведения множества вершин и дуг; A – совокупность темпоральных ассерций, как подмножество вершин транзакционного графа $A \subseteq L$. Метод поиска функциональных нарушений (ФН) блока операторов кода использует предварительно построенную таблицу ФН $B = [B_{ij}]$, где строка есть отношение между тестовым сегментом и подмножеством программных блоков $T_i = (B_{i1}, B_{i2}, \dots, B_{ij}, \dots, B_{in})$ с возможными ФН. Столбец таблицы формирует отношение между программным блоком и тестовыми сегментами $B_j = (T_{1j}, T_{2j}, \dots, T_{ij}, \dots, T_{pj})$, которые могут проверять блок с ФН. На стадии моделирования определяется обобщенная реакция $m = \{m_1, m_2, \dots, m_i, \dots, m_p\}$ механизма ассерций F на тест, путем формирования $m_i = (A_1 \vee A_2 \vee \dots \vee A_i \vee \dots \vee A_k)$, $A_i = \{0, 1\}$ как реакции ассерций на тест-сегмент T_i . Поиск ФН основан на определении хог-операции между вектором состояния ассерций и столбцов таблицы ФН $m \oplus (B_1 \vee B_2 \vee \dots \vee B_j \vee \dots \vee B_n)$. Выбор решения определяется совокупностью векторов B_j с минимальным числом единичных координат

$$B = \min_{j=1, n} [B_j = \sum_{i=1}^p (B_{ij} \oplus m_i)],$$

формирующих программные блоки с ФН, проверяемые на тестовых сегментах.

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

3. В отличие от конечных автоматов модель DRTLQ способна одновременно осуществлять конвейерный анализ нескольких пересекающихся вычислительных путей. Цикл работы DRTLQ модели обеспечивается пошаговым транспортированием событий от входов модели к выходному ассерционному монитору. Понятие левой цепочки (кольцо транспортирования) обеспечивает линейную сложность продвижения событий в зависимости от количества инициированных потоков и степени их пересечения по ходу анализа. Правая цепочка (кольцо активации) обеспечивает быстрое досрочное завершение вычислительно-го потока, что особенно эффективно в случае наличия сильных разветвлений.

4. Модель DRTLQ способна эффективно реализовать предложенный режим глобального времени, предполагающий полноценную интерпретацию сложных темпоральных операторов. Событийная модель обработки конструкций-свойств и вычислительные процедуры, контролирующие множества инициированных вычислительных путей, обеспечивают эффективную реализацию всех LTL-операторов с линейной сложностью. Производительность достигается иерархической декомпозицией событийных потоков. Теоретическая древовидность низкоуровневых вычислений, определяемая семантикой LTL, подменяется линейной нагрузкой на последовательностный уровень, а фактический учет сложных ветвлений эффективно реализуется на высоком уровне абстракции. Вычислительная сложность реализации такова, что применение на практике режима глобального времени оказывается существенно быстрее (от 1,25 до 5 раз) по сравнению с традиционным режимом интерпретации.

5. *Научная новизна полученных результатов.* Впервые предложена аналитическая модель верификации, которая характеризуется использованием динамических регистровых очередей для анализа темпоральных ассерций в процессе моделирования тестов для цифровых систем на кристаллах, что обеспечивает высокое быстродействие моделирования и заданную глубину диагностирования ошибок кода.

6. *Практическое значение полученных результатов.* Разработанная аналитическая модель верификации цифровых систем на основе использования динамических регистровых очередей для анализа ассерций линейной темпоральной логики доведена до практической реализации в виде программного обеспечения в составе верификационной системы Riviera (Aldec Inc.). В сочетании с другими реализованными методами верификации, данный программный продукт представляет собой комплексное полноценное решение по верификации System-on-Chip. Продукт занимает лидирующие позиции на мировом рынке программных средств в сфере EDA.

Список литературы: 1. *Clarke E., Grumberg O., Peled D.* Model Checking. 6th edition, MIT Press, 2008. 313p. 2. *Burch J.R., Clarke E.M., Long D.E.* Representing circuits more efficiently in symbolic model checking // 28th ACM/IEEE Design Automation Conference, 1991. P. 403-407. 3. *Pnueli A.* The Temporal Logic of Programs // Proceedings of the 18th IEEE Symposium Foundations of Computer Science (FOCS 1977). 1977. P. 46-57. 4. *Buchi J.R.* On a decision method in restricted second order arithmetic // Proceedings of International Congress on Logic, Methodology and Philosophy of Science, Stanford University Press, 1960. P. 1-11. 5. *He A., Wu J., Li L.* An Efficient Algorithm for Transforming LTL Formula to Buchi Automaton // International Conference on Intelligent Computation Technology and Automation. 2008. P. 1215-1219. 6. *Gastin P., Oddoux D.* Fast LTL to Buchi automata translation // 13th International Conference on Computer Aided Verification (CAV 2001), volume 2102 of Lecture Notes in Computer Science. P. 53-65. 7. *Giannakopoulou D., Lerda F.* From states to transitions: Improving translation of LTL formulae to Buchi automata // 22nd International Conference on Formal Techniques for Networked and Distributed Systems (FORTE 2002), volume 2529 of Lecture Notes in Computer Science. 2002. P. 308-326. 8. *Etessami K., Holzmann G. J.* Optimizing Buchi automata // 11th International Conference on Concurrency Theory (CONCUR 2000), volume 1877 of Lecture Notes in Computer Science, 2000. P. 153-167. 9. *Saffra S.* On the complexity of omega-automata // 29th IEEE Symposium on Foundations of Computer Science, 1988. P. 319-327. 10. *Valmari A.* The state explosion problem // Lectures on Petri Nets I: Basic Models, volume 1491 of Lecture Notes in Computer Science, 1998. P. 429-528. 11. *Fernandez J.C., Jard C., Jeron T., Viho A.* Using on-the-fly verification techniques for the generation of test suits // Proceedings of 1996 Workshop of Computer-Aided Verification. 1996. P. 348-359. 12. *Voeten J.P.M., Van der Putten P.H.A., Geilen M.C.W., Stevens M.P.J.* Formal modelling of reactive hardware/software systems // ProRISC/IEEE'97, Utrecht: STW, Technology Foundation, 1997. P. 663-670. 13. *Ribeiro O., Fernandes J., Pinto L.* Model Checking Embedded Systems with PROMELA // 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05). 2005. P. 378-385. 14. *Зайченко С.А.* Модель интерпретации высокоуровневых операторов LTL-логики // АСУ и приборы автоматики. 2009. Вып. 149. С. 96-111.

Поступила в редколлегию 12.10.2009

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

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

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ РАСПРЕДЕЛЕНИЯ ЧАСТИЦ ПОРОШКОВОЙ МАССЫ ПО РАЗМЕРАМ

Рассматривается процесс измельчения твёрдых частиц в сепараторах. Исследуется способ распределения твёрдых частиц по размерам при помощи воздушно-центробежных сепараторов. Описывается процесс сепарации и определяются силы, действующие на мелкодисперсные частицы. Выводятся формулы для сил, действующих на частицы в зависимости от дисперсных характеристик рассматриваемых частиц. Определяются факторы, которые наиболее сильно влияют на точность процесса сепарации.

1. Введение и постановка задачи

Рассмотрим процесс распределения твёрдых мелкодисперсных частиц при помощи механических аппаратов. Измельчение крупных твёрдых частиц до частиц малых размеров обычно производится в целях ускорения процесса перемешивания порошковых масс. Так как на практике все материалы, применяемые в промышленных целях, обычно представлены в твёрдых крупнодисперсных состояниях, то измельчение является одним из основных способов их переработки. В работах [1-5] были рассмотрены различные способы измельчения твёрдых крупнодисперсных материалов. Данная статья является продолжением исследований, начатых в [1]. Далее механический способ измельчения будет рассмотрен как наиболее простой и экономичный из всех методов диспергирования.

Условия формирования порошковых масс зависят от физико-химических свойств исходного материала и дисперсных свойств частиц, физико-химических характеристик среды. Зачастую требуется получить мелкодисперсные порошковые смеси с заданными характеристиками (диаметр частиц, свойства поверхности и т.д.) при минимальных затратах энергии и времени. С этой целью изучается воздействие на различные виды материалов разных возмущений (механических, гидравлических, химических, энергетических [4,5]).

Технологический процесс измельчения твёрдых частиц в специальных аппаратах можно разделить на два этапа. Первый этап – измельчение частиц под действием возмущений, направленных из внешней среды на исходный материал. Второй этап – собственно формирование порошковой массы с заданными характеристиками. Задачей исследования процессов измельчения является, в частности, изучение физико-химических и поверхностных свойств материала (знание этих свойств необходимо для того, чтобы определить влияние воздействия частиц порошковой массы друг на друга). Знание физико-химических свойств поверхностей частиц в ряде случаев необходимо для определения условий, при которых они могут быть применены на практике.

Целью работы является: а) исследование процесса измельчения твёрдых частиц порошковой смеси в различных частях сепаратора; б) исследование сил, действующих на частицы в зависимости от их дисперсных характеристик; в) установление факторов, которые наиболее сильно влияют на процесс распределения частиц внутри сепаратора.

Задача работы состоит в построении математической модели процесса распределения мелкодисперсных частиц в воздушно-центробежных сепараторах.

2. Распределение частиц порошковой массы по их размерам

Принцип действия воздушно-центробежных сепараторов состоит в следующем: частицы исходного материала подвергаются одновременно действию двух сил – центробежной, которая создаётся вращением потока, и радиальной, создаваемой движением потока к центру сепарационной камеры. При рассмотрении стационарного случая можно прийти к выводу о равенстве двух указанных сил, которое определяется радиусом круговой траектории частиц порошковой массы заданных размеров. Если же частицы будут больших диаметров, чем расчётные, то они будут перемещаться по спирали к краям сепарационной камеры, а частицы с диаметрами меньше заданного – также по спирали к центру сепаратора.

Исследуем случай, когда выполняются условия:

1) Любую фиксированную частицу можно рассматривать как частицу сферической формы (или формы, которую можно привести к сферической при помощи использования определённого эквивалентного сфере ее размера и коэффициента формы) в контексте теории сепарации в центробежно-воздушных полях.

2) Предполагаем, что частицы порошковой массы не соприкасаются с движущимися деталями сепаратора, т.е. частица рассматривается как свободно подвешенная в потоке.

Рассмотрим основные силы, действующие на процесс.

Сила потока, который переносит частицу:

$$\varphi_{\text{str}} = \frac{1}{8} k \pi \theta d^2 \rho_{\text{cp}} v_r^2. \quad (1)$$

Центробежная сила, действующая на сферическую частицу диаметром d , равна:

$$\varphi_{\text{cent}} = \pi d^3 (\rho - \rho_{\text{cp}}) \frac{v_\tau}{6R}. \quad (2)$$

Здесь ρ и ρ_{cp} – соответственно плотность частицы и среды, в которой производится сепарация; R – радиус траектории; v_τ и v_r – тангенциальная и радиальная составляющие скорости потока на расстоянии R от центра; θ – безразмерный коэффициент, учитывающий отклонение формы частицы от сферической:

$$k = 24 \text{Re} \quad \text{при} \quad 0 < \text{Re} < 1, \quad (3)$$

$$k = 13 \text{Re}^{\frac{1}{3}} \quad \text{при} \quad 1 < \text{Re} < 10, \quad (4)$$

где параметр Re – число Рейнольдса.

Равенство сил φ_{cent} и φ_{str} определяет граничный размер частиц d_0 , находящихся на круговой траектории заданного радиуса.

Для частиц минимальных размеров и небольших скоростей в соответствии с условием (3) имеем

$$d_0 = 4,24 \left(\frac{\rho}{\rho - \rho_{\text{cp}}} \theta \right)^{\frac{1}{2}} (R v_\tau)^{\frac{1}{2}} v_\tau^{-1}. \quad (5)$$

В случае частиц средних размеров и достаточно больших скоростей v_τ в соответствии с (4) имеем:

$$d_0 = 4,54 \left[\left(\frac{\rho}{\rho - \rho_{\text{cp}}} \theta \right)^2 \right]^{\frac{1}{3}} \frac{R^{\frac{2}{3}} v_r}{v_\tau^{\frac{4}{3}}}, \quad (6)$$

$$v_r = \frac{V}{2\pi} R h, \quad v_\tau = v_{\tau 0} \left(\frac{R_0}{R} \right)^\delta. \quad (7)$$

Здесь V – объемный расход воздуха; h – высота зоны разделения [6]; $v_{\tau 0}$ – тангенциальная составляющая у внешней границы зоны; R_0 – расстояние внутренней границы зоны до центра. В зависимости от способа формирования потока величина δ может изменяться в пределах $-1 \leq \delta \leq 1$.

При $\delta = 0$ и $h(R) = \text{const}$ для частиц наименьших размеров $v_\tau = \text{const}$ и d_0 не зависит от R . Следовательно, процесс сепарации обычно распределяет частицы только по одной из их характеристик, в зависимости от их дисперсности (т.е. по размеру в зависимости от величины диаметра). Круговые траектории движения частиц одинаковых размеров в пределах зоны сепарации являются устойчивыми, а значит частицы с достаточно близкими дисперсными характеристиками могут находиться в этой зоне достаточно долго. Данный случай является идеальным для работы сепаратора. Для такого процесса необходимым

условием является следующее утверждение: угловая скорость потока увеличивается пропорционально $\frac{1}{R}$ по мере приближения потока к центру. Любые частицы, размеры которых больше d_0 , попадая в рассматриваемую нами зону, будут по спиралям удаляться к краям сепарационной зоны, а частицы диаметров мельче d_0 – также по спиралям к центру.

Если $0 < \delta \leq 1$, то на круговых орбитах, приближающихся к центру, частицы с близкими дисперсными характеристиками будут больших размеров, а при $-1 \leq \delta < 0$ – меньших. Отсюда получаем следующее утверждение: при $\delta \neq 0$ в сепараторе существует несколько зон распределения частиц порошковой массы, а некоторую область можно определить при помощи формул (5) и (6).

Движение частиц (одинаковых по своим дисперсным характеристикам) по круговым траекториям при $\delta \neq 0$ является неустойчивым. Возмущения потока и случайные перемещения частиц будут переводить их на спиральные траектории и направлять в один из продуктов сепарации.

С помощью формул (5)–(7) можно получить следующее соотношение:

$$\frac{d_{O1}}{d_{O2}} = \left(\frac{R_1}{R_2} \right)^\delta. \quad (8)$$

Здесь R_1 и R_2 – граничные радиусы области сепарации; d_{O1} и d_{O2} – равнозначные размеры частиц на внешней и внутренней границах зоны распределения. Частицы с размерами $d_{O1} < d < d_{O2}$ в зависимости от места их попадания в зону сепарации могут попасть как в крупнодисперсный порошок, так и в мелкодисперсную порошковую смесь. Поэтому в мелкодисперсную смесь будут входить все частицы мельче d_{O2} , а в крупнодисперсную – все частицы, дисперсные характеристики которых больше d_{O1} .

Доля частиц заданного размера, попадающих в определенную зону сепарации, равновероятна (при условии, что попадание частицы в любую точку зоны пропорционально ширине этой зоны).

Из приведенных выше условий и на основании (8):

$$\frac{\pi R^2 - \pi R_1^2}{\pi R_2^2 - \pi R_1^2} = \frac{\left(\frac{R}{R_1} \right)^2 - 1}{\left(\frac{R_2}{R_1} \right)^2 - 1} = \frac{\left(\frac{d}{d_{O1}} \right)^\delta - 1}{\left(\frac{d_{O2}}{d_{O1}} \right)^\delta - 1}. \quad (9)$$

Состав мелкодисперсной части порошковой массы представляет собой:

$$J(d) = \int_0^{d_{O1}} F(d)dd + \int_{d_{O1}}^{d_{O2}} \frac{\left(\frac{d}{d_{O1}} \right)^\delta - 1}{\left(\frac{d_{O2}}{d_{O1}} \right)^\delta - 1} F(d)d(d). \quad (10)$$

На основании (9) и (10) можно вывести коэффициенты полезного действия сепаратора:

$$\eta_1 = \frac{J(d)}{\int_0^{d_{O2}} F(d)d(d)}, \quad (11)$$

$$\eta_2 = \frac{\int_0^{d_{O1}} F(d)d(d)}{J(d)}. \quad (11')$$

Если параметр δ заметно отличается от нуля, а отношение $\frac{R_2}{R_1}$ велико (порядка 2–5), то точность распределения частиц по их размерам и КПД классификации низкие (при условии, что на $[d_{O_1}, d_{O_2}]$ приходится значительная часть порошковой массы).

В общем случае точность процесса сепарации при фиксированных параметрах распределения определяется изменениями сил, которые действуют на частицы критических размеров. Можно сделать вывод, что точность распределения частиц зависит от следующего условия: *чем быстрее изменяются действующие на компоненты порошковой массы возмущения, тем точнее будет конечное их распределение.* Это свойство также определяет предпочтительное использование воздушно-центробежных сепараторов в сравнении с гравитационными при распределении малых частиц.

Действительно, для гравитационной силы и противодействующей ей силы воздушного потока

$$\begin{aligned}\varphi_{vp} &= \frac{1}{6} \pi(\rho - \rho_{cp})gd^3 - 3\pi\mu d v_{potoka} = \\ &= \frac{1}{6} \pi(\rho - \rho_{cp})g(d^3 - dd_{ravn}^2),\end{aligned}\quad (12)$$

где μ – динамическая вязкость среды; v_{potoka} – скорость потока относительно частицы, а d_{ravn} – равновесное значение размера частиц для $\varphi = 0$:

$$d_{ravn}^2 = \frac{18\mu v_{potoka}}{(\rho - \rho_{cp})g}.$$

Величина производной и дифференциал d в точке $d = d_{ravn}$ равны

$$\begin{aligned}\frac{d\varphi_{vp}}{d(d)} &= \frac{1}{6} \pi(\rho - \rho_{cp})g(3d^2 - d_{ravn}^2); \\ d(d)|_{d=d_{ravn}} &= \frac{3d\varphi}{\pi(\rho - \rho_{cp})gd_{ravn}^2}.\end{aligned}\quad (13)$$

Для центробежных полей

$$\begin{aligned}\varphi_{cent} &= \frac{1}{6} \pi d^3 (\rho - \rho_{cp}) \omega^2 R - \frac{13}{8} \pi \rho_{cp} v^2 d^{\frac{1}{2}} d^{\frac{3}{2}} v^{\frac{3}{2}} = \\ &= \frac{1}{6} \pi (\rho - \rho_{cp}) \omega^2 R (d^3 - d^{\frac{3}{2}} d_0^{\frac{3}{2}}),\end{aligned}\quad (14)$$

где ω – угловая скорость вращения на расстоянии R от центра. Равновесное значение размера частиц на расстоянии R от центра вращения равно

$$d_{ravn} = \left(\frac{39 v_f^{\frac{3}{2}} \rho_{cp} v^{\frac{1}{2}}}{4 \omega^2 R (\rho - \rho_{cp})} \right).$$

Величина производной и дифференциал в точке $d = d_{ravn}$ для центробежных полей равны

$$\begin{aligned}\frac{d\varphi_{cent}}{d(d)} &= \frac{1}{6} \pi (\rho - \rho_{cp}) \omega^2 R (3d^2 - \frac{3}{2} d_{ravn}^{\frac{3}{2}} d^{\frac{1}{2}}); \\ d(d)|_{d=d_{ravn}} &= \frac{4d\varphi}{(\rho - \rho_{cp}) \omega^2 R d_{ravn}^2}.\end{aligned}\quad (15)$$

Сравнивая формулы (13) и (15), можно видеть, что величина дифференциала dd при равных значениях $d\varphi$ в случае центробежных полей меньше, чем в случае гравитационного поля.

Из формул (13) и (15) следует, что с уменьшением граничного размера величина случайного отклонения размеров частиц по обе стороны границы их разделения увеличивается достаточно быстро. Следовательно, точность сепарации с ростом дисперсности резко падает, а экспериментальные трудности реализации процесса распределения возрастают.

На качестве сепарации сказывается также значение разности плотностей сепарируемых частиц и среды. Очевидно, качество сепарации тем лучше, чем плотность частиц выше и чем менее плотной является сепарационная среда.

3. Заключение

Научная новизна. Исследован процесс распределения твёрдых частиц порошковых смесей в воздушно-центробежных сепараторах. Рассмотрен процесс движения частиц в сепараторе с течением времени. Выведены формулы для сил, действующих на частицы в зависимости от их дисперсных характеристик. Установлены факторы, которые наиболее существенно влияют на процесс распределения частиц внутри сепаратора.

Практическая ценность такого метода обработки мелкодисперсных частиц состоит в том, что мы можем получить порошковую смесь с определённым распределением частиц по их размерам.

Результаты исследований могут быть применены для получения массивов частиц, используемых на производстве. Следует учитывать, что рассматривался идеальный случай движения частиц в сепараторе. Поэтому полученные результаты следует предварительно опробовать в лабораторных условиях.

Список литературы: 1. *Вовк А.В., Дикарев В.А., Подгорбунский Н.С.* Управление процессом формирования порошковых смесей // АСУ и приборы автоматики. 2008. Вып. 143. С. 50- 57. 2. *Аганова И.С., Дикарев В.А., Подгорбунский Н.С.* Эволюция скачков и изломов импульсов при их распространении в информационном канале // АСУ и приборы автоматики. 2008 Вып. 142. С. 57- 63. 3. *Коузов П.А.* Основы анализа дисперсного состава промышленных пылей и измельчённых материалов. Л.: Химия, 1987. 264с. 4. *Гора Н.Н., Вовк А.В.* Вывод системы дифференциальных уравнений, описывающей процесс обработки многокомпонентной смеси // Вестник НТУ ХПИ, тематический выпуск «Информатика и моделирование», В23, 2006. С. 19-28. 5. *Вовк А.В.* Процесс формирования порошковых масс в объёме активной смеси // Радиоэлектроника и информатика. 2007. С. 141-144. 6. *Авдеев Н.Я.* Расчёт гранулометрических характеристик полидисперсных систем. Ростов-на-Дону: Ростовское книжн. изд-во, 1966. 54 с.

Поступила в редколлегию 10.12.2009

Подгорбунский Никита Сергеевич, аспирант кафедры прикладной математики ХНУРЭ. Научные интересы: теория вероятностей, случайные процессы. Адрес: Украина, 61195, Харьков, ул. Метростроителей, 15, кв. 23, тел. 716-02-70.

АДАПТИВНАЯ МОДЕЛЬ НАДЕЖНОСТИ И РАБОТОСПОСОБНОСТИ ТЕХНИЧЕСКОЙ СИСТЕМЫ

Предлагается модель надежности и работоспособности технической системы. Предполагается, что в процессе эксплуатации из-за случайных процессов износа и разрегулировки работоспособность системы снижается. Характеристики этого процесса известны не полностью. Предлагается метод последовательного улучшения стратегии управления работоспособностью в процессе наблюдения за поведением системы. Управлениями считаются виды регламентных работ, проводимые в заданные моменты времени.

1. Общая постановка задачи, актуальность и цель исследования

Важной задачей при эксплуатации технической системы на неограниченном интервале времени является обеспечение оптимального уровня ее работоспособности [1]. Основным показателем работоспособности часто может быть охарактеризован величиной прибыли, которую обеспечивает работающая система. Для поддержания оптимального уровня работоспособности предусмотрено проведение регламентных работ, направленных на обновление системы. Под обновлением системы будем понимать все виды ремонтных работ, проведение регулировок и других действий, обеспечивающих такое состояние системы, которое в конечном итоге увеличивает прибыль и эффективность ее использования. Регламентные работы различаются по глубине воздействия на систему и, следовательно, по стоимости.

Актуальной задачей является выбор такой стратегии применения регламентных работ и их типов, которые обеспечивали бы максимальную прибыль от эксплуатации системы.

Пусть введено в рассмотрение понятие состояния системы. Состояние системы определяется набором контролируемых параметров, которому соответствует определенный уровень эффективности и работоспособности. Выбор набора информативных параметров и определение с их помощью состояния могут быть проведены методами теории распознавания образов [2]. Далее будем считать, что множество состояний $\Xi = \{x\}$ определено и ранжировано по уровню работоспособности системы. Положим, что это множество совпадает с множеством неотрицательных вещественных чисел. Состоянию $x = 0$ соответствует наивысшая эффективность функционирования системы, при возрастании значения x эффективность монотонно уменьшается.

Предполагается, что регламентом установлено проведение профилактических ремонтов после ее работы в течение интервала времени τ . Тип регламентных работ далее будем называть управлением. Множество всех различных управлений обозначим через $Y = \{y_1, y_2, \dots, y_l\}$. Допускаем, что применение управления y_1 в любом состоянии x не меняет этого состояния. Применение управления y_k , $k > 1$, в состоянии x приведет к переходу системы в одно из лучших, по сравнению с x , состояний. Это состояние определяется заданным вероятностным распределением на множестве $[0, x)$. Будем предполагать, что среднее значение обновления системы тем больше, чем больше номер управления. Ниже это свойство будет уточнено.

Реализация управления занимает определенное время, в течение которого система простаивает, и требует определенных затрат.

Состояние системы в момент контроля и принятия решения о выборе управления является реализацией случайной величины. Вероятностное распределение этой случайной величины неизвестно. Выбор типа распределения для конкретной рассматриваемой системы возможен лишь в том случае, если доступна необходимая статистическая информация. Выберем для моделирования тот класс систем, для которых это распределение является гамма-распределением. Такой выбор достаточно общий, так как при определенных значениях параметров гамма-распределение определяет экспоненциальное распределение, рас-

пределение χ^2 и другие. Предлагаемый метод моделирования и оптимизации потребует очевидных изменений при использовании другого типа распределения.

Целью исследования является разработка метода оптимизации уровня работоспособности технической системы в условиях неполной информации о ее характеристиках. В работе предлагается адаптивный подход к решению этой задачи. Наблюдения, получаемые в процессе работы системы, последовательно используются для уточнения стратегии управления.

2. Математическая модель

В основу моделирования выбранной системы положим марковский процесс принятия решений [3]. Для этого необходимо определить следующие элементы: E – множество состояний, Y – множество управлений, Q – переходная функция, определяющая одношаговые переходы на множестве состояний E , π – стратегия управления, w – функция непосредственных доходов.

Положим, что множество состояний E конечно. Для этого проведем дискретизацию множества $\Xi = [0, \infty)$. Пусть h – шаг дискретизации. Выберем количество состояний, задав число N . Тогда множество состояний $E = \{x_0, x_1, \dots, x_N\}$. Если в момент контроля состояние системы попадает в полуинтервал $[ih, (i+1)h)$, $0 \leq i \leq N-1$, то будем считать, что наблюдалось состояние $x_i = ih$. Наблюдению значения из полуинтервала $[Nh, \infty)$ поставим в соответствие состояние x_N .

Пусть множество управлений Y состоит из l элементов: $\{y_1, y_2, \dots, y_l\}$, каждое из которых может быть применено в любом состоянии.

Образование $\omega: E \rightarrow Y$ назовем решающей функцией, последовательность решающих функций $\pi = \{\omega_1, \omega_2, \dots\}$ – стратегией управления. Стратегия управления π для каждого состояния x в момент t_k определяет выбор управления $\omega_k(x) \in Y$.

Стратегию вида $\pi = \omega^{(\infty)} = \{\omega, \omega, \dots\}$ назовем стационарной.

Для определения переходной функции Q проведем вспомогательные построения. Пусть в момент контроля t_k идентифицировано значение $x \in \Xi$. Применение управления $y_k \in Y$ в этом состоянии за время T_k переведет систему в некоторое случайное состояние $z \in [0, x)$ в соответствии с плотностью распределения вероятностей $f_k(z|x)$, $k=2, \dots, l$. Для $k=1$ величина простоя системы T_1 равна 0, а плотность f_1 вырождена в точке x . Все плотности $f_k(z|x)$, $k=2, \dots, l$ определяются конкретными регламентными работами, оцениваются статистическими методами. Будем считать их заданными.

Управления различаются по степени обновления системы. Положим, что они упорядочены следующим образом. Если $2 \leq k < n \leq l$, то

$$\int_0^x z f_k(z|x) dz > \int_0^x z f_n(z|x) dz$$

для всех $x \in \Xi$.

Последнее неравенство означает, что чем больше номер управления, тем в большей степени в среднем оно обновляет систему.

Рассмотрим теперь действие управлений на множестве E . Если наблюдаемое значение x попало в полуинтервал $[x_i, x_{i+1}) \subset \Xi$, то будем считать, что система находится в состоянии $x_i \in E$. Пусть в этом состоянии было применено управление $y_k \in Y$. Тогда через интервал времени T_k система перейдет в состояние x_j , $j=0, \dots, i$, с вероятностью

$$p_{i0}^{(k)} = \int_0^{x_i} f_k(z|x) dz, \quad p_{i1}^{(k)} = \int_{x_1}^{x_2} f_k(z|x) dz, \quad \dots, \quad p_{ii}^{(k)} = \int_{x_i}^x f_k(z|x) dz. \quad (1)$$

Очевидно, что $p_{00}^{(k)} = 1$ для любого $y_k \in Y$, $p_{ii}^{(1)} = 1$ для любого $x_i \in E$. Заметим, что длительность перехода T равна 0 только для управления y_1 .

После перехода в состояние z под действием управления $y_k \in Y$, система работает в течение периода времени τ . С работающей системой связана интенсивность дохода v_ξ , зависящая от состояния $\xi \in \Xi$. По предположению случайная величина $(x-z)$ имеет гамма-распределение с неизвестными параметрами α и β , где $x \in \Xi$ – наблюдение состояния системы в конце указанного периода. Плотность гамма-распределения имеет следующий вид [4]:

$$f(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, \quad x \geq 0, \quad \alpha, \beta > 0.$$

Определим соответствующие переходы на дискретном множестве состояний E и вероятности переходов.

Пусть $x_i \in E$ – состояние системы, в которое управление ее переводит. Обозначим искомые вероятности через q_{ij} , $x_i, x_j \in E$. По предположению, состояние системы в процессе работы не может улучшиться. Поэтому

$$q_{i0} = 0, \quad q_{i1} = 0, \dots, \quad q_{ii-1} = 0, \quad q_{ii} = \int_0^h f(x|\alpha, \beta) dx, \quad q_{ii+1} = \int_h^{2h} f(x|\alpha, \beta) dx, \dots, \\ q_{iN-1} = \int_{(N-i)h}^{(N-i+1)h} f(x|\alpha, \beta) dx, \quad q_{iN} = \int_{(N-i)h}^{\infty} f(x|\alpha, \beta) dx. \quad (2)$$

Теперь легко определить переходную функцию Q марковского процесса принятия решений. Пусть в момент контроля наблюдалось состояние $x_i \in E$, в котором применено управление $y_k \in Y$. Через интервал времени T_k это управление переведет систему в состояние $x_s \in E$ с вероятностью $p_{is}^{(k)}$. Затем через интервал времени τ состояние $x_j \in E$ будет наблюдаться с вероятностью q_{sj} . Поэтому вероятность перехода за один период, с учетом формул (1) и (2), в этом случае составит величину

$$Q_{ij}^{(k)} = \sum_{s=0}^{\min(i,j)} p_{is}^{(k)} q_{sj}. \quad (3)$$

Рассмотрим далее определение функции непосредственных доходов w . Для этого необходимо вычислить среднюю прибыль системы на одном периоде как функцию от состояния в начале периода перед применением управления и управления. Вычислим сначала средний доход на интервале времени τ , получаемый при работе системы. Пусть $x_s \in E$ – состояние системы после применения управления $y_k \in Y$. Предположив, что на интервале времени $(0, \tau)$ износ и падение эффективности работы системы происходит по линейному закону, получим среднюю величину дохода:

$$V(x_s) = \sum_{j=s}^N q_{sj} (v_s \tau + \frac{1}{2} (v_{j+1} - v_j) \tau),$$

где $v_{N+1} = 0$.

Величина непосредственного дохода в единицу времени на одном периоде при условии, что в начале периода в состоянии $x_i \in E$ применено управление $y_k \in Y$, равна

$$w(x_i, y_k) = \frac{1}{T_k + \tau} \left(\sum_{s=0}^i p_{is}^{(k)} V(x_s) - r(x_i, y_k) \right), \quad (4)$$

здесь $p_{is}^{(k)}$ определены в (1); $r(x_i, y_k)$ – стоимость управления $y_k \in Y$, примененного в состоянии $x_i \in E$.

Решающей функции ω поставим в соответствие матрицу переходных вероятностей $Q(\omega)$ с элементами $Q_{ij}^{(\omega(x_i))}$, $i, j = 1, \dots, N$, $\omega(x_i) \in Y$, и вектор-столбец непосредственных доходов $w(\omega)$ с компонентами $w(x_i, \omega(x_i))$, $i = 1, \dots, N$, $\omega(x_i) \in Y$. Пусть система управляется стационарной стратегией $\pi = \omega^{(\infty)}$. Тогда средний доход в единицу времени, приносимый системой при неограниченном времени работы, составит величину

$$\varphi(\pi) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=0}^{n-1} Q^k(\omega) w(\omega) . \quad (5)$$

Известно [3], что если множество состояний образует один эргодический класс, то вектор-столбец $\varphi(\pi)$ состоит из одинаковых компонент. Это означает, что величина среднего дохода в единицу времени не зависит от начального состояния. В нашем случае это условие выполнено.

Задача состоит в отыскании стационарной стратегии π , для которой компоненты вектора $\varphi(\pi)$ максимальны. Решение этой задачи дает алгоритм Ховарда [3]. Применить непосредственно этот алгоритм не удастся, так как модель определена не полностью. Параметры α и β , определяющие распределение приращений траектории процесса износа и разрегулировки системы, неизвестны. Предлагается применить адаптивный подход к выбору оптимальной стратегии управления. Именно, наблюдая состояние системы после применения “априорной” стратегии управления, уточнить оценки параметров, получить улучшенную “апостериорную” стратегию, применить ее и получить очередное наблюдение. Затем вновь улучшить стратегию. Улучшение стратегии понимаем здесь в смысле критерия φ . Учитывая, что стационарных стратегий конечное число, а оценки параметров α и β в определенном смысле сходятся к истинным значениям, можно доказать, что за конечное число улучшений будет достигнута ε -оптимальная стратегия. Далее рассмотрим метод оценки неизвестных параметров.

Выбор байесовского подхода к решению задачи оценки параметров в нашем случае удобен, так как легко позволяет последовательно уточнять оценки параметров по мере поступления дополнительной информации о распределении.

Заметим, что для практического применения байесовского метода оценки параметра распределения необходимо для оцениваемого параметра найти сопряженное семейство распределений [4]. Оно существует, если для этого параметра найдется достаточная статистика конечной размерности, не зависящая от объема выборки [4].

Рассмотрим метод получения оценки параметра β гамма-распределения по повторной выборке $x = \{x_1, x_2, \dots, x_n\}$ объема n . Легко видеть, что достаточной статистикой для этого параметра является $t(x) = \sum_{i=1}^n x_i$ и, следовательно, существует сопряженное семейство распределений параметра β .

Докажем, что если задана выборка x , априорное распределение параметра β есть гамма-распределение с параметрами α_0, β_0 , то апостериорное распределение параметра β есть гамма-распределение с параметрами $\alpha' = \alpha_0 + n\alpha$, $\beta' = \beta_0 + \sum_{i=1}^n x_i$, α – параметр распределения наблюдений.

Действительно, функция правдоподобия имеет вид:

$$L(\beta) = \prod_{i=1}^n \frac{\beta^\alpha}{\Gamma(\alpha)} x_i^{\alpha-1} e^{-\beta x_i} = \frac{\prod_{i=1}^n x_i^{\alpha-1}}{(\Gamma(\alpha))^n} \beta^{n\alpha} e^{-\beta \sum_{i=1}^n x_i} .$$

Если параметр β считать переменной, а $\sum_{i=1}^n x_i$ константой, то с точностью до множителя, не содержащего β , функция L пропорциональна плотности гамма-распределения, т.е. $L(\beta) \propto \beta^{n\alpha} e^{-\beta \sum_{i=1}^n x_i}$, где \propto – символ пропорциональности. Поэтому сопряженным семейством распределений параметра β является гамма-распределение [4]. Пусть априорное распределение $\xi(\beta)$ параметра β есть гамма-распределение с параметрами α_0, β_0 , т.е. $\xi(\beta) \propto \beta^{\alpha_0-1} e^{-\beta_0 \beta}$. Тогда по формуле Байеса апостериорная плотность $\xi(\beta|x)$ распределения параметра β пропорциональна произведению априорной плотности на функцию правдоподобия:

$$\xi(\beta|x) \propto \beta^{\alpha_0-1} e^{-\beta_0\beta} \beta^{n\alpha} e^{-\beta \sum_{i=1}^n x_i} = \beta^{\alpha_0+n\alpha-1} e^{-\beta(\beta_0+\sum_{i=1}^n x_i)}.$$

Полученная апостериорная плотность распределения параметра β совпадает с плотностью гамма-распределения с параметрами

$$\alpha' = \alpha_0 + n\alpha, \quad \beta' = \beta_0 + \sum_{i=1}^n x_i. \quad (6)$$

Заметим, что здесь параметр α является параметром распределения наблюдений. Далее оценки параметров распределения наблюдений будем обозначать $\tilde{\alpha}$ и $\tilde{\beta}$, а оценки параметров распределения β через: α_0, β_0 – априорные, α', β' – апостериорные. Для вычисления оценки $\tilde{\alpha}$ отметим следующее. При заданных значениях параметров α и β средняя величина приращения траектории за время τ составляет величину $\frac{\alpha}{\beta}$, а дисперсия – величину $\frac{\alpha}{\beta^2}$. Поэтому их оценки, в соответствии с методом моментов, могут быть получены из системы

$$\begin{aligned} \frac{\alpha}{\beta} &= \frac{1}{n} \sum_{i=1}^n \eta_i, \\ \frac{\alpha}{\beta^2} &= \frac{1}{n-1} \left(\sum_{i=1}^n \eta_i^2 - n\bar{\eta}^2 \right), \end{aligned} \quad (7)$$

где $\bar{\eta} = \frac{1}{n} \sum_{i=1}^n \eta_i$, $\eta_i, i=1, \dots, n$ – наблюдения приращений процесса.

Из (7) получим оценки: $\tilde{\beta} = \frac{\tilde{\alpha}}{\bar{\eta}}$,

$$\tilde{\alpha} = \frac{(n-1)\bar{\eta}^2}{\sum_{i=1}^n \eta_i^2 - n\bar{\eta}^2}. \quad (8)$$

Априорные значения параметров α_0 и β_0 выберем из условия $\frac{\alpha_0}{\beta_0} = \tilde{\beta}$. В модели предполагается, что параметр распределения случайного приращения α вычисляется по формуле (8), а параметр β – по формуле Байеса.

Оценки $\tilde{\alpha}$ и $\tilde{\beta}$ позволяют задать все элементы модели, найти оптимальную (в условиях имеющейся информации) стратегию, применить ее и получить новые наблюдения $\eta_1, \eta_2, \dots, \eta_s$.

Далее можно получим оценку $\tilde{\alpha}$ по формуле (8), учитывая, что n – объем всех наблюдений. Подставив полученное $\tilde{\alpha}$ в (6) и, если были апостериорные значения α', β' , подставив их вместо α_0, β_0 , получим новые апостериорные оценки α', β' распределения параметра β . Далее действия повторяются.

В основу метода вычисления оптимальной стратегии управления положен итерационный алгоритм Ховарда [3]. Каждое наблюдение за управляемым процессом содержит дополнительную информацию о рассматриваемой системе. Эта информация используется для улучшения по критерию φ вычисленной ранее стратегии управления. Если на текущей итерации, за счет обработки очередной порции информации, величина критерия φ изменилась на малую величину ε , где $\varepsilon > 0$ заранее задано, то предлагается алгоритм остановить.

3. Оптимизационный алгоритм

Пусть перед началом процесса управления и оптимизации системы имеется информация о наблюдаемых величинах приращений процесса износа и разрегулировки: $\eta_1, \eta_2, \dots, \eta_n$. Выберем величину $\varepsilon > 0$.

Алгоритм, реализующий адаптивный подход к управлению системой, состоит в выполнении следующих действий.

1. Выбрать решающую функцию ω_1 , определив для каждого $x_i \in E$ управление $\omega_1(x_i) \in Y$.

2. С учетом (8) найти $\tilde{\alpha}$ и $\tilde{\beta} = \frac{\tilde{\alpha}}{\eta}$, выбрать положительные значения α_0, β_0 из условия $\frac{\alpha_0}{\beta_0} = \tilde{\beta}$.

k -й шаг алгоритма.

3. По формулам (3) и (4) вычислить матрицу переходных вероятностей $Q(\omega_k)$ с элементами $Q_{ij}^{(\omega_k(x_i))}$, $i, j = 1, \dots, N$, и вектор-столбец $w(\omega_k)$ непосредственных доходов с компонентами $w(x_i, \omega_k(x_i))$, $i = 1, \dots, N$, соответствующих стационарной стратегии $\omega_k^{(\infty)}$. При вычислениях использовать полученные оценки $\tilde{\alpha}$ и $\tilde{\beta}$ в плотности распределения вероятностей $f(x|\alpha, \beta)$.

4. Решить систему линейных уравнений

$$w(\omega_k) + Q(\omega_k)R(\omega_k) - R(\omega_k) = \varphi(\omega_k)$$

относительно компоненты вектора $\varphi(\omega_k)$ и компонент вектора $R(\omega_k)$, у которого первая компонента равна 0. Если $k = 1$, обозначить $\varphi_1 = \varphi(\omega_k)$.

Вычислить $U(\omega') = w(\omega') + Q(\omega')R(\omega_k) = \max_{\omega} (w(\omega) + Q(\omega)R(\omega_k))$ и проверить неравенство $U(\omega') - w(\omega_k) - Q(\omega_k)R(\omega_k) > 0$.

Если оно выполнено, обозначить ω' через ω_k и перейти в 3. Если оно не выполнено, перейти в 5.

5. Если $k \geq 2$, обозначить $\varphi_k = \varphi(\omega_k)$ и перейти в 6. Если $k = 1$, перейти в 7.

6. Проверить условие: $|\varphi_k - \varphi_{k-1}| < \varepsilon$. Если оно выполнено, перейти в 8. Если оно не выполнено, перейти в 7.

7. Получить наблюдения приращения траектории управляемого процесса $\eta_1, \eta_2, \dots, \eta_s$, $s \geq 1$, применив полученную стационарную стратегию $\pi = \omega_k^{(\infty)}$. Вычислить оценку $\tilde{\alpha}$ по формуле (8), в которой n – количество всех наблюдений. Обозначить α' через α_0, β' через β_0 и найти параметры апостериорного распределения β по формулам:

$$\alpha' = \alpha_0 + s\tilde{\alpha}, \quad \beta' = \beta_0 + \sum_{i=1}^s \eta_i.$$

Увеличить k на единицу. С уточненными значениями оценок параметров $\tilde{\alpha}$ и $\tilde{\beta} = \frac{\alpha'}{\beta'}$ перейти в 3.

8. Стратегия $\pi = \omega_k^{(\infty)}$ является искомой. Эта стратегия обеспечивает средний доход в единицу времени, равный φ_k .

4. Заключение

Построена модель надежности и работоспособности технической системы, характеристики которой заданы не полностью. Эта модель основана на понятии состояния, которое и определяет текущий уровень надежности и работоспособности системы. Использование состояния в задачах моделирования надежности является новым и перспективным, так как позволяет строить более адекватные модели по сравнению с существующими. Для модели предложен оптимизационный алгоритм, в основе которого лежит марковский процесс принятия решений и адаптивный подход улучшения начальной стратегии, выбранной в условиях ограниченной информации об объекте. Алгоритм реализуется в реальном масштабе времени. По мере поступления информации о результатах применения текущей стратегии управления алгоритм обеспечивает уточнение этой стратегии в смысле выбранного критерия φ .

При моделировании реальной системы, как правило, информация о процессе ее износа практически отсутствует. Мы предположили, что этот процесс случаен, и за фиксированный интервал времени ухудшение характеристик системы может быть описано гамма-

распределением. Выбор другого распределения, который в реальных условиях во многом зависит от состояния системы, т.е. набора информативных параметров, не изменит принципиально модель и предложенный алгоритм.

Статистический метод оценки неизвестных параметров, в нашем случае байесовский, обеспечивает сходимость этих оценок по вероятности к истинным значениям. Это означает, что мы не можем гарантировать, что за конечное число итераций алгоритма будет достигнута оптимальная стратегия. Однако вероятность этого события с учетом конечного числа состояний и управлений возрастает с увеличением количества итераций. Предложенный метод остановки алгоритма основан на практическом требовании: продолжение алгоритма должно обеспечивать улучшение стратегии по выбранному критерию на каждой итерации не менее, чем на заданную величину ε . Оценка вероятности достижения оптимальной стратегии в момент остановки алгоритма имеет больше теоретический характер и может быть основана на оценке, полученной в [6].

Список литературы: 1. *Вопросы математической теории надежности* / Под ред. Б.В. Гнеденко. М.: Радио и связь, 1983. 376с. 2. *Горелик А.Л., Скрипкин В.А.* Методы распознавания. М.: Высшая школа, 1984. 208 с. 3. *Майн Х., Осаки С.* Марковские процессы принятия решений. М.: Наука, 1977. 175с. 4. *Де Гроот М.* Оптимальные статистические решения. М.: Мир, 1974. 493с. 5. *Крамер Г.* Математические методы статистики. М.: Мир, 1975. 648с. 6. *Подцькин Н.С.* Оптимизация периода наблюдений в марковском процессе принятия решений // Вісник Харківського національного університету. Серія “Математичне моделювання. Інформаційні технології. Автоматизовані системи управління”. 2004. Вип. 3. № 629. С.25-32.

Поступила в редколлегию 11.03.2010

Подцькин Николай Серафимович, канд. техн. наук, доцент кафедры математического моделирования и программного обеспечения Харьковского национального университета им. В.Н. Каразина. Научные интересы: математическое моделирование управляемых стохастических систем и методы их оптимизации. Адрес: Украина, 61115, Харьков, ул. 2-й Пятилетки, 2-Г, кв. 115, тел.: 707-54-68.

УДК 519.24:62.50

Н.Д. КОШЕВОЙ, О.Л. БУРЛЕЕВ, Е.М. КОСТЕНКО

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ОПТИМИЗАЦИИ МНОГОФАКТОРНЫХ ПЛАНОВ ЭКСПЕРИМЕНТА

Показывается эффективность оптимального по стоимостным затратам планирования эксперимента на примерах исследования технологических процессов, приборов и систем. Доказывается, что синтез оптимальных планов для количества факторов $k > 3$ целесообразно осуществлять методом ветвей и границ, а при количестве факторов $k \leq 3$ оптимальные планы, полученные методом ветвей и границ и методом анализа перестановок строк матрицы планирования, совпадают.

Постановка проблемы. При решении задач оптимизации и управления различными объектами возникает проблема получения математических моделей указанных объектов. При этом целесообразно получать эти модели при минимальных стоимостных и временных затратах. Особенно эта задача актуальна при исследовании дорогостоящих и длительных процессов.

1. Анализ последних исследований и публикаций. Известны методы синтеза оптимальных по стоимостным и временным затратам планов эксперимента [1-3], основанные на использовании следующих видов оптимизации: анализ перестановок строк матрицы планирования, случайный поиск, метод ветвей и границ [4]. В работе [4] показано, что при количестве факторов $k \leq 3$ для исследования технологических процессов, приборов и систем оптимальные планы, полученные методом ветвей и границ и методом анализа перестановок строк матрицы планирования, совпадают. Целесообразно сравнить результаты, полученные различными методами оптимизации, для объектов исследования с количеством факторов $k > 3$.

Цель исследования: провести сравнительный анализ методов оптимизации многофакторных планов эксперимента.

2. Основные результаты исследований. В работе [5] синтезированы оптимальные планы для экспериментальных исследований вихретоковых преобразователей. В исследование были включены следующие факторы: X_1 – диаметр сердечника, мм; X_2 – высота сердечника, мм; X_3 – количество витков; X_4 – частота питающего напряжения, Гц. По убыванию стоимости реализации эти планы можно ранжировать следующим образом: план, полученный методом анализа перестановок строк матрицы планирования (112,85 усл. ед.); план, полученный методом случайного поиска (101,8 усл. ед.); план, полученный методом итерационного планирования эксперимента (86,3 усл. ед.).

При оптимизации исходного плана эксперимента методом ветвей и границ получен оптимальный план (табл.1), стоимость реализации которого составляет 53,05 усл. ед.

Таблица 1

Оптимальные планы эксперимента для исследования вихретоковых преобразователей

| Метод итерационного планирования эксперимента (анализ перестановок) | | | | | Метод ветвей и границ | | | | |
|---|----------------------|-------|-------|-------|-----------------------|----------------------|-------|-------|-------|
| Номер опыта | Обозначение факторов | | | | Номер опыта | Обозначение факторов | | | |
| | X_1 | X_2 | X_3 | X_4 | | X_1 | X_2 | X_3 | X_4 |
| 4 | +1 | +1 | +1 | +1 | 4 | +1 | +1 | +1 | +1 |
| 5 | +1 | +1 | -1 | +1 | 5 | +1 | +1 | -1 | +1 |
| 2 | -1 | +1 | -1 | -1 | 9 | +1 | +1 | -1 | -1 |
| 8 | -1 | +1 | +1 | -1 | 12 | +1 | +1 | +1 | -1 |
| 7 | -1 | -1 | +1 | +1 | 14 | -1 | +1 | +1 | -1 |
| 3 | -1 | -1 | -1 | +1 | 8 | -1 | +1 | +1 | -1 |
| 1 | +1 | -1 | -1 | -1 | 13 | -1 | +1 | -1 | +1 |
| 6 | +1 | -1 | +1 | -1 | 2 | -1 | +1 | -1 | -1 |
| 11 | +1 | -1 | +1 | +1 | 3 | -1 | -1 | -1 | +1 |
| 10 | +1 | -1 | -1 | +1 | 15 | -1 | -1 | -1 | -1 |
| 9 | +1 | +1 | -1 | -1 | 7 | -1 | -1 | +1 | +1 |
| 12 | +1 | +1 | +1 | -1 | 16 | -1 | -1 | +1 | -1 |
| 14 | -1 | +1 | +1 | +1 | 11 | +1 | -1 | +1 | +1 |
| 13 | -1 | +1 | -1 | +1 | 6 | +1 | -1 | +1 | -1 |
| 15 | -1 | -1 | -1 | -1 | 10 | +1 | -1 | -1 | +1 |
| 16 | -1 | -1 | +1 | -1 | 1 | +1 | -1 | -1 | -1 |

При исследовании технологического процесса меднения печатных плат в качестве доминирующих факторов были выбраны: X_1 – концентрация CuSO_4 в растворе электролита, г/л; X_2 – концентрация H_2SO_4 в растворе электролита, г/л; X_3 – плотность тока в гальванической ванне, А/дм^2 ; X_4 – время обработки плат в гальванической ванне, мин. В работе [6] синтезированы оптимальные планы для исследования данного технологического процесса, которые получены с использованием следующих методов: анализ перестановок строк матрицы планирования (стоимость оптимального плана 86,62 усл. ед., проанализирован 102741 вариант); случайный поиск (стоимость оптимального плана 70,15 усл. ед., проанализирован 42601 вариант). При оптимизации исходного плана методом ветвей и границ получен оптимальный план (табл.2), стоимость реализации которого составляет 43,84 усл. ед.

При исследовании устройства для контроля качества диэлектрических материалов доминирующими факторами, влияющими на энергопотребление устройства, были выбраны: X_1 – напряжение питания, $U_{\text{пит}}$; X_2 – значение начальной емкости эталонного конденсатора в плечах мультивибратора, C_0 ; X_3 – число витков основной обмотки магниточувствительного преобразователя, W ; X_4 – число витков дополнительной мультивибраторной обмотки, W_1 (обмотка 3 или 4). При исследовании точности устройства в качестве фактора X_2 выбрано значение сопротивления, подсоединенного к выходу операционного усилителя. Факторы X_1 , X_3 и X_4 остаются такими же как и при исследовании энергопотребления устройства.

Таблица 2

Оптимальные планы эксперимента для исследования технологического процесса меднения печатных плат

| Метод случайного поиска | | | | | Метод ветвей и границ | | | | |
|-------------------------|----------------------|----------------|----------------|----------------|-----------------------|----------------------|----------------|----------------|----------------|
| Номер опыта | Обозначение факторов | | | | Номер опыта | Обозначение факторов | | | |
| | X ₁ | X ₂ | X ₃ | X ₄ | | X ₁ | X ₂ | X ₃ | X ₄ |
| 15 | -1 | -1 | -1 | -1 | 16 | -1 | -1 | +1 | -1 |
| 7 | -1 | -1 | +1 | +1 | 15 | -1 | -1 | -1 | -1 |
| 14 | -1 | +1 | +1 | +1 | 3 | -1 | -1 | -1 | +1 |
| 8 | -1 | +1 | +1 | -1 | 7 | -1 | -1 | +1 | +1 |
| 2 | -1 | +1 | -1 | -1 | 13 | -1 | +1 | -1 | +1 |
| 13 | -1 | +1 | -1 | +1 | 14 | -1 | +1 | +1 | +1 |
| 11 | +1 | -1 | +1 | +1 | 8 | -1 | +1 | +1 | -1 |
| 6 | +1 | -1 | +1 | -1 | 2 | -1 | +1 | -1 | -1 |
| 10 | +1 | -1 | -1 | +1 | 9 | +1 | +1 | -1 | -1 |
| 9 | +1 | +1 | -1 | -1 | 12 | +1 | +1 | +1 | -1 |
| 12 | +1 | +1 | +1 | -1 | 4 | +1 | +1 | +1 | +1 |
| 5 | +1 | +1 | -1 | +1 | 5 | +1 | +1 | -1 | +1 |
| 4 | +1 | +1 | +1 | +1 | 10 | +1 | -1 | -1 | +1 |
| 1 | +1 | -1 | -1 | -1 | 11 | +1 | -1 | +1 | +1 |
| 3 | -1 | -1 | -1 | +1 | 6 | +1 | -1 | +1 | -1 |
| 16 | -1 | -1 | +1 | -1 | 1 | +1 | -1 | -1 | -1 |

В работе [8] синтезированы оптимальные планы для исследования энергопотребления и точности данных устройств, которые получены с использованием следующих методов: анализ перестановок строк матрицы планирования (стоимость оптимального плана для исследования энергопотребления – 24,9 усл. ед., проанализирован 131501 вариант; стоимость оптимального плана для исследования точности – 24,7 усл. ед., проанализировано 7777777 вариантов); случайный поиск (стоимость оптимального плана для исследования энергопотребления – 30,0 усл. ед., проанализировано 7143854 варианта; стоимость оптимального плана для исследования точности 29,4 усл. ед., проанализировано 22654466 вариантов).

Использование метода итерационного планирования эксперимента дало оптимальные планы эксперимента, реализация которых составляет 40,4 усл. ед.

При оптимизации исходного плана эксперимента методом ветвей и границ получены оптимальные планы: для исследования энергопотребления (табл.3) стоимостью реализации 20,8 усл. ед.; для исследования точности (табл.4) стоимостью реализации 20,8 усл. ед.

Выводы. На примерах исследования технологических процессов, приборов и систем доказана эффективность оптимального по стоимостным затратам планирования эксперимента. Показано, что для исследования объектов с количеством факторов $k > 3$ оптимальные планы эксперимента целесообразно получать методом ветвей и границ, а при количестве факторов $k \leq 3$ оптимальные планы, полученные методом ветвей и границ и методом анализа перестановок строк матрицы планирования, совпадают.

Список литературы: 1. Кошевой Н.Д. Метод итерационного планирования оптимальных по стоимостным и временным затратам экспериментов / Н.Д. Кошевой, Е.М. Костенко // Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. К.: ВКНУ, 2009. Вип. №19. С.44-48. 2. Кошевой Н.Д. Оптимальное планирование эксперимента для исследования динамических объектов / Н.Д. Кошевой, Е.М. Костенко // Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. К.: ВКНУ, 2009. Вип. №20. С.57-62. 3. Кошевой Н.Д. Метод оптимального по стоимостным и временным затратам последовательного планирования эксперимента / Н.Д. Кошевой, Е.М. Костенко, В.А. Дергачев // Автоматизация: проблемы, поиски, решения. Материалы межд. научн.-техн. конф., 7-12 сентября 2009. Севастополь: изд. Сев. НГУ, 2009. С.21-23. 4. Кошевой Н.Д. Применение метода ветвей и границ для оптимизации многофакторных экспериментов / Н.Д. Кошевой, О.Л. Бурлеев, Е.М. Костенко // Радіоелектронні і комп'ютерні системи. 2010. №1(42). С.67–70. 5. Кошевой Н.Д. Оптимальное планирование экспериментальных исследований вихретоковых преобразователей / Н.Д. Кошевой, В.А. Дергачев, Е.М. Костенко, М.В. Цеховской //

Открытые информационные и компьютерные интегрированные технологии: сб. научн. тр. Нац. аэрокосм. ун-та им. Н.Е. Жуковского «ХАИ». Харьков, 2009. Вып. 42. С.147-151. 6. *Koshevoy N.D.* Appraisal of optimum efficiency by cost expenses of the experiment's planning / N.D. Koshevoy, V.A. Dergachev, E.M. Kostenko // Вісник Черкаського державного технологічного університету. 2009. Спецвипуск. С.132-134. 7. *Заболотный А.В.* Разработка, исследование и оптимизация устройства для контроля качества диэлектрических покрытий / А.В. Заболотный, Н.Д. Кошевой // Приборы и системы. Управление, контроль, диагностика. 2004. №1. С.39-42. 8. *Кошевой Н.Д.* Оптимальное планирование эксперимента при исследовании устройства для контроля качества диэлектрических материалов / Н.Д. Кошевой, Е.М. Костенко, А.В. Заболотный // АСУ и приборы автоматики. 2009. Вып. 147. С. 38-41.

Таблица 3

Оптимальные планы для исследования энергопотребления устройств для контроля качества диэлектрических материалов

| Анализ перестановок | | | | | Метод ветвей и границ | | | | |
|---------------------|----------------------|----------------|----------------|----------------|-----------------------|----------------------|----------------|----------------|----------------|
| Номер опыта | Обозначение факторов | | | | Номер опыта | Обозначение факторов | | | |
| | X ₁ | X ₂ | X ₃ | X ₄ | | X ₁ | X ₂ | X ₃ | X ₄ |
| 2 | -1 | +1 | +1 | +1 | 2 | -1 | +1 | +1 | +1 |
| 6 | -1 | +1 | -1 | +1 | 4 | +1 | +1 | +1 | +1 |
| 5 | -1 | -1 | -1 | +1 | 3 | +1 | -1 | +1 | +1 |
| 7 | +1 | -1 | -1 | +1 | 1 | -1 | -1 | +1 | +1 |
| 8 | +1 | +1 | -1 | +1 | 5 | -1 | -1 | -1 | +1 |
| 4 | +1 | +1 | +1 | +1 | 7 | +1 | -1 | -1 | +1 |
| 3 | +1 | -1 | +1 | +1 | 8 | +1 | +1 | -1 | +1 |
| 1 | -1 | -1 | +1 | +1 | 6 | -1 | +1 | -1 | +1 |
| 9 | -1 | -1 | +1 | -1 | 14 | -1 | +1 | -1 | -1 |
| 10 | -1 | +1 | +1 | -1 | 16 | +1 | +1 | -1 | -1 |
| 11 | +1 | -1 | +1 | -1 | 15 | +1 | -1 | -1 | -1 |
| 12 | +1 | +1 | +1 | -1 | 13 | -1 | -1 | -1 | -1 |
| 13 | -1 | -1 | -1 | -1 | 9 | -1 | -1 | +1 | -1 |
| 14 | -1 | +1 | -1 | -1 | 11 | +1 | -1 | +1 | -1 |
| 15 | +1 | -1 | -1 | -1 | 12 | +1 | +1 | +1 | -1 |
| 16 | +1 | +1 | -1 | -1 | 10 | -1 | +1 | +1 | -1 |

Таблица 4

Оптимальные планы для исследования точности устройств для контроля качества диэлектрических материалов

| Анализ перестановок | | | | | Метод ветвей и границ | | | | |
|---------------------|----------------------|----------------|----------------|----------------|-----------------------|----------------------|----------------|----------------|----------------|
| Номер опыта | Обозначение факторов | | | | Номер опыта | Обозначение факторов | | | |
| | X ₁ | X ₂ | X ₃ | X ₄ | | X ₁ | X ₂ | X ₃ | X ₄ |
| 3 | +1 | -1 | +1 | +1 | 3 | +1 | -1 | +1 | +1 |
| 7 | +1 | -1 | -1 | +1 | 1 | -1 | -1 | +1 | +1 |
| 5 | -1 | -1 | -1 | +1 | 2 | -1 | +1 | +1 | +1 |
| 6 | -1 | +1 | -1 | +1 | 4 | +1 | +1 | +1 | +1 |
| 8 | +1 | +1 | -1 | +1 | 8 | +1 | +1 | -1 | +1 |
| 4 | +1 | +1 | +1 | +1 | 6 | -1 | +1 | -1 | +1 |
| 2 | -1 | +1 | +1 | +1 | 5 | -1 | -1 | -1 | +1 |
| 1 | -1 | -1 | +1 | +1 | 7 | +1 | -1 | -1 | +1 |
| 9 | -1 | -1 | +1 | -1 | 15 | +1 | -1 | -1 | -1 |
| 10 | -1 | +1 | +1 | -1 | 13 | -1 | -1 | -1 | -1 |
| 11 | +1 | -1 | +1 | -1 | 14 | -1 | +1 | -1 | -1 |
| 12 | +1 | +1 | +1 | -1 | 16 | +1 | +1 | -1 | -1 |
| 13 | -1 | -1 | -1 | -1 | 12 | +1 | +1 | +1 | -1 |
| 14 | -1 | +1 | -1 | -1 | 10 | -1 | +1 | +1 | -1 |
| 15 | +1 | -1 | -1 | -1 | 9 | -1 | -1 | +1 | -1 |
| 16 | +1 | +1 | -1 | -1 | 11 | +1 | -1 | +1 | -1 |

Поступила в редколлегию 01.03.2010

Кошевой Николай Дмитриевич, д-р техн. наук, профессор, зав. кафедрой, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ». Научные интересы: автоматизация экспериментальных исследований, проектирование измерительных преобразователей, автоматизация производственных процессов. Адрес: Украина, 61070, Харьков, ул. Чкалова, 17, тел. 057-707-43-03.

Бурлеев Олег Леонидович, магистр, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ». Научные интересы: автоматизация экспериментальных исследований, проектирование измерительных преобразователей. Адрес: Украина, 61070, Харьков, ул. Чкалова, 17, тел. 057-707-43-03.

Костенко Елена Михайловна, канд. техн. наук, доцент, проректор по учебно-педагогической и инновационной работе, Полтавская государственная аграрная академия. Научные интересы: автоматизация экспериментальных исследований, охрана труда. Адрес: Украина, 36003, Полтава, ул. Сковороды, 1/3, тел. 05322-7-36-93.

УДК 681. 325 (088. 8)

Л.В. ЛАРЧЕНКО, В.Г. ЛОБОДА

ЦИФРОВОЙ ФУНКЦИОНАЛЬНЫЙ ПРЕОБРАЗОВАТЕЛЬ ДВУХ ЧАСТОТ

В настоящее время преобразование двух входных частот в выходную, равную корню квадратному из суммы их квадратов, осуществляется в преобразователях последовательно во времени посредством выполнения четырех арифметических операций. В предлагаемом устройстве это преобразование осуществляется за время выполнения трех из них при более простой технической реализации.

1. Введение

В [1] предложен метод непрерывного формирования приращений ступенчатых функций с числоимпульсной формой представления аргумента, а в [2] – его техническая реализация на примере специализированного цифрового вычислителя для извлечения корня квадратного из суммы квадратов. Его достоинством является возможность воспроизведения и вычисления значений функций в реальном масштабе времени в темпе формирования кода аргумента при более простой технической реализации. Там же названы некоторые цифровые измерительные приборы, в которых рассмотренный вычислитель фиксирует результат измерения соответствующей физической величины в цифровом коде. Однако выполняемая им функция не ограничивает названную в [2] область его применения. Известны, например, цифровые преобразователи частотно-импульсных сигналов, в которых реализуемая устройством функция выполняется не над кодами входных чисел, а над их частотами. В качестве примера можно назвать преобразователи для формирования выходной частоты f , определяемой выражением

$$f = \sqrt{f_1^2 + f_2^2}, \quad (1)$$

где f_1 и f_2 – входные частоты преобразователя.

Такие устройства широко применяются в векторных анализаторах с частотными входами и выходом, при частотном представлении модуля рассогласования двух процессов, а также при построении преобразователей форм представления информации с промежуточным время-импульсным преобразованием для специализированных устройств обработки информации, цифровых контрольно-измерительных приборах, в системах автоматического управления и регулирования, работающих в комплексе с ЭВМ, повышая быстродействие всей системы.

Анализ функции (1) показывает, что в таких преобразователях в качестве основного узла может быть использована модификация предложенного в [2] вычислителя. Полученное при этом техническое решение функционального преобразователя частоты по сравнению с известными устройствами аналогичного назначения [3-5] проще в реализации и обеспечивает минимально возможное время преобразования частот f_1 и f_2 входных сигналов в выходную частоту f следования импульсов.

Поступила в редколлегию 01.03.2010

Кошевой Николай Дмитриевич, д-р техн. наук, профессор, зав. кафедрой, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ». Научные интересы: автоматизация экспериментальных исследований, проектирование измерительных преобразователей, автоматизация производственных процессов. Адрес: Украина, 61070, Харьков, ул. Чкалова, 17, тел. 057-707-43-03.

Бурлеев Олег Леонидович, магистр, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ». Научные интересы: автоматизация экспериментальных исследований, проектирование измерительных преобразователей. Адрес: Украина, 61070, Харьков, ул. Чкалова, 17, тел. 057-707-43-03.

Костенко Елена Михайловна, канд. техн. наук, доцент, проректор по учебно-педагогической и инновационной работе, Полтавская государственная аграрная академия. Научные интересы: автоматизация экспериментальных исследований, охрана труда. Адрес: Украина, 36003, Полтава, ул. Сковороды, 1/3, тел. 05322-7-36-93.

УДК 681. 325 (088. 8)

Л.В. ЛАРЧЕНКО, В.Г. ЛОБОДА

ЦИФРОВОЙ ФУНКЦИОНАЛЬНЫЙ ПРЕОБРАЗОВАТЕЛЬ ДВУХ ЧАСТОТ

В настоящее время преобразование двух входных частот в выходную, равную корню квадратному из суммы их квадратов, осуществляется в преобразователях последовательно во времени посредством выполнения четырех арифметических операций. В предлагаемом устройстве это преобразование осуществляется за время выполнения трех из них при более простой технической реализации.

1. Введение

В [1] предложен метод непрерывного формирования приращений ступенчатых функций с числоимпульсной формой представления аргумента, а в [2] – его техническая реализация на примере специализированного цифрового вычислителя для извлечения корня квадратного из суммы квадратов. Его достоинством является возможность воспроизведения и вычисления значений функций в реальном масштабе времени в темпе формирования кода аргумента при более простой технической реализации. Там же названы некоторые цифровые измерительные приборы, в которых рассмотренный вычислитель фиксирует результат измерения соответствующей физической величины в цифровом коде. Однако выполняемая им функция не ограничивает названную в [2] область его применения. Известны, например, цифровые преобразователи частотно-импульсных сигналов, в которых реализуемая устройством функция выполняется не над кодами входных чисел, а над их частотами. В качестве примера можно назвать преобразователи для формирования выходной частоты f , определяемой выражением

$$f = \sqrt{f_1^2 + f_2^2}, \quad (1)$$

где f_1 и f_2 – входные частоты преобразователя.

Такие устройства широко применяются в векторных анализаторах с частотными входами и выходом, при частотном представлении модуля рассогласования двух процессов, а также при построении преобразователей форм представления информации с промежуточным время-импульсным преобразованием для специализированных устройств обработки информации, цифровых контрольно-измерительных приборах, в системах автоматического управления и регулирования, работающих в комплексе с ЭВМ, повышая быстродействие всей системы.

Анализ функции (1) показывает, что в таких преобразователях в качестве основного узла может быть использована модификация предложенного в [2] вычислителя. Полученное при этом техническое решение функционального преобразователя частоты по сравнению с известными устройствами аналогичного назначения [3-5] проще в реализации и обеспечивает минимально возможное время преобразования частот f_1 и f_2 входных сигналов в выходную частоту f следования импульсов.

2. Математическое обоснование структуры преобразователя частоты

При синтезе функционального преобразователя частот, реализующего выражение (1), прежде всего нужно учесть диапазон возможных значений f_1 и f_2 .

Преобразование высоких входных частот в выходную осуществляют путем их промежуточного преобразования в код с помощью подсчета числа импульсов этих частот за фиксированный интервал времени, извлечения квадратного корня из суммы их квадратов с последующим преобразованием полученного значения в частоту следования импульсов. В этом случае описанный в [2] вычислитель должен оперировать с числами N_1 и N_2 , пропорциональными частотам f_1 и f_2 соответственно.

Если f_1 и f_2 лежат в области низких и инфранизких частот, преобразование частот f_1 и f_2 в N_1 и N_2 усложняется, так как в этом случае на первом этапе периоды $T_1 = \frac{1}{f_1}$ и $T_2 = \frac{1}{f_2}$ входных сигналов преобразуют в числовые эквиваленты $N_{T_1} = f_0 T_1$ и $N_{T_2} = f_0 T_2$

путем заполнения их импульсами высокой частоты f_0 , на втором – получают квадраты $N_{T_1}^2$ и $N_{T_2}^2$ этих чисел, на третьем – вычисляют значения $N_1 = \left[\frac{k^2 f_0^2}{N_{T_1}^2} \right]$, $N_2 = \left[\frac{k^2 f_0^2}{N_{T_2}^2} \right]$, где k – коэффициент пропорциональности, и на четвертом – извлекают корень квадратный из их суммы.

В результате получают число

$$N = \left[\sqrt{\left[\frac{k^2 f_0^2}{N_1^2} \right] + \left[\frac{k^2 f_0^2}{N_2^2} \right]} \right], \quad (2)$$

которое преобразуют в выходную частоту f .

В связи с тем, что известные функциональные преобразователи частоты, реализующие выражение (2), имеют значительно более сложную структуру, остается актуальной задача их совершенствования как с точки зрения упрощения технической реализации, так и с точки зрения уменьшения времени функционального преобразования входных частот f_1 и f_2 в числовой эквивалент N .

Анализ исходного выражения (1) показывает, что улучшить названные характеристики преобразователя можно путем реализации (1), представленного в виде, отличном от (2), исключив операции деления в подкоренном выражении.

Для получения такого вида преобразуем исходное выражение (1) следующим образом. Перепишем его в виде

$$f = \sqrt{\frac{1}{T_1^2} + \frac{1}{T_2^2}}.$$

С учетом частоты заполнения f_0 эталонного генератора оно преобразуется к виду

$$f = \frac{f_0}{f_0} \sqrt{\frac{1}{T_1^2} + \frac{1}{T_2^2}} = f_0 \frac{\sqrt{N_1^2 + N_2^2}}{N_1 N_2}, \quad (3)$$

где $N_1 = f_0 T_1$ и $N_2 = f_0 T_2$ – коды периодов T_1 и T_2 соответственно.

Из последнего выражения следует, что выходная частота f функционального преобразователя может быть получена путем умножения эталонной f_0 в дробное число раз. При этом числитель дроби может быть реализован предложенным в [2] специализированным вычислителем. А так как на практике $N_1^2 N_2^2 \gg N_1^2 + N_2^2$, уже при $N_1 = N_2 = 2$ имеет место $f < f_0$. Следовательно, числоимпульсная последовательность частоты f может

быть сформирована из числоимпульсной последовательности частоты f_0 путем выборки определенных ее импульсов.

Так как числа N_1 и N_2 в выражении (3) целые, вычисление произведения $N_1 N_2$ и квадратов N_{T1}^2 и N_{T2}^2 может быть выполнено в цифровом преобразователе без погрешности. Погрешность же вычисления корня квадратного может быть обеспечена не хуже $|\delta_{\max}| = 0,5$. В этом случае подлежащее реализации в преобразователе математическое выражение принимает вид

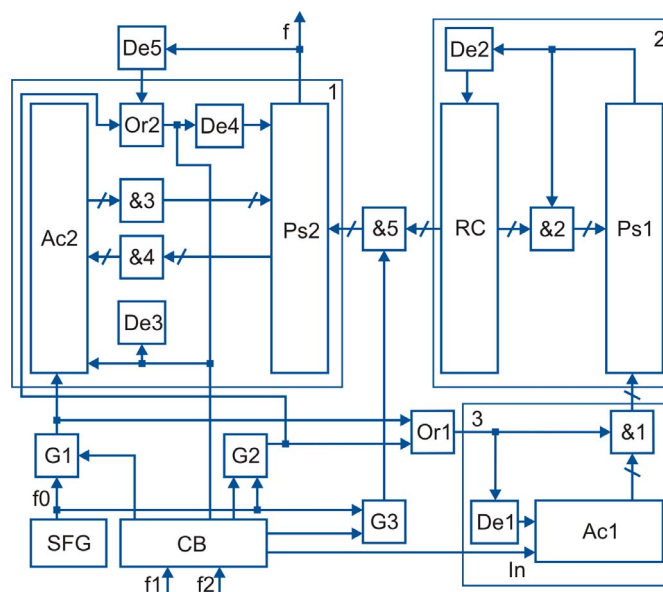
$$f = f_0 \frac{[\sqrt{N_1^2 + N_2^2} + 0,5]}{N_1 N_2}. \quad (4)$$

Целью разработки является уменьшение времени функционального преобразования двух входных частот преобразователя в выходную при более простой его технической реализации.

Для достижения поставленной цели решается задача – реализовать в техническом устройстве выражение (4) на базе известных унифицированных блоков посредством введения новых связей между ними.

3. Структурная схема функционального преобразователя двух частот

На рисунке приведена структурная схема преобразователя частоты, который реализует выражение (4).



Структурная схема функционального преобразователя двух частот

Устройство содержит блок управления СВ, генератор эталонной частоты SFG, делитель частоты (ДЧ) 1, устройство для извлечения корня квадратного (УИК) 2, квадрататор 3, управляемые вентили G1, G2, G3, логический элемент Or1 и группу элементов &5.

В состав квадрататора входят суммирующий счетчик Ac1, группа элементов &1 и элемент задержки De1. Выход De1 соединен со счетным входом второго триггера счетчика со стороны младших разрядов. Следовательно, при появлении импульса на выходе элемента De1 в счетчик записывается число 2.

В состав УИК входят параллельный сумматор Ps1, счетчик результата RC, группа элементов &2 и элемент задержки De2.

При подаче с блока управления СВ одиночного импульса на шину In счетчика квадрататора его триггеры устанавливаются в исходное состояние.

Группа элементов &1 квадрататора поразрядно соединяет выходы триггеров суммирующего счетчика со счетными входами триггеров сумматора Ps1 УИК. При наличии импуль-

са на выходе элемента Or1 посредством группы &1 прямой код числа, записанного в счетчике квадратора, заносится в сумматор Ps, где суммируется с его содержимым.

В УИК при наличии выходного импульса сумматора Ps1 посредством группы &2 в сумматор заносится число, хранящееся в счетчике результата RC, в обратном коде со сдвигом на один разряд в сторону старших разрядов. Следовательно, если в счетчике RC записано число n , в сумматор Ps1 заносится число $2^i - 2(n+1)$, где i – число триггеров сумматора Ps1.

В состав делителя частоты 1 входят суммирующий счетчик As2, параллельный сумматор Ps2, две группы элементов &3, &4, поразрядно соединяющие триггеры счетчика и сумматора по счетным входам, элемент Or2 и элементы задержки De3, De4, De5.

В исходном состоянии с блока управления СВ на вентили G1, G2, G3 поданы запирающие потенциалы. В счетчике As1 квадратора записано число 1, в сумматоре Ps1 – число $2^i - 1$, а показания счетчика результата RC, сумматора Ps2 и счетчика As2 – нулевые.

На первом этапе работы устройства, предшествующему началу генерирования импульсов частоты f , посредством блока управления СВ и управляемых им вентилях G1, G2, G3 осуществляется запись в счетчик результата RC УИК, счетчик As2 и сумматор Ps2 чисел $[\sqrt{N_1^2 + N_2^2} + 0,5]$ и $2^k - N_1 N_2$ соответственно для последующего преобразования частоты f_0 эталонного генератора SFG в выходную f . Здесь k – число триггеров счетчика As2 и сумматора Ps2.

Функционирование преобразователя на этом этапе осуществляется следующим образом.

Блок управления СВ открывает клапан G1 на период $T_1 = \frac{1}{f_1}$ первого входного сигнала частоты f_1 . За это время в счетчик As2 ДЧ поступит число импульсов N_1 частоты f_0 эталонного генератора, равное $N_1 = f_0 T_1$. Это же число импульсов поступит на вход квадратора через элемент Or1.

При этом каждый импульс пачки N_1 заносит в сумматор Ps1 число, хранящееся в счетчике As1, и увеличивает содержимое счетчика As1 на два. В результате со стороны счетчика As1 квадратора в сумматор Ps1 будут последовательно вводиться числа 1, 3, 5, ..., а по цепи обратной связи со стороны счетчика результата RC в сумматор Ps1 – числа $2^i - 2$, $2^i - 4$, $2^i - 6$... При этом каждому входному импульсу квадратора будет соответствовать только один импульс переполнения сумматора Ps1. Покажем это.

С поступлением первого импульса пачки N_1 на вход квадратора в сумматоре Ps1 число $2^i - 1$ суммируется с числом 1. В результате на выходе сумматора Ps1 появится первый импульс переполнения, а его показания станут равными нулю. Импульс переполнения по цепи обратной связи со стороны счетчика результата RC вводит в сумматор Ps1 число $2^i - 2$ и записывает в счетчик результата RC УИК число 1.

С приходом второго импульса пачки N_1 на вход квадратора в сумматоре Ps1 суммируются числа $2^i - 2$ и 3. В результате на его выходе появится второй импульс переполнения, а в нем останется записанным число 1 в прямом коде. Аналогично предыдущему импульс его переполнения посредством обратной связи записывает в сумматор Ps1 число $2^i - 4$ и поступает на вход счетчика результата RC. В результате показания сумматора Ps1 станут равными $2^i - 3$, а счетчика результата RC – равными двум.

С приходом третьего импульса показания сумматора Ps1 станут $2^i - 4$, а счетчика результата RC УИК – равными трем.

При поступлении на вход квадратора очередных импульсов пачки N_1 описанные процессы в этой части схемы будут циклически повторяться. Так как после каждого очередного переполнения сумматора Ps1 в нем остается записанным в прямом коде число меньше заносимого по цепи обратной связи со стороны счетчика результата RC в дополнительном коде, каждому входному импульсу квадратора соответствует один и только один

импульс переполнения сумматора Ps1. Следовательно, по окончании ввода в квадратор числа импульсов N_1 в счетчике результата RC будет также зафиксировано число N_1 , что не противоречит равенству $\sqrt{N_1^2} = N_1$.

По окончании периода T_1 первого входного сигнала с блока управления СВ на шину In счетчика As1 квадратора подается единичный импульс, который устанавливает его триггеры в исходное состояние.

Одновременно блок управления СВ открывает клапан G2 на период $T_2 = \frac{1}{f_2}$ второго входного сигнала. За это время через клапан на квадратор и ДЧ поступит $N_2 = f_0 T_2$ импульсов частоты f_0 эталонного генератора SFG. Проанализировав работу квадратора и УИК при обработке второй пачки импульсов N_2 , можно убедиться в том, что по окончании ввода ее в квадратор в счетчике результата RC будет записано число $[\sqrt{N_1^2 + N_2^2} + 0,5]$ в прямом коде.

Остановимся теперь на процессах, которые протекают в ДЧ в течение времени T_2 .

На этом промежутке каждый импульс пачки N_2 , прошедший через элемент &2, посредством группы элементов &3 со стороны счетчика As2 заносит в сумматор Ps2 число N_1 . Следовательно, по окончании периода T_2 второго входного сигнала в сумматоре As2 будет записано число $N_1 N_2$.

В это же время блок управления CD формирует единичный импульс, который выполняет следующие операции: устанавливает триггеры счетчика As2 в нулевое состояние; пройдя через элемент задержки De3, посредством группы &4 заносит в счетчик As2 дополнительный код числа $N_1 N_2$, записанного в сумматоре Ps2 (в результате его показания станут равными $2^k - N_1 N_2$, где k – число триггеров счетчика и сумматора); пройдя через элемент задержки De4, устанавливает триггеры сумматора Ps2 в нулевое состояние; пройдя через элемент Or2, посредством группы &3 записывает в сумматор Ps2 число $2^k - N_1 N_2$.

Таким образом, по окончании периода второго входного сигнала как в счетчике As2, так и в сумматоре Ps2 делителя частоты будет записано число $2^k - N_1 N_2$.

Условия правильной и надежной работы в этой части схемы будут обеспечены, если время задержки элемента De4 будет больше, чем De3.

На втором этапе работы устройства частота f_0 эталонного генератора SFG преобразуется в выходную f следующим образом.

По окончании периода второго входного сигнала блок управления СВ открывает клапан G3. С этого момента импульсы частоты f_0 эталонного генератора SFG начинают поступать на группу &5. При этом каждый импульс данной последовательности посредством группы &5 число $[\sqrt{N_1^2 + N_2^2} + 0,5]$, записанное в счетчике RC, заносит в сумматор Ps2. Так как $[\sqrt{N_1^2 + N_2^2} + 0,5] < N_1 N_2$, первое переполнение сумматора Ps2 произойдет при поступлении на вход группы &5 импульса с номером n , при котором $n [\sqrt{N_1^2 + N_2^2} + 0,5] \geq N_1 N_2$. В этот момент на выходе сумматора Ps2 появится первый импульс переполнения, а в нем останется записанным число $\Delta_1 = n [\sqrt{N_1^2 + N_2^2} + 0,5] - N_1 N_2$. Импульс переполнения сумматора Ps2 поступает на выход преобразователя и в цепь обратной связи. Пройдя через элементы De5 и Or2, он посредством группы &3 вновь записывает в сумматор Ps2 число $2^k - N_1 N_2$. В результате показания сумматора Ps2 станут равными $2^k - N_1 N_2 + \Delta_1$. Так как $N_1 N_2 > \Delta_1$, импульс переполнения сумматора Ps2 будет отсутствовать.

В дальнейшем при поступлении очередных импульсов частоты f_0 на вход группы &5 описанные процессы в этой части схемы будут циклически повторяться. Нетрудно видеть, что за время поступления на вход группы &5 $N_1 N_2$ импульсов на выходе сумматора Ps2 появится ровно $[\sqrt{N_1^2 + N_2^2} + 0, 5]$ импульсов переполнения. При этом временные интервалы между соседними выходными импульсами преобразователя могут принимать два значения: либо $[\frac{N_1 N_2}{\sqrt{N_1^2 + N_2^2} + 0, 5}] T_0$, либо $[\frac{N_1 N_2}{\sqrt{N_1^2 + N_2^2} + 0, 5}] T_0 + T_0$, где $T_0 = \frac{1}{f_0}$.

Из этих выражений следует, что если выполнить условия $N_1 N_2 \gg [\sqrt{N_1^2 + N_2^2} + 0, 5]$ и $T_0 \rightarrow 0$, временные интервалы между двумя выходными импульсами преобразователя будут практически равны между собой, что может быть достигнуто соответствующим выбором эталонной частоты f_0 и использованием быстродействующих сумматоров Ps1 и Ps2.

Повышение эталонной частоты SFG способствует не только более равномерному следованию выходных импульсов преобразователя, но и уменьшению погрешности преобразования временных интервалов T_1 и T_2 в числовые эквиваленты N_1 и N_2 .

Нами был рассмотрен вариант преобразователя частоты, в котором целое число $[\sqrt{N_1^2 + N_2^2} + 0, 5]$ заменяет истинное дробное $\sqrt{N_1^2 + N_2^2}$, что приводит к появлению еще одной составляющей погрешности формирования частоты f , обусловленной этим округлением. Очевидно, эту составляющую можно уменьшить за счет более точного вычисления корня квадратного, например, до одной, двух или более значащих цифр после запятой. Естественно, что повышение точности вычислений при этом, как и в случае преобразования периодов T_1 и T_2 в числовые эквиваленты N_1 и N_2 , будет достигаться увеличением разрядности счетчиков и сумматоров преобразователя.

4. Выводы

Предложен цифровой функциональный преобразователь двух входных частот в выходную частоту, равную корню квадратному из суммы их квадратов.

Научная новизна состоит в разработке алгоритма преобразования, при котором обеспечивается минимально возможное время преобразования.

Практическая значимость предложенного функционального преобразователя состоит в существенном упрощении его технической реализации.

Дальнейшие исследования будут направлены на разработку специализированных цифровых устройств для вычисления векторов в пространстве и частотным их представлением.

Список литературы: 1. Ларченко Л.В. Метод формирования приращений при функциональной обработке единичных кодов // Радиоэлектроника и информатика. 2001. № 3(16). С.61 – 63. 2. Ларченко Л.В., Хаханова А.В. Специализированный вычислитель для извлечения корня квадратного из суммы квадратов // Радиоэлектроника и информатика. 2010. №1. С.71 – 74. 3. Джексон Р. Г. Новейшие датчики. 2 – е изд., доп. Москва: Техносфера, 2008. 400с. 4. Волков В.Г. Устройство для измерения модуля рассогласования двух процессов. Описание изобретения по авторскому свидетельству № 993277, G 06 G, 7/14. Бюллетень №4, 1983. 5. Ларченко Л.В., Лобода В.Г., Черкесов А.Б. О системном проектировании ФОРУ систем управления // Изв. вузов, сер. Электромеханика. 1997. №1–2. С.110–112.

Поступила в редколлегию 15.01.2010

Ларченко Лина Викторовна, канд. техн. наук, доцент кафедры АПВТ ХНУРЭ. Научные интересы: функционально-ориентированные устройства, частотно-импульсные вычислительные устройства. Увлечения: путешествия, музыка, кино, театр. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.

Лобода Виталий Гаврилович, канд. техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: функционально-ориентированные устройства, частотно-импульсные вычислительные устройства. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.

ВЛИЯНИЕ СТЕПЕНИ ЧЕРНОТЫ ПОВЕРХНОСТИ НА ТЕПЛОВОЙ РЕЖИМ ТЕЛ ПРОСТЕЙШЕЙ ГЕОМЕТРИЧЕСКОЙ ФОРМЫ, ФУНКЦИОНИРУЮЩИХ В РАЗРЕЖЕННОЙ СРЕДЕ

Описываются экспериментальные исследования по изучению влияния степени черноты поверхности на тепловой режим тел простейшей геометрической формы, функционирующих в разреженной среде.

1. Актуальность

Постоянное усложнение радиоэлектронной аппаратуры вызвано значительным расширением областей её применения и непрерывным внедрением новых достижений в науке и технике.

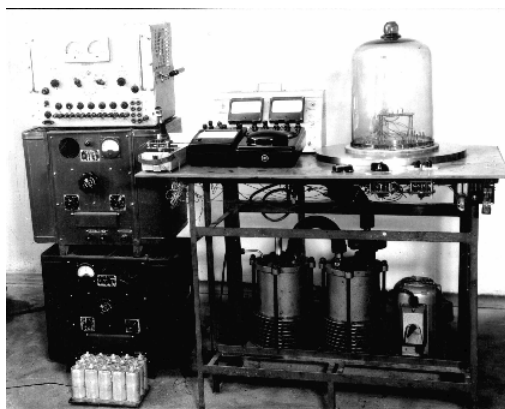
Надёжность работы радиоэлектронной аппаратуры, как известно, зависит от многих факторов: конструктивного исполнения, применения новейших разработанных элементов и узлов, новейших технологических процессов, условий эксплуатации в разных климатических зонах, влияния внешних факторов. Но одно из наибольших воздействий на стабильность параметров радиоэлектронной аппаратуры оказывает температура. Поэтому без поддержания заданного температурного режима трудно обеспечить длительную работоспособность, а также необходимую надёжность радиоэлектронного оборудования и его элементов [1-6].

При исследовании температурных полей сложных систем (каковыми являются все радиоэлектронные аппараты) приходится иметь дело с радиоэлементами, обладающими различной степенью черноты. Поэтому цель исследования – изучить влияние степени изменения черноты радиоэлементов на температурное поле функционирующего аппарата в условиях пониженного давления.

2. Постановка задачи

Путём экспериментального исследования изучить влияние степени черноты поверхности на тепловой режим тел простейшей геометрической формы (цилиндр конечных размеров, шар), функционирующих в разреженной среде. Для решения поставленной задачи была создана экспериментальная установка.

Она состоит из (рисунки):



Экспериментальная установка

1. Двухступенчатого пластинчато-роторного форвакуумного насоса типа Z - 30 производительностью 30 м³/ч до разрежения 5·10⁻⁴ мм рт. ст.
2. Вакуумной камеры.

3. Опорной плиты, через сверления в которой пропускались 44 герметичных разъёма:
- кольцевая прокладка толщиной 10 мм, изготовленная из вакуумной резины;
 - сменные стеклянные вакуумные колпаки с размерами: $D=250$ мм и $h=450$ мм; $D=450$ мм и $h=725$ мм;
 - пульт управления, на котором был смонтирован вакуумметр ВИТ – 1А- П;
 - зеркальный миллиамперметр типа МА-011/5 (ГОСТ1845 – 62);
 - зеркальный вольтметр типа М – 45 (ГОСТ 1845 – 62);
 - переключатель термодпар двухполюсный на 30 положений;
 - термометр со шкалой $-20^0 \div +20^0$;
 - зеркальный гальванометр типа ГЗП – 47 (ГОСТ 1845 – 65).

Вакуум измерялся термодпарной манометрической лампой типа ЛТ – 2 (ТУ – СУЗ 390 000 ТУ), входящей в комплект прибора ВИТ – А – П.

Опытное тело было выполнено в виде шара – электрокалориметра. Температурное поле поверхности шара по сравнению с поверхностью цилиндра имеет равномерный характер распределения температурного поля.

Неравномерность температурного поля шара составляла $\pm 3 \div 6$ °С. Опытный шар, изготовленный из латуни, имел диаметр 40,25 мм. Для размещения нагревательного элемента было предусмотрено специальное цилиндрическое отверстие в теле шара. Нагреватель помещался внутри шара, отверстия закрывались крышками, выполненными из латуни. Степень черноты составляла $\varepsilon \approx 0,22$. Для установления влияния степени черноты на тепловой режим шара в вакууме поверхность его затем была покрыта высокотемпературным блестящим чёрным лаком с известной степенью черноты $\varepsilon = 0,8$.

Питание электрокалориметра осуществлялось постоянным током от селенового выпрямителя ВСА – 6 М.

Исследование теплоотдачи электрокалориметра производилось методом стационарного режима. Температура электрокалориметра измерялась медно-константановыми термопарами диаметром 0,1 мм, длиной 1 м, которые были приклеены на поверхности клеем БФ – 2. На поверхности шара было размещено равномерно 8 термодпар. Для уменьшения потерь тепла через электроды термодпар последние были изготовлены из медной и константановой проволоки, покрытой лаком и шелковой изоляцией, термодпары подсоединялись при помощи многопозиционного переключателя к зеркальному гальванометру ГЗП – 47. Перед экспериментами была произведена градуировка термодпар. При дополнительном внешнем сопротивлении 240 Ом одно деление гальванометра составляло 0,58 градуса.

Потребляемая электрокалориметром мощность определялась по методу вольтметра – амперметра. Ток измерялся амперметром М253, напряжение – вольтметром В1500/5. Потребляемая мощность для каждого калориметра (цилиндра, шара) была постоянной на протяжении всех экспериментов.

Температура оболочки, т.е. стеклянного колпака, определялась с помощью двух термодпар, приклеенных клеем БФ-2 на внутренней поверхности. Температура окружающего воздуха определялась по ртутному термометру с ценой деления 0,5 °С ГОСТ- 215-17.

Степень разрежения под колпаком измерялась вакуумметром ВИТ-1А 2 с помощью температурной манометрической лампы типа ЛТ-2.

Методика проведения экспериментальных исследований. Исследование влияния степени черноты поверхности на тепловой режим в разреженной среде было произведено на шаре – электрокалориметре. Степень черноты поверхности составляла $\varepsilon \approx 0,22$ и $\varepsilon \approx 0,8$.

Среднеповерхностная температура шара была определена по среднеарифметическому значению.

Потребляемая мощность опытным электрокалориметром составляла ($P = 2,42$ Вт).

Мощность, рассеиваемая шаром, во всех экспериментах составляла $P_{ш} = 2,2$ Вт:

$$P_{ш} = P - P_{п}, \quad (1)$$

где P – потребляемая мощность, Вт; $P_{п}$ – мощность, рассеиваемая токопроводящими проводами, Вт.

Мощность, рассеиваемая токопроводящими проводами, вычислялась по выражению:

$$P_{\text{п}} = \sqrt{\alpha_{\text{п}} \cdot \lambda_{\text{п}} S_{\text{п}} F_{\text{п}}} \cdot (t_{\text{п}} - t_{\text{ф}}) \cdot t_{\text{ф}} \cdot y \cdot \ell_{\text{пт}}, \quad (2)$$

где $\alpha_{\text{п}}$ – коэффициент теплоотдачи токопроводящих проводов, Вт/м²·град.; λ – коэффициент теплопроводности, Вт/м·град; $S_{\text{п}}$ – длина параметра сечения провода, м; $F_{\text{п}}$ – площадь сечения провода, м²; $t_{\text{п}}$ – средняя температура провода, °С; $t_{\text{ф}}$ – температура окружающей среды, °С; $y = \sqrt{\alpha_{\text{п}} \cdot S_{\text{п}} / \lambda_{\text{п}} \cdot F_{\text{п}}}$ – безразмерный параметр; $\ell_{\text{пт}}$ – длина токопроводящих проводов, м.

Конвективный коэффициент теплоотдачи шара определялся по формуле:

$$\alpha_{\text{к}} = \alpha - \alpha_{\text{л}} = P_{\text{к}} / F(t_{\text{ср.}} - t_{\text{ф}}) = \varepsilon \cdot C_0 \frac{\left(\frac{T_{\text{ср.}}}{100}\right)^4 - \left(\frac{T_{\text{ф}}}{100}\right)^4}{T_{\text{ср.}} - T_{\text{ф}}}, \quad \text{Вт/м}^2 \cdot \text{град.} \quad (3)$$

Здесь F – площадь поверхности шара, м²; ε – степень черноты поверхности шара; C_0 – коэффициент лучеиспускания абсолютно чёрного тела, Вт/м²·К⁴; $T_{\text{ср.}} = (t_{\text{ср.}} + 273)$ – средняя температура поверхности, К; $T_{\text{ф}}$ – температура окружающей среды, К.

Результаты экспериментального исследования теплоотдачи шара при различных степенях разрежения приведены в табл. 1,2.

Как видно из табл. 1,2, температура перегрева шара при степени черноты $\varepsilon = 0,8$ при остальных одинаковых параметрах снизилась на 50% по сравнению с температурой перегрева при степени черноты поверхности $\varepsilon = 0,22$.

Роль конвективного коэффициента теплоотдачи с увеличением степени черноты поверхности снижается, особенно это сказывается при низких давлениях.

Научные и практические результаты. Впервые по результатам проведенных экспериментальных исследований радиоэлементов, функционирующих в разреженной среде, установлено, что при расчёте температурных полей для деталей, имеющих геометрическую форму в виде шара или в виде ограниченного цилиндра, при давлении среды $H = 10^{-3}$ мм рт. ст. и степени черноты поверхности $\varepsilon < 0,22$, конвективную составляющую коэффициента теплоотдачи можно не учитывать, а расчёты производить с учётом только лучистой составляющей.

Анализ экспериментальных данных (табл. 1, 2) позволяет сделать вывод, что покрытие поверхностей радиодеталей веществом с высокой степенью черноты приводит к увеличению общего коэффициента теплоотдачи. Учёт полученных результатов позволит получить

Таблица 1

| Н мм. рт. ст. | $\varepsilon = 0,22$ | | | | | | |
|--------------------|------------------------|--------------------------|-------------|--|---|---|------------------------------------|
| | $t_{\text{ф}}$, °С | $t_{\text{ср.}}$, °С | t , °С | α , Вт/ м ² град | $\alpha_{\text{л}}$, Вт/ м ² град | $\alpha_{\text{к}}$, Вт/ м ² град | $\alpha_{\text{к}}/\alpha$ 100% |
| 750 | 21,5 | 75,5 | 54 | 8,03 | 1,63 | 7,40 | 92,2 |
| 10 ⁻¹ | 20,0 | 153 | 133 | 3,26 | 2,38 | 0,88 | 27 |
| 10 ⁻² | 20,0 | 173 | 153 | 2,83 | 2,62 | 0,21 | 7,4 |
| 6·10 ⁻³ | 20,0 | 175 | 155 | 2,78 | 2,67 | 0,11 | 3,96 |

Таблица 2

| Н мм. рт. ст. | $\varepsilon = 0,8$ | | | | | | |
|--------------------|------------------------|--------------------------|-------------|--|---|---|------------------------------------|
| | $t_{\text{ф}}$, °С | $t_{\text{ср.}}$, °С | t , °С | α , Вт/ м ² град | $\alpha_{\text{л}}$, Вт/ м ² град | $\alpha_{\text{к}}$, Вт/ м ² град | $\alpha_{\text{к}}/\alpha$ 100% |
| 750 | 20 | 62,4 | 42,4 | 10,2 | 6,0 | 4,2 | 41 |
| 10 ⁻¹ | 21 | 87,7 | 66,7 | 6,5 | 6,4 | 0,1 | 1,5 |
| 10 ⁻² | 20 | 89,1 | 69,1 | 6,3 | 6,3 | 0 | 0 |
| 6·10 ⁻³ | 20 | 90,4 | 70,4 | 6,1 | 6,1 | 0 | 0 |

уточнённые расчёты температурных полей РЭА, работающих в этих же средах, на начальных этапах проектирования при расчёте температурных полей РЭА или при их синтезе.

Сравнение с лучшими аналогами. В литературных источниках [1-6] приведены сведения степеней черноты для различных материалов. Данные о степенях черноты для радиоэлементов и о влиянии степени черноты радиоэлементов, функционирующих в разреженных средах, в нормативных документах и в литературе отсутствуют. Поэтому полученные в результате проведенных экспериментов данные позволяют конструктору на начальной стадии проектирования оптимального по перегреву аппарата (особенно при синтезе) учитывать закономерности влияния степени черноты радиоэлементов функционирующих в условиях пониженного давления.

Анализ полученных экспериментальных результатов (табл. 1,2) позволяет сделать следующие выводы.

1. При расчёте теплового режима радиодеталей, имеющих геометрическую форму в виде шара или в виде ограниченного цилиндра, при давлении среды $H = 10^{-3}$ мм рт. ст. и степени черноты поверхности $\epsilon = 0,22$ конвективную составляющую коэффициента теплоотдачи можно не учитывать, а расчёт производить с учётом только лучистой составляющей.

2. Конвективный коэффициент теплоотдачи для ограниченных цилиндров при свободной конвекции в разреженной среде можно определять по формулам, представленным в [8].

3. Из табл. 1,2 видно, что для увеличения общего коэффициента теплоотдачи радиодеталей необходимо их поверхности покрывать веществом с высокой степенью черноты.

Список литературы: 1. *Иванов О.А.* Охлаждение аппаратуры РЛС. М., Военное издательство министерства обороны СССР, 1975. 96 с. 2. *Дульнев Г.Н., Тарнавский Н.Н.* Тепловые режимы электронной аппаратуры. Л.: Энергия, 1971. 287 с. 3. *Михеев М.А.* Основы теплопередачи. Госэнергоиздат. 1956. 315 с. 4. *Тёмкин А.Г.* Обратные методы теплопроводности. М.: Энергия, 1973. 464 с. 5. *Мучник Г.Ф., Рубаинов И.Б.* Методы теории теплообмена. ч. 1. М.: Высшая школа, 1970. 288 с. 6. *Лыков А.В.* Теория теплопроводности. Госэнергоиздат. 1952. 392 с. 7. *Майко И.М., Синотин А.М.* Экспериментальное определение эффективной теплопроводности нагретых зон РЭА // Вопросы радиоэлектроники. ТРТО. 1972. №2. С. 23-25. 8. *Майко И.М., Синотин А.М., Дётинов Ю.М.* О теплофизическом конструировании одноблочных радиоэлектронных аппаратов с заданным тепловым режимом // Вопросы радиоэлектроники. ТРТО. 1974. № 1. С. 14-18. 9. *Синотин А.М., Семенец В.В.* Метод определения эффективных теплопроводностей сложных систем тел // АСУ и приборы автоматики. 2004. Вып. 127. С. 48 - 52

Поступила в редколлегию 10.11.2009

Синотин Анатолий Мефодиевич, канд. техн. наук, доцент кафедры социальной информатики ХНУРЭ. Научные интересы: проектирование, автоматизация и производство радиоэлектронной аппаратуры. Адрес: Украина, 31052, Харьков, ул. Полтавский Шлях, 41, кв. 2, тел. 7021 – 494.

Колесникова Татьяна Анатольевна, канд. техн. наук, доц. каф. ИКГ ХНУРЭ. Научные интересы: проектирование и производство биомедицинской радиоэлектронной аппаратуры. Адрес: Украина, Харьков, пр. Победы, 57 Г, кв. 35, тел. 3 - 37-61-47.

ПОСТРОЕНИЕ КРОСС-ПРОЦЕССОВ В АСУ ОРГАНИЗАЦИОННОЙ СЕТЬЮ

Предлагается формальная постановка и подход к решению задачи построения кросс-процессов организационной сети. Применяется аддитивная свёртка нормализованных частных критериев. Задача состоит в поиске на графе организационной сети подграфа, изоморфного графу-шаблону и имеющего лучшее значение критерия.

1. Введение

Высокие темпы развития информационных технологий и ускорение протекания экономических процессов предопределяют переход от иерархических организационных структур, слабо адаптирующихся к изменениям среды, к гибким и мобильным сетевым [1]. К таким структурам относится организационная сеть.

Организационная сеть (оргсеть) – это координированная совокупность предприятий и организаций, которая обычно выступает как целостный субъект рынка, как система. Координирование заключается в согласовании действий участников сети, которое выполняется на основе общих правил или стандартов ведения дел. Каждое предприятие сети представляет собой ресурс сети [2, 3].

Для организационной сети, включающей большое количество предприятий, координирующие действия выполняет специально созданная интеллектуальная компания - координатор. Для небольшого числа предприятий в качестве координатора может выступать эксперт или группа экспертов.

Оргсеть образуют разнотипные предприятия. Это позволяет формировать группы предприятий, реализующих технологии производства и поставки продукта конечному потребителю. Такой процесс, в котором участвуют несколько предприятий, будем называть кросс-процессом. В оргсети может функционировать много кросс-процессов. Отдельное предприятие может быть участником нескольких кросс-процессов. Кросс-процесс носит временный характер: он осуществляется, пока в нем существует потребность, определяемая состоянием внешней среды (покупатели, технологии и т.п.). В рамках оргсети происходят постоянные формирования и «распады» кросс-процессов [4]. Поэтому одна из важных задач координатора состоит в организации (формировании, построении) кросс-процессов.

Цель исследования: разработка моделей и методов автоматизации построения структур совместной деятельности предприятий организационной сети.

Задача исследования: создание модели и постановка задачи построения кросс-процесса в организационной сети.

2. Постановка задачи

Кросс-процесс реализует технологию производства и продвижения изделия к покупателю. Графическую модель (шаблон) кросс-процесса представим ориентированным графом $H(V_h, D_h)$, который назовем технологическим. Вершинами этого графа являются технологические единицы (макрооперации). Технологическая единица или макрооперация – это совокупность операций, в результате которых получают компоненту конечного изделия или готовое изделие (сборка) [5]. Каждому предприятию оргсети ставится в соответствие одна или несколько технологических единиц. Дуги технологического графа указывают последовательность выполнения макроопераций технологического процесса, а также означают перемещение материальных единиц между макрооперациями.

Структура оргсети представлена в виде графа $G(V, D)$, в котором вершины и дуги имеют такой же смысл, как и для технологического графа $H(V_h, D_h)$.

В зависимости от конкретной ситуации для выделения лучшего кросс-процесса принимается один из следующих вариантов критериев: 1) минимальные материальные затраты на создание и продвижение материального потока по звеньям оргсети; 2) минимальные

временные затраты на создание и продвижение материального потока по звеньям оргсети;
3) минимальные материальные и временные затраты [6].

В общем случае задача выбора лучшего кросс-процесса является многокритериальной, и для поиска обобщенных многокритериальных скалярных оценок будем использовать обобщенный критерий $P(x)$ [7]:

$$P(x) = \sum_{i=1}^n a_i k_i^H(x), \quad (1)$$

где x – решение, принадлежащее множеству допустимых решений X ; $k_i^H(x)$ – нормализо-

ванные частные критерии решения x : $k_i^H(x) = \left(\frac{k_i(x) - k_{i\delta}}{k_{i\epsilon} - k_{i\delta}} \right) \alpha_i$; a_i – безразмерные весо-

вые коэффициенты относительной важности частных критериев, для которых выполняется

ограничение $0 \leq a_i \leq 1$, $\sum_{i=1}^n a_i = 1$; $k_i(x)$ – исходный i -й частный критерий: $k_{i\epsilon}$, $k_{i\delta}$ –

соответственно лучшее и худшее значения частного критерия; n – количество частных критериев.

Параметр α_i определяет вид зависимости: выпуклая вверх ($0 < \alpha_i < 1$), линейная ($\alpha_i = 1$), выпуклая вниз ($\alpha_i > 1$).

В зависимости от вида экстремума

$$k_{i\epsilon} = \begin{cases} \max_{x \in X} k_i(x), & \text{если } k_i(x) \rightarrow \max, \\ \min_{x \in X} k_i(x), & \text{если } k_i(x) \rightarrow \min, \end{cases}$$

$$k_{i\delta} = \begin{cases} \min_{x \in X} k_i(x), & \text{если } k_i(x) \rightarrow \max, \\ \max_{x \in X} k_i(x), & \text{если } k_i(x) \rightarrow \min. \end{cases}$$

Если известны точные количественные значения весовых коэффициентов a_i частных критериев $k_i(x)$, то оптимальное значение $x^0 \in X$ равно

$$x^0 = \arg \operatorname{extr}_{x \in X} \sum_{i=1}^n a_i k_i^H(x), \quad i = \overline{1, n}, \quad \sum_{i=1}^n a_i = 1. \quad (2)$$

Оптимальное значение x^0 представляет собой подграф G_T графа G такой, что: а) $G_T(V_T, D_T) \sim N(V_h, D_h)$; б) функция (1) достигает своего экстремума.

3. Определение параметров кросс-процесса

Основными параметрами кросс-процесса являются общие затраты на производство и продвижение продукции, а также его продолжительность.

Для каждого предприятия известны:

- 1) статистическая информация об уровнях загрузки (на текущий момент и за предыдущие периоды работы предприятия);
- 2) продолжительность выполнения работы (время обработки сырья, изготовление изделий, производство продукции и т.д.);
- 3) затраты на выполнение работы (суммарные затраты предприятия на выполнение работы, включающие, помимо затрат на ресурсы, затраты на хранение, заработную плату сотрудникам, амортизационные отчисления и т.д.);
- 4) транспортные затраты (временные и материальные затраты на перемещение материального потока между ресурсами оргсети).

На основе перечисленных выше данных можно определить следующие функции. Для каждой вершины графа оргсети (технологической единицы) строятся 2 функции:

$$C_{vi} = f_{v1i}(W_i, O); T_{vi} = f_{v2i}(W_i, O), \quad i \in I_V, \quad O_{vi}^{\min} \leq W_i + O \leq O_{vi}^{\max},$$

где I_V – множество индексов вершин графа G ; C_{vi} – затраты предприятия на создание необходимого количества изделий; W_i – текущая загрузка i -й технологической единицы; O – объем производимой продукции; T_{vi} – продолжительность выполнения макрооперации; O_{vi}^{\min} , O_{vi}^{\max} – соответственно минимальный и максимальный объем продукции, которая может производиться i -й технологической единицей.

Для каждой дуги определяются следующие 2 функции:

$$C_{dj} = f_{d1j}(L_j, O); T_{dj} = f_{d2j}(L_j, O), j \in J_V,$$

$$O_{dj}^{\min} \leq O \leq O_{dj}^{\max},$$

где J_V – множество индексов дуг графа G ; C_{dj} – затраты на перемещение продукции по j -й дуге; L_j – расстояние (длина j -й дуги); O – объем перемещаемой продукции; T_{dj} – продолжительность перемещения продукции по j -й дуге; O_{dj}^{\min} , O_{dj}^{\max} – соответственно минимальный и максимальный объем продукции, который может перемещаться через j -ю дугу.

Каждому l -му кросс-процессу, представленному соответствующим графом $G_{\Pi}(V_{\Pi}, D_{\Pi})$, можно поставить в соответствие число, показывающее величину затрат:

$$C_l = \sum_{i \in I_{V_{\Pi}}} C_{vi} + \sum_{j \in J_{D_{\Pi}}} C_{dj},$$

$I_{V_{\Pi}}, J_{D_{\Pi}}$ – множества индексов, соответственно, вершин и дуг графа $G_{\Pi}(V_{\Pi}, D_{\Pi})$, а также число, показывающее продолжительность кросс-процесса:

$$T_l = \sum_{i \in I_{V_{\Pi}}^{KP}} T_{vi} + \sum_{j \in J_{D_{\Pi}}^{KP}} T_{dj},$$

$I_{V_{\Pi}}^{KP}, J_{D_{\Pi}}^{KP}$ – множества индексов, соответственно, вершин и дуг графа $G_{\Pi}(V_{\Pi}, D_{\Pi})$, составляющих критический путь.

Числа C_l и T_l являются значениями $k_1(x_1)$ и $k_2(x_1)$ критериев $k_1(x)$ и $k_2(x)$, где решение x_1 – граф $G_{\Pi}(V_{\Pi}, D_{\Pi})$.

4. Подход к решению задачи

Суть задачи состоит в том, что требуется на графе оргсети найти подграф, соответствующий некоторому шаблону (технологическому графу) и удовлетворяющий условию (2). Поскольку технологических звеньев каждого типа может быть несколько, то X – множество допустимых решений будут составлять всевозможные комбинации технологических звеньев, соответствующие шаблону (технологическому графу). Таким образом, имеем комбинаторную задачу на сети. Для сокращения перебора применяется принцип динамического программирования [7]. Технологический граф разбивается на стадии (рис.1).

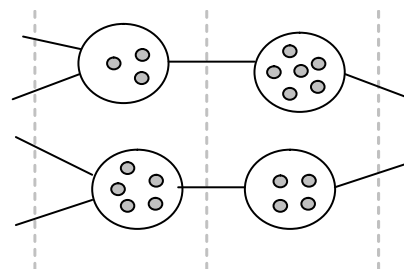
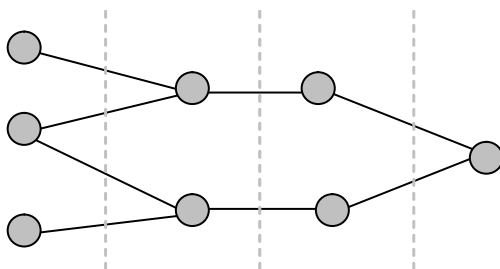


Рис. 1. Пример простейшего технологического графа Рис. 2. Фрагмент шаблона (рис. 1) с группами однотипных технологических единиц

Каждая стадия – шаг процедуры динамического программирования. В соответствии с шаблоном строится вспомогательная структура, фрагмент которой показан на рис. 2. Для каждой технологической единицы шаблона формируется множество подобных ей технологических единиц оргсети. Затем на этой вспомогательной структуре реализуется процедура динамического программирования.

Выводы

В разные периоды времени к кросс-процессам могут предъявляться какие-то новые, дополнительные или иные требования. Рассмотренная модель и подход к решению задачи позволяет учитывать это обстоятельство. Изменения будут состоять в модификации кортежа частных критериев и добавлении соответствующих функций вершинам и дугам сети. А сама структура модели остается неизменной.

Научная новизна: Предложена модель, предполагающая универсальную схему построения наилучшего возможного кросс-процесса с учетом технологии производства и перемещения продукции.

Практическая значимость: Уменьшение затрат на реорганизацию бизнес-процессов в социально-экономических системах и повышение загруженности предприятий оргсети.

Список литературы: 1. *Биггиеро Л.* Сети малого бизнеса. В кн.: Информационные технологии в бизнесе / Под ред. М. Желены. СПб.: Питер, 2002. С. 322-332. 2. *Стрелец И.А.* Новая экономика и информационные технологии. М.: Экзамен, 2003. 254 с. 3. *Башмаков А.И., Башмаков И.А.* Интеллектуальные информационные технологии. М.: МГТУ им. Н.Э. Баумана, 2005. 304 с. 4. *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов. М.: Мир, 1979. 536 с. 5. *Месарович М., Такахара Я.* Общая теория систем: математические основы. М.: Мир, 1978. 311 с. 6. *Боцман А.Ю., Вишняк М.Ю.* Сравнительный анализ методов поиска оптимального кросс-процесса в организационной сети // 3-й Международный радиоэлектронный форум «Прикладная радиоэлектроника. Состояние и перспективы развития», МРФ-2008. Сборник научных трудов. Международная конференция «Информационные компьютерные технологии и системы». Харьков: АНПРЭ, ХНУРЭ. 2008. 7. *Петров Э.Г., Новожилова М.В., Гребенник И.В., Соколова Н.А.* Методы и средства принятия решений в социально-экономических и технических системах. Учеб. пособие // Под общей редакцией Петрова Э.Г. Херсон: ОЛДІ-плюс, 2003. 380 с.

Поступила в редколлегию 02.03.2010

Боцман Андрей Юрьевич, аспирант кафедры системотехники ХНУРЭ. Научные интересы: исследование интеллектуальных систем, поддерживающих технологию управления бизнес-процессами. Адрес: Украина, 61166, Харьков, пр. Ленина, 14. тел. (057) 702-10-06, (050) 228-63-88, E-mail: Andrew_Botsman@mail.ru.

Вишняк Михаил Юрьевич, канд. техн. наук, доцент кафедры системотехники ХНУРЭ. Научные интересы: современные технологии управления организационными сетями. Адрес: Украина, 61166, Харьков, пр. Ленина, 14. тел. (057) 702-10-06, (057) 345-12-57, (095) 887-44-09, E-mail: mvishnyak95@gmail.com.

УДК 681.325.53: 37: 004.5

В.В. ВАРЕЦА

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ДВУХ СПОСОБОВ РЕАЛИЗАЦИИ ФОРМИРОВАТЕЛЕЙ ЭКВИВАЛЕНТОВ В ПРЕОБРАЗОВАТЕЛЯХ КОДОВ

Предлагается алгоритм построения таблиц законов функционирования основного нестандартного узла (формирователя эквивалентов) в преобразователях кодов по методу накопления эквивалентов. Описываются формулы для числа строк ФЭ.

1. Постановка задачи

Достоинством преобразователей кодов (ПК) по методу накопления эквивалентов по сравнению с другим методом является достаточно высокое быстродействие и возможность изменения в широких пределах соотношения между быстродействием и аппаратурными затратами.

Регулирование соотношения между быстродействием и затратами достигается путем выбора числа шагов преобразования (1,2,3,4 или более), выбора значений шагов, варианта декомпозиции многоразрядного ПК на блоки, а также благодаря использованию последовательной или параллельной стратегии обнуления входных разрядных счетчиков в многошаговых ПК.

Целью данной работы является анализ структуры функционирования ПК и формулировка алгоритма построения таблиц законов функционирования ФЭ для ПК с различным числом шагов преобразования:

- определение числа строк ФЭ, аппаратные затраты которых пропорциональны их числу;
- сравнительный анализ способов реализации ФЭ.

2. Последовательная стратегия использования шагов преобразования

Последовательная стратегия использования шагов схемно реализуется более просто. Максимальное число тактов преобразования любого k -ичного входного числа в двоичный код на выходе ПК для 2-шагового N_2 , 3-шагового N_3 и 4-шагового N_4 ПК определяется соответственно по формулам:

$$N_2 = \lceil (K-1)/a \rceil + a - 1;$$

$$N_3 = \lceil (K-1)/b \rceil + \lceil (b-1)/a \rceil + a - 1; \quad (1)$$

$$N_4 = \lceil (K-1)/c \rceil + \lceil (c-1)/b \rceil + \lceil (b-1)/a \rceil + a - 1,$$

где K – основание системы счисления на входе; a – второй шаг преобразования ($a > 1$); b – третий шаг преобразования ($a < b < c$); c – четвертый шаг преобразования; $\lceil \rceil$ – скобки означают округление до меньшего целого. Первый шаг всегда равен 1.

При последовательной стратегии использования шагов обнуление входных разрядных счетчиков вначале ведется с наибольшим шагом [1].

Затем, когда все разряды имеют значение меньше наибольшего шага, происходит переход на более меньший шаг и на последнем этапе выполняется преобразование с шагом 1.

Таблица 1

| Шаги | | | |
|-------|-------|-------|-------|
| 1 | 2 | 3 | 4 |
| 1 | a | b | c |
| C_i | D_i | E_i | F_i |

Для получения информации о превышении цифрой X_i шага преобразования используются триггеры регистров состояния (табл. 1) и дешифраторы превышения.

Дешифраторы превышения реализуются на базе комбинационных схем. Условие формирования значений сигналов с выходов дешифраторов превышения определяется выражениями:

$$C_i = \begin{cases} 1, & X_i \geq 1; \\ 0, & X_i < 1, \end{cases} \quad (2)$$

$$D_i = \begin{cases} 1, & X_i \geq a; \\ 0, & X_i < a, \end{cases} \quad (3)$$

$$E_i = \begin{cases} 1, & X_i \geq b; \\ 0, & X_i < b, \end{cases} \quad (4)$$

$$F_i = \begin{cases} 1, & X_i \geq c; \\ 0, & X_i < c. \end{cases} \quad (5)$$

3. Параллельная стратегия использования шагов преобразования

При применении параллельной стратегии использования различных шагов преобразования одновременно в различных разрядах может вестись уменьшение показаний разрядных счетчиков на величину значения наибольшего из шагов c, b, a или 1 (для 4-шагового преобразователя кодов)[2,3].

При этом значение цифры

$$X_i(t+1) = X_i(t) - \gamma, \quad (6)$$

где $\gamma = \max\{c, b, a, 1\}$.

Применение параллельной стратегии использования различных шагов преобразования одновременно в различных разрядах уменьшает максимальное число тактов преобразования в 2-шаговом ПК – от 5-ти до 4-х, а в 3-шаговом ПК от 4-х до 3-х. Чтобы определить

целесообразность использования параллельной стратегии, необходимо получить ряд оценок сложности ФЭ для двух стратегий, а также алгоритм построения закона функционирования ФЭ[2].

Оптимальные значения шагов преобразования для параллельной стратегии можно получить путем моделирования (табл. 2) или используя специальное программное обеспечение «Converter» [2]. В табл.2 NT означает номер такта преобразования.

Для одношагового ФЭ при числе разрядов $p=2$ имеем следующую таблицу ФЭ (табл. 3). В табл. 3 S означает общий вид эквивалента.

При построении таблиц законов функционирования ФЭ для параллельной стратегии используется полный лексико-графический перебор всех значений шагов преобразования (табл. 4-6).

Из анализа табл.3 следует, что в верхней ее части для двухшагового ПК располагается 2^p комбинаций младшей группы переменных.

Затем следует, что для каждой комбинации старшей группы с числом единиц j следует записать 2^{p-j} различных комбинаций в младшей группе, имеющих $p-j$ нулей, и перебрав все значения этих разрядов, получим 2^{p-j} комбинаций с единицами.

Число таких всевозможных комбинаций с j единицами в старшей группе равняется C_p^j .

Таким образом, для двухшагового ПК получим число строк ФЭ по формуле.

$$N_2 = 2^p + \sum_{j=1}^{p-1} C_p^j \cdot 2^{p-j} + 1 \quad (6)$$

После исследования числа строк таблиц ФЭ для различного количества разрядов (рисунок) было получено выражение:

$$N_m^{\Phi\Xi} = (m+1)^p, \quad (7)$$

где m – число различных шагов преобразования.

Определение числа строк ФЭ является комбинаторной задачей.

Количество и значения коэффициентов C_p^j определяются числами треугольника Паскаля (табл. 7).

Результаты расчета N_m для $m=1,4$ и $p=1,5$ представлены в табл. 8

Таблица 4

| i | D ₂ D ₁ | C ₂ C ₁ | S |
|---|-------------------------------|-------------------------------|-----------------------------------|
| 0 | 0 0 | 0 0 | 0 |
| 1 | 0 0 | 0 1 | K ⁰ |
| 2 | 0 0 | 1 0 | K ¹ |
| 3 | 0 0 | 1 1 | K ¹ + K ⁰ |
| 4 | 0 1 | 0 1 | aK ⁰ |
| 5 | 0 1 | 1 1 | K ¹ + aK ⁰ |
| 6 | 0 1 | 1 0 | aK ¹ |
| 7 | 1 0 | 1 1 | aK ¹ + K ⁰ |
| 8 | 1 0 | 1 1 | aK ¹ + aK ⁰ |

Таблица 2

| NT | K=12 Набор шагов 1,4 | | | | | | | | | | | |
|----|------------------------|----|---|---|---|---|---|---|---|---|---|---|
| | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 2 | 1 | 0 | 0 |
| 2 | 3 | 2 | 1 | 0 | 2 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 3 | 2 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| NT | K=12 Набор шагов 1,2,4 | | | | | | | | | | | |
| | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 1 | 0 | 0 | 0 |
| 2 | 3 | 2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Таблица 3

| i | Шаг1 | | S |
|---|----------------|----------------|---------------------------------|
| | C ₂ | C ₁ | |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | K ⁰ |
| 2 | 1 | 0 | K ¹ |
| 3 | 1 | 1 | K ¹ + K ⁰ |

Таблица 5

| i | D ₂ D ₁ | C ₂ C ₁ | S |
|---|-------------------------------|-------------------------------|-----------------------------------|
| 0 | 0 0 | 0 0 | 0 |
| 1 | 0 0 | 0 1 | K ⁰ |
| 2 | 0 0 | 1 0 | K ¹ |
| 3 | 0 0 | 1 1 | K ¹ + K ⁰ |
| 4 | 0 1 | x 1 | aK ⁰ |
| 5 | 1 0 | 1 x | aK ¹ |
| 6 | 1 1 | 1 1 | aK ¹ + aK ⁰ |

Таблица 6 График выигрыша (экономии) числа строк параллельного ФЭ по сравнению со способом непосредственного выделения всех комбинаций ФЭ в параллельном ПК соответственно для двухшагового, трехша-

| i | E ₂ E ₁ | D ₂ D ₁ | C ₂ C ₁ | S |
|----|-------------------------------|-------------------------------|-------------------------------|-----------------------------------|
| 0 | 0 0 | 0 0 | 0 0 | 0 |
| 1 | 0 0 | 0 0 | 0 1 | K ⁰ |
| 2 | 0 0 | 0 0 | 1 0 | K ¹ |
| 3 | 0 0 | 0 0 | 1 1 | K ¹ + K ⁰ |
| 4 | 0 0 | 0 1 | 0 1 | aK ⁰ |
| 5 | 0 0 | 0 1 | 1 1 | K ¹ + aK ⁰ |
| 6 | 0 0 | 1 0 | 1 0 | aK ¹ |
| 7 | 0 0 | 1 0 | 1 1 | aK ¹ + K ⁰ |
| 8 | 0 0 | 1 1 | 1 1 | aK ¹ + K ⁰ |
| 9 | 0 0 | 0 1 | 0 1 | bK ⁰ |
| 10 | 0 1 | 0 1 | 1 1 | K ¹ + bK ⁰ |
| 11 | 0 1 | 1 1 | 1 1 | aK ¹ + bK ⁰ |
| 12 | 1 0 | 1 0 | 1 0 | bK ¹ |
| 13 | 1 0 | 1 0 | 1 1 | bK ¹ + K ⁰ |
| 14 | 1 0 | 1 1 | 1 1 | bK ¹ + aK ⁰ |
| 15 | 1 1 | 1 1 | 1 1 | bK ¹ + bK ⁰ |

гов, четырехшагового ПК определяется формулами:

Таблица 7

| | |
|-----|---------------|
| p=0 | 1 |
| p=1 | 1 1 |
| p=2 | 1 2 1 |
| p=3 | 1 3 3 1 |
| p=4 | 1 4 6 4 1 |
| p=5 | 1 5 10 10 5 1 |

Таблица 8

| m | p | | | | |
|----------------|-------|-------|------|------|-------|
| | 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 4 | 8 | 16 | 32 |
| 2 | 3 | 9 | 27 | 81 | 243 |
| 3 | 4 | 16 | 64 | 256 | 1024 |
| 4 | 5 | 25 | 125 | 625 | 3125 |
| k ₂ | 0,75 | 1,125 | 1,69 | 2,53 | 3,8 |
| k ₃ | 0,67 | 1,33 | 2,67 | 5,33 | 10,66 |
| k ₄ | 0,625 | 1,56 | 3,91 | 9,76 | 24,41 |

гового, четырехшагового ПК определяется формулами:

$$k_2 = \frac{N_m}{2 \cdot N_1} = \frac{N_2}{2 \cdot N_1} = \frac{3^p}{2 \cdot 2^p}; \quad (8)$$

$$k_3 = \frac{4^p}{3 \cdot 2^p}; \quad (9)$$

$$k_4 = \frac{5^p}{4 \cdot 2^p}. \quad (10)$$

Обобщенная формула экономии числа строк параллельного ФЭ для любого числа шагов преобразования m описывается выражением:

$$k_i = \frac{N_m}{m \cdot N_1}. \quad (11)$$

Для двухшагового ПК с различным числом разрядов p (значения p приводятся в скобках) имеем систему формул :

$$N_2(1) = 2^p + 1,$$

$$N_2(2) = 2^2 + C_2^1 \cdot 2^{2-1} + 1 = 2^2 + 2 \cdot 2 + 1 = 4 + 4 + 1 = 9,$$

$$\begin{aligned} N_2(3) &= 2^3 + C_3^1 \cdot 2^{3-1} + C_3^2 \cdot 2^{3-2} + 1 = \\ &= 2^3 + 3 \cdot 2^2 + 3 \cdot 2 + 1 = 8 + 12 + 6 + 1 = 27, \end{aligned} \quad (12)$$

$$\begin{aligned} N_2(4) &= 2^4 + C_4^1 \cdot 2^3 + C_4^2 \cdot 2^2 + C_4^3 \cdot 2 + 1 = 2^4 + 4 \cdot 2^3 + 6 \cdot 2^2 + 4 \cdot 2 + 1 = \\ &= 16 + 32 + 24 + 8 + 1 = 81 \end{aligned}$$

$$\begin{aligned} N_2(5) &= 2^5 + C_5^1 \cdot 2^4 + C_5^2 \cdot 2^3 + C_5^3 \cdot 2^2 + C_5^4 \cdot 2 + 1 = \\ &= 32 + 5 \cdot 2^4 + 10 \cdot 2^3 + 10 \cdot 2^2 + 5 \cdot 2 + 1 = 32 + 80 + 80 + 40 + 10 + 1 = 243. \end{aligned}$$

Анализ таблиц ФЭ для других значений m показывает, что число строк $N_m^{\Phi Э}$ определяется формулой:

$$N_m = (m + 1)^p; m = 1, 2, \dots \quad (13)$$

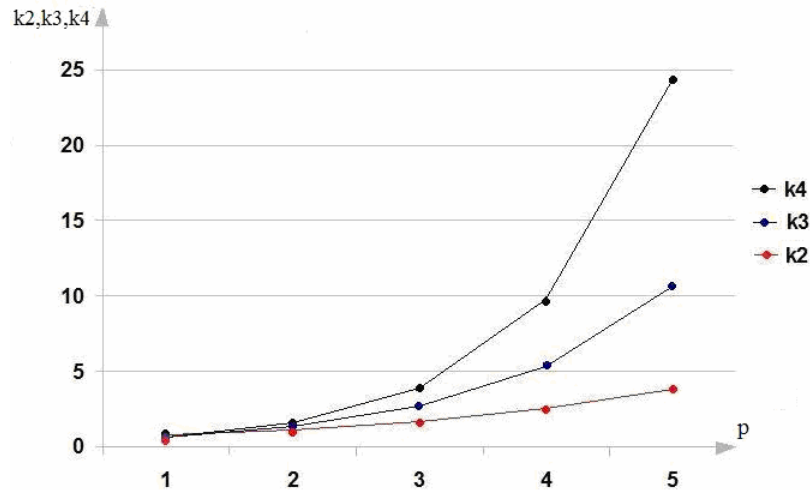


График изменения k_2, k_3, k_4 в функции от p числа разрядов ФЭ

4. Алгоритм построения таблиц законов функционирования ФЭ

Анализ табл. 3-6 позволяет сформулировать следующий алгоритм построения законов функционирования ФЭ в ПК с параллельным использованием шагов преобразования:

1. Выполнить полный перебор (начиная с нулевой комбинации) всех p разрядных двоичных комбинаций младшей группы переменных состояний. При этом в оставшихся $m-1$ старших группах (m -число различных шагов преобразования) все p разрядов переменных каждой группы принимают нулевые значения.

2. После получения единичной комбинации переменных группы, где выполнялся перебор, перейти к перебору значений переменных соседней старшей группы, исключив при этом нулевую комбинацию.

3. Определить число нулей в очередной комбинации значений переменных текущей группы. В столбце переменных текущей группы очередную комбинацию повторить 2^i раза. Во всех p разрядах переменных групп, более старших по отношению к текущей группе, следует выписать нулевые значения, а в разрядах, где переменные текущей группы имеют единичные значения, во всех более младших группах следует также записать единицы, перебрав все различные двоичные значения (от 000...0 до 111...1) в i нулевых разрядах очередной комбинации текущей группы.

4. После получения единичной комбинации переменных текущей группы следует сравнить набор текущей группы j с числом шагов преобразования m . Если $i < m$, то следует перейти к перебору значений переменных более старшей группы с номером $i+1$ (исключая нулевую комбинацию переменных). Если $i = m$, то переходят к вычислению эквивалентов.

5. Для получения десятичных значений эквивалента строки j таблицы закона функционирования ФЭ следует просуммировать степени основания k , соответствующие единичным значениям разрядов двоичных комбинаций переменных состояний текущей группы, а затем умножить сумму на величину шага преобразования, соответствующего текущей группе.

6. Выполнить перевод десятичного значения эквивалента в двоичную систему счисления.

7. Дооформить таблицу закона функционирования ФЭ путем указания всех исходных данных: основания системы счисления на входе, числа разрядов на входе, на выходе, числа шагов преобразования, значений шагов преобразования и стратегию использования шагов – параллельную.

Детальное системное проектирование многошаговых и многоблочных преобразователей кодов выполняется путем анализа аппаратурных затрат различных вариантов декомпозиции многоуровневого ПК на блоки в соответствии с методикой [4].

Выводы

Основные результаты: 1) Рассмотрены структуры и функционирование преобразователя кодов по методу накопления эквивалентов с применением параллельной стратегии использования шагов преобразования, что по сравнению с последовательной стратегией позволяет в двухшаговом параллельном ПК сократить число тактов преобразования ее с 5-ти до 4-х (повышать быстродействие на 20%), а в трехшаговом параллельном ПК сократить число тактов преобразования с 4-х до 3-х (повышать быстродействие на 25%).

2) Получено аналитическое выражение для определения числа строк в таблицах законов функционирования ФЭ в ПК с параллельной стратегией использования шагов преобразования в функции от основания системы счисления на входах, числа различных шагов преобразования m и числа переменных p на ФЭ.

3) Проанализированы два способа реализации ФЭ в ПК с параллельной стратегией использования шагов: а) на основе совокупности дешифраторов ДШ, выделяющих все строки ФЭ (затраты пропорциональны числу строк ФЭ) и б) на основе двух дешифраторов ДШ1 и ДШ2, функционирующих раздельно, младшие из которых управляются состоянием триггеров старшего регистра состояний.

Практическая значимость. Проведенные исследования позволяют ускорить этап системного проектирования ПК с параллельной стратегией использования шагов (повысить быстродействие ПК и уменьшить аппаратурные затраты по построению основного нестандартного узла ФЭ и всего ПК в целом).

Список литературы: 1. А.С. 1647908 5НОЗМ 7/12. Преобразователь двоично-К-ичного кода в двоичный код // *Н.Я. Какурин, Ю.К. Кирьяков, А.Н. Макаренко* // Открытия, изобретения. 1984. №17. С. 262-263. 2. *Какурин Н.Я., В.В. Варца, С.Н. Коваленко.* Параллельная стратегия использования шагов в двухшаговых преобразователях кода // АСУ и приборы автоматики. 2007. Вып. 141. С. 29-36. 3. *Коваленко С.Н.* Двухшаговый преобразователь кодов с параллельным использованием шагов преобразования // АСУ и приборы автоматики. 2007. Вып. 140. С. 103-109. 4. *Какурин Н.Я., Лопухин Ю.В., Макаренко А.Н., Замалеев Ю.С.* Системное проектирование преобразователей кодов дробных чисел по методу накопления эквивалентов // АСУ и приборы автоматики. 2009. Вып. 146. С. 33-39.

Поступила в редколлегию 28.02.2010

Варца 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.

УДК 534.014.4

А.В. МАМОНТОВ

МЕТОДИКА РАСЧЕТА ОПТИМАЛЬНЫХ ЗВУКОПОГЛОЩАЮЩИХ ПОКРЫТИЙ ПОМЕЩЕНИЙ

Предлагается методика расчета звукопоглощающего покрытия с оптимальным временем реверберации и минимальной себестоимостью. Составляется целевая функция и ограничения. Разрабатывается алгоритм и компьютерная программа решения оптимизационной задачи в среде Mathcad. Приводятся данные, свидетельствующие об эффективности предложенной методики.

1. Введение

В зависимости от характера звуковой передачи требуется оптимальное значение стандартного времени реверберации (далее времени реверберации), которое определено для студий, киноконцертных и театральных залов и некоторых других помещений, в зависимости от типа звука (речи, жанра, музыкального стиля) [1-3]. Речевые передачи проводятся в

помещениях студий с малым временем реверберации, а музыкальные – с большим. Иначе речь будет неразборчивой, а музыка «сухой» (менее мелодичной).

Эта проблема проявляется и во многих производственных условиях, где необходимо воспринимать и дифференцировать звуковые сигналы в процессе работы. К таким помещениям можно отнести учебные классы, лекционные аудитории, помещения диспетчерских служб, участки диагностики, регулировки и настройки аппаратуры, участки изготовления прецизионных деталей и т. д.

Согласно [4] одним из вредных производственных факторов, влияющих на качество труда и состояние здоровья человека, является нагрузка на слуховой анализатор при необходимости воспринимать и дифференцировать звуковые сигналы. Он характеризуется разборчивостью слов и сигналов, и измеряется в процентах. Оптимальными считаются условия, в которых разборчивость сигналов находится в пределах 90...100%, допустимыми – 90...70%, вредными (2 степень) – менее 50%.

Таким образом, создание комфортных условий труда также связано с обеспечением оптимального (заданного) времени реверберации для определенного характера трудовой деятельности. Эти условия могут быть достигнуты, благодаря применению звукопоглощающего покрытия поверхностей помещения с определенным коэффициентом звукопоглощения и площадью.

В настоящее время существует возможность широкого выбора звукопоглощающих материалов с различными физическими свойствами и стоимостью. Случайный выбор какого-либо материала позволит улучшить условия труда, обеспечив требуемое (оптимальное) время реверберации. Однако с экономической точки зрения он не будет оптимальным. Учитывая большие площади производственных помещений, требуются как правило большие затраты, которые резко ограничивают применение звукопоглощающих покрытий с этой целью. *Актуальность* данной работы объясняется снижением себестоимости покрытия при оптимальном времени реверберации, которое позволит улучшить условия труда.

Оптимизация заключается в получении минимальной стоимости покрытия. Поставленная задача относится к нелинейному дискретночисленному программированию.

Цель исследования состоит в разработке методики расчета звукопоглощающего покрытия помещения, обеспечивающего минимальную себестоимость при заданном значении времени реверберации.

При этом необходимо решить следующие задачи:

1. Составить целевую функцию и ограничения минимизации стоимости звукопоглощающего покрытия.
3. Составить методику расчета оптимального покрытия для различных типов звука.
4. Составить компьютерную программу расчета покрытия в среде Mathcad.
5. Оценить экономическую эффективность применения оптимального покрытия.

2. Целевая функция, ее ограничения и постановка задачи

Целевой функцией в данной задаче является стоимость покрытия, которая определяется произведением удельной стоимости материала (с учетом стоимости крепежных элементов и работ по установке) на его площадь. Так как площадь звукопоглощающего материала зависит от нескольких величин, в общем случае целевая функция имеет вид:

$$F = C_j \cdot S_j = F(C_j; \alpha_{i,j}(S_j); T_{\text{опт}i}), \quad (1)$$

где C_j – удельная себестоимость материала, грн./м²; j – номер материала в перечне доступных материалов; S_j – площадь покрытия материалом, м²; $\alpha_{i,j}$ – коэффициент звукопоглощения ограничивающей поверхности; i – номер октавной полосы; j – номер материала в перечне доступных материалов; $T_{\text{опт}i}$ – оптимальное время реверберации в октавной полосе под номером i , с.

Ограничениями являются: объем помещения; максимально возможная площадь покрытия; функциональная зависимость коэффициента звукопоглощения от площади звукопоглощающего материала, объема помещения и времени реверберации; функциональная зависимость времени реверберации от объема помещения; допустимое отклонение времени реверберации от оптимального значения в октавных полосах частот.

Объем помещения V задается действительным числом. Максимально возможная площадь покрытия S_{\max} – это суммарная площадь потолка, стен и других ограничивающих поверхностей, которые допускается покрывать звукопоглощающим материалом. Ограничение по площади запишется следующим образом:

$$S_j \leq S_{\max} . \quad (2)$$

Рассмотрим известные функциональные зависимости оптимального времени реверберации от объема помещения, разработанные для звуковых студий [1; 2]. Если первичный источник звука и слушатель находится в одном и том же помещении, величина оптимальной реверберации определяется по приближенным формулам с погрешностью не более 10% в соответствии с действующими нормативами:

– для речевых передач

$$T_{\text{ОПТ}} = 0,3 \lg V - 0,05 ; \quad (3)$$

– для малых музыкальных форм и оперных театров

$$T_{\text{ОПТ}} = 0,4 \lg V - 0,15 ; \quad (4)$$

– для симфонической музыки

$$T_{\text{ОПТ}} = 0,5 \lg V - 0,3 , \quad (5)$$

где $T_{\text{ОПТ}}$ – оптимальное время реверберации в октавной полосе №3 (на частоте 500 Гц), с;
 V – объем помещения, м^3 .

Согласно формуле Эйринга время реверберации зависит от объема помещения, реверберационного коэффициента поглощения и площади ограничивающих поверхностей:

$$T = \frac{0,161V}{S\alpha'} , \quad (6)$$

здесь S – суммарная площадь ограничивающих поверхностей, м^2 ; α' – реверберационный коэффициент поглощения:

$$\alpha' = -\ln(1 - \alpha_{\text{CP}}) , \quad (7)$$

α_{CP} – средний коэффициент поглощения:

$$\alpha_{\text{CP}} = \frac{\frac{\sum_1^n S_a \alpha_a + S_j \alpha_j + (S_{\text{MAX}} - S_j) \alpha_b}{S}}{S_{\text{MAX}} + \frac{\sum_1^n S_a}{1}} , \quad (8)$$

где n – количество непокрываемых поверхностей (поверхностей, которые не могут быть покрыты, например, пола, окон, дверей, оборудование; S_a и α_a – соответственно, площадь и коэффициент звукопоглощения материала непокрываемой поверхности на частоте 500 Гц; a – номер непокрываемой поверхности; S_j и α_j – соответственно, площадь и коэффициент звукопоглощения материала покрытия на частоте 500 Гц; α_b – коэффициент звукопоглощения покрываемых поверхностей (стен и потолка) на частоте 500 Гц до покрытия. В данной работе значение α_b для всех покрываемых поверхностей упрощенно считается одинаковым.

Решив уравнения (3), (6), (7) совместно, получим функциональную зависимость коэффициента звукопоглощения материала от его площади и времени реверберации для речевых передач:

$$\alpha_j = \frac{(1 - \exp(-\frac{0,16V}{(S_{\text{MAX}} + \frac{\sum_1^n S_a}{1}) \cdot (0,3 \lg V - 0,05)})) (S_{\text{MAX}} + \frac{\sum_1^n S_a}{1}) - \sum_1^n S_a \alpha_a - (S_{\text{MAX}} - S_j) \alpha_b}{S_j} . \quad (9)$$

В общем случае, с учетом выражений (3)-(5), функциональная зависимость коэффициента звукопоглощения материала от его площади и времени реверберации равна

$$\alpha_j = \frac{(1 - \exp(-\frac{0,16V}{(S_{MAX} + \sum_1^n S_a) \cdot T_{\text{opt}}})) \cdot (S_{MAX} + \sum_1^n S_a) - \sum_1^n S_a \alpha_a - (S_{MAX} - S_j) \alpha_b}{S_j}. \quad (10)$$

Условия допустимого отклонения времени реверберации от оптимального значения в октавных полосах частот представлены в относительных единицах [1, 2]:

$$\delta_{\text{MIN}} \leq \frac{T_{\text{opt}i}}{T_{\text{opt}}} \leq \delta_{\text{MAX}}, \quad (11)$$

где δ_{MIN} и δ_{MAX} – соответственно, нижняя и верхняя границы допуска;

$$\frac{T_{\text{opt}i}}{T_{\text{opt}}} = \frac{0,16V}{(S_{MAX} + \sum_1^n S_a) \ln(1 - \frac{\sum_1^n S_a \alpha_{ia} + S_j \alpha_{ij} + (S_{MAX} - S_j) \alpha_{ib}}{(S_{MAX} + \sum_1^n S_a)})} \cdot T_{\text{opt}}. \quad (12)$$

На рис. 1 приведены частотные зависимости $\delta_{\text{MIN}i}$ и $\delta_{\text{MAX}i}$ (в октавных полосах) для речевых передач и музыкальных программ [2, 3].

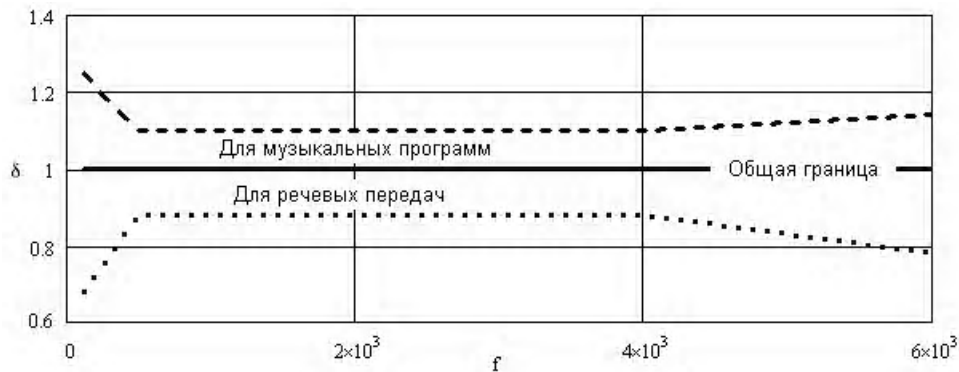


Рис. 1. Частотные зависимости $\delta_{\text{MIN}i}$ и $\delta_{\text{MAX}i}$

В данной задаче величины S_j ; $\alpha_{i,j}$ и $T_{\text{opt}i}$ заданы в виде дискретных значений. Требуется найти оптимальное сочетание величин α_j и S_j , при котором целевая функция (1) принимает минимальное значение.

3. Методика расчета оптимального покрытия

Традиционный инженерный способ расчета звукопоглощающего покрытия помещения основан на том, что материал и площадь покрытия выбираются произвольно, а дальнейшие расчеты сводятся к проверке соблюдения допуска на время реверберации в октавных полосах. Очевидна необходимость создания методики и компьютерной программы расчета оптимального покрытия.

Данная задача является оптимизационной и относится к нелинейному дискретночисленному программированию. Среди существующих методов ее решения наиболее подходящим является метод «ветвей и границ» [5; 6]. Однако применение данного метода в обычном виде затруднено тем, что целевая функция F зависит от S_j неявно.

Для вычисления целевой функции требуется предварительно вычислять S_j . Для этого организуем цикл вычислений, приравняв левую часть выражения (9) заданному значению $\alpha_{i,j}$. Площадь будем изменять с заданным шагом до тех пор, пока не будет получено тождество. Значение, при котором выражение обращается в тождество, будем считать площадью звукопоглощающего материала S_j (рис. 2).

Методика расчета покрытия включает в себя:

- выбор первого звукопоглощающего материала из перечня доступных;
- численное определение необходимой площади покрытия из условия (10);
- проверка условия (11);
- расчет стоимости покрытия (при выполнении условия (11));
- перебор остальных имеющихся материалов с выполнением перечисленных выше этапов;
- выбор материала, при котором целевая функция (1) имеет минимум.

На рис. 2, соответственно, показаны графики для коэффициента звукопоглощения при нефиксированном и фиксированном объеме помещения.

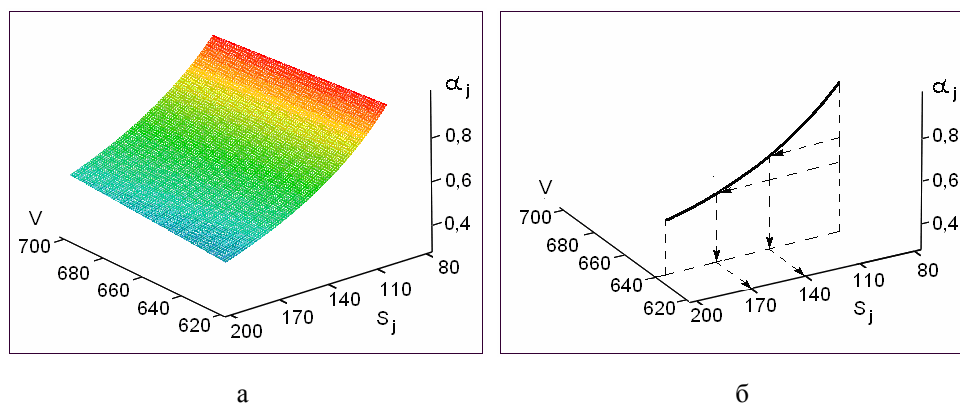


Рис. 2. Коэффициент звукопоглощения

Характеристики помещения, соответствующие рассмотренному примеру, приведены в компьютерной программе (рис. 3).

4. Компьютерная программа и расчет оптимального покрытия

Компьютерная программа разработана на основе предложенной методики в широко распространенной среде Mathcad-14 [7]. Исходные и выходные данные представлены в виде текста и матриц. Программа содержит необходимые пояснения, а также упрощения, которые не имеют принципиального значения.

Наименование материалов, данные об их физических свойствах и стоимости заимствованы из [8]. Толщина покрытий не превышает 30 мм.

В данной методике и программе не учитывались такие характеристики материалов, как теплопроводность, горючесть, влагостойкость и эстетические свойства. Однако работа в этом направлении не прекращается, и в дальнейшем эти требования могут быть учтены.

Помещение, для которого выполнялся расчет, имеет размеры 8 x 20 x 4 м. Потолок и стены оштукатурены (коэффициент звукопоглощения 0,1). Пол бетонный (коэффициент звукопоглощения 0,05). Покрывать можно стены и потолок.

В данной статье программа приведена в целях возможности воспроизводства всеми заинтересованными лицами.

5. Оценка экономической эффективности оптимального покрытия

Оценку экономической эффективности проведем путем сравнения стоимости звукопоглощающих покрытий помещения различными материалами. Результаты расчета стоимости покрытий приведены на рис. 4. Расчет проводился с использованием программы (см. рис. 3).

Анализируя полученные данные, видим, что стоимость оптимального покрытия составляет 6615 грн. Стоимость остальных покрытий находится в пределах 9765...15350 грн. Таким образом, экономический эффект от применения звукопоглощающего покрытия в данном помещении будет в пределах 3150...8735 грн. Учитывая то, что многие предприятия, учреждения и организации испытывают необходимость подобной реконструкции сразу в нескольких помещениях, экономический эффект по предприятию в целом увеличится как минимум на порядок.

$V := 640$ $S1 := 184$ $Smax := 360$ $Tотт := 0.3 \cdot \log(V) - 0.05$ $Tдоп := (0.68 \ 0.75 \ 0.88 \ 0.88 \ 0.88 \ 0.88 \ 0.78)$ $\delta min := Tдоп^T$

| | | | | | | | | | | |
|---------------|--------------------------------|------|------|------|------|------|------|------|----------------------------|--|
| $\alpha f :=$ | "Обознач. мат-ла\ Частота, Гц" | 125 | 250 | 500 | 1000 | 2000 | 4000 | 6000 | "Уд. стоимость, грн./кв.м" | $\alpha 1 := 0.05$ $\alpha 2 := 0.1$ $\Delta \alpha := 0.1$ $\delta max := 1$ |
| | "Шуманет-EM" | 0.03 | 0.17 | 0.68 | 0.98 | 0.86 | 0.45 | 0.30 | 74.8 | |
| | "URSA" | 0.4 | 0.85 | 0.98 | 1.0 | 0.93 | 0.97 | 0.80 | 108.5 | |
| | "Isover" | 0.3 | 0.77 | 0.92 | 0.98 | 0.86 | 0.80 | 0.72 | 102.2 | |
| | "Rockwool" | 0.03 | 0.18 | 0.66 | 0.99 | 0.88 | 0.73 | 0.45 | 77.4 | |
| | "Heraklite" | 0.18 | 0.15 | 0.71 | 0.93 | 0.95 | 0.56 | 0.45 | 102.2 | |
| | "Ecofone" | 0.05 | 0.44 | 0.98 | 0.90 | 0.79 | 0.45 | 0.32 | 73.5 | |
| | "Izolone" | 0.08 | 0.26 | 0.66 | 0.78 | 0.76 | 0.65 | 0.36 | 100.3 | |

```

for i ∈ 0
  for j ∈ 0
    Ki,j ← "Обозначение мат-ла"
  for j ∈ 1..last[(αf)(0)]
    Ki,j ← αfj,i
  for i ∈ 1
    for j ∈ 1..last[(αf)(0)]
      for Sj ∈ 1..Smax
        for n ∈ 0..6
          for p ∈ 1..7
            for q ∈ 1..7
              αSj,V ←  $\frac{-0.161 \cdot V}{1 - e^{(S1+Smax) \cdot (Tотт)}} \cdot (S1 + Smax) - S1 \cdot \alpha 1 - (Smax - Sj) \alpha 2$ 
              Ki,j ← Sj if |αfj,3 - αSj,V| < Δα ∧ δminn <  $\frac{-0.161 \cdot V}{\ln \left[ 1 - \frac{S1 \cdot \alpha 1 + (Smax - Sj) \alpha 2 + Sj \cdot \alpha f_{(p,q)}}{(S1 + Smax)} \right]} \cdot (S1 + Smax) \cdot Tотт$  < δmax
    for i ∈ 2
      for j ∈ 1..last[(αf)(0)]
        Ki,j ← αfj,8
    for i ∈ 3
      for j ∈ 0..last[(αf)(0)]
        Ki,j ← Ki-2,j · Ki-1,j
    for j ∈ 0
      for i ∈ 1
        Ki,j ← "Площадь покрытия, кв. м"
      for i ∈ 2
        Ki,j ← "Уд. стоимость, грн./кв. м"
      for i ∈ 3
        Ki,j ← "Стоимость покрытия, грн."
    for i ∈ 0
      for j ∈ 0..last[(KT)(0)]
        b ← augment(K(0), K(j)) if K3,j ≠ 0 ∧ K3,j = min[submatrix[K, 3, 3, 1, last[(αf)(0)]]]

```

Рис. 3. Компьютерная программа и расчет покрытия

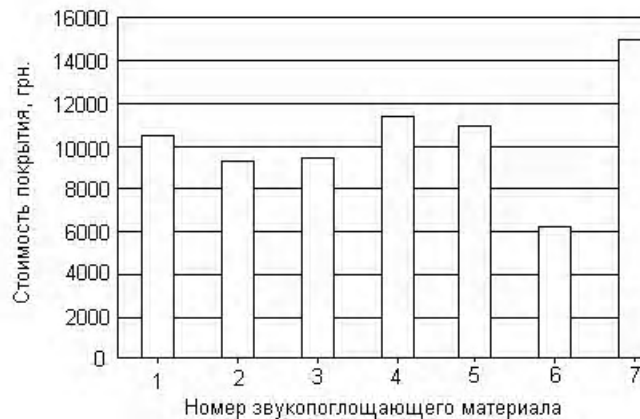


Рис. 4. Стоимость покрытий различными звукопоглощающими материалами: 1 – «Шуманет-БМ»; 2 – «URSA»; 3 – «Isover»; 4 – «Rockwool»; 5 – «Heraklite»; 6 – «Ecofone»; 7 – «Izolone»

6. Выводы

Наряду с традиционным инженерным способом расчета звукопоглощающего покрытия помещения, при котором материал и площадь покрытия выбирались произвольно, а дальнейшие расчеты сводились к проверке соблюдения допуска на время реверберации, стала очевидной необходимость создания методики и компьютерной программы расчета оптимального покрытия. Также стала очевидной необходимость нормирования времени реверберации для различных видов деятельности, которая будет способствовать более высокой разборчивости звуковых сигналов и низкой утомляемости. В результате проделанной работы поставленные задачи были решены.

Научная новизна заключается в постановке и решении оптимизационной задачи расчета покрытия. Новизна заключается также и в том, что в данной работе сформулирована необходимость нормирования времени реверберации для различных видов трудовой деятельности в целях улучшения условий труда.

Практическая значимость заключается в возможности достижения существенного экономического эффекта при акустической обработке помещений с различным характером звуковых передач, а также в улучшении условий труда.

Список литературы: 1. *Иофе В. К и др.* Справочник по акустике. М.: Связь, 1979, 312 с. 2. *Выходец А. В. и др.* Радиовещание и электроакустика. М.: Радио и связь, 1989, 430 с. 3. *Сапожков М. А.* Электроакустика. М.: Связь, 1978, 272 с. 4. ГН 3.3.5-8-6.6.1 2002 р. Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу. 5. *Акулич И. Л.* Математическое программирование в примерах и задачах. М.: Высшая школа, 1993. 275 с. 6. *Деньдобренко Б. Н. и др.* Автоматизация конструирования РЭА. М.: Высшая школа, 1980. 384 с. 7. *Гурский Д., Турбина Е.* Вычисления в Mathcad -12. М.: Питер, 2006. 544 с. 8. *Никифоров А. С.* Звукопоглощающие и звукоизоляционные материалы. М.: Связь, 2006. 322 с.

Поступила в редколлегию 16.03.2010

Мамонтов Александр Викторович, ст. пр. кафедры «Охрана труда» ХНУРЭ. Научные интересы: оптимизация защиты человека от вибрации, шума и радиации. Увлечения: музыка, садоводство, фото- и видеосъемка. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-13-60.

АРХИТЕКТУРА ПОДСИСТЕМЫ ВИЗУАЛИЗАЦИИ ДАННЫХ РАСПРЕДЕЛЕННОЙ ИМИТАЦИОННОЙ СИСТЕМЫ МОДЕЛИРОВАНИЯ GRID

Описываются исследования для решения проблемы разработки максимально гибкой и расширяемой архитектуры подсистемы визуализации данных, предназначенной для распределённой имитационной системы моделирования GRID. Разрабатывается оригинальный подход в реализации задачи создания графического интерфейса на основе распределённой модульной архитектуры с применением событийно-управляемой системы отношений между объектами типа «подписчик-слушатель».

1. Постановка задачи

В современных программных системах одним из основных элементов является интерфейс пользователя, который оказывает сильное влияние на скорость освоения приложений, время выполнения прикладных задач, стоимость программных разработок, производительность отдельных сотрудников и целых организаций. В последнее время появилось множество научных публикаций в этой предметной области [1,2]. Наиболее актуальной и обсуждаемой является построение визуального интерфейса распределённых и параллельных систем [3-5], ярким представителем которой является GRID-инфраструктура [6].

Цель данной работы – описание модели визуального интерфейса для распределённой имитационной системы моделирования GRID-инфраструктуры – GRASS (GRID Advanced Simulation System), разрабатываемой в Харьковском национальном университете радиоэлектроники [7,8].

2. Состав системы визуализации

В распределённой имитационной модели GRID функциональность визуализации данных не является непосредственно частью структуры системы, а наряду с другими подсистемами выделена в отдельные модули: “Windows Manager” и множество модулей-адаптеров.

Модуль “Windows Manager” представляет собой основу подсистемы визуализации, которая при запуске системы в графическом режиме создает главное окно, а также предоставляет методы, позволяющие другим компонентам системы регистрировать собственные графические элементы, панели инструментов и пункты меню. “Windows Manager” также занимается размещением и группировкой этих графических элементов внутри главного окна.

Модули-адаптеры, являющиеся вспомогательными компонентами системы, создают графические элементы, регистрируют их в главном окне с помощью модуля “Windows Manager” и обеспечивают взаимодействие графического интерфейса пользователя с основными модулями системы, а также своевременное обновление данных, предоставляемых этими модулями, в главном окне.

3. Модуль “Windows Manager”

Как и любой модуль в системе, “Windows Manager” поддерживает и реализует интерфейс Framework::IPlugin [8]. Вся логика работы менеджера окон расположена непосредственно в классе WindowsManager, с которым связан класс плагина (plug-in) [8] отношения агрегирования (рис. 1) [9].

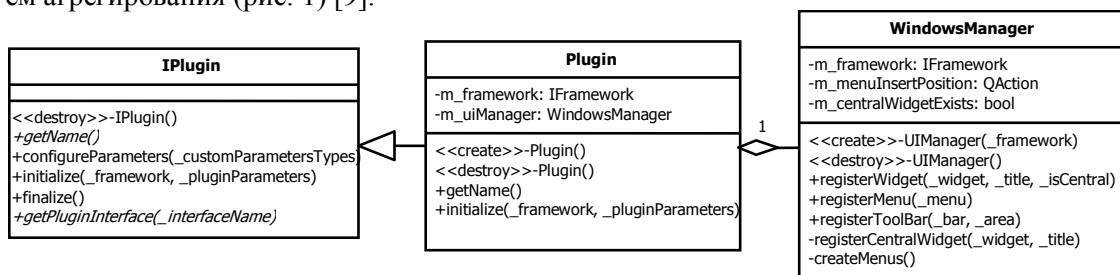


Рис. 1. Диаграмма классов модуля «Windows Manager»

Система конфигурируется таким образом, что при запуске в графическом режиме плагин менеджера окон загружается автоматически, создавая при этом главное окно и ожидая регистрации графических элементов другими плагинами. Класс Plugin — основной класс, описывающий данный модуль в системе распределенного имитационного моделирования GRID. Данный модуль реализует интерфейс IPlugin, описание методов которого представлено ниже. Метод initialize() вызывается при запуске системы и загрузке данного модуля. Все действия, которые необходимо выполнить до непосредственного начала работы модуля, описаны в этом методе.

Для получения имени модуля необходимо вызвать метод getName(). Метод getPluginInterface() в свою очередь позволяет получить один из интерфейсов модуля по его имени: в качестве параметра он принимает название интерфейса. Такая двухуровневая схема доступа к интерфейсу модуля позволяет одному модулю поддерживать несколько интерфейсов для взаимодействия.

Класс Windows Manager наследуется от класса QMainWindow библиотеки Qt [10], а также предоставляет интерфейс, описанный в IWindows Manager и содержащий нижеприведенные методы.

Метод registerWidget() позволяет любому другому модулю системы зарегистрировать в главном графическом окне системы собственный элемент. Ответственность за расположение этого графического элемента ложится на Windows Manager, в то время как обновление состояния должен производить модуль-адаптер на своё усмотрение.

Методы register Menu() и register ToolBar() предоставляют возможность другим модулям системы добавлять в главное окно собственные пункты меню и собственные панели инструментов соответственно. Как и в случае с регистрацией графического элемента, обработка событий для добавляемых меню/панелей осуществляется модулем, инициирующим регистрацию.

Таким образом, WindowsManager играет пассивную роль и занимается лишь правильным отображением и размещением графических элементов в главном окне, а инициировать процесс добавления этих элементов должен непосредственно каждый модуль в отдельности.

4. Модули-адаптеры

Модули-адаптеры используются для того, чтобы снизить связанность (coupling) системы и повысить связность (cohesion) каждого отдельно взятого модуля, следуя правилу эффективной разработки «low coupling – high cohesion». В стандартной ситуации каждому модулю системы помимо выполнения своей основной работы пришлось бы «заботиться» и о выводе информации на экран оператора системы. С добавлением нового функционала сложность модуля растет экспоненциально, т.е. уровень связности падает, а уровень связанности, наоборот, возрастает вследствие корреляции этих двух параметров, что приводит к затруднению дальнейшей модернизации и сопровождения системы.

Модули-адаптеры, призванные решить эту проблему, представляют собой дополнительную прослойку между графической и аналитической подсистемами, т.е. играют роль связующего звена.

Для каждого модуля, чьи данные необходимо выводить в графическом режиме работы системы, создается дополнительный модуль-адаптер, который так же, как и все, наследует и реализует интерфейс IPlugin. При этом сам модуль-адаптер не предоставляет собственного интерфейса, так как всю работу выполняет на этапе загрузки. Таким образом, система должна быть сконфигурирована так, чтобы все необходимые модули-адаптеры автоматически загружались при старте. Сама же подсистема регистрации элементов графического интерфейса, т.е. модуль Windows Manager, описанный выше, загружается «по первому требованию», что позволяет избавиться от дополнительных накладных расходов на процессорное время и память, необходимые для процесса загрузки и инициализации, в случае если система сконфигурирована таким образом, что ни один модуль-адаптер не запускается автоматически.

Каждый модуль-адаптер может иметь различное внутреннее устройство, однако должен выполнять как минимум одну функцию из трех возможных: инициировать регистрацию графического элемента, инициировать добавление пунктов меню и/или инициировать добавление панели инструментов.

Все регистрируемые компоненты хранятся внутри самого модуля-адаптера, поэтому типичная реализация модуля, выполняющего все три функции, подразумевает наличие внутреннего класса `Widget`, а также набора действий, инкапсулированных в классах `QAction`, на основе которых могут создаваться как пункты меню, так и кнопки панелей инструментов.

При старте системы и последующей инициализации модуля-адаптера вызывается его конструктор, который обеспечивает создание всех необходимых графических компонентов и их регистрацию в менеджере окон, после чего за их отображение отвечает “Windows Manager”. Обновление данных в уже зарегистрированном и отображаемом графическом элементе производит модуль-адаптер. Связь адаптера с модулем, предоставляющим данные, производится на основе паттерна (pattern) проектирования “Наблюдатель” (Observer). Согласно этому паттерну, модуль, предоставляющий данные, реализует метод `addListener()`, а модуль-адаптер реализует метод `notify()` и подписывается на события, происходящие в основном модуле. Соответствующая диаграмма классов приведена на рис. 2.

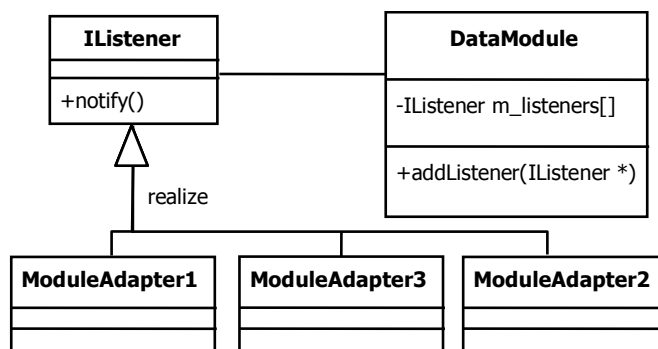


Рис. 2. Диаграмма отношений основного модуля с модулями-адаптерами на основе паттерна проектирования “Наблюдатель”

Опишем схему работы: модуль-адаптер подписывается [11] на события в предоставляющем данные модуле, посредством вызова метода `addListener()`, где в качестве “наблюдателя” или “слушателя” указывает себя. Этот модуль сохраняет слушателя в соответствующей коллекции, и когда данные обновляются, он вызывает для всех зарегистрированных в коллекции слушателей (модулей-адаптеров) метод `notify()`. Оповещенный модуль-адаптер повторно считывает данные из основного модуля через открытый интерфейс.

Такой подход дает возможность не только присоединять несколько модулей-адаптеров к одному основному модулю, позволяя выводить информацию от одного и того же модуля в различном виде, но и присоединять один модуль-адаптер к нескольким основным модулям, чтобы выводить общую информацию, поступившую от каждого из них.

Выводы

Предложена новая архитектура визуального интерфейса для распределённой имитационной системы моделирования GRID-инфраструктуры — GRASS. Описанное решение позволило выделить визуализацию данных в отдельную подсистему, уменьшить связанность модулей в системе и увеличить их внутреннюю связность. Одним из основных достоинств архитектуры является возможность подключения новых модулей со своими оригинальными функциями визуализации без перекомпиляции всего проекта. Гибкость разработанной архитектуры позволяет расширять или модифицировать модули визуализации в целях проведения более сложных экспериментов.

Научная новизна результатов состоит в получении модели взаимодействия подсистемы визуализации данных и построении пользовательского графического интерфейса с остальными компонентами распределённой имитационной системы моделирования GRID, реализуемого на основе принципа «подписчик — слушатель».

Практическая значимость результатов заключается в возможности быстрого построения графических интерфейсов пользователя для имитационной распределённой системы моделирования GRID, используя при этом модульный подход, что позволяет наращивать элементы подсистемы визуализации данных по мере необходимости без опасений поте-

рять контроль над сложностью проекта из-за многочисленных внутренних межкомпонентных связей в распределённой имитационной системе моделирования.

Список литературы: 1. *Грибова В.В., Тарасов А.В.* Модель онтологии предметной области “графический пользовательский интерфейс” // Информатика и системы управления. Амурский государственный университет. Благовещенск, 2005. №1(9). С.80-90. 2. *Манаков Д.В.* Анализ параллельных визуальных технологий // Вычислительные технологии. 2007. Т. 12, № 1. С. 45-60 3. *Brodie K., Brooke J., Chen M. et al.* Visual Supercomputing Technologies, Applications and Challenges, Eurographics. STAR Reports. 2004. P. 37–68. 4. *Байдалин А.Ю.* Опыт разработки полнофункциональной системы визуализации параллельного программирования // Тр. 35-й Регион. школы-конф. “Проблемы теоретической и прикладной математики”. Екатеринбург. 26.01–30.01. 2004. С. 303–308. 5. *Karpov A.N.* Data visualization on parallel computer systems // Тр. 15-й Междунар. конф. по компьютерной графике и зрению. ГрафиКон-2005, 20–24 июля 2005. Новосибирск. 2005. С. 211–214. 6. *Thiebaux M., Tangmunarunkit H., Czajkowski K., Kesselman C.* Scalable Grid- Based Visualization Framework. Technical Report ISI-TR-2004-592, USC/Information Sciences Institute, June 2004. 7. *Волк М.А.* Структура программного комплекса имитационного моделирования элементов GRID-систем для научных исследований // Системы обработки информации. Вип. 3(77). 2009. С.125-128 8. *Волк М.А., Горенков А.С.* Архитектура имитационной модели GRID – системы, основанная на подключаемых модулях // Проблеми інформатики і моделювання. Матеріали дев’ятої міжнародної науково-технічної конференції. Х.:НТУ “ХПІ”, 2009. 92 с. 9. *Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж.* Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер, 2008. 37 с. 10. *Шлеер М.* Qt4. Профессиональное программирование на C++. СПб: БХВ-Петербург, 2007. 553 с. 11. *Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж.* Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер, 2008. 280 с.

Поступила в редколлегию 11.03.2010

Олендаренко Сергей Сергеевич, студент ХНУРЭ. Научные интересы: разработка архитектур распределенных приложений, построение компиляторов, функциональные языки программирования, ГИС-системы. Хобби: программирование игр, велосипедный спорт. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 (66) 494-11-67.

Волк Максим Александрович, канд. техн. наук, доцент кафедры ЭВМ ХНУРЭ. Научные интересы: распределенное имитационное моделирование, GRID – системы, системное программирование. Хобби: туризм, спортивное ориентирование, автомобили. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 (50) 343-42-90.

Гридель Ростислав Николаевич, аспирант ХНУРЭ. Научные интересы: распределенные имитационные системы моделирования. Хобби: компьютерная техника. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 (57) 7021-354.

УДК 519.713

Ю.С. МАРЧЕНКО

ПРОГРАММИРОВАНИЕ. ПРОБЛЕМЫ В ОБУЧЕНИИ И МЕТОДИЧЕСКИЕ ОШИБКИ. СООБЩЕНИЕ 1

Описывается многолетний опыт преподавания различных курсов программирования. Анализируются проблемы, возникающие при обучении, обсуждаются возможные методические ошибки, которые допускают обучающие этой дисциплине.

1. Введение

Курсы информатики и основ программирования сегодня в программах общеобразовательных школ, средних специальных и высших учебных заведений практически всех специальностей. Основные навыки работы на компьютере, знание основ программирования необходимы сегодня врачу и строителю, биологу и механику, т.е. специалистам самых разных специальностей, когда необходимо выполнить обработку статистических данных, инженерные расчеты строительных конструкций, механических узлов, электронных схем.

2. Анализ литературных источников

Проанализировав лучшие, наиболее востребованные книги по программированию, издающиеся огромными тиражами, приходишь к выводу, что в них отсутствуют темы, дающие ключ к пониманию тех глав, с которых фактически начинаются эти книги.

Так, Г. Шилдт [1] сразу переходит к рассмотрению основных элементов языка, типов данных, инструкций управления, работе с массивами. Далее рассматриваются темы «Указатели» и «Функции».

Книга С. Прата [2], прежде всего, предлагает освоить базовую структуру программы на C++, и только после этого перейти к изучению деталей, например, циклов, объектов. В этой же главе рассматриваются функции и классы, которым в последующих главах будет уделено особое внимание. Далее автор переходит к рассмотрению встроенных типов данных и указывает на возможность языка C++ разрабатывать собственные типы данных.

В книге Р. Лафоре [3] изложение ведется в следующей последовательности. Вначале рассмотрены основы процедурного программирования, а затем – непосредственно изучение классов. Рассмотрению классов предшествует рассмотрение структур. Причиной этого является то, что класс представляет собой синтаксическое расширение структуры. Заметим, язык C++ предоставляет программисту возможность работать не только с данными, но и с адресами этих данных. Программа на языке C++ с использованием указателей более эффективна, чем вариант этой же программы без использования указателей. Сравнительная эффективность программ оценивается по времени выполнения и требуемой для программы памяти. Однако указатели не являются необходимыми при изучении основ объектно-ориентированного программирования (ООП) и, как правило, представляют трудность для понимания. Поэтому в [3] вполне обоснованно указатели рассматриваются позже, чем в большинстве книг по программированию на C++, пока не будут рассмотрены основные концепции ООП.

В книге авторов Х.М. Дейтел и П.Дж. Дейтел [4] вначале рассмотрены несколько простых программ с подробным их комментарием. Такое начало в изложении курса программирования является хорошим методическим приемом, дающим обучающему возможность увидеть и получить общее впечатление от программы на языке C++. Авторы называют это подходом «живого кода» [4, с.41]. Далее они очень кратко определяют понятие алгоритма [4, с. 152] и упоминают [4, с. 154] работу Бома и Джакопини [5], согласно которой алгоритм любой сложности может быть реализован с использованием трех типовых алгоритмических структур – последовательной, выбора и повторения. После этого авторы переходят к рассмотрению основных операторов языка.

Книга У. Савитча [6] представляет собой учебный курс по программированию на C++. Ее отличает взвешенный методический подход, который позволяет использовать классы с первых шагов в программировании. Облегчает изучение C++ большое количество примеров, учебных программных проектов, подсказки и предостережения. Автор отмечает, что разработчики больших систем программного обеспечения (например, операционных систем, компиляторов), прикладные программисты подразделяют процесс разработки на ряд этапов [6, с. 34], которые называют жизненным циклом программного обеспечения. У. Савитч описывает ряд общих принципов [6, с. 31-34], которые можно использовать при разработке программ независимо от языка программирования. Но, обратите внимание на «скромные» 4 страницы в книге объемом 704 страницы, которые отведены этой важной теме!

Общим недостатком этих, бесспорно лучших, книг по программированию на языке C++ является невосприятие многими обучающимися такой последовательности в изложении. Рассмотрению основных конструкций языка должны предшествовать темы, к обсуждению которых мы и переходим.

Таким образом, целью данной работы является обоснование последовательности изложения лекционного материала и практических занятий, при которой достигается оптимальное восприятие дисциплины «Основы программирования».

Прежде всего, начиная изучать программирование, обучаемый должен знать определение алгоритма и его свойства, понимать этапы, по которым следует провести решаемую на компьютере задачу. Ведь авторы указанных книг уверяют, что для освоения излагаемого материала не требуется предварительного опыта в программировании.

3. Алгоритм. Основные свойства

Предварительно алгоритм можно определить как последовательность некоторых правил (действий), приводящих к получению результата. Так, инструкция пользования телефо-

ном-автоматом, рецептура приготовления лекарств, кулинарные рецепты – это алгоритмы, которые окружают нас. При решении различных задач каждый алгоритм предполагает наличие некоторых исходных данных и приводит к получению искомого результата. Каждый алгоритм предписывает некоторую конечную цепочку действий (шагов). Такая цепочка называется алгоритмическим процессом.

Рассмотрим основные свойства алгоритмов. Как правило, алгоритм должен быть допустим для различных вариантов исходных данных (допустимые исходные данные). Это свойство называется массовостью. Так, вычисляя сумму элементов массива, тип массива может быть, например, целым, вещественным, логическим; количество элементов массива – произвольным конечным числом.

Под детерминированностью (определенностью) алгоритма понимают однозначность определения каждого шага (действия). Каждый шаг алгоритма должен быть сформулирован так, чтобы не допускать двусмысленности толкования. Так, высказывание «Мать любит дочь» может быть истолковано двояко. Такое высказывание недопустимо при определении алгоритма, так как его реализация с таким шагом может привести к различным результатам.

В начале XX века алгоритмы стали объектом изучения. Результаты, полученные в теории алгоритмов, служат фундаментом компьютерной технологии. Мы же кратко рассмотрим алгоритмы в их прикладном понимании.

4. Основные этапы решения задач на ЭВМ

В настоящее время на ЭВМ решают самые разнообразные задачи:

- расчет баллистических траекторий космических объектов;
- разработка компьютерных игр;
- обработка статистических данных;
- сложные инженерные расчеты в строительстве, механике, электронике;
- автоматизированное проектирование и многие другие задачи.

Но, несмотря на огромное разнообразие этих задач, усматривается определенная общность в процессе разработки, и можно выделить последовательность этапов, через которые нужно провести решаемую на ЭВМ задачу.

5. Постановка задачи

Под постановкой задачи понимают математическую или иную строгую формулировку решаемой задачи. Этот этап включает определение целей создаваемой программы, а также ограничений, предъявляемых к программе. При постановке задачи должны быть определены требования:

- ко времени решения поставленной задачи;
- к объему необходимых ресурсов, например, оперативной памяти (ОП);
- к необходимой точности вычислений.

Так, часто не принципиально за 10 или за 15 секунд получено решение задачи. Если же ЭВМ работает в режиме реального времени – управляет технологическим процессом (химическое производство, ядерные исследования), под управлением ЭВМ осуществляется запуск космического корабля, то на время ответа вычислительной системы могут накладываться жесткие временные ограничения, и это должно быть учтено при проектировании программы.

Если, например, объем ОП недостаточен для размещения всей программы, она размещается во внешней памяти и отдельными фрагментами загружается в ОП. Языковые средства C++ позволяют по-разному строить такую программу. Так, последующий фрагмент может замещать (перекрывать) предыдущий или располагаться в ОП вслед за предыдущим фрагментом. Таким образом, структура разрабатываемой программы определяется на этапе постановки задачи с возможной коррекцией на последующих этапах.

Вычисления, проводимые программой, например, на C++, могут быть выполнены с обычной, двойной точностью и особо точные. Это обеспечивает соответственно точность вычислений до 7, 15 и 19 значащих десятичных цифр. Поэтому точность вычислений должна быть обсуждена при постановке задачи.

6. Проектирование программы

На этом этапе следует выбрать метод расчета, если задача вычислительного характера. Если разрабатывается компьютерная игра, то должен быть определен ее сценарий. В любом случае следует выбрать или создать формальную модель, которая реализуется в разрабатываемой программе. На этапе проектирования определяют базовые и производные типы данных, с которыми будет работать программа. Наконец, определяют основные модули, из которых будет состоять программа и характер связей между этими модулями.

Рассмотрим пример вычисления $\sin x$. Запишем разложение этой функции в ряд:

$$\sin x \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + \frac{(-1)^{n+1} x^{n+1}}{(2n-1)!}.$$

В данном случае выбор численного метода заключается в том, что вычисление $\sin x$ сведено к вычислению алгебраической суммы членов ряда. Формирование суммы будем продолжать до тех пор, пока очередное слагаемое по абсолютной величине станет меньше заданной точности вычисления. Заметим, что при любом x , начиная с некоторого слагаемого, они будут монотонно убывать по абсолютной величине. Таким образом, вычисление тригонометрической функции сведено к вычислению алгебраической суммы членов ряда, где используются операции, которые выполняет ЭВМ.

7. Разработка алгоритма

Алгоритм – ключевое понятие в программировании. На этом этапе следует разработать детали проекта программы, когда программирование отдельных ее частей не вызывает никаких затруднений. Таким образом, алгоритм определяют как точное предписание, определяющее процесс преобразования за конечное число шагов исходных данных в искомый результат.

Существуют различные формы записи алгоритмов – словесно-формульная, графическая.

8. Кодирование алгоритма

После разработки алгоритма его записывают на языке, понятном ЭВМ. Это могут быть:

- язык символического представления реального машинного кода – Ассемблер;
- алгоритмические языки среднего и высокого уровней.

Разработка программы на языке Ассемблер требует знания логической организации (архитектуры) ЭВМ – адресность, регистры, флаги. Но программа, написанная на машинном языке, более эффективна, чем вариант этой программы на языке высокого уровня – она требует меньше памяти и быстрее выполняется.

Алгоритмические языки высокого уровня (например, Pascal) наиболее «дружелюбны» к программисту и выглядят как обычный текст на английском языке с использованием общепринятых математических знаков.

Языки среднего уровня (C, C++) сочетают «дружелюбность» языков высокого уровня с возможностью работы с отдельными битами, байтами, адресами, т.е. базовыми элементами ЭВМ, что характерно при программировании на Ассемблере. Код программы на C++ по времени выполнения и требуемой памяти сравним с программами, написанными на Ассемблере. Заметим, что язык C++ предусматривает в коде возможность применения ассемблерных вставок.

Использование алгоритмических языков предполагает наличие в компьютере соответствующего программного обеспечения (ПО). Так, преобразование программ с языка ассемблер в машинный код выполняет системная программа транслятор. Системные программы для преобразования последовательности операторов на языке высокого или среднего уровня в машинный код называются компиляторами.

ПО ЭВМ также должно включать компоновщики (редакторы связей), программные библиотеки (например, библиотеку стандартных функций) и другие системные программы.

9. Отладка и тестирование программы

Отладка – это устранение синтаксических и смысловых ошибок, которые могли быть допущены на всех этапах разработки программы. Синтаксические ошибки обнаруживают и исправляют, выполняя пробную трансляцию программы. Логические ошибки можно

выявить в процессе проверки программы на специально подобранных наборах входных данных. Поэтому тестирование определяют как выполнение программы в целях обнаружения скрытых ошибок. Ряд наборов исходных данных задаются такими, чтобы выполнялись все операторы (команды Ассемблера) и все ветви программы.

Тестированию уделяют в настоящее время большое внимание. «Работающая» программа проявит свои ошибки после длительного времени эксплуатации, если тестирование не было выполнено в полном объеме.

Завершая рассмотрение основных этапов решения задач на ЭВМ, заметим, что разделение этого процесса на 5 этапов носит условный характер. В случае простых программ некоторые этапы сливаются, например, проектирование с разработкой алгоритма или кодирование алгоритма с отладкой. В случае сложных программ могут добавиться новые этапы, например, проектирование базы данных или разработка интерфейса пользователя. При разработке сложной программы в процессе тестирования могут быть обнаружены ошибки, для исправления которых необходимо пересмотреть все этапы, возможно, и саму постановку задачи.

10. Выводы

Проанализировав самые востребованные учебники по программированию на C++, приходим к выводу, что в них отсутствуют, или освещены очень скупо темы, дающие ключ к пониманию самого процесса программирования. Рассмотрение основных конструкций языка, классов следует начинать лишь после того, как станут понятными этапы программирования. Последующими темами должны быть «Основы алгоритмизации» и «Типовые алгоритмические структуры».

Новизна и практическая значимость данной работы определяется следующим. Многолетний опыт преподавания дисциплины «Основы программирования» дает основания указать ту последовательность в изложении материала, при которой достигается наилучшее восприятие дисциплины, которая часто у обучающихся вызывает значительное затруднение в усвоении излагаемого материала.

В работе [5] показано, что алгоритм любой сложности может быть реализован лишь с использованием типовых алгоритмических структур, которые комбинируются различным образом в сложном алгоритме. Глубокое рассмотрение этих тем поможет обучающимся постичь самый сложный и ключевой этап в программировании – составление алгоритма. Эти темы планируется рассмотреть в сообщении 2 этой статьи.

Список литературы: 1. Шилдт Г. C++: базовый курс. 3-е издание. М.: Издательский дом «Вильямс», 2006. 624с. 2. Прата С. Язык программирования C++. Лекции и упражнения. 5-е издание. М.: ООО «И.Д. Вильямс», 2007. 1184 с. 3. Лафоре Р. Объектно-ориентированное программирование в C++. Классика Computer Science. 4-е издание. СПб.: Питер, 2005. 924с. 4. Дейтел Х.М., Дейтел П. Дж. Как программировать на C++. 5-е малое издание. М.: ООО «Бином-Пресс», 2007. 800с. 5. Bohm C., Jacopini G. Flow diadrams, Turing machines and languages with only twofomation rules. Communications of the ACM. 1966. Vol.9. P. 366-371. 6. Савитч У. Язык C++. Курс объектно-ориентированного программирования. 3-е издание. М.: Издательский дом «Вильямс», 2001. 704 с.

Поступила в редколлегию 09.03.2010

Марченко Юрий Сергеевич, канд. техн. наук, доцент кафедры программного обеспечения ЭВМ ХНУРЭ. Научные интересы: алгоритмические языки и программирование. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 8(057)7021446, дом. тел. 8(057)7026005.

ИССЛЕДОВАНИЕ МЕТОДОВ РЕШЕНИЯ ЗАДАЧ НАХОЖДЕНИЯ ОПТИМАЛЬНОГО РЕШЕНИЯ С ПРИМЕНЕНИЕМ МНОГИХ КРИТЕРИЕВ

Исследуются методы многокритериальной оценки, которые позволяют оптимизировать деятельность предприятия по нескольким критериям, где каждый критерий является составляющей технологического процесса.

Введение

В настоящее время прослеживается широкий спрос на технологии, которые позволяют решить проблему принятия решений, их интенсивное использование во всех сферах как метода автоматизации интеллектуальной деятельности человека. Результаты исследования задач планирования и управления показывают, что в реальной постановке эти задачи являются многокритериальными. Так, часто встречающееся выражение «достичь максимального эффекта при наименьших затратах» уже означает принятие решения при двух критериях. Оценка деятельности предприятий и планирования как системы принятия решений производится на основе более десятка критериев: выполнение плана производства по объему, по номенклатуре, плана реализации, прибыли по показателям рентабельности, производительности труда.

Это предоставило дополнительный импульс изучению и формализации процессов принятия решений и широкого круга задач оптимизации.

Актуальность работы. Задачи проектирования сложных систем актуальны в самых различных областях науки. Системное представление объектов и процессов используется в экономике, физике, но наиболее часто такой подход встречается в механике. При проектировании производственных процессов используется их модель, представленная в виде системы уравнений, описывающих рабочие характеристики. Математическая модель может корректироваться в процессе разработки, и для этого проектировщику необходимо производить большое количество сложных расчетов для проверки результатов изменений. Несмотря на глобальную компьютеризацию и высокий уровень информационных технологий, на данный момент большинство расчетов инженеры производят практически вручную. Компьютерные же технологии применяются только для создания чертежей уже спроектированной системы. Поэтому создание программного комплекса, позволяющего упростить расчеты на этапе проектирования системы и оптимизировать параметры уже готовой системы, является актуальным. Возникающие при этом научные и практические задачи требуют широкого применения методов теории оптимизации, ориентированных на определение и идентификацию наилучших вариантов решения, не прибегая к проверке всех возможных решений путем реализации их на практике.

Состояние проблемы. Среди многочисленных проблем, возникновение которых обусловлено бурно развивающимся научно-техническим прогрессом, наиболее важной является проблема совершенствования управления во всех звеньях народного хозяйства. Современные промышленные предприятия и научно-производственные комплексы, научно-исследовательские и опытно-конструкторские центры, комбинаты бытового обслуживания, т. е. самые разнообразные по характеру своей деятельности организации представляют собой сложные системы «человек - машина», эффективность функционирования которых существенно зависит от качества организационного управления этими системами. При формировании как стратегических, так и многих тактических решений руководителю далеко не всегда бывает достаточно личного опыта, интуиции и организаторских способностей в их традиционном понимании. В этих случаях на помощь приходит теория оптимизации, а именно многокритериальные задачи, ведь руководитель вынужден учитывать многочисленные, нередко взаимно противоречивые соображения и опираться на сложные критерии эффективности путей достижения целей.

Целью работы является оптимизация функционирования предприятия путем создания подсистемы поддержки принятия решений с использованием информационных технологий, ЭВМ, системного программного обеспечения и математических моделей. Создать систему, которая поможет определить наилучший (оптимальный) вариант действий при наличии ограничений технико-экономического или любого другого характера. При условиях неполноты нужного количества ресурсов необходимо их распределить таким образом, чтобы наиболее лучше удовлетворить всех потребителей и при этом получить максимальный эффект (максимальная прибыль или минимальный убыток).

В данной работе проведена оценка классических и современных методологий решения многокритериальных задач, выявление их достоинств и недостатков, разработка и исследование математической модели, разработка методов и алгоритмов, позволяющих найти решение указанной выше математической модели, создание программного продукта, спроектированного как полноценная система, способная решать любые оптимизационные задачи, разрешимые разработанными методами и алгоритмами.

Задачей исследования является нахождение оптимального плана действий, того или иного технологического процесса предприятия, который оценивается по нескольким критериям: максимизация прибыли и качества выпускаемой продукции, минимизация загрязнения окружающей среды.

1. Постановка задачи

Анализ предметной области, представление ее с помощью методов теории принятия решений, методов математического программирования.

К основным особенностям этого класса задач можно отнести следующее:

Многокритериальность – это главное свойство. Большинство авторов старается свести многокритериальные задачи к однокритериальным. Например, производство – безусловно, важный показатель. Но всегда ли нужно требовать его максимального значения? Самое главное, что при однокритериальной постановке задачи без ответа остаются важные вопросы: какой ценой это достигается, в какой мере ухудшаются при этом другие критерии и почему мы отдаем предпочтение этому, а не другому критерию? Стараясь свести многокритериальную задачу к однокритериальной, авторы заменяют одну задачу другой, которая очень отличается от начальной. Хорошо известен тот факт, что при оптимизации по одному критерию получаем одни результаты, по другому – другие, сразу по двум – третьи и т.д.

Построение допустимого множества – основной этап в постановке и решении задачи оптимального производства. Его неформальный анализ является неотъемлемой частью решения задачи.

Обычным есть то, что человек ставит задачу, а ЭВМ используется для ее решения. В этом случае такая система не работает, так как в очень редких случаях возможно априори, до принятия решения, поставить задачу производства. Постановка и решение – единый процесс. Определить допустимое множество возможно лишь в процессе решения задачи.

В ряде задач заказчик по тем или другим причинам не может формализовать основные критерии производства. Как правило, лишь после построения допустимого (или значительно большего) множества решений и его анализа с учетом критериев, которые нельзя формализовать, он отбирает наиболее оптимальное решение.

Математические модели – это сложные системы: линейные и нелинейные, детерминированные и стохастические, с распределенными и сосредоточенными параметрами. Параметры модели в большинстве случаев непрерывные. Область поиска может быть несвязной, отсутствует информация о гладкости функций цели, последние могут быть недифференцируемыми, хотя по обыкновению они непрерывные, размерность векторов параметров и критериев достигает иногда многих десятков. Основным преимуществом метода является то, что в ходе анализа пространства параметров поступает ценная информация о качестве математической модели. Ее корректируют, а при необходимости строят новую. В этом случае метод выступает в роли индикатора достоверности модели из позиций исследуемых показателей. Этот фактор не менее важный, чем сами результаты оптимизации, так как мы получаем ответы на ключевые вопросы: насколько содержательная модель по тем или другим критериям, какая значимость полученных результатов и т.п.

2. Математическая модель

Главная цель деятельности каждого предприятия в условиях рыночной экономики – максимизация прибыли или минимизация затрат. Эффективность работы предприятия в значительной мере зависит от информации о формировании себестоимости. Затраты на производство продукции являются базой для установления цены. Информация о себестоимости лежит в основе прогнозирования и управления производством.

Большинство задач, решаемых методами оптимизации, может быть сформулировано так: максимизировать или минимизировать $F(x_1, x_2, \dots, x_n)$ при ограничениях

$$\begin{aligned} g_1(x_1, \dots, x_n) &\leq b_1; \\ g_2(x_1, \dots, x_n) &\leq b_2; \\ &\dots\dots\dots \\ g_m(x_1, \dots, x_n) &\leq b_m; \end{aligned}$$

где $f(x_1, x_2, \dots, x_n)$ – целевая функция, или критерий эффективности (например, прибыль от производства каких-либо видов продукции, стоимость перевозок и т.п.); $X = \{x_1, \dots, x_n\}$ – варьируемые параметры; $g_1(x), \dots, g_m(x)$ – функции, которые задают ограничения на имеющиеся ресурсы. Именно при таких условиях и был промоделирован критерий максимизации прибыли для поставленной задачи.

Прибыль предприятия: с формально-математической точки зрения задача линейного программирования может быть описана таким образом: найти неизвестные величины, удовлетворяющие

$$F_1(x) = \sum_{j=1}^n C_j x_j \rightarrow \max, \quad (1)$$

$$x_j \in Q = \{1, 2, \dots, n\}; \quad (2)$$

где Q – множество видов продукции, выпускаемых предприятием.

При условии выполнения системы неравенств

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\leq b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\leq b_2; \\ &\dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\leq b_m; \quad x \geq 0. \end{aligned} \quad (3)$$

Уравнение (1) описывает функцию цели: максимизация прибыли при технико-экономических ограничениях неравенства (3).

Качество выпускаемой продукции: модель задачи квадратичного программирования имеет следующую структуру: найти максимальное значение функции (4) показателя качества продукта при линейных ограничениях (5):

$$F_2(x) = C^t X + 1/2 X^t D X = \sum_{j=1}^n C_j x_j + 1/2 \sum_{i=1}^m \sum_{j=1}^n d_{ij} x_i x_j \rightarrow \max; \quad (4)$$

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\leq b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\leq b_2; \\ &\dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\leq b_m. \end{aligned} \quad (5)$$

Уровень загрязнения окружающей среды: в связи с хозяйственной деятельностью объекта любой области в конкретном регионе непременно возникает определения природно-техническая система, которая включает как влияние объекта на окружающую среду, так и изменение состояние последнего. Задача минимизации загрязнения на производстве попадает под задачу минимизации функции (7) $F_3(X)$ при ограничениях:

$$AX \leq B, X \geq 0; \quad (6)$$

$$F_3(X) = \sum_{j=1}^n C_j x_j \rightarrow \min. \quad (7)$$

Параметры $C_j^{(3)}$ этой функции не известны и зависят от характеристик оборудования и материальных ресурсов. Модель объекта представляется в виде $F_3 = X_3 C^3 + E$, где $F_3^t = \{F_{31}, F_{32}, \dots, F_{3s}\}$ – вектор наблюдений; $C^{(3)} = \{C_1, C_2, \dots, C_n\}$ – вектор параметров, подлежащих оцениванию; $X_3 = X_{ij}, i=1, n, j=1, s$ – матрица экспериментальных значений независимых переменных; $E^t = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n\}$ – вектор ошибок.

Ограничением на выпуск продукции различных типов служат продовольственные ресурсы $B = \{b_1, b_2, b_3, \dots, b_m\}$. С учетом норм затрат ресурсов на единицу каждого вида продукции указанные ограничения можно записать в виде: выражение (6) описывает условия, которые необходимо учесть в годовой производственной программе, строкам матрицы A соответствуют все виды ресурсов, рассматриваемые в задачах. Соответствующие строкам матрицы A компоненты вектора B указывают ограничения видов ресурсов или объемов производства, которые установлены для годовой производственной программы предприятия. Эти неравенства представляют собой обычные условия неотрицательности, вытекающие из физического смысла задач.

$A = \{a_{ij}\}, i=1, m, j=1, n$ – матрица норм затрат ресурсов на единицу каждого вида продукции.

Общая постановка задачи состоит в следующем: определить вектор $X^{(0)}$, обеспечивающий компромисс между величиной прибыли (1), качеством выпускаемой продукции (4) и уровнем загрязнения окружающей среды (7), а также удовлетворяющий ограничениям (3), (5), (6).

Один из возможных методов решения состоит в том, что вначале находятся три оптимальных вектора производства $X^{(i)}, i=1, 3$, каждый из которых соответствует одному из локальных критериев. Затем определяется выпуклая линейная комбинация $X^{(0)}$, представляющая собой оптимальную (компромиссную) программу относительно указанных критериев:

$$X^{(0)} = v_1 x^{(1)} + v_2 x^{(2)} + v_3 x^{(3)}, \sum_{i=1}^3 v_i = 1, v_i \geq 0. \quad (8)$$

Для нахождения v_1, v_2, v_3 используется игровая модель, для решения которой был выбран матричный метод:

$$g_{kl} = \left| \frac{F_1(X^{(1)}) - F_1(X^{(k)})}{F_1(X^{(1)})} \right|, (k, l). \quad (9)$$

Для программной реализации разработанной математической модели была написана программа, которая позволяет решать любые уравнения. Данное приложение позволяет выбрать тип поставленной задачи – максимизация или минимизация, задать количество переменных в целевой функции и ввести их коэффициенты. Также приложение позволяет добавлять уравнения в систему ограничений. В основе программы лежит симплекс-метод, он основан на расчете симплекс-таблиц. Его идея заключается в последовательном продвижении по базисам опорных планов вплоть до получения оптимального решения или доказательства неразрешимости задачи. При этом значение целевой функции должно увеличиваться. Система ограничений приводится к каноническому виду. По найденной системе ограничений и целевой функции строится симплекс-таблица. Программа работает с предварительно обработанными данными, т.е. при разработке прикладной программы предполагалось, что данные по каждому из критериев подаются на вход, нормализованные и приведенные к каноническому виду. Это объясняется тем, что целью данной работы было экспериментальное моделирование оптимизационной модели, а также тем, что критерии оценки в реальном технологическом процессе можно разложить на сложные математические модели. Поэтому во время исследования и разработки оптимизационной модели этот факт был отброшен, т.е. считается, что данные на входе – это готовые результаты работы каждой такой модели.

Программа загружает данные и пошагово ищет оптимальное решение для заданного критерия.

3. Выводы

При разработке проектов сложных систем, в частности автоматических систем управления АСУ, перед проектировщиком возникает проблема принятия решений при наличии одновременно нескольких показателей качества. Поэтому разработка методов принятия решений при нескольких критериях оптимальности и в условиях неопределенности по-прежнему остается одной из главных задач исследования операций.

В результате данного исследования были построены математические модели производственных процессов с их технико-экономическими ограничениями, выбраны методы решения этих моделей, разработано программное приложение

Научной новизной результатов исследований, приведенных в работе, является метод решения полученной математической модели, наиболее полностью охватывающий ее на базе теории оптимизации. Также избранный метод легко реализуется на вычислительной машине и при всем это разрешает найти оптимальное решение. Под оптимальным решением понимается один из возможных вариантов решения данной проблемы такой, что будет иметь наивысшую оценку.

Практическим значением является информационная подсистема, которая позволяет найти оптимальный план деятельности предприятия, которое оценивается по нескольким критериям.

Список литературы: 1. Уланов Г.М. и др. Методы разработки интегрированных АСУ промышленными предприятиями. М.: Энергоатомиздат, 1983. 320с. 2. Таха Хэмди А. Введение в исследование операций-М.: Мир, 2001. 903с. 3. Вагнер Г. Основы исследования операций. М.: Мир, 1972. 329с. 4. Батищев Д.И. Методы оптимального проектирования. М.: Радио и связь, 1984. 246 с. 5. Понов И.Г. Математические методы в планировании отраслей. М., 1981. С. 54-83.

Поступила в редколлегию 05.03.2010

Гвоздинский Анатолий Николаевич, канд. техн. наук, профессор кафедры искусственного интеллекта ХНУРЭ. Научные интересы: оптимизация процедур принятия решений в сложных системах управления. Адрес: Украина, 61166, Харьков, ул. Акад. Ляпунова, 7, кв. 9, тел. 32-69-08.

Назаренко Роман Геннадиевич, студент группы ИСПР-06-1 ХНУРЭ. Научные интересы: методы принятия решений в системах искусственного интеллекта. Адрес: Украина, 61118, пр. 50 лет ВЛКСМ, 68, кв. 206, тел. 710-55-94.

УДК 519.7

А.Н. ГВОЗДИНСКИЙ, А.А. КУЛИКОВА

ОБ ОДНОМ ПОДХОДЕ К РЕШЕНИЮ ЗАДАЧ СОСТАВЛЕНИЯ РАСПИСАНИЯ В СИСТЕМАХ УПРАВЛЕНИЯ ОБЪЕКТАМИ ТРАНСПОРТНОГО ТИПА

Рассматривается задача составления расписания в системах управления объектами транспортного типа, в частности, воздушного транспорта, с учетом различных факторов, характеризующих эту систему. Также рассматриваются проблемы составления математических моделей в таких задачах и применение симплекс-метода с искусственным базисом.

1. Введение

В настоящее время в системах массового обслуживания особое место занимают задачи транспортного типа, среди которых – системы управления городским транспортом, железнодорожным, водным и воздушным. В каждой из этих систем есть свои особенности и специфика, однако общим для них является проблема математического моделирования и методы решения задач указанных типов с использованием современных математических и вычислительных средств.

Авиаперевозки – динамично развивающаяся отрасль мирового транспорта. Изменяется не только авиационная техника, но и методы, способы, модели авиационного бизнеса.

Актуальность данной задачи состоит в оптимизации маршрутов и оптимальном выборе используемых на них типов транспортных средств из доступных в парке с учетом

текущих грузо- и пассажиропотоков. Вследствие использования данной системы возможно уменьшение топливных, трудовых, прочих амортизационных затрат, требующихся на перевозку единицы груза, что не только сокращает расходы, но и дает возможность расширения деятельности перевозчиков без серьезных капиталовложений, а также благоприятно влияет на конкурентоспособность компании.

2. Цель исследования, описание объекта исследования, постановка задачи, составление математической модели

Целью данного исследования является анализ методов создания математических моделей и вычислительных алгоритмов для решения задачи составления расписания в системах управления объектами транспортного типа.

Описание объекта исследования. Различные типы самолетов гражданской авиации, например, такие как АН-74, ЯК-42, ТУ-154, А-300, Боинг-747 отличаются друг от друга такими показателями, как грузоподъемность, вместимость, себестоимость рейса, количество самолетов данного типа т.п. Каждый самолет, который выполняет соответствующий рейс на авиалинии, перевозит пассажиров и груз. Во время рейса он тратит некоторое количество ресурсов (топливо, смазочные материалы), которые составляют в стоимостном виде себестоимость рейса. Последняя зависит от типа самолета, который выполняет этот рейс. Поскольку тарифная плата не зависит от типа самолета, то прибыль авиакомпании зависит от назначения самолетов и авиалинии.

Задача исследования заключается в рассмотрении экстремальной задачи распределения самолетов между воздушными линиями при условии выполнения запланированных показателей перевозок при минимальной общей стоимости перевозок.

Составление математической модели. Для построения математической модели введем такие условные обозначения. Число самолетов j -го типа – n_j , количество типов самолетов – $j=\overline{1,n}$, количество авиалиний: $i=\overline{1,m}$, где i – номер авиалинии. Вместимость самолета j -го типа обозначим как R_j , а грузоподъемность – V_j , C_{ij} – себестоимость эксплуатации самолета j -го типа на i -й линии, а S_{ij} – максимальное количество рейсов, совершаемых самолетом j -го типа на i -й авиалинии. Введем также коэффициент исправности самолета j -го типа – k_j , a_i – план перевозок пассажиров по i -й авиалинии за единицу времени, d_i – план перевозок грузов. Искомая переменная X_{ij} – количество самолетов j -го типа, которые должны быть назначены на i -ю авиалинию.

Тогда модель распределения самолетов по авиалиниям будет иметь следующий вид. Функция цели (минимальная суммарная месячная стоимость эксплуатации самолетов):

$$F(\overline{X}) = \sum_{i=1}^m \sum_{j=1}^n C_{ij} X_{ij} S_{ij} \longrightarrow \min .$$

Определим ограничения, присутствующие в условии задачи. Ограничение по перевозке

пассажиров: $\sum_{j=1}^n S_{ij} R_j X_{ij} \geq a_i, i=\overline{1,m}$. Ограничение по перевозке грузов: $\sum_{j=1}^n S_{ij} V_j X_{ij} \geq d_i, i=\overline{1,m}$.

Хотя самолет перевозит и пассажиров и груз одновременно, ограничения по выполнению перевозок для пассажиров и для груза можно рассматривать отдельно, так как вместительность и грузоподъемность самолета независимы друг от друга. Также в качестве ограничений выступают ограничения на количество самолетов каждого типа, которые имеются у авиакомпании, и ограничение неотрицательности искомых переменных. Учитывая тот факт, что некоторые самолеты могут иметь неисправности, эти ограничения имеют следующий вид.

Ограничение по количеству действующих самолетов: $\sum_{i=1}^m X_{ij} \leq k_j n_j, j=\overline{1,n}$.

Условие неотрицательности и целочисленности: $X_{ij} \geq 0, X_{ij} = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$.

3. Применение симплекс метода с искусственным базисом для решения поставленной задачи

Данная задача является задачей линейного программирования, для решения которой применяется симплекс-метод с искусственным базисом, который является наиболее известным и широко применяемым на практике методом решения общей задачи линейного программирования. Этот алгоритм достаточно эффективен, однако имеет экспоненциальную сложность, причина которой заключается в комбинаторном характере симплекс-метода, последовательно перебирающего вершины многогранника допустимых решений при поиске оптимального решения.

Общая идея симплексного метода (метода последовательного улучшения плана) для решения задач линейного программирования заключается в следующих принципах.

1. Умение находить начальный опорный план.
2. Наличие признака оптимальности опорного плана.
3. Умение переходить к нехудшему опорному плану.

Пусть ЗЛП представлена системой ограничений в каноническом виде:

$$\sum_{j=1}^n a_{ij}x_j = b_i, \quad b_i \geq 0 \quad (i=1, \dots, m).$$

Ограничение ЗЛП имеет предпочтительный вид, если при неотрицательной правой части ($b_i \geq 0$) левая часть ограничений содержит переменную, входящую с коэффициентом, равным единице, а в остальные ограничения равенства – с коэффициентом, равным нулю.

Пусть система ограничений имеет вид:

$$\sum_{j=1}^n a_{ij}x_j \leq b_i, \quad b_i \geq 0 \quad (i=1, \dots, m).$$

Сведем задачу к каноническому виду. Для этого прибавим к левым частям неравенств дополнительные переменные $x_{n+1} \geq 0 \quad (i = 1, \dots, m)$. Получим систему, эквивалентную исходной:

$$\sum_{j=1}^n a_{ij}x_j + x_{n+1} = b_i, \quad b_i \geq 0 \quad (i=1, \dots, m),$$

которая имеет предпочтительный вид: $x_0 = (0; \underbrace{0; \dots; 0}_n; \underbrace{b_1; b_2; \dots; b_m}_m)$.

В целевую функцию дополнительные переменные вводятся с коэффициентами, равными нулю: $c_{n+1} = 0 \quad (i = 1, \dots, m)$.

Пусть далее система ограничений имеет вид: $\sum_{j=1}^n a_{ij}x_j \geq b_i, \quad b_i \geq 0 \quad (i=1, \dots, m)$.

Сведём её к эквивалентной вычитанием дополнительных переменных $x_{n+1} \geq 0 \quad (i = 1, \dots, m)$ из левых частей неравенств системы. Получим систему:

$$\sum_{j=1}^n a_{ij}x_j - x_{n+1} = b_i, \quad b_i \geq 0 \quad (i=1, \dots, m).$$

Однако теперь система ограничений не имеет предпочтительного вида, так как дополнительные переменные x_{n+1} входят в левую часть (при $b_i \geq 0$) с коэффициентами, равными -1 , поэтому базисный план $x_0 = (0; \dots; 0; \underbrace{-b_1; -b_2; \dots; -b_m}_n)$ не является допустимым. В этом случае вводится так называемый искусственный базис. К левым частям ограничений-равенств, не имеющих предпочтительного вида, добавляют искусственные переменные ω_i . В целевую функцию переменные ω_i вводят с коэффициентом M в случае решения

задачи на минимум и с коэффициентом M – для задачи на максимум, где M – большое положительное число. Полученная задача называется M -задачей, соответствующей исходной. Она всегда имеет предпочтительный вид.

Пусть исходная ЗЛП имеет вид: $F(\bar{X}) = \sum_{j=1}^n C_j X_j \longrightarrow \min$, $\sum_{j=1}^n a_{ij} x_j = b_i$, $b_i \geq 0$ ($i=1, \dots, m$),
 $x_j \geq 0$ ($j=1, \dots, n$),

причём ни одно из ограничений не имеет предпочтительной переменной. M -задача запишется так:

$$\begin{aligned} \max(\min) \bar{Z} &= \sum_{j=1}^n c_j x_j - (+) \sum_{i=1}^m M \omega_i, \\ \sum_{j=1}^n a_{ij} x_j + \omega_i &= b_i, \quad (i=1, \dots, m) \\ x_j &\geq 0 \quad (j=\overline{1, n}), \quad \omega_i \geq 0, \quad (i=1, \dots, m). \end{aligned}$$

Задача имеет предпочтительный план. Её начальный опорный план имеет вид:
 $x_0 = (\underbrace{0; 0; \dots; 0}_n; \underbrace{b_1; b_2; \dots; b_m}_m)$.

Если некоторые из уравнений имеют предпочтительный вид, то в них не следует вводить искусственные переменные.

Если в оптимальном плане M -задачи все искусственные переменные $\omega_i = 0$ ($i=1, \dots, m$), то план $x = (x_1; x_2; \dots; x_n)$ является оптимальным планом исходной задачи:

$$\bar{x} = (x_1; x_2; \dots; x_n; \omega_1; \omega_2; \dots; \omega_m).$$

Для того чтобы решить задачу с ограничениями, не имеющими предпочтительного вида, вводят искусственный базис и решают расширенную M -задачу, которая имеет начальный опорный план $\bar{x}_0 = (\underbrace{0; 0; \dots; 0}_n; b_1; b_2; \dots; b_m)$.

Решение исходной задачи симплексным методом путем введения искусственных переменных ω_i называется симплексным методом с искусственным базисом.

Если в результате применения симплексного метода к расширенной задаче получен оптимальный план, в котором все искусственные переменные $\omega_i^* = 0$, то его первые n компоненты дают оптимальный план исходной задачи.

Теорема. Если в оптимальном плане M -задачи хотя бы одна из искусственных переменных отлична от нуля, то исходная задача не имеет допустимых планов, т.е. ее условия несовместны.

Признаки оптимальности. Пусть исходная задача решается на максимум. Если для некоторого опорного плана все оценки Δ_j ($j=\overline{1, n}$) неотрицательны, то такой план оптимален. Если исходная задача решается на минимум и для некоторого опорного плана все оценки Δ_j ($j=\overline{1, n}$) неположительны, то такой план оптимален.

Была разработана программа, куда были введены исходные данные: целевая функция и соответствующие ограничения. Исходная симплекс – таблица приведена ниже.

Исходя из вычисленных оценок $\Delta_j = C_b \times P_j - C_j$, в которых содержатся положительные, делается вывод, что данный опорный план не является оптимальным. Номер ведущего столбца вычисляется из соотношения $\operatorname{argmax} \{ \Delta_j \}$, так как рассматривается задача на минимизацию. Номер ведущей строки выбирается как $\operatorname{argmax} \{ \Theta_i \}$, $\Theta_i = \min \left\{ \frac{P_0 [i]}{P_k [i]} \right\}$.

Дальнейшее решение предполагает использование итераций симплекс-метода для нахождения оптимального плана перевозок (рис. 1).

| № | B | Ci/Cj | X0=P0 | 1 | 2 | 3 | 4 | 5 | € |
|----|-----|-------|--------------|-----------|-----------|-----------|-----------|-----------|---|
| | | | | 350 | 480 | 282 | 440 | 260 | € |
| 1 | P27 | M | 2,8499999046 | 480 | 352 | 492 | 800 | 600 | € |
| 2 | P28 | M | 3,5 | 0 | 0 | 0 | 0 | 0 | € |
| 3 | P29 | M | 3 | 0 | 0 | 0 | 0 | 0 | € |
| 4 | P30 | M | 80 | 10 | 8 | 18 | 40 | 16 | € |
| 5 | P31 | M | 75 | 0 | 0 | 0 | 0 | 0 | € |
| 6 | P32 | M | 80 | 0 | 0 | 0 | 0 | 0 | € |
| 7 | P22 | 0 | 7,0019998550 | 1 | 0 | 0 | 0 | 0 | € |
| 8 | P23 | 0 | 14,010000228 | 0 | 1 | 0 | 0 | 0 | € |
| 9 | P24 | 0 | 12 | 0 | 0 | 1 | 0 | 0 | € |
| 10 | P25 | 0 | 5,0040001869 | 0 | 0 | 0 | 1 | 0 | € |
| 11 | P26 | 0 | 7 | 0 | 0 | 0 | 0 | 1 | € |
| 12 | Fj= | | 244,35000610 | 490M 0 | 360M 0 | 510M 0 | 840M 0 | 616M 0 | € |
| 13 | dj= | | | 490M -350 | 360M -480 | 510M -282 | 840M -440 | 616M -260 | € |

Рис. 1. Исходная симплекс-таблица

На седьмом шаге искусственные переменные выведены из базиса, нахождение оптимального плана продолжается согласно алгоритму симплекс-метода до получения оптимального плана. Основная задача исследования заключалась в рациональном распределении самолетов по воздушным авиалиниям, при котором обеспечивается минимум суммарных затрат, в результате проектирования разработанная программа представила следующий результат: $X^* = (0,0;0,2;0;0; 0; 0; 1.004,1.87;0,0,0,2,0..0)$.

Соответствующее ему значение целевой функции: $F(X^*) = 2238.40$.

В соответствии с задачей полученные данные округляются в сторону большего целого числа: $X^* = (0,0;0,2;0;0; 0; 0; 2,2;0,0,0,2,0..0)$, $F(X^*) = 2584$ дес. млн ден. ед.

В оптимальном плане было получено количество самолетов j-го типа, которые должны быть назначены на i-ю авиалинию (рис.2).

Согласно начальным данным, для минимизации общей стоимости перевозок 2 самолета типа А-300 должны быть назначены на первую авиалинию, 2 самолета типа А-300 – на вторую авиалинию, 2 самолета типа Боинг-747 – на вторую авиалинию, 2 самолета типа А-300 – на третью авиалинию (рис.2).

| № | B | Ci/Cj | X0=P0 | 1 | 2 | 3 | 4 | 5 | € |
|----|-----|-------|--------------|---------------|---------------|---------------|--------|---------------|---|
| | | | | 350 | 480 | 282 | 440 | 260 | € |
| 1 | P4 | 440 | 1,9999998807 | 0,2500000298 | 0,2000000029 | 0,4500000178 | 1 | 0,4000000059 | € |
| 2 | P10 | 360 | 1,8699992895 | 0,3125000298 | 0,25 | 0,5625 | 0 | 0,5 | € |
| 3 | P14 | 192 | 1,9999998807 | 0 | 0 | 0 | 0 | 0 | € |
| 4 | P16 | 0 | 1597,1499023 | -280 | -192 | -132 | 0 | -280 | € |
| 5 | P17 | 0 | 1159,9000244 | -56,25000381 | -44,99999618 | -101,25 | 0 | -89,99999237 | € |
| 6 | P18 | 0 | 1597 | 0 | 0 | 0 | 0 | 0 | € |
| 7 | P22 | 0 | 7,0019998550 | 1 | 0 | 0 | 0 | 0 | € |
| 8 | P23 | 0 | 14,010000228 | 0 | 1 | 0 | 0 | 0 | € |
| 9 | P24 | 0 | 12 | 0 | 0 | 1 | 0 | 0 | € |
| 10 | P9 | 300 | 1,0040005445 | -0,2500000029 | -0,2000000002 | -0,4500000017 | 0 | -0,4000000059 | € |
| 11 | P26 | 0 | 5,1300005912 | -0,312500029 | -0,25 | -0,5625 | 0 | 0,5 | € |
| 12 | Fj= | | 0M 2238,399 | 0M 147,5000 | 0M 118 | 0M 265,5 | 0M 440 | 0M 236 | € |
| 13 | dj= | | | 0M -202,4999 | 0M -362 | 0M -16,5 | 0M 0 | 0M -24 | € |

Рис. 2. Конечный результат работы программы

4. Выводы

В работе исследованы и предложены методы разработки математических моделей и вычислительных алгоритмов для решения задач составления расписания в системах управления объектами транспортного типа, особое внимание уделено оптимизационным методам поиска наилучших решений. На конкретных примерах (воздушный транспорт) исследуемые методы опробированы и была доказана их эффективность, также рассмотрены

вопросы использования методов для применения в различных системах управления объектами транспортного типа.

Научная новизна. Исследовано применение одного из самых универсальных методов решения задач линейного программирования – симплекс-метода с искусственным базисом – в системах управления воздушным транспортом для составления расписания полетов.

Практическое значение. Численное оптимальное решение позволяет детально проанализировать распределение финансовых затрат на перевозки авиакомпаниями, а также составить расписание полетов. При известных ценах на билеты и тарифах перевозки грузов возможно построение аналогичной математической модели, оптимизация которой позволит максимизировать прибыль авиакомпании. Возможность использования данной системы позволяет уменьшить топливные, трудовые, прочие амортизационных затраты, требующиеся на перевозку единицы груза, что не только сокращает расходы, но и дает возможность расширения деятельности перевозчиков без серьезных капиталовложений, а также благоприятно влияет на конкурентоспособность компании.

Список литературы: 1. *Сакович В.А.* Исследование операций (детерминированные методы и модели). Мн.: Высш. шк., 1985. 256с. 2. *Гвоздинський А.М., Якімова Н.А., Губін В.О.* Методи оптимізації в системах прийняття рішень: Навч. посібник. Харків: ХНУРЕ, 2006. 324с. 3. *Таха, Хемди А.* Введение в исследование операций, 7-е издание.: Пер. с англ. М.: Издательский дом «Вильямс», 2005. 912 с. 4. *Бондаренко М. Ф., Гвоздинський А. М.* Оптимізаційні задачі в системах прийняття рішень. Харків: ХТУРЕ, 1998. 216с. 5. *Гвоздинський А.М.* Оптимізаційні задачі в організаційному управлінні. У 3-х ч. Харків, ХТУРЕ, 1997. 6. *Зайченко Ю.П.* Исследование операций. 3-е изд., переработанное и дополненное. К.: Высш. шк., 1988. 552 с.

Поступила в редколлегию 05.03.2010

Гвоздинский Анатолий Николаевич, канд. техн. наук, профессор кафедры искусственного интеллекта ХНУРЭ. Научные интересы: оптимизация процедур принятия решений в сложных системах управления. Адрес: Украина, 61166, Харьков, ул. Акад. Ляпунова, 7, кв. 9, тел. 702-38-23.

Куликова Анна Александровна, студентка кафедры искусственного интеллекта ХНУРЭ. Научные интересы: интеллектуальный анализ данных, машинное обучение. Адрес: Украина, 61204, Харьков, ул. Ахсарова, 17, кв. 383.

УДК 519.7

А.Н. ГВОЗДИНСКИЙ, М.А. ГАВРЮШЕНКО

**ИССЛЕДОВАНИЕ И РАЗРАБОТКА МЕТОДОВ РЕШЕНИЯ ЗАДАЧ
ЗАКРЕПЛЕНИЯ ПОТРЕБИТЕЛЕЙ ЗА ПОСТАВЩИКАМИ С
УЧЕТОМ ВОЗВРАТА ТРАНСПОРТНЫХ СРЕДСТВ**

Исследуется алгоритм, с помощью которого возможно определить оптимальный план перевозок груза от поставщиков к потребителю и маршруты движения транспортных средств.

Введение

В настоящее время, в условиях развитого свободного рынка, острой проблемой является оптимизация производства и бизнеса. Из-за достаточно сильной конкуренции идет стремление минимизировать затраты на издержки, тем самым повысить конкурентоспособность товара или услуги.

Огромное количество возможных вариантов перевозок затрудняет получение достаточно экономного плана эмпирическим или экспертным путем. Применение математических методов и вычислительных возможностей в планировании перевозок дает большой экономический эффект.

Актуальность исследования. Транспортная задача является представителем класса задач линейного программирования, поэтому обладает всеми качествами линейных оптимизационных задач, но одновременно она имеет и ряд дополнительных полезных свойств, которые позволили разработать специальные методы ее решения.

Под названием “транспортная задача” объединяется широкий круг задач с единой математической моделью. Классическая транспортная задача – задача о наиболее экономном плане перевозок однородного продукта или взаимозаменяемых продуктов из пунктов производства в пункты потребления, встречается чаще всего в практических приложениях линейного программирования.

Среди задач транспортного типа в производственной деятельности особое значение имеет задача закрепления поставщиков за потребителями, которая получила в настоящее время широкое распространение в теоретических обработках и практическом применении на транспорте и в промышленности. Особенно большое значение она имеет в рационализации постановок важнейших видов промышленной и сельскохозяйственной продукции, а также оптимального планирования грузопотоков и работы различных видов транспорта.

При массовом перевозе грузов от производителя к потребителю возникает необходимость не только определить закрепление поставщиков за потребителями, но и выбор маршрутов перевозок таким образом, чтобы обеспечить минимум порожних пробегов и возврат транспортных средств. Такие задачи означают необходимость нахождения оптимального плана обратной доставки транспорта от потребителей к поставщикам после перевозки основного груза.

Целью данного исследования является оптимизация функционирования транспортных систем, уменьшение затрат на перевозку грузов от поставщиков к потребителям, расчет маршрутов движения транспортных средств, а также разработка алгоритма оптимизации с помощью исследуемого метода потенциалов и реализация его программными средствами.

Задача исследования – составить математическую модель в классе экстремальных задач, на основе которой определить нужды в автомобилях, которые необходимы для выполнения плана перевозок, обратные потоки транспортных средств и маршруты их движения.

1. Методы решения

Транспортные модели – специальный класс задач линейного программирования. Эти модели часто описывают перемещение (перевозку) какого-либо товара из пункта отправления (исходный пункт, например место производства) в пункт назначения (склад, магазин, грузохранилище). Назначение транспортной задачи – определить объем перевозок из пунктов отправления в пункты назначения с минимальной суммарной стоимостью перевозок. При этом должны учитываться ограничения, налагаемые на объемы грузов, имеющих в пунктах отправления (предложения), и ограничения, учитывающие потребность грузов в пунктах назначения (спрос). В транспортной модели предполагается, что стоимость перевозки по какому-либо маршруту прямо пропорциональна объему груза, перевозимого по этому маршруту. В общем случае транспортную модель можно применять для описания ситуаций, связанных с управлением запасами, управлением движением капиталов, составлением расписаний, назначением персонала и др.

Транспортную задачу, по сути, можно решить с помощью симплекс-метода. Однако специфическая структура условий задачи позволила разработать более эффективные вычислительные методы. Т-задачу можно решить распределенным методом, методом потенциалов, дифференциальных рент и т.д. Для решения Т-задач ручными методами необходимо пройти такие этапы: определить начальный опорный план задачи, оценить его, перейти к следующему, лучшему плану путем замены одной из базисной переменной на свободную переменную. Существует несколько простых методов нахождения начального опорного плана Т-задачи: северо-западного угла, минимальной стоимости, двойных меток, метод Фогеля и т.д.

Для разработки исследуемого алгоритма был использован метод потенциалов и метод минимального элемента.

2. Разработка математической модели и вычислительного алгоритма

Рассмотрим перевоз однородной продукции. Задача решается в 3 этапа.

1. С помощью модели закрепленных поставщиков за потребителями решаем обычную транспортную задачу без учета возврата транспортных средств:

$$F(X) = \sum_{i=1}^m \sum_{j=1}^n C_{ij} X_{ij} \rightarrow \min; \quad (1)$$

$$\sum_{j=1}^n X_{ij} = a_i, i=1, m; \quad (2)$$

$$\sum_{i=1}^m X_{ij} = b_j, j=1, n. \quad (3)$$

где C_{ij} – затраты на перевозку единицы продукции от i -го поставщика к j -му потребителю; a_i – количество вывозов груза от i -го поставщика; b_j – количество необходимого груза потребителю; X_{ij} – количество груза, перевезённого от i -го поставщика к j -му потребителю.

Решением этой задачи являются транспортные потоки X_{ij} между потребителем и поставщиком. Зная вместимость транспортного средства, можно определить количество транспортных средств, необходимых для перевоза груза объемом X_{ij} :

$$n_{ij} = \frac{X_{ij}}{a}, i=1, m; j=1, n; \quad (4)$$

количество транспортных средств, прибывших к j -му потребителю:

$$n_j = \sum_{i=1}^m n_{ij}, j=1, n; \quad (5)$$

количество транспортных средств, отправленных от i -го поставщика:

$$n_i = \sum_{j=1}^n n_{ij}, i=1, m. \quad (6)$$

2. С помощью решения обычной транспортной задачи определяем оптимальные обратные потоки транспортных средств:

$$F(X) = \sum_{i=1}^m \sum_{j=1}^n C_{ij} y_{ij} \rightarrow \min, \quad (7)$$

при ограничениях:

$$\sum_{i=1}^m y_{ji} = n_j, j=1, n; \quad (8)$$

$$\sum_{j=1}^n y_{ji} = n_i, i=1, m; \quad (9)$$

$$y_{ji} \geq 0, i=1, m \quad j=1, n. \quad (10)$$

Здесь C_{ij} – стоимость возврата единицы транспортного средства от j -го потребителя к i -му поставщику; Y_{ji} – количество транспортных средств, отправленных от j -го потребителя к i -му поставщику.

3. Определяем маршруты следования транспортных средств.

В результате решения транспортной задачи первого этапа получаем объемы поставок и потребления, а также оптимальные транспортные потоки грузов. Требуется определить потребность в автомобилях n_{ij} , n_i , n_j .

В качестве искомой переменной выберем X_{ij} – количество единиц перевозимого груза из i -го в j -й пункт.

Условие неотрицательности искомых переменных выполняется из природы самой задачи, т.е. отрицательных грузов, подлежащих перевозке не бывает.

Особенностью систем ограничений транспортной задачи, в отличие от общей задачи линейного программирования, является то, что матрица коэффициентов в левой части единична, а вектор констант правой части состоит из двух частей a_i и b_j и общее их количество $(m+n)$. Общее количество неизвестных всей таблицы X_{ij} равно mn , а число неизвестных составляющих маршрут включает в себя $(m+n) X_{ij} > 0$.

С учетом выражения $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ следует, что в системе $(m+n)$ равенств число независимых равенств равно $(m+n+1)$, следовательно, для решения задачи необходимо найти не $(m+n)$, а $(m+n+1)$ неизвестных – это и будет невырожденный план.

Для того чтобы найти любой опорный план, который всегда является допустимым, нам необходимо было бы путем полного перебора найти $(m-1)*(n-1)$ вариантов. Это комбинаторная задача. Поэтому среди всех опорных планов нужно находить такие, чтобы они позволяли на их множестве найти оптимальный. Учитывая, что Т-задачи не требуют предварительного определения ОДР, а любой опорный план является и допустимым, разработаны специальные методы построения исходных опорных планов.

С помощью известного метода минимального элемента определяем начальный опорный план X_0 и вычисляем предварительные потенциалы $v_j^{(0)}, u_i^{(0)}$.

По найденному плану X_0 строим схему Т-задачи из основных коммуникаций плана. Далее образуем следующие множества: J_1 – множество индексов всех пунктов V_j , которые связаны с пунктом A_1 основными коммуникациями; I_1 – множество индексов тех пунктов производства A_i , которые связаны с множеством J_1 ; наконец J_2 – множество индексов пунктов потребления, которые связаны основными коммуникациями с множеством J_1 и т.д.

Образование таких множеств продолжается до тех пор, пока не получат пустое множество.

Поскольку на выполнение условий оптимального плана влияют лишь разности $v_j - u_i$, то за начало отсчета (нуль) можно принять потенциал любого из пунктов.

Полагаем для определенности $u_1^{(0)} = 0$ и вычислим систему потенциалов относительно A_1 . Тогда $v_j^{(0)} = c_{1j} - u_1^{(0)} = c_{1j}$, где $j \in J_1$.

По значениям $v_j^{(0)}$ ($j \in J_1$) определяем потенциалы пунктов A_i ($i \in I_1$):

$$u_i^{(0)} = v_j^{(0)} - c_{ij} \quad (j \in J_1; i \in I_1). \quad (11)$$

Аналогично вычисляем потенциалы $v_j^{(0)}$ ($j \in J_2$) и т.д. После того, как потенциалы всех пунктов найдены, строим матрицу

$$C_1 = \left\| c_{ij} - (v_j^{(0)} - u_i^{(0)}) \right\|. \quad (12)$$

Очевидно, позиции матрицы C_1 , отвечающие базисным элементам плана X_0 , будут заняты нулями.

Если матрица C_1 не содержит отрицательных элементов, то X_0 – оптимальный план. В противном случае X_0 – неоптимальный план, который может быть улучшен. Тогда переходят к выполнению однотипных итераций.

(k+1)-я итерация. Каждая итерация кроме первой, где отсутствует первый этап, состоит из двух этапов. Предположим, что уже проведено k итераций ($k=1, \dots$), в результате которых получен план X_k и матрица C_k .

Цель (k+1)-й итерации – построение матрицы C_{k+1} , а также либо установление оптимальности плана X_k , либо нахождения более экономичного плана X_{k+1} .

Вычисляем матрицу C_{k+1} . Преобразование матрицы C_k в матрицу C_{k+1} состоит в следующем. Выбираем наибольший по модулю отрицательный элемент матрицы C_k . Пусть это

элемент $c_{\lambda\mu}^k = \Delta_k$. Выделяем строку, в которой содержится элемент $c_{\lambda\mu}^k$, а множество

существенных элементов этой строки, не совпадающих с данным элементом, обозначают G_1 . При этом X_k – существенными элементами называют те элементы матрицы C_k , которые отвечают базисным перевозкам плана X_k .

Затем выделяем столбцы матрицы C_k , которые содержат элементы множества G_1 . Множество X_k – существенных элементов, которые находятся в столбцах матрицы C_k и отличны от G_1 элементов множества, обозначают G_2 .

Процесс выделения продолжается до тех пор, пока очередное множество не окажется пустым. Поскольку каждая строка и столбец не могут быть выделены дважды, то весь процесс заканчивается за $l=m+n-1$ шагов.

Далее строим матрицу C_{k+1} . Для этого величину Δ_k прибавляем ко всем выделенным столбцам и вычитаем из всех выделенных строк матрицы C_k . При этом все выделенные X_k – существенные элементы матрицы C_k остаются равными нулю, а кроме того, в нуль превращается и элемент $c_{\lambda\mu}$.

Если все элементы матрицы C_{k+1} окажутся неотрицательными, то X_k – оптимальный план, и на этом процесс заканчивается. В противном случае производим улучшение плана X_k . Выбираем наибольший по модулю отрицательный элемент матрицы C_{k+1} . Пусть это элемент $c_{i_0j_0}^{(k+1)} = \Delta_{k+1} < 0$. Затем составляем, применив, например метод вычеркивания, цепочку из положительных элементов плана X_k , которая замыкается на $x_{i_0j_0}$.

После того как цепочка построена, в ней находим минимальный нечетный по порядку следования элемент:

$$\Theta_{k+1} = \min_{0 \leq \mu \leq S} x_{i\mu j\mu+1}^{(k)} \quad (13)$$

Прибавляем Θ_{k+1} ко всем четным элементам цепочки и к элементу $x_{i_0j_0}$ и вычитаем Θ_{k+1} из всех нечетных элементов. Остальные элементы X_k оставляем без сомнения.

Новый план X_{k+1} построен. Он является опорным, так как число его ненулевых перевозок не изменилось.

Пусть L_k – величина транспортных издержек, отвечающих плану X_k . Тогда новое значение целевой функции, отвечающее плану X_{k+1} , находим по отношению:

$$L_{k+1} = L_k + \Theta_{k+1} \Delta_{k+1}. \quad (14)$$

Так как $\Theta_{k+1} > 0$ и $\Delta_{k+1} < 0$, то $L_{k+1} < L_k$. Поэтому X_{k+1} является улучшенным опорным планом.

Затем производим аналогичную (k+2)-ю итерацию.

Выводы

Научную новизну имеют следующие результаты исследования:

- 1) Сформулированы и обоснованы основные методы повышения эффективности перевозки грузов с минимальными затратами.
- 2) Предложен метод анализа маршрутов транспортных средств от поставщиков к потребителям и обратно.
- 3) Разработана оптимальная схема грузовых потоков, которая позволяет снизить затраты на перевозки.
- 4) Предложен вычислительный алгоритм, его программная реализация с использованием современных вычислительных средств. Вычислительный алгоритм разработан в среде Visual Studio 2008. Программа выполняет такие действия: находит опорный план перевозок по методу минимального элемента, а также опорный план возврата по методу минимального элемента, оптимальный план перевозок по методу потенциалов, оптимальный план возврата по методу потенциалов, находит потребности в автомобилях, считает маршруты для автомобилей.

Результаты программы представлены на рисунке.

Практическое значение. После исследования алгоритмов решения транспортных задач можно сказать, что они являются мощным математическим аппаратом для решения определённого рода задач, реализация которых не занимает много времени. Такие математические аппараты могут быть использованы в различных системах транспортного типа различного назначения, так как с их помощью можно решить поставленную задачу с минимальными затратами, т.е. найти самое оптимальное решение.

Без оптимизации, чтобы обеспечить маятниковые поставки, требуется практически в два раза больше автомобилей, чем с оптимизацией. С помощью вычислительного алгоритма, который реализован программно (рисунок), можно легко посчитать маршруты от поставщиков к потребителям, количество транспорта, необходимого для совершения маятниковых поставок, а также снизить затраты на автомобили и на перевозки.

Запасы поставщиков A1 ... AN: 100 15 70 55 20 Груза в автомобиле: 5

Заявки получателей B1 ... BM: 30 40 55 55 45 10 25

Стоимость перевозок (A x B):
 3 5 9 7 4 6 5
 3 6 8 4 3 3 4
 8 10 4 5 6 8 6
 7 3 7 8 5 3 8
 9 4 3 4 6 7 6

Стоимость возврата (B x A):
 1 4 4 1 2
 3 2 2 5 1
 4 3 3 7 3
 2 3 5 5 1
 4 3 2 2 3

Посчитать

Оптим. план перев. (A x B): 1020 Потребность в автомобилях (A x B): Оптим. план возвр. (B x A): 103

3 0 0 0 0 4 5 0 2 5
 0 0 0 1 5 0 0 0
 0 0 3 5 3 5 0 0 0
 0 4 0 0 5 0 1 0 0
 0 0 2 0 0 0 0 0 0

6 0 0 0 9 0 5
 0 0 0 3 0 0 0
 0 0 7 7 0 0 0
 0 8 0 1 0 2 0
 0 0 4 0 0 0 0

2 0 0 4 0
 0 3 1 0 4
 0 0 1 1 0 0
 1 1 0 0 0 0
 0 0 2 7 0

Маршруты для автомобилей:

Все маршруты:
 A1 → B1 = 6 ; A1 → B5 = 9 ; A1 → B7 = 5 ; A2 → B4 = 3 ; A3 → B3 = 7 ; A3 → B4 = 7 ; A4 → B2 = 8 ; A4 → B4 = 1 ;
 A4 → B6 = 2 ; A5 → B3 = 4 ; B1 → A1 = 2 ; B1 → A4 = 4 ; B2 → A2 = 3 ; B2 → A3 = 1 ; B2 → A5 = 4 ; B3 → A3 = 11 ;
 B4 → A1 = 11 ; B5 → A3 = 2 ; B5 → A4 = 7 ; B6 → A1 = 2 ; B7 → A1 = 5

Маятниковые маршруты, которые можно исключить:
 A1 → B1 = 2 ; B1 → A1 = 2 ; A1 → B7 = 5 ; B7 → A1 = 5 ; A3 → B3 = 7 ; B3 → A3 = 7

Оставшиеся маршруты:
 A1 → B1 = 4 ; A1 → B5 = 9 ; A2 → B4 = 3 ; A3 → B4 = 7 ; A4 → B2 = 8 ; A4 → B4 = 1 ; A4 → B6 = 2 ; A5 → B3 = 4 ;

Экранная форма выполнения программы

Список литературы: 1. *Вентцель Е.С.* Исследование операций. М.: 1972. 552 с. 2. *Бондаренко М. Ф., Гвоздинський А. М.* Оптимізаційні задачі в системах прийняття рішень. Харків: ХТУРЕ, 1998. 216с. 3. *Гвоздинський А.М., Якімова Н.А., Губін В.О.* Методи оптимізації в системах прийняття рішень: Навч. посібник. Харків: ХНУРЕ, 2006. 324с. 4. *Большакова И.В., Кураленко М.В.* Линейное программирование. Мн., 2004. 148 с.

Поступила в редколлегию 05.03.2010

Гвоздинский Анатолий Николаевич, канд. техн. наук, профессор кафедры искусственного интеллекта ХНУРЭ. Научные интересы: оптимизация процедур принятия решений в сложных системах управления. Адрес: Украина, 61166, Харьков, ул. Ак. Ляпунова, 7, кв. 9, тел. 702-38-23

Гаврюшенко Мария Александровна, студентка группы ИСПР-06-1 ХНУРЭ. Научные интересы: теория принятия решений, методы принятия решений в системах искусственного интеллекта. Адрес: Украина, 61140, Харьков, пр.Гагарина, 40, кв. 59.

УДК 519.7

О.Н. БЕЗГИНОВА, А.Н. ГВОЗДИНСКИЙ, М.Д. ПОВАЖНАЯ

ИССЛЕДОВАНИЕ И РАЗРАБОТКА МЕТОДОВ РЕШЕНИЯ ЗАДАЧ ПРОИЗВОДСТВЕННОГО ПЛАНИРОВАНИЯ

Исследуются методы решения задач в производственном планировании современного промышленного предприятия. Предлагается оптимизационный метод решения задач производственного планирования, позволяющий оптимально распределять ресурсы и выявлять «узкие места» предприятия.

1. Введение

Актуальность исследования. В современном мире человек, стараясь сделать все наилучшим образом, в основу любой целенаправленной деятельности закладывает процессы принятия решений. Теория принятия решений – быстро развивающаяся наука. Задачи,

которыми она занимается, порождены практикой управленческих решений на различных уровнях - от отдельного подразделения или малого предприятия до государств и международных организаций. Процессы принятия решений являются важным этапом при проектировании новой техники, разработке технологии ее создания и последующей эксплуатации. В экономике – обеспечивают оптимальное функционирование и взаимодействие производственных и хозяйственных организаций. Именно на этой области будет более подробно рассмотрено применение данных процессов, так как задача планирования организационной деятельности предприятий приобретает все более широкие масштабы.

Совершенствование производственного планирования представляет собой первоочередную актуальную задачу при управлении современным предприятием. В условиях жесткой конкуренции промышленные предприятия сталкиваются с проблемой эффективного размещения производственных ресурсов в целях получения наибольшей прибыли, расходуя при этом минимум затрат. Именно поэтому так высока роль построения такой стратегии управления ресурсами, которая была бы направлена на формирование оптимальной и эффективной производственной программы предприятия в условиях динамично меняющейся экономической среды и большой конкуренции.

Состояние проблемы. Ввиду ограниченности производственных ресурсов и времени, в управлении современным предприятием большое значение имеет рациональное их использование, которое определяет комплекс задач производственного планирования. К данному комплексу относятся такие важные задачи, как решение вопросов себестоимости, уровня загруженности оборудования, фонда заработной платы, производственных площадей, ассортимента и объема выпускаемой продукции. Для решения этих задач разработано множество методов планирования организационной деятельности предприятия. Однако мы должны помнить, что необходимо не просто составить оптимальную производственную программу (план выпуска продукции) на планируемый период так, чтобы полностью удовлетворить спрос потребителей. Также требуется правильно проанализировать полученное решение, выявив «узкие места» на производстве, и дать рекомендации по их разрешению. Ведь нередко бывает так, что после анализа выявляется либо существование дефицита определенного вида продукции, либо то, что выпуск продукции данного типа является экономически нецелесообразным.

Цель исследования – рассмотреть существующие методы решения задач принятия решений в управлении производственным планированием, на основе которого создать оптимальную производственную программу работы предприятия.

Необходимо рассмотреть деятельность производственного предприятия, определить эффективные методы планирования для его производственной деятельности.

Основными задачами данного исследования являются:

– экспериментальное моделирование оптимизации производственного процесса на предприятии. Процесс оптимизации должен включать: определение критериев оценки эффективности производственного процесса, формулировку задач оптимизации на основе данных критериев, исследование, выбор и применение методов для решения соответствующих задач;

– анализ полученного решения, выявление «узких мест» на производстве; дать рекомендации по их разрешению;

– разработка программного продукта, позволяющего практически реализовать поставленные задачи.

Объектом исследования данной работы является производственный процесс некоторого предприятия при заданных ограничениях таких производственных ресурсов, как фонд производственного времени, затраты материалов на производство продукции, комплектующие изделия, затраты труда, фонд заработной платы, объем выпуска и себестоимость продукции.

2. Формальная постановка задачи

Для решения поставленной задачи следует составить математическую модель производственного процесса предприятия, т.е. сформировать систему ограничений, основанную на имеющихся производственных ресурсах, и сформировать функции цели, нахождение оптимальных значений которых приведет к получению оптимальной программы производ-

ства. При составлении функций цели должны быть учтены все показатели эффективности работы предприятия. Для выявления узких мест на предприятии следует составить двойственную задачу для каждой функции цели. После того, как математическая модель будет сформирована, для получения окончательных результатов следует применить методы линейного программирования, в частности симплекс-метод.

Рассмотрим на конкретном практическом примере построение математической модели некоторого предприятия.

Значения ограничений ресурсов рассматриваемого предприятия приведены в табл. 1, 2.

Таблица 1. Плановые задания

| Лимитированные ресурсы и показатели | Номенклатура изделий | | | | |
|-------------------------------------|----------------------|----------------|----------------|----------------|----------------|
| | B ₁ | B ₂ | B ₃ | B ₄ | B ₅ |
| Ограничения по ассортименту | ≥ 10000 | ≤ 30000 | ≥ 15000 | ≥ 20000 | ≥ 30000 |

Таблица 2. Нормы затрат ресурсов

| Лимитированные ресурсы и показатели | Номенклатура изделий | | | | | R _j , x 10 ⁵ |
|---|----------------------|----------------|----------------|----------------|----------------|---------------------------------------|
| | B ₁ | B ₂ | B ₃ | B ₄ | B ₅ | |
| Фонд производственного времени, станко-часов | 2 | 3 | 1 | 1 | 3 | 9 |
| Материалы, кг | 1 | 2 | 3 | 4 | 2 | 8 |
| Комплекующие изделия, шт. | 3 | 1 | 3 | 2 | 4 | 6 |
| Трудозатраты, часов | 3 | 4 | 1 | 2 | 3 | 5 |
| Фонд заработной платы | 3 | 2 | 3 | 4 | 1 | 9,5 |
| Выпуск продукции в стоимостном выражении, грн | 10 | 12 | 8 | 16 | 11 | 25 |
| Себестоимость, грн | 3 | 4 | 4 | 2 | 1 | |

Обозначим количество каждой i-й продукции, которую выпускает данное предприятие, как X_i, фонд производственного времени – T_i, материалы – M_i, комплекующие изделия – k_i, трудозатраты – L_i, фонд заработной платы – S_i, выпуск продукции в стоимостном выражении – P_i, себестоимость – C_i, ограничение по ассортименту – A_i, объем ресурсов – R_j.

Количество каждой i-й продукции, которую выпускает данное предприятие – X_i.

Ограничение по фонду производственного времени в общем виде представлено формулой

$$\sum_{i=1}^5 T_i * X_i \leq R_1.$$

Подставив значения из табл. 1, мы получим первое уравнение системы ограничений: $2x_1 + 2x_2 + x_3 + x_4 + 3x_5 \leq 9 * 10^5$.

Ограничение по объему используемых материалов в общем виде представлено формулой

$$\sum_{i=1}^5 M_i * X_i \leq R_2.$$

Запишем второе уравнение системы ограничений: $x_1 + 2x_2 + 3x_3 + 4x_4 + 2x_5 \leq 8 * 10^5$.

Ограничение по количеству комплекующих изделий в общем виде представлено формулой

$$\sum_{i=1}^5 k_i * X_i \leq R_3.$$

Таким образом, третье уравнение системы ограничений будет иметь вид:

$$3x_1 + x_2 + 3x_3 + 2x_4 + 4x_5 \leq 6 * 10^5.$$

Ограничение по затратам труда в общем виде представлено формулой $\sum_{i=1}^5 L_i * X_i \leq R_4$.

Четвертое уравнение системы ограничений будет иметь вид:

$$3x_1 + 4x_2 + x_3 + 2x_4 + 2x_5 \leq 5 * 10^5.$$

Ограничение по фонду заработной платы труда в общем виде представлено формулой

$$\sum_{i=1}^5 S_i * X_i \leq R_5.$$

Следующее уравнение системы ограничений будет иметь вид:

$$3x_1 + 2x_2 + 3x_3 + 4x_4 + \dots + 1x_5 \leq 9,5 \cdot 10^5.$$

Ограничение по выпуску продукции в стоимостном выражении в общем виде представлено формулой $\sum_{i=1}^5 P_i \cdot X_i \leq R_6$.

И, наконец, шестое уравнение системы ограничений будет иметь вид:

$$10x_1 + 12x_2 + \dots + 8x_3 + 16x_4 + 11x_5 \leq 25 \cdot 10^5.$$

Данные об ассортименте дают нам следующие ограничения:

$$x_1 \geq 10000, x_2 \leq 30000, x_3 \geq 15000, x_4 \geq 20000, x_5 \geq 30000.$$

Таким образом, система ограничений для данной задачи имеет вид:

$$\begin{aligned} 2x_1 + 2x_2 + x_3 + x_4 + 3x_5 &\leq 9 \cdot 10^5, \\ x_1 + 2x_2 + 3x_3 + 4x_4 + 2x_5 &\leq 8 \cdot 10^5, \\ 3x_1 + x_2 + 3x_3 + 2x_4 + 4x_5 &\leq 6 \cdot 10^5, \\ 3x_1 + 4x_2 + x_3 + 2x_4 + 2x_5 &\leq 5 \cdot 10^5, \\ 3x_1 + 2x_2 + 3x_3 + 4x_4 + x_5 &\leq 9,5 \cdot 10^5, \\ 10x_1 + 12x_2 + 8x_3 + 16x_4 + 11x_5 &\leq 25 \cdot 10^5, \\ x_1 \geq 10000, x_2 \leq 30000, x_3 \geq 15000, \\ x_4 \geq 20000, x_5 \geq 30000. \end{aligned}$$

При этом необходимо учитывать условие $X_i \geq 0, i = \overline{1,5}$.

Теперь составим функции цели, которые позволят нам определить наиболее оптимальное использование ресурсов и оптимизировать работу предприятия.

Целью работы каждого предприятия является получение максимально возможной прибыли. Этого можно добиться рядом способов, например, максимизируя объем выпуска продукции, минимизируя себестоимость, максимизируя производственное время.

Исходя из сказанного, запишем функции цели для данной задачи.

В качестве первой функции цели будет выступать максимизация прибыли предприятия.

Запишем ее в общем виде: $\sum_{i=1}^5 (P_i - C_i) \cdot X_i \rightarrow \text{MAX}$.

Для конкретных значений функция цели будет иметь вид:

$$F_1(X) = 7x_1 + 8x_2 + 4x_3 + \dots + 14x_4 + 10x_5 \rightarrow \text{MAX}.$$

Вторая функция цели будет выражать максимизацию валового объема выпуска продукции в денежном выражении. Запишем ее в общем виде $\sum_{i=1}^5 P_i \cdot X_i \rightarrow \text{MAX}$.

Для конкретных значений функция цели примет вид:

$$F_2(X) = 10x_1 + 12x_2 + 8x_3 + 16x_4 + \dots + 11x_5 \rightarrow \text{MAX}.$$

Третьей функцией цели представим минимизацию себестоимости, которая имеет общий вид: $\sum_{i=1}^5 C_i \cdot X_i \rightarrow \text{MIN}$.

Запишем эту функцию с конкретными значениями: $F_3(X) = 3x_1 + 4x_2 + 4x_3 + 2x_4 + \dots + x_5 \rightarrow \text{MIN}$.

И наконец, в роли четвертой функции цели будет выступать максимизация производственного времени: $\sum_{i=1}^5 T_i \cdot X_i \rightarrow \text{MAX}$ и $F_4(X) = 2x_1 + 2x_2 + x_3 + x_4 + 3x_5 \rightarrow \text{MAX}$.

Совокупность системы ограничений и функций цели составляет математическую модель задачи оптимизации производственного процесса данного предприятия. Чтобы решить поставленную задачу, следует найти оптимальное решение для каждой функции цели.

3. Методы исследования

Поскольку поставленная задача относится к классу экстремальных задач, для ее решения целесообразным будет применение методов математического программирования.

Математическое программирование (оптимальное программирование) — область прикладной математики, объединяющая различные математические методы и дисциплины. Это математическая дисциплина, изучающая теорию и методы решения задач о нахождении экстремумов функций на множествах конечномерного векторного пространства, определяемых линейными и нелинейными ограничениями (равенствами и неравенствами).

Общая задача математического программирования состоит в нахождении оптимального (максимального или минимального) значения целевой функции, причем значения переменных должны принадлежать некоторой области допустимых значений.

Методы исследования зависят непосредственно от рода деятельности того или иного предприятия. Ведь в зависимости от этого мы стараемся добиться разных целей: то ли мы находим оптимальный состав смеси, то ли вычисляем наиболее выгодный путь доставки продукции, то ли распределяем самолеты на посадочных полосах и т.д. Вспомним также, что в зависимости от задач предприятия применяются различные типы решаемых задач: задача о назначении, о смесях, о распределении ресурсов. Однако все они сводятся к одному разделу: задачам логического программирования.

Существуют следующие виды математического программирования: линейное, нелинейное, дискретное, динамическое, выпуклое программирование и др.

Наиболее известным и широко применяемым на практике для решения общей задачи линейного программирования является симплекс-метод. Симплекс-метод, известный также под названием метода последовательного улучшения плана, позволяет переходить от одного допустимого базисного решения к другому, причем так, что значения целевой функции непрерывно возрастают. В результате оптимальное решение находят за конечное число шагов (исключая вырожденные задачи, когда наблюдается явление «зацикливания»). Для выбранного объекта исследования симплекс-метод позволяет дать оценку полученному решению, на базе которой можно оценить производственный план предприятия, выявив «узкие места» на производстве и дав рекомендации по их разрешению.

Нелинейное программирование - это раздел математического программирования, изучающий методы решения таких экстремальных задач, в которых результаты возрастают или убывают непропорционально изменению масштабов использования ресурсов. В краткой форме задачу нелинейного программирования можно записать так: $\max G(x)$ при условиях $g(x) \leq b$, $x \geq 0$, где x – вектор искомых переменных, $G(x)$ – целевая функция, $g(x)$ – функции ограничений, b – вектор констант ограничений.

Иначе говоря, задача состоит в выборе таких неотрицательных значений переменных, подчиненных системе ограничений в форме неравенств, при которых достигается максимум или минимум данной функции. При этом не оговаривается форма ни целевой функции, ни неравенств. Возможны разные случаи: целевая функция нелинейна, а ограничения линейны; целевая функция линейна, а ограничения — нелинейны; и целевая функция, и ограничения нелинейны. Нелинейные задачи сложны, часто их упрощают тем, что приводят к линейным. Для этого условно принимают, что на том или ином участке целевая функция возрастает или убывает пропорционально изменению независимых переменных.

В задачах дискретного программирования переменные принимают только дискретные значения, например, целочисленные. Одними из основных методов решения задач дискретного программирования являются метод ветвей и границ и динамическое программирование.

Метод ветвей и границ — общий алгоритмический метод для нахождения оптимальных решений различных задач оптимизации, особенно дискретной и комбинаторной оптимизации. По существу, метод является вариацией полного перебора с отсевом подмножеств допустимых решений, заведомо не содержащих оптимальных решений.

Динамическое программирование в математике и теории вычислительных систем — метод решения задач с оптимальной подструктурой и перекрывающимися подзадачами. Динамическое программирование обычно придерживается двух подходов к решению задач:

– нисходящее динамическое программирование: задача разбивается на подзадачи меньшего размера, они решаются и затем комбинируются для решения исходной задачи. Используется запоминание для решений часто встречающихся подзадач;

$$\begin{aligned}
2y_1 + y_2 + 3y_3 + 3y_4 + 3y_5 + 10y_6 - y_7 &\geq 10, \\
2y_1 + 2y_2 + y_3 + 4y_4 + 2y_5 + 12y_6 + y_8 &\geq 12, \\
y_1 + 3y_2 + 3y_3 + y_4 + 3y_5 + 8y_6 - y_9 &\geq 8, \\
y_1 + 4y_2 + 2y_3 + 2y_4 + 4y_5 + 16y_6 - y_{10} &\geq 16, \\
3y_1 + 2y_2 + 4y_3 + 2y_4 + y_5 + 11y_6 - y_{11} &\geq 11, \\
Y &\geq 0.
\end{aligned}$$

Решение двойственной задачи получается из $m+1$ строки оптимального плана и имеет вид: $Y^* = \| 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0 \|$.

Проанализируем полученный результат. Двойственная оценка первой переменной – количества выпускаемой продукции первого типа – равна 0. Это означает, что количество выпускаемой продукции этого типа находится на оптимальном уровне, и дальнейшее увеличение выпуска данной продукции не повлияет на значение целевой функции. Если бы двойственная оценка переменной была больше нуля, это свидетельствовало бы о том, что данный ресурс полностью используется, т. е. о его дефиците.

4. Выводы

В ходе данного исследования были рассмотрены методы решения задач производственного планирования. На конкретном примере рассмотрено решение задачи данного класса. Построена математическая модель задачи оптимизации производственного процесса современного предприятия. Рассмотрены методы решения оптимизационных задач, из которых выбран и применен наиболее подходящий метод для решения построенной модели.

Научной новизной результатов исследований, приведенных в статье, является метод решения полученной математической модели производственного процесса современного предприятия с учетом таких производственных ресурсов, как фонд производственного времени, затраты материалов на производство продукции, комплектующие изделия, затраты труда, фонд заработной платы, объем выпуска и себестоимость продукции. На основе обширного анализа методов решения задач производственного планирования предложен вычислительный алгоритм, в основе которого лежит симплекс-метод. С помощью данного метода можно не только получить план эффективной производственной программы предприятия, но также дать оценку его работы.

Практическое значение. Исследование задач производственного планирования позволяет глубже рассмотреть проблемы оптимизации производственного процесса на современном предприятии. Нахождение оптимального плана производства позволяет повысить такие показатели его эффективности, как прибыль предприятия, валовой объем выпуска продукции, себестоимость, уровень загруженности оборудования. Полученные результаты исследования могут найти применение при решении реальных задач в будущем. Разработанная программная реализация соответствующего метода может быть применена при планировании работы как малых частных фирм, так и крупных государственных предприятий. Благодаря этому будет достигнуто получение наибольшей прибыли при минимальном количестве затрат. Также будет обеспечена высокая конкурентоспособность предприятия, что очень важно в условиях современного рынка. Рассмотрен конкретный пример, показана сфера его использования.

Список литературы: 1. Угланов Г.М и др. Методы разработки интегрированных АСУ промышленными предприятиями. М.: Энергоатомиздат, 1983. 380с. 2. Таха, Хэмди А. Введение в исследование операций. М.: Мир, 2001. 400с. 3. Карманов В.Г. Математическое программирование. М.: Наука, 1980. 256с. 4. Химмельблау Д. Прикладное нелинейное программирование: Пер. с англ. М.: Мир, 1975. 534с.

Поступила в редколлегию 05.03.2010

Безгинова Ольга Николаевна, студентка факультета КН ХНУРЭ. Адрес: Украина, 61085, Харьков, ул. Астрономическая, 44а, кв. 15, тел. 720-56-19.

Гвоздинский Анатолий Николаевич, канд. техн. наук, профессор кафедры искусственного интеллекта ХНУРЭ. Научные интересы: оптимизация процедур принятия решений в сложных системах управления. Адрес: Украина, 61166, Харьков, ул. Ак. Ляпунова, 7, кв. 9, тел. 702-38-23.

Поважная Марина Дмитриевна, студентка факультета КН ХНУРЭ. Адрес: Украина, 61033, Харьков, ул. Саперная, 20, кв. 121, тел. 717-26-45.

**МОДЕЛИРОВАНИЕ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА
ФОРМИРОВАНИЯ ПОВЕРХНОСТНОГО СЛОЯ
ОБРАБАТЫВАЕМЫХ ДЕТАЛЕЙ С УЧЁТОМ ПОСЛЕДУЮЩИХ
ОПЕРАЦИЙ**

Моделируется технологический процесс механической обработки высокопрочных деталей на станках с учетом режимов и выходных параметров готового изделия и влияние качества поверхностного слоя на последующие операции.

1. Введение

Изготовление деталей для различных приборов автоматики в значительной степени определяется финишными операциями формообразования и режимами их обработки [1]. Поэтому возникает необходимость моделирования процесса в целях нахождения оптимальных условий для формирования поверхностного слоя при механической обработке. Обработка деталей из известных материалов производилась при исследовании следующих параметров процесса.

Исследовались следующие параметры: P – удельное давление (кг/см^2); v – линейная скорость перемещения деталей (м/мин); t – продолжительность обработки (мин); $R_{\text{аисх}}$ – исходная шероховатость поверхности детали (мкм); z – зернистость пасты (мкм); R_a – достигаемая шероховатость поверхности (мкм); Q – производительность процесса (съем поверхностного слоя) (мкм/мин).

Из предусмотренных выходных параметров (R_a и Q) за главный принималась достигаемая шероховатость поверхности деталей (R_a), которая определяет качество поверхности деталей и зависит от входных параметров ($P, V, t, z, R_{\text{аисх}}$).

2. Задача

Исследовать зависимость $R_a = f(P, V, t, R_{\text{аисх}}, z)$ в целях получения уравнения регрессии и последующего использования его при определении оптимальных режимов обработки с учетом обеспечения наиболее высокой производительности процесса.

Для выяснения зависимости $R_a = f(P, V, t, R_{\text{аисх}}, z)$ необходимо провести эксперимент с последующим статическим анализом полученных результатов. Указанную задачу можно решать традиционными методами планирования эксперимента [2]. Однако для сокращения количества опытов применялась теория подобия [2]. Исходя из этого предусматривались следующие этапы решения задачи:

- 1) приведение зависимости $R_a = f(P, V, t, R_{\text{аисх}}, z)$ к критериальному виду;
- 2) проведение экспериментов и обработка их результатов в целях получения регрессии для R_a ;
- 3) анализ уравнения регрессии;
- 4) определение путей использования полученной зависимости.

2. Решение задачи

1. Приведение зависимости $R_a = f(P, V, t, R_{\text{аисх}}, z)$ к критериальному виду.
Введем обозначение: $[F]$ – размерность силы; $[L]$ – размерность длины; $[T]$ – размерность времени.

В уравнении $R_a = f(P, V, t, R_{\text{аисх}}, z)$ размерность параметров такова:

$$[R_a] = [L]; [P] = [F] \cdot [L]^{-2}; [V] = [L] \cdot [T]^{-1}; [t] = [T]; [R_{\text{аисх}}] = [L]; [z] = [L]. \quad (1)$$

Так как только параметр P содержит $[F]$, то безразмерные критерии подобия не могут быть получены (нет параметра, «компенсирующего» размерность). Для того чтобы полу-

читать безразмерные критерии подобия, введем в число независимых переменных «компенсирующий» параметр μ :

$$\mu = 1 \cdot F.$$

Тогда зависимость (1) принимает вид

$$R_a = f(P, V, t, R_{\text{аисх}}, z, \mu). \quad (2)$$

Введем обозначения для шести независимых переменных уравнения (2) (табл. 1).

Размерность переменной X_i имеет вид

$$[X_i] = [F]^{\lambda_i^{(F)}} \cdot [L]^{\lambda_i^{(L)}} \cdot [T]^{\lambda_i^{(T)}}. \quad (3)$$

Таблица 1

| Параметр | Обозначение | $[F]^{\lambda_i^{(F)}}$ | $[L]^{\lambda_i^{(L)}}$ | $[T]^{\lambda_i^{(T)}}$ |
|-------------------|-------------|-------------------------|-------------------------|-------------------------|
| P | X_1 | 1 | -2 | 0 |
| V | X_2 | 0 | 1 | -1 |
| T | X_3 | 0 | 0 | 1 |
| $R_{\text{аисх}}$ | X_4 | 0 | 1 | 0 |
| z | X_5 | 0 | 1 | 0 |
| ? | X_6 | 1 | 0 | 0 |

Показатели степеней $\lambda_i^{(F)}$, $\lambda_i^{(L)}$, $\lambda_i^{(T)}$ сведены в табл. 1. На основании π -теоремы подобия [2] зависимость (2) может быть представлена в критериальном виде. Каждый из критериев подобия π имеет вид безразмерного комплекса

$$\pi = \prod_{i=1}^6 X_L^{a_i},$$

т.е. отыскание критериев подобия сводится к определению показателей степеней a_i при соответствующих переменных X_L . Для этого преобразуем критерий π следующим образом:

$$\pi = \prod_{i=1}^6 X_L^{a_i} = k \prod_{i=1}^6 [X_i]^{a_i}, \quad (4)$$

где k – безразмерная величина.

Подставим в (4) значение для $[X_i]$ из (3):

$$\begin{aligned} \pi &= \prod_{i=1}^6 ([F]^{\lambda_i^{(F)}} \cdot [L]^{\lambda_i^{(L)}} \cdot [T]^{\lambda_i^{(T)}})^{a_i} = \\ &= k \prod_{i=1}^6 [F]^{\lambda_i^{(F)} a_i} \prod_{i=1}^6 [L]^{\lambda_i^{(L)} a_i} \prod_{i=1}^6 [T]^{\lambda_i^{(T)} a_i} = k [F]^{\sum_{i=1}^6 \lambda_i^{(F)} a_i} \cdot [L]^{\sum_{i=1}^6 \lambda_i^{(L)} a_i} \cdot [T]^{\sum_{i=1}^6 \lambda_i^{(T)} a_i}. \end{aligned}$$

Поскольку π – безразмерные комплексы, имеем систему уравнений относительно a_i :

$$\left. \begin{aligned} \sum_{i=1}^6 \lambda_i^{(F)} a_i &= 0, \\ \sum_{i=1}^6 \lambda_i^{(L)} a_i &= 0, \\ \sum_{i=1}^6 \lambda_i^{(T)} a_i &= 0. \end{aligned} \right\} \quad (5)$$

Для нахождения критериев подобия необходимо найти фундаментальную систему решений системы (5).

Матрица М коэффициентов этой системы составляется по табл. 1:

$$M = \begin{pmatrix} \lambda_1^{(F)} & \lambda_2^{(F)} & \dots & \lambda_6^{(F)} \\ \lambda_1^{(L)} & \lambda_2^{(L)} & \dots & \lambda_6^{(L)} \\ \lambda_1^{(T)} & \lambda_2^{(T)} & \dots & \lambda_6^{(T)} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ -2 & 1 & 0 & 1 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

С учётом вида матрицы М система (5) имеет вид

$$\begin{cases} a_1 + a_2 = 0, \\ -2a_1 + a_2 + a_4 + a_5 = 0, \\ -a_2 + a_3 = 0. \end{cases} \quad (6)$$

Ранг матрицы М

$$\text{rg}M = 3,$$

т.е. фундаментальная система решений системы (6) состоит из трех решений $(6 - 3) = 3$.

Обозначим решение системы (6) \bar{a} : $\bar{a} = (a_1, a_2, \dots, a_6)$.

Решив систему (6), получим одну из фундаментальных систем решений:

$$\left. \begin{aligned} \bar{a}_1 &= (0, -1, -1, 1, 0, 0), \\ \bar{a}_2 &= (0, -1, -1, 0, 1, 0), \\ \bar{a}_3 &= (-1, -2, -2, 0, 0, 1). \end{aligned} \right\} \quad (7)$$

Используя фундаментальную систему решений (7), составим три категории подобия по формуле

$$\pi = \prod_{i=1}^6 X_L^{a_i},$$

$$\left. \begin{aligned} \pi_1 &= X_2^1 X_3^{-1} X_4 = \frac{R_{\text{aux}}}{Vt}, \\ \pi_2 &= X_1^{-1} X_2^{-2} X_3^{-2} \dots X_6 = \frac{\mu}{PV^2 t^2}, \\ \pi_3 &= X_2^{-1} X_3^{-1} X_5 = \frac{z}{Vt}. \end{aligned} \right\} \quad (8)$$

Таким образом, зависимость (2) сводится к критериальному виду

$$R_a = \varphi(\pi_1, \pi_2, \pi_3), \quad (9)$$

где π_1, π_2, π_3 – критерии подобия, связанные с основными переменными формулы (8).

2. Проведение эксперимента в целях получения интерполяционной формулы для функции

$$R_a = \varphi(\pi_1, \pi_2, \pi_3).$$

Планирование эксперимента. Будем искать зависимость (9) в виде полинома

$$\bar{R}_a = \sum_{i=1}^6 b_i \pi_i, \quad (10)$$

где R_a – среднее значение, R_a , $\pi_4 = \pi_1 \cdot \pi_2$, $\pi_5 = \pi_1 \cdot \pi_3$, $\pi_6 = \pi_2 \cdot \pi_3$, b_i ($i = 0 \dots 6$) – коэффициенты регрессии, π_0 – фиктивная переменная, $\pi_0 = 1$.

Для устранения взаимной корреляции параметров $\pi_1 \pi_2 \pi_3$ (8) примем следующие положения:

1) многократные эксперименты показали, что оптимальный результат получается при значении параметра t , близкого к 30м/мин;

2) зафиксируем параметры V и t : $V = 30$ м/мин, $t = 2$ мин, а значения параметров π_1, π_2, π_3 будем варьировать, используя параметры.

В силу принятых положений параметры окажутся некоррелированными. Параметры V и t будут входить в исходную зависимость, так как они входят в критерии подобия, и уравнение (10) будет адекватно описывать в области факторного пространства с центром в выбранных значениях параметров V и t .

Для определения коэффициентов регрессии проводим полный факторный эксперимент 2^3 . Факторы π_1, π_2, π_3 варьируются на двух уровнях – максимальном и минимальном (с учетом принятого положения 2). Диапазоны варьирования параметров $R_{\text{ансх}}, P, z$, указаны в табл. 2.

Таблица 2

| Параметры | min | Max |
|-----------------------|------|-----|
| (мкм) | 0,63 | 2,5 |
| (кг/см ²) | 0,5 | 40 |
| (мкм) | 7 | 28 |

Обозначим знаком «+» максимальный уровень каждого из параметров π_1, π_2, π_3 , а знаком «-» – минимальный.

Уровни параметров π_1, π_2, π_3 определяются по правилу знаков [2].

Каким значениям параметров $R_{\text{ансх}}, P, z$ соответствуют уровни + и – значения параметров π_1, π_2, π_3 определяем из формулы (8).

Составим матрицу планирования и в соответствии с ней проведем полный факторный эксперимент 2^3 . Число параллельных опытов примем равным трем.

Значения параметра оптимизации R_a , полученные в трех параллельных экспериментах, обозначим соответственно $R_a^{(1)}, R_a^{(2)}, R_a^{(3)}$.

Вычисление коэффициентов интерполяционного полинома. Среднее значение параметра оптимизации R_a для каждого из восьми опытов вычислим по формуле

$$R_a = \frac{\sum_{i=1}^6 R_a^{(i)}}{3}.$$

Значения R_a указаны в (2).

Так как матрица планирования полного факторного эксперимента ортогональна, метод наименьших квадратов приводит к следующей формуле для вычисления коэффициентов полинома (10):

$$b_j = \frac{\sum_{i=1}^N \pi_j^{(i)} R_{ai}}{N}, \quad (11)$$

где N – число опытов ($2^3 = 8$), $j=0; 1; 2; \dots, k$ – индексы коэффициентов b_j ($K=6$).

При вычислении коэффициентов по этой формуле значения параметров π_1, π_2, π_3 на крайних уровнях (+ и –) принимаются соответственно +1 и -1.

Вычисленные по формуле (11) коэффициенты сведены в табл. 3.

Таблица 3

| b_0 | b_1 | b_2 | b_3 | b_4 | b_5 | b_6 |
|-------|-------|-------|-------|--------|-------|-------|
| 0,74 | 0,33 | -0,02 | 0,08 | -0,013 | -0,04 | -0,04 |

Таким образом, полином (10) принимает вид

$$\bar{R}_a = 0,74 + 0,33\pi_1 - 0,02\pi_2 + 0,08\pi_3 - 0,013\pi_1 \pi_2 - 0,04\pi_1 \pi_3 - 0,04\pi_2 \pi_3. \quad (12)$$

Выполнение постулатов регрессивного анализа. Перед проведением статистической обработки полученного результата уравнения (12) необходимо убедиться в выполнении постулатов регрессивного анализа [4].

1) Принимаем, что R_a – случайная величина с нормальным законом распределения.

2) Для проверки однородности дисперсии R_a предварительно вычислим дисперсии изменчивости по каждому опыту по формуле

$$S_i = \frac{\sum_{q=1}^n (R_{ai}^{(q)} - R_{ai})}{n-1} \quad (i=1, \dots, 8),$$

где $n=3, n-1=2$ – число степеней свободы [2].

Однородность дисперсий изменчивости проверяется с помощью F – критерия Фишера:

$$F = \frac{S^2_{\max}}{S^2_{\min}},$$

где $S^2_{\max} = 6 \cdot 10^{-4}$, $S^2_{\min} = 10^{-4}$ [2]. Табличное значение критерия при 5% уровне значимости 4 19,2. В нашем случае $F = 6 < 19,2$, т.е. дисперсии изменчивости однородны.

3) Принимаем, что дисперсии факторов незначительны по сравнению с дисперсиями параметра оптимизации, т.е. считаем факторы π_1, π_2, π_3 неслучайными величинами.

4) Параметры π_1, π_2, π_3 считаем некоррелированными (с учетом принятых ранее положений).

Проверка значимости коэффициентов. В силу однородности дисперсий изменчивости дисперсия воспроизводимости вычисляется по формуле

$$S^2_{\{R_a\}} = \frac{\sum_{i=1}^N \sum_{q=1}^n (R_{ai}^{(q)} - R_{ai})^2}{N(n-1)},$$

или в нашем случае

$$S^2_{\{R_a\}} = \frac{\sum_{i=1}^6 \sum_{q=1}^3 (R_{ai}^{(q)} - R_{ai})^2}{q(3-1)} = 3 \cdot 10^{-4}. \quad (13)$$

Дисперсию коэффициентов регрессии вычисляем по формуле

$$S_{\{b_j\}} = \frac{S^2_{\{R_a\}}}{N}, \quad (j=0, \dots, 6).$$

Имеем $S_{\{b_j\}} = 0.61 \times 10^{-2}$.

Доверительные интервалы для каждого из коэффициентов регрессии равны между собой и вычисляются по формуле

$$\Delta b_j = t \cdot S_{\{b_j\}} \quad (j=0, \dots, 6),$$

где t – табличное значение критерия Стьюдента при числе степеней свободы, с которыми определялась дисперсия $S^2_{\{R_a\}}$ (в нашем случае 2), и выбранном уровне значимости (5%) $t = 4,303$ (определено t – критерия [3]):

$$\Delta b_j = 0,026 \quad (j=0, \dots, 6).$$

Коэффициент b_j считаем значимым, если $|b_j| > \Delta b_j$.

Сравнивая $|b_j|$ (табл.3) с доверительным интервалом, получаем, что коэффициенты b_2 и b_4 незначимы.

Уравнение регрессии приобретает вид

$$R_a = 0,74 + 0,33\pi_1 + 0,08\pi_3 - 0,04\pi_1\pi_3 - 0,04\pi_2\pi_3. \quad (14)$$

Проверка адекватности полученной зависимости. Для проверки адекватности зависимости (14) вычислим по этой формуле значения $R_a(\widehat{R}_a)$ в каждом из восьми опытов (табл.4).

Таблица 4

| № опыта | R_a | \overline{R}_a | $\Delta R_a = \overline{R}_a \cdot \widehat{R}_a$ | ΔR_a^2 |
|---------|-------|------------------|---|----------------|
| 1 | 0,31 | 0,25 | 0,06 | 0,0036 |
| 2 | 0,53 | 0,57 | -0,04 | 0,0016 |
| 3 | 0,27 | 0,33 | -0,06 | 0,0036 |
| 4 | 0,97 | 0,98 | -0,01 | 0,0001 |
| 5 | 1,08 | 1,07 | 0,01 | 0,0001 |
| 6 | 0,53 | 0,49 | 0,04 | 0,0016 |
| 7 | 1,24 | 1,16 | 0,08 | 0,0064 |

Невязки ΔR_a определяются из формулы $\Delta R_a = \overline{R}_a \cdot \widehat{R}_a$ (табл.4). Квадраты невязок ΔR_a также указаны в табл. 4. Сумма квадратов невязок $\sum_{i=1}^N \Delta R_a^2 = 0,00234$.

Дисперсия адекватности вычисляется по формуле $S_{ag}^2 = \frac{\sum_{i=1}^N \Delta R_{ai}^2}{f}$, где в числителе – сумма квадратов невязок, в знаменателе – число степеней свободы, которые определяются так:

$$f = N - (k + 1) = 1$$

(здесь k – количество вычисленных коэффициентов регрессии).

Таким образом, дисперсия адекватности равна $S_{ag}^2 = 0,0234$.

Адекватность модели (14) проверим с помощью F – критерия Фишера (для 50 уровня значимости):

$$F = \frac{S_{ag}^2}{S_{\{R_a\}}^2},$$

где $S_{\{R_a\}}^2$ вычисляются по формуле (12).

Табличное значение F – критерия 199,5 [4]. В нашем случае $F = 78 < 199,5$.

Таким образом, построенная модель(14) адекватно описывает процесс.

3. Анализ полученной зависимости

Полученная зависимость (14)

$$R_a = 0,74 + 0,33\pi_1 + 0,08\pi_3 - 0,04\pi_1\pi_3 - 0,04\pi_2\pi_3,$$

где π_1, π_2, π_3 вычисляется по формуле (8), позволяет проследить влияние факторов π_1, π_2, π_3 на величину \overline{R}_a . Для минимизации благоприятно одновременное уменьшение значений факторов π_1, π_2, π_3 .

Необходимо учесть, что в уравнении(14) значение параметров π_1, π_2, π_3 кодированы (изменяются от -1 до $+1$). Переход от натуральных значений параметров к кодированным осуществляется по формуле

$$\pi_j = \frac{\widehat{\pi}_j - \widehat{\pi}_{j0}}{I_j}, \quad j = 1, 2, 3,$$

где π_j – кодированное значение параметров; $\widehat{\pi}_j$ – натуральное значение параметров; $\widehat{\pi}_{j0}$ – натуральное значение основного уровня; I_j – интервал варьирования.

Натуральные значения основных уровней параметров и интервалов варьирования определяются, исходя из данных, и приведены в табл. 5

Таблица 5

| Параметры V (м/мин) | t (мин) | p (кг/см ³) | R _{ансх} мкм | z мкм | ? _{j0} | I _j |
|------------------------|------------|----------------------------|--------------------------|----------|-----------------------|-----------------------|
| 30 | 2 | – | 1,55 | – | 2,6x10 ⁻⁸ | 1,6x10 ⁻⁸ |
| 30 | 2 | 0,98 | – | – | 0,27x10 ⁻⁷ | 0,27x10 ⁻⁷ |
| 30 | 2 | – | – | 18 | 3x10 ⁻⁷ | 1,6x10 ⁻⁷ |

Уравнение (14) адекватно описывает процесс, если кодированные значения параметров π_1, π_2, π_3 не выйдут за пределы интервала $[-1; +1]$, поэтому, варьируя натуральные параметры $V, t, P, R_{ансх}, z$, необходимо следить, чтобы $\pi_1, \pi_2, \pi_3 \in [-1; +1]$, где π_1, π_2, π_3 – кодированные значения параметров.

4. Выводы

1) Полученная зависимость может использоваться для прямого вычисления \bar{R}_a по натуральным значениям параметров, как это было описано выше.

2) В условиях производства обычно известны исходная шероховатость поверхности $R_{ансх}$ и шероховатость после обработки детали R_a . Тогда, используя зависимость (14), можно выяснить, какого типа пасту необходимо применить (параметр z) и при каком давлении и скорости вести обработку (параметры P и V), чтобы достичь наиболее высокой производительности процесса (параметр t).

3) Зависимость (14) может быть использована при работе технологических линий с программным управлением. При этом ЭВМ планирует параметры обработки, исходя из зависимости (14) с учетом обеспечения наиболее высокой производительности процесса.

4) Зависимость (14) может быть использована для построения специальных таблиц режимов обработки по классам входной точности поверхности, по типам паст, скорости и т.д. Таблицы предназначены для использования технологии при выборе режимов обработки деталей.

5) По выходным параметрам (R_a) можно планировать качество адгезии защитного покрытия.

Список литературы: 1. Невлюдов И.Ш., Анпилогов Е.М. Технология финишной обработки магнитопродов. ХЦТИ, 1978. 2. Дзюндзюк Б.В., Анпилогов С.М., Мегель Ю.Е., Анпилогова И.Е. Об одном методе моделирования технологического процесса формирования поверхностного слоя обрабатываемых деталей // Збірник наукових праць Харківського університету Повітряних Сил. Вип. 61/12. Харків, 2006. С. 112-116. 3. Налимов В.В., Чернова Н.А. Статистические методы планирования экстремальных экспериментов. М.: Наука, 1969. С. 340. 4. Седов Л.И. Методы подобия и размерностей в механике. М.: Наука, 1967. 428 с. 5. ЩигOLEV Б.М. Математическая обработка наблюдений. М.: Наука, 1969. 344 с.

Поступила в редколлегию 02.03.2010

Дзюндзюк Борис Васильевич, д-р техн. наук, профессор, зав. кафедрой охраны труда ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-13-60.

Анпилогов Евгений Михайлович, канд. техн. наук, доцент кафедры охраны труда ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-13-60.

Савин Валерий Витальевич, ассистент кафедры охраны труда ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-13-60.

Анпилогова Ирина Евгеньевна, инженер кафедры ТАПР ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-13-60.

Герасименко Олег Викторович, студент ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Ленина, 14.

ДИАГНОСТИКА КОМПЕТЕНТНОСТИ ПОЛЬЗОВАТЕЛЕЙ КОМПЬЮТЕРНЫХ СИСТЕМ

Рассматривается диагностический эксперимент анализа компетентности пользователей компьютерных систем. Предлагается модель структуры ИКТ-компетентностей, формы квалификационных заданий и способы их оценивания. Для оценивания заданий открытой формы с развернутым ответом предлагается использовать математический аппарат правил и процедур нечеткой логики.

Введение

Независимо от сложности компьютерных систем (КС) человек остается главным звеном системы «человек – машина». Именно он ставит цели перед системой, планирует, направляет и контролирует весь процесс ее функционирования. От его деятельности зависит правильность функционирования КС, которая характеризует безошибочность (правильность) решения задач, стоящих перед пользователем КС. Одной из основных причин неправильного функционирования КС предлагается рассматривать ошибки пользователей КС. Последние связывают с уровнем его компетентности в области информационных технологий. В условиях информатизации общества специалист должен обладать не только набором знаний, умений и навыков, но и уметь применять их для решения ряда профессиональных задач, т.е. обладать информационно-коммуникационно-технологической компетентностью (профессиональной ИКТ-компетентностью).

Под профессиональной компетентностью принято понимать совокупность знаний, умений, навыков и способов их применения, которые позволяют личности адекватно воспринимать и обрабатывать информацию в своей предметной области, постигать сущность связей между объектами профессиональной деятельности и принимать адекватные решения в различных стандартных и нестандартных ситуациях [1].

Кроме профессиональной компетентности на рынке труда широко используется понятие «квалификация». Квалификация – это степень профессиональной подготовленности к выполнению определенного вида работы, которая выражается в уровне знаний, умений и навыков и устанавливается в виде разряда или категории, которые в свою очередь определены государственными квалификационными характеристиками. Таким образом, квалификация – это комплексная характеристика, позволяющая определить степень соответствия работника должности, его профессиональный уровень. Исследования в области профессионального образования показали [2], что компетентность включает в себя не только элементы квалификации (знания, умения, навыки и способности их применения), но и социально-коммуникативные личностные характеристики, обеспечивающие самостоятельность профессиональной деятельности.

Учитывая, что личностные характеристики индивидуума могут проявляться только в процессе реальной профессиональной деятельности, на этапе анализа компетентности пользователя КС, можно ограничиться только квалификационной частью компетентности, а для ее оценки пользоваться квалификационными заданиями специального вида.

Целью данной работы является описание способов формального оценивания квалификационных заданий различных форм в системе диагностирования компетентности пользователей КС.

1. Диагностический эксперимент по анализу компетентности

Процесс анализа компетентности пользователя КС можно рассматривать как диагностический эксперимент (ДЭ) в человеко-машинной системе, в ходе которого с помощью специальных квалификационных заданий проверяется уровень знаний, умений и навыков пользователя КС, а также его способности адекватно реагировать на различные ситуации, возникающие в процессе работы КС. ДЭ анализа компетентности состоит из подготовки

специальных квалификационных заданий и эталонных (правильных) реакций испытуемого, процесса проведения испытания и сравнения ответов (принятых решений) с эталоном, а затем принятия решения о результате испытания (рис.1).

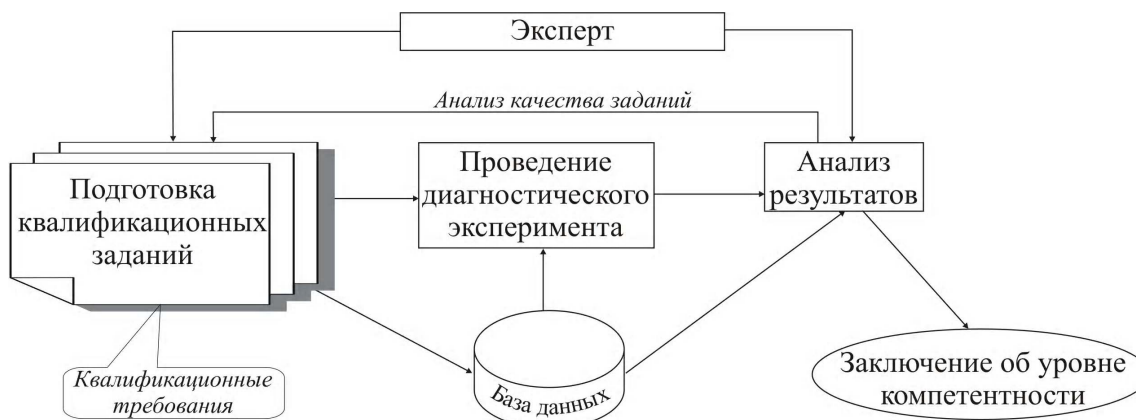


Рис. 1. Структура диагностического эксперимента по проверке компетентности

Если не учитывать возможность проведения натуральных экспериментов на реальном оборудовании, а ограничиться только моделированием работы человеко-машинной системы, то такой ДЭ на первый взгляд похож на классическое тестирование знаний в педагогических измерениях. Но ДЭ по анализу компетентности вместе с тем имеет ряд существенных отличий от классического тестирования знаний:

1. Тестирование проверяет, как правило, уровень полученных и усвоенных знаний. Анализ компетентности является более широким процессом и проверяет возможности пользователя применять полученные знания, умения и навыки по многим учебным дисциплинам в штатных и нештатных режимах функционирования КС.

2. Для тестирования используются специальные формы тестовых заданий [3], а для анализа компетентности, кроме тестовых форм могут использоваться квалификационные задания, которые трудно отнести к одной из наиболее распространенных тестовых форм. Такие задания по своей форме и дидактичной нагрузке наиболее близки к заданиям открытой формы с развернутым ответом, которые используются в современном внешнем независимом оценивании (ЗНО) в Украине [4]. Не ставя целью в данной работе обстоятельно рассматривать содержание указанных заданий, отметим, что это могут быть задания по анализу и синтезу текста, на установление и обоснование последовательности математических или графических действий, на написание программного кода и т.д.

3. Для оценки тестовых заданий закрытых форм и заданий открытой формы с коротким ответом используются формальные процедуры, которые могут быть легко автоматизированные, а для оценки квалификационных заданий открытой формы с развернутым ответом невозможно обойтись без эксперта (экспертной системы), так как оценить подобные задания по формальным признакам достаточно сложно.

2. Модель компетентности пользователей КС

В соответствии с компетентностной парадигмой образования каждая из ключевых компетентностей, в частности ИКТ-компетентность, имеет три измерения: предметное измерение (что нужно знать), деятельностное измерение (что нужно уметь) и практическое измерение (в частности, как использовать приобретенные знания и умения при эксплуатации КС) [5].

Структура базы данных квалификационных заданий анализа ИКТ-компетентности пользователя КС определяется моделью предметной области. Предметная область ИКТ-компетентности определяет перечень знаний, умений и навыков, которые дают возможность индивидууму квалифицированно использовать компьютерные системы, сети и технологии в хозяйственной и учебной деятельности, в научных исследованиях и в быту, в общении и обмене информацией с другими людьми.

При анализе ИКТ-компетентности используется четырехуровневая иерархическая модель предметной области, определяющая структуру соответствующей компетентности:

«КАТЕГОРИЯ КОМПЕТЕНТНОСТИ» -> « ИМЯ РАЗДЕЛА» -> «ИМЯ ТЕМЫ»-> «ЗАДАНИЯ ТЕМЫ».

«КАТЕГОРИЯ КОМПЕТЕНТНОСТИ» – это определенный набор знаний, умений и навыков из разных областей знаний, которые в совокупности определяют уровень ИКТ-компетентности пользователя. Количество категорий компетентности, как правило, фиксировано и определяется квалификационными характеристиками, предъявляемыми к пользователям КС. Предлагается следующая структура категорий ИКТ-компетентности пользователей КС, которая отвечает содержательной структуре ИКТ-компетентности.

1. Информационная компетенция, которая определяет знание основных понятий информатики, функционирования компьютерных систем и сетей (КСС).

2. Техническая компетенция, которая определяет знание основ технического и программного обеспечения КСС.

3. Технологическая компетенция, которая определяет знание технологий, а также аппаратных и программных средств для обработки и преобразования информации.

4. Алгоритмическая компетенция, которая определяет знание основ алгоритмизации и алгоритмических языков.

5. Сетевая и телекоммуникационная компетенция, которая определяет знание основных принципов построения и использования локальных сетей и глобальной сети Internet.

6. Методическая и исследовательская компетенция, которая определяет знания основных принципов использования компьютеров в жизни и деятельности человека, а также в научных исследованиях.

7. Компетенция в вопросах информационной безопасности, которая определяет знания основных информационных атак в КС, принципы защиты данных в КС, аппаратные программные методы защиты информации в КС.

«ИМЯ РАЗДЕЛА» обычно соответствует конкретной области профессиональной деятельности пользователя и соответствует перечню знаний, умений и навыков, определяемых учебными дисциплинами и тренинговыми циклами одного направления. Количество разделов не фиксировано и определяется, как правило, глубиной анализа компетентности.

Например, категория «сетевая и телекоммуникационная компетенция» может быть разделена на следующие разделы:

- принципы и топологии построения локальных сетей;
- правила обмена в сети (протоколы);
- сетевое оборудование и сетевые операционные системы;
- World Wide Web (WWW) и ее основные сервисы;
- клиент-серверные технологии.

«ИМЯ ТЕМЫ» обычно соответствует структурной единице раздела, определенной в рабочей программе учебной дисциплины или разделе квалификационной характеристики.

«ЗАДАНИЯ ТЕМЫ» – это задания разных форм, содержащие текстовые, графические и мультимедийные компоненты, а также расчетные задания.

Если рассматривать деятельностное измерение ИКТ-компетентности, то можно говорить о таких умениях (способностях):

- алгоритмические, т.е. умения алгоритмизировать задачи и процессы, составлять оптимальные алгоритмы и на их основе писать программный код;
- технологические, т.е. умения использовать готовые программные продукты, устанавливать и эффективно использовать операционные системы;
- технические, т.е. умения выбирать, устанавливать и настраивать аппаратные средства КС;
- исследовательские, т.е. умения собирать и обрабатывать данные, строить компьютерные модели, оценивать корректность и эффективность принятых решений.

Практическое измерение ИКТ-компетентности связано с такими направлениями применения (использования) полученных знаний и умений:

- коммуникативное (аппаратное и программное обеспечение работы в компьютерной сети, организация доступа к сетевым ресурсам);
- комплексировочное (оптимальный выбор и установка аппаратных и программных средств, необходимых для решения поставленных задач);

- эксплуатационное (поддержка работоспособности КС);
- диагностическое (определение технического состояния КС и устранение, при необходимости, возникающих дефектов).

В соответствии с моделью предметной области формируется тестовая база квалификационных заданий разных форм, которые используются при проведении диагностического эксперимента (сеанса диагностирования). Таким образом, тестовая база данных состоит из семи категорий (предметное измерение), внутри которых находятся задания, структурированные по разделам и темам. Внутри каждой темы задания структурированы по остальным двум измерениям (уметь, использовать). Сеанс диагностирования, как правило, состоит из заданий одного раздела со случайным выбором заданий из разных тем, которые в свою очередь должны покрывать все измерения ИКТ-компетентности. Не касаясь в данной работе способов формирования сеанса диагностирования, рассмотрим формы заданий и способы их оценивания.

3. Оценивание квалификационных заданий и шкалирование итоговых оценок

Диагностический эксперимент (сеанс диагностирования) для каждого испытуемого состоит из N квалификационных заданий разных форм. Рассмотрим принципы оценивания указанных заданий различных форм, пользуясь при этом опытом оценивания заданий разных форм в компьютерной системе тестирования знаний OpenTEST2 [6].

В заданиях с выбором одного правильного ответа правильный ответ оценивается в 1 балл, а неправильный – 0 баллов. Ответы на задания множественного выбора (соответствия) оцениваются в долю балла, которая обусловлена количеством выбранных правильных альтернатив. Для заданий закрытых форм итоговая оценка корректируется на вероятность угадывания правильного ответа. В заданиях открытой формы с коротким ответом правильный ответ оценивается в 1 балл, а неправильный – 0 баллов, но корректировки на вероятность угадывания не выполняется, так как вероятность угадывания для таких заданий принимается равной нулю.

Для решения проблемы оценивания заданий открытой формы с развернутым ответом воспользуемся аналогией с заданиями с множественным выбором, т.е. ответ на задание с развернутым ответом может оцениваться в долю балла, которая обусловлена степенью правильности выполнения этого задания. Эту долю балла можно представить как процент правильности выполнения такого задания. Для этого нужно определить признаки, по которым определяется степень правильности выполнения задания и градации степени выполнения, которые может определять эксперт. По аналогии со значениями лингвистической переменной «компетентность» [7] можно определить пятиуровневую градацию степеней правильности выполнения задания: «очень низкий», «низкий», «средний», «достаточный», «высокий». Процентные пределы этих диапазонов могут быть фиксированы или определяться экспертами для каждого задания и испытания отдельно. Кроме того, если выполнение задания состоит из нескольких этапов (шагов), эта оценка считается накопительной: чем больше этапов задания выполнил испытуемый, тем выше итоговая оценка за выполнение задания. Признаки, по которым оценивается степень выполнения задания, определяются для каждого задания совместно автором задания и экспертом. При оценивании такого задания эксперт (экспертная система) анализирует эти признаки и принимает решение об отнесении результата к одному из определенных диапазонов.

Каждое квалификационное задание может иметь вес B_i , выраженный целым положительным числом больше 1. Вес каждого задания в базе квалификационных заданий (B_i) определяется автором задания (экспертом) на этапе составления системы квалификационных заданий раздела соответствующей категории предметной области и отражает его субъективную точку зрения на влияние i -го задания на общую оценку за сеанс диагностирования.

За результат выполнения каждого квалификационного задания испытуемый получает балл от 0 до B_i . Выражение $\sum_{i=1}^N B_i$ определяют балл, набранный испытуемым в ходе всего

ДЭ, где N – количество заданий в сеансе диагностирования. Выражение $B_{\max} = \sum_{i=1}^N B_i^{\max}$

определяет максимально возможный балл за сеанс диагностирования. Исходная («первичная») оценка за сеанс диагностирования B_H (набранный балл) определяется как процент правильности выполнения квалификационных заданий:

$$B_H (\%) = \frac{\sum_{i=1}^N B_i}{\sum_{i=1}^{B_i^{\max}} B_i} \cdot 100\% .$$

Такое нормирование позволяет свести оценку за сеанс диагностирования к диапазону [0...100] баллов (процентов) независимо от форм, веса и количества заданий в сеансе диагностирования.

Под шкалированием тестовых оценок понимается их отнесение к одному из выбранных диапазонов оценивания. Одним из главных факторов, влияющих на способ шкалирования, является форма распределения исходных оценок по оцениваемому диапазону. В педагогических измерениях (рассматриваемый ДЭ также является разновидностью педагогических измерений) принято считать, что большинство распределений исходных оценок лучше всего аппроксимируются кривой нормального закона распределения, с ярко выраженной «центральной тенденцией». Шкалирование оценок (с учетом предположения о нормальном распределении результатов тестирований по шкале оценивания) осуществляется путем деления площади нормализованной кривой распределения испытуемых на D равных частей, где D – диапазон шкалы оценивания. Для реализации такого подхода в базу данных закладывается таблица значений z_i (функция Лапласа) для 99 верхних границ диапазонов 100-балльной шкалы оценивания в порядке возрастания от 1 до 99 [8].

Для отнесения «первичной» оценки B_i к соответствующему диапазону шкалы оценивания необходимо произвести следующие преобразования.

1. Выполняется стандартизация «исходной» оценки в доверительном интервале 95%:

$$z_i = \frac{Y_k - M}{1.53 * \sigma} .$$

2. По таблице значений сотых долей z_i в базе данных находится процентное значение оценки в линейной шкале β .

3. Оценка в соответствующей линейной шкале оценивания будет определяться номером диапазона, который вычисляется как ближайшее наибольшее целое от номера диапазона шкалы оценивания, т.е.

$$]i[= \left\lceil \frac{\beta}{100} \right\rceil \left[\frac{\beta * D}{100} \right] .$$

Как уже упоминалось выше, итоговая оценка, как правило, шкалируется в соответствии с принятой пятиуровневой градацией ($D=5$) определения уровня компетентности: «очень низкая», «низкая», «средняя», «достаточная», «высокая».

4. Оценивание заданий открытой формы с развернутым ответом

В соответствии с принятой концепцией оценивания сеанса диагностирования оценивание заданий открытой формы с развернутым ответом также должно сводиться к определению «процента правильности ответа». Это необходимо учитывать уже на этапе формулировки заданий путем выделения соответствующих критериев оценивания и определения диапазонов оценки за указанные критерии. Веденная система критериев является основой для работы эксперта (экспертной системы) при оценивании таких заданий. Формулировка задания должна соответствовать критериям оценивания экспертов. Это означает, что испытуемый после прочтения задания должен понять, какую задачу ему предстоит выполнить, и с какой полнотой он должен дать ответ для получения максимального балла.

Согласно предложенной концепции при оценке таких заданий будем опираться на оценку эксперта. Информация, предоставляемая экспертами, содержит понятия естественного языка, трудно выражаемые количественными отношениями. Для перехода от словесного описания к численным показателям предлагается использовать аппарат нечёткой логики. Данный подход позволит не только оценить результаты тестирования, но и будет использоваться на этапе экспертной подготовки заданий. Механизм нечёткого вывода в своей основе имеет базу знаний, формируемую специалистами (экспертами) предметной области в виде совокупности нечётких продукционных правил.

Исходя из содержательной основы квалификационных заданий на проверку компетентности, их можно разделить на две группы: коммуникативно-технические и логико-технологические. К первой группе можно отнести работу с программной и технической документацией, составление технических проектов, диагностирование технического состояния объектов. В таких заданиях, как правило, главным является соответствие результатов выбранным критериям, а порядок выполнения действий не является определяющим. Ко второй группе можно отнести алгоритмизацию и решение математических и логических задач, применение технологий обработки информации, соблюдение правил работы с аппаратурой. Для этого класса заданий порядок выполнения действий является определяющим, наряду с правильным результатом выполнения отдельных процедур. Поэтому решающие процедуры нечеткой логики также можно разделить на два класса:

1) одноуровневые процедуры (накопительные), учитывающие степень соответствия выбранным критериям;

2) двухуровневые процедуры (на последовательность действий), учитывающие степень правильности выполнения действия и их последовательность.

Данное деление не является абсолютным: квалификационные задания первой группы, в большинстве своем, являясь одноуровневыми, могут быть и двухуровневыми, а второй группы, в свою очередь, наоборот. Рассмотрим особенности оценивания каждой из групп квалификационных заданий по оценке компетентности.

При применении одноуровневых процедур оценивания выбираются критерии оценки K_i и диапазоны оценивания, а эксперт оценивает каждый из критериев в заданном диапазоне. Итоговая оценка V вычисляется в соответствии с одной из моделей нечеткого вывода (Мамдами, Сугено, Цукамото).

Рассмотрим в качестве примера задание первой группы следующего типа: «Проанализировать программный код». Для его оценивания введем следующие критерии, по которым эксперт будет определять степень выполнения задания:

- критерий А (синтаксис) – [0...10] баллов;
- критерий В (использование типов данных) – [0...10] баллов;
- критерий С (выполняемость) – [0...20] баллов;
- критерий D (правильность результата) – [0...40] баллов;
- критерий Е (представление результата) – [0...20] баллов.

Каждому критерию автор (эксперт) ставит в соответствие нечеткое подмножество в виде соответствующего диапазона баллов. Из условия задания определим пять входных лингвистических переменных (пять критериев оценки задания) и одну выходную переменную (степень правильности выполнения задания или оценка). Каждая из входных нечетких переменных будет иметь по три терма (значения): «плохо», «удовлетворительно» и «хорошо». Изменяться будут лишь границы диапазонов для каждого терма. Примем границы диапазонов всех критериев равными и в пересчете на 100-балльную шкалу определим: «плохо» – [0...40] баллов, «удовлетворительно» – [30...70] баллов и «хорошо» – [60...100] баллов. Выходная нечеткая переменная (оценка) имеет пять термов (значений): «очень низкая», «низкая», «средняя», «достаточная», «высокая» [7]. Границы диапазонов для нее с учетом «центральной тенденции» в пересчете на 100-балльную шкалу определим следующими: «очень низкая» – [0...30] баллов, «низкая» – [25...45] баллов, «средняя» – [40...60] баллов, «достаточная» – [55...75] баллов и «высокая» – [70...100] баллов. В соответствии с выбранной моделью нечеткого вывода, будут вычисляться функции принадлежности для всех лингвистических переменных и для всех правил. Один из возможных вариантов продукционных правил представлен ниже:

ЕСЛИ (A = удовлетворительно) И (B = хорошо) И (C = плохо) И (D = плохо) И (E = плохо) ТО (оценка = очень низкая).

ЕСЛИ (A = удовлетворительно) И (B = хорошо) И (C = удовлетворительно) И (D = плохо) И (E = плохо) ТО (оценка = низкая).

ЕСЛИ (A = хорошо) И (B = хорошо) И (C = удовлетворительно) И (D = удовлетворительно) И (E = плохо) ТО (оценка = средняя).

ЕСЛИ (A = хорошо) И (B = хорошо) И (C = хорошо) И (D = удовлетворительно) И (E = удовлетворительно) ТО (оценка = достаточная).

ЕСЛИ (A = хорошо) И (B = хорошо) И (C = хорошо) И (D = хорошо) И (E = удовлетворительно) ТО (оценка = высокая).

Особенностью составления этих продукционных правил является то, что каждое правило должно содержать оценку по всем критериям. Приведенная система правил не является полной, но достаточно отображает принятую концепцию.

При применении двухуровневых процедур оценивания задается последовательность действий для достижения результата, критерии оценивания каждого действия и решающее правило для получения результата.

Рассмотрим в качестве примера задание второй группы следующего типа: «Минимизировать булеву функцию (БФ) методом карт Карно».

Для того чтобы его оценить, рассмотрим следующие признаки, по которым эксперт будет определять степень правильности выполнения задания:

- действие 1 (д1): составление карты Карно по числу переменных БФ – [0...10] баллов;
- действие 2 (д2): заполнение карты Карно по таблице истинности – [0...10] баллов;
- действие 3 (д3): выполнение обведения групп единиц (нулей) – [0...40] баллов;
- действие 4 (д4): составления дизъюнкций (конъюнкций) – [0...20] баллов;
- действие 5 (д5): запись минимизированной БФ в форме ДНФ (КНФ) – [0...20] баллов.

Аналогично с предыдущим примером происходит фаззификация входных лингвистических переменных (д1 – д5) и выходной переменной (оценка), и выбирается соответствующая модель нечеткого вывода, от которой будет зависеть результат обработки продукционных правил.

Представим один из возможных вариантов продукционных правил:

ЕСЛИ (д1 = плохо) ТО (оценка = очень низкая).

ЕСЛИ (д1 = хорошо) И (д2 = плохо) ТО (оценка = низкая).

ЕСЛИ (д1 = хорошо) И (д2 = хорошо) И (д3 = удовлетворительно) ТО (оценка = средняя).

ЕСЛИ (д1 = хорошо) И (д2 = хорошо) И (д3 = удовлетворительно) И (д4 = удовлетворительно) ТО (оценка = достаточная).

ЕСЛИ (д1 = хорошо) И (д2 = хорошо) И (д3 = хорошо) И (д4 = хорошо) И (д5 = удовлетворительно) ТО (оценка = высокая).

Как и в предыдущем примере, приведенная система правил не является полной, но достаточно отображает принятую концепцию. Особенностью же составления правил для двухуровневых процедур оценивания является то, что каждое последующее правило должно содержать положительную оценку всех предыдущих действий.

Для того чтобы можно было использовать предложенные решающие процедуры нечеткой логики при определении результата диагноза в ДЭ, когда входы и выход системы (конечный результат) имеют реальные значения, наиболее простым путем является использование фаззификатора на входе и дефаззификатора на выходе системы нечеткой логики (рис. 2).

Использование этих двух блоков предоставляет возможность естественного перехода от выводов эксперта к нечетким продукционным правилам. Базис разработанных продукционных правил является основной частью системы нечеткой логики, а все другие компоненты используются только для интерпретации этих правил при решении конкретной задачи. Предложенные продукционные правила могут быть положены в основу экспертной системы по оцениванию заданий открытой формы с развернутым ответом. В ходе оценивания результатов диагноза эксперт подает на вход системы нечеткого вывода оценку степени выполнения задания для каждого критерия в границах установленных им диапазо-

нов, а экспертная система формирует итоговую оценку за выполнения задания в диапазоне [0...100] баллов (процент правильности ответа) на основании выбранной модели нечеткого вывода.



Рис. 2. Система нечеткого вывода

Рассмотрим в качестве примера выполнение и оценивание задания с развернутым ответом (первой группы) на проверку сетевой и телекоммуникационной компетенции.

Задание: «предложите проект локальной офисной сети и обоснуйте выбор аппаратного и программного обеспечения для офиса на 20 рабочих мест, расположенных в одном помещении, для работы с программами MS Office и бухгалтерской системой 1С».

Выделим следующие признаки (критерии) оценивания данного задания:

- а) работоспособность модели полученной сети – [0...30] баллов;
- б) организация одновременной работы с базой данных бухгалтерской системы 1С – [0...20] баллов;
- в) обоснование к выбору топологии сети, аппаратного и программного обеспечения для решения конкретной задачи – [0...20] баллов;
- г) простота реализации (минимизации потраченных ресурсов без ущерба для поставленной задачи) – [0...20] баллов;
- д) масштабируемость модели сети и возможность обслуживания – [0...10] баллов.

Пример производственных правил для оценки данного задания приведен ниже.

ЕСЛИ (а = плохо) И (б = плохо) И (в = хорошо) И (г = хорошо) И (д = хорошо) ТО (оценка=очень низкая).

ЕСЛИ (а = удовлетворительно) И (б = удовлетворительно) И (в = удовлетворительно) И (г = плохо) И (д = плохо) ТО (оценка = низкая).

ЕСЛИ (а = хорошо) И (б = хорошо) И (в = удовлетворительно) И (г = удовлетворительно) И (д = удовлетворительно) ТО (оценка = средняя).

ЕСЛИ (а = хорошо) И (б = хорошо) И (в = хорошо) И (г = хорошо) И (д = удовлетворительно) ТО (оценка = достаточная).

ЕСЛИ (а = хорошо) И (б = хорошо) И (в = хорошо) И (г = хорошо) И (д = хорошо) ТО (оценка = высокая).

Испытуемый дал следующий ответ.

«Для офиса на 20 рабочих мест, расположенных в одном помещении и работающих с программами MS Office и бухгалтерской системой 1С, предлагается использовать в качестве топологии – топологию “иерархическая звезда”. Выбор был остановлен на данной топологии исходя из того, что строим модель сети с использованием нескольких коммутаторов, иерархически соединенных между собой связями типа “звезда”. Для обеспечения работоспособности такой сети предлагается использовать следующее аппаратное и программное обеспечение:

- 1) 20 персональных компьютеров с установленными ОС Windows, MS Office и бухгалтерской системой 1С: Бухгалтерия;
- 2) 1 сервер;
- 3) 16-портовые коммутаторы – 2 шт.

Выбор данного обеспечения обуславливается главной задачей, которую должна решать данная сеть, т.е. обеспечение возможности одновременной работы 20 рабочих мест с одним сервером. 16-портовые коммутаторы в количестве 2 штук выбраны по той причине, что это дает нам возможность выполнить коммутацию 20 компьютеров. Выбор программного обеспечения обуславливается характером решаемых задач, поэтому на каждом рабочем месте должна быть установлена операционная система (Windows 2000/XP/vista/7)

для корректной работы с офисом (MS Office 2003/2007/2010), а на сервере – серверная операционная система (Windows Server 2000/2003/2010). База 1С: Бухгалтерия должна располагаться на одном из персональных компьютеров, а для обеспечения доступа к ней на каждое рабочее место должна быть установлена программа-клиент 1С: Бухгалтерия.

Эксперт при оценивании данного задания по каждому критерию выставил следующие баллы: а) 20 баллов; б) 6 баллов; в) 20 баллов; г) 10 баллов; д) 10 баллов.

Для анализа результатов диагностирования использовалась математическая система MATLAB, а именно, специальный пакет нечеткого вывода Fuzzy Logic Toolbox. В качестве модели нечеткого вывода была использована модель Мамдани, согласно которой для обработки продукционных правил используется минимаксная композиция нечетких множеств, а в качестве форм функций принадлежности для всех лингвистических переменных баллы выбрана гауссова кривая.

Подавая эти значения на вход системы нечеткого вывода, содержащей 243 продукционных правила ($3^5 = 243$ – максимально возможное число правил), получаем, что оценка за выполнение задания принадлежит к диапазону «средняя». Используя в качестве метода дефаззификации метод COG (center of gravity, «центр тяжести»), получим результат диагноза: 52,1 балла по 100-балльной шкале, что соответствует диапазону «удовлетворительно» (47 – 54%) при пятидиапазонной (квинтельной) стандартизации нормального распределения.

Выводы

Разработанная методика проведения ДЭ по анализу компетентности пользователей КС позволяет перейти к автоматизации этого достаточно сложного и неформального процесса. Предложенная система оценивания квалификационных заданий различных форм позволяет применить единый подход при формировании итоговой оценки (результата диагноза). Наиболее сложной проблемой при этом является оценивание заданий открытой формы с развернутым ответом. На сегодняшний день, несмотря на то, что подобные задания используются даже в ЗНО, их оценивание является непрозрачной и достаточно субъективной процедурой. Предложенная процедура, когда эксперт оценивает формальные критерии в заданных диапазонах, а экспертная система выставляет итоговую оценку, позволит, с точки зрения авторов, построить автоматизированную экспертную систему для получения результата диагноза компетентности пользователей КС.

Список литературы: 1. Концепція тесту загальної навчальної компетентності (ТЗНК) випускників загальноосвітніх навчальних закладів (Проект) // Вісник. Тестування і моніторинг в освіті. 2009. № 9. С. 5–28. 2. Ильязова М.Д. Компетентность, компетенция, квалификация – основные направления современных исследований // Научные исследования в образовании. 2008. № 1. С. 28–31. 3. Аванесов В.С. Форма тестовых заданий. Учебное пособие, 2 изд. переработанное и расширенное. М.: Центр тестирования, 2005. 156 с. 4. Горох В.П. Готуємося до зовнішнього оцінювання з математики // Вісник. Тестування і моніторинг в освіті. 2009. № 1. С. 25–35. 5. Раков С.А., Ващуленко О.П., Горох В.П., Милянник А.І., Пузирьов В.В. Три виміри логіко-математичної компетентності // Вісник. Тестування і моніторинг в освіті. 2009. № 12. С. 6–17. 6. Шкиль А., Каук В., Напрасник С., Цимбалюк Е., Хоменко Р. Новые функциональные возможности компьютерной системы тестирования знаний OPENTEST2 // Педагогические измерения. 2009. № 2. С. 86–103. 7. Кривуля Г.Ф., Кучеренко Д.Е. Интеллектуальные средства диагностирования состояний компьютерных систем управления // Інформаційно-керуючі системи на залізничному транспорті. №4. 2009. С. 23–28. 8. Кривуля Г.Ф., Шкиль А.С., Напрасник С.В., Гаркуша Е.В. Представление результатов тестирования в компьютерной системе тестирования знаний OpenTEST2 // Інформаційно-керуючі системи на залізничному транспорті. 2008. №2. С. 25–30.

Поступила в редколлегию 25.02.2010

Кривуля Геннадий Федорович, д-р техн. наук, проф. кафедры АПВТ ХНУРЭ. Научные интересы: автоматизация проектирования и диагностика цифровых устройств. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-13-26. E-mail: krivoulya@i.ua.

Шкиль Александр Сергеевич, канд. техн. наук, доцент кафедры АПВТ ХНУРЭ. Научные интересы: дистанционное образование, педагогические измерения, диагностика цифровых систем. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-13-26. E-mail: shkil@opentest.com.ua.

Кучеренко Дария Ефимовна, аспирантка кафедры АПВТ ХНУРЭ. Научные интересы: диагностика компьютерных неисправностей. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-13-26. E-mail: d_zin@ukr.net.

Гаркуша Елена Владимировна, аспирантка кафедры АПВТ ХНУРЭ. Научные интересы: педагогические измерения, анализ компетентностей пользователей КС. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-13-26.