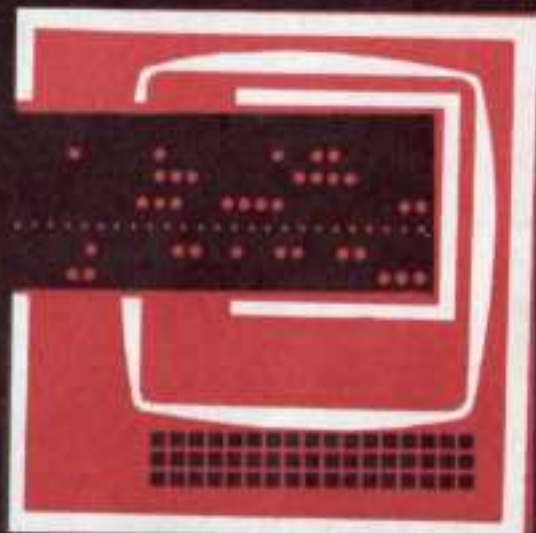


АВТОМАТИЗИ- РОВАННЫЕ СИСТЕ- МЫ УПРАВЛЕНИЯ

И ПРИБОРЫ АВТОМАТИКИ

74|85



681.5(06)
A22

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО
СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ УССР

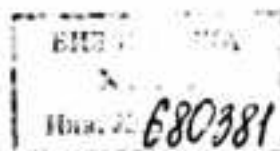
ХАРЬКОВСКИЙ ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ
ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ ИМЕНИ М. К. ЯНГЕЛЯ

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И ПРИБОРЫ АВТОМАТИКИ

Республиканский
межведомственный
научно-технический
сборник

Основан в 1965 г.

В Ы П У С К 74



ХАРЬКОВ
ИЗДАТЕЛЬСТВО ПРИ ХАРЬКОВСКОМ
ГОСУДАРСТВЕННОМ УНИВЕРСИТЕТЕ
ИЗДАТЕЛЬСКОГО ОБЪЕДИНЕНИЯ
„ВИЩА ШКОЛА“
1985

Автоматизированные системы управления и приборы автоматики: Респ. междувед. науч.-техн. сб.—Х.: Вища шк. Изд-во при Харьк. ун-те, 1985.— Вып. 74, 144 с.

В сборнике представлены результаты научных исследований в области создания и совершенствования автоматизированных систем управления технологическими процессами. Рассмотрены вопросы оптимизации, идентификации, моделирования сложных систем, организации баз данных алгоритмического обеспечения систем автоматизированного проектирования.

Нормативные материалы приведены по состоянию на 1 января 1985 г. Для научных работников и специалистов.

Редакционная коллегия: Е. П. Путятин (отв. ред.), В. Ф. Шостак (зам. отв. ред.), Н. А. Лысенков (отв. секр.), В. И. Васильев, Э. А. Дедиков, В. В. Дубровский, А. Г. Евдокимов, И. В. Кузьмин, С. Ф. Кацадан, Н. Д. Коноваленко, Е. Н. Лаваренко, В. И. Мелешко, В. М. Михайленко, И. Ф. Огороднейчук, Э. Г. Петров, Л. А. Растриниц, В. В. Свиридов, Ю. Г. Стоин, И. Н. Шнырев

Адрес редакционной коллегии: 310141, Харьков-141, пр. Ленина, 14, институт радиоэлектроники, тел. 40-94-59

Редакция естественнонаучной литературы

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И ПРИБОРЫ АВТОМАТИКИ

Выпуск 74

Редактор *А. П. Гужва*
Художественный редактор *Т. П. Воробийко*
Технический редактор *Г. П. Александрова*
Корректоры *Е. В. Сергина, И. В. Балакирева*

ИБ № 9499

Сдано в набор 22.06.84. Подп. в печать 18.10.84. БЦ 00298. Формат 60×90/16. Бумага типогр. № 3. Лит. гарн. Вмс. печать. 9 печ. л. 9,25 кр.-отт. 10 уч.-изд. л. Тираж 1000 экз. Изд. № 1206. Зак. 4-239. Цена 1 р. 40 к.

Издательство при Харьковском государственном университете издательского объединения «Вища школа», 310003, Харьков-3, ул. Университетская, 16

Харьковская книжная фабрика «Коммунист», 810012, Харьков-12, ул. Энгельса, 11

0604020101-005
А М226(04)-85 478-85

© Издательское объединение
«Вища школа», 1985

АВТОМАТИЗАЦИЯ ПОСТРОЕНИЯ МЕТОДОМ R -ФУНКЦИЙ АНАЛИТИЧЕСКИХ ОПИСАНИЙ МНОГОГРАННЫХ ГЕОМЕТРИЧЕСКИХ ОБЪЕКТОВ ТРЕХМЕРНЫХ СЦЕН

В работах [1, 2] на основе теории R -функций [3] созданы методика и алгоритмы формирования визуальной обстановки на экране телевизионного монитора по аналитической модели динамической пространственной сцены. Пространственная сцена состоит из m движущихся и неподвижных (образующих фон и препятствия) пространственно-односвязных n -гранных объектов D_i с границей D_i^0 , в общем случае с криволинейными гранями Γ_i^l , $\bigcup_{l=1}^n \Gamma_i^l = D_i^0$, l -источников света и промежуточной среды. Аналитическая модель сцены включает в себя системы функций $\{S_k\}$, описывающих характеристики промежуточной среды и освещенности, $\{Q_i^l\}$, описывающих отражательные способности граней Γ_i^l , и построенные методом R -функций аналитические описания $\{f^i(x, y, z)\}$ геометрических объектов D_i , обладающие свойствами

$$(V(x, y, z) \in D_i^0) (f^i(x, y, z) = 0);$$

$$(V(x, y, z) \in (R^3 \setminus D_i^0)) (f^i(x, y, z) < 0), \quad i = 1, \dots, m. \quad (1)$$

Автоматизировать построение аналитической модели пространственной сцены — значит автоматизировать построение аналитических описаний f^i со свойствами (1) для геометрических объектов D_i .

Статья посвящена разработке алгоритмов автоматического построения методом R -функций аналитического описания $f(x, y, z)$ со свойствами (1) для трехмерного геометрического объекта, представляющего собой ограниченную пространственно-односвязную область $D \subset R^3$, граница D^0 которой состоит из участков заданных плоскостей. Будем рассматривать произвольный n -гранник D указанного класса с гранями Γ_i ($i = 1, \dots, n$), которые есть произвольные ограниченные односвязные или многосвязные многоугольники. Пусть для данного многогранника выполняется условие, что любые его две смежные грани не принадлежат одной и той же плоскости.

В соответствии с определенной [1, 3] формальной процедурой аналитическое описание $f(x, y, z)$ многогранника будем строить с помощью достаточно полной системы R_3 -функций $\{xV_iy, x\Lambda_iy, \bar{x}\}$ в форме

$$f(x, y, z) = \bigvee_{i=1}^n f_i(x, y, z), \quad (2)$$

где n — количество граней многогранника, f_i^* — аналитическое описание грани Γ_i

$$\begin{aligned} & (\forall (x, y, z) \in \Gamma_i) (f_i^*(x, y, z) = 0); \\ & (\forall (x, y, z) \in (R^3 \setminus \Gamma_i)) (f_i^*(x, y, z) < 0), \quad i \in \{1, \dots, n\}. \end{aligned} \quad (3)$$

Аналитическое описание $f_i^*(x, y, z)$ грани Γ_i построим в виде

$$f_i^*(x, y, z) = (-|f_i|) \wedge \varphi_i. \quad (4)$$

Здесь f_i — левая часть уравнения плоскости, участок которой является гранью Γ_i ; φ_i — непрерывная в R^3 функция, которая

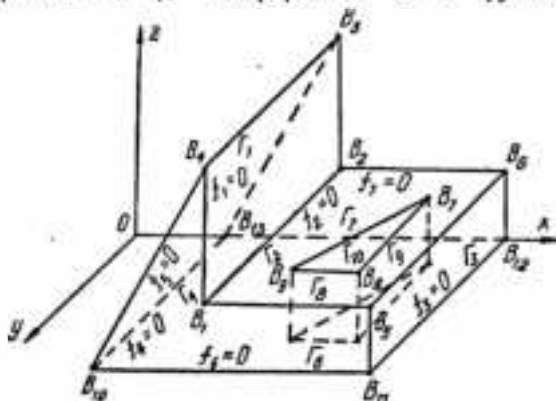


Рис. 1. Пример пространственно-односвязного многогранника с плоскими гранями

описывает некоторую область $\Omega_i \subset R^3$, содержащую грань Γ_i и не содержащую остальные точки плоскости $f_i = 0$. Для выполнения условий (3) функция φ_i в (4) в точках области Ω_i должна быть неотрицательной, т. е.

$$\begin{aligned} & (\forall (x, y, z) \in \Omega_i) (\varphi_i(x, y, z) \geq 0); \\ & (\forall (x, y, z) \in (R^3 \setminus \Omega_i)) (\varphi_i(x, y, z) < 0). \end{aligned} \quad (5)$$

Как следует из формальных построений (2)—(4) аналитического описания $f(x, y, z)$, удовлетворяющего условиям (1), существо их полной автоматизации на ЭВМ заключается в автоматизации построения функций φ_i в уравнении (4).

Основные принципы алгоритмизации построения функций φ_i , $i = 1, \dots, n$, для заданного n -гранника состоят в следующем. Пусть задан n -гранник D , например 10-гранник, изображенный на рис. 1. Каждая из граней Γ_i является участком плоскости, описываемой уравнением $f_i = 0$. Плоскость $f_i = 0$ разбивает пространство R^3 на две открытые области D_i^+ , D_i^- , определяемых неравенствами $f_i > 0$, $f_i < 0$ соответственно.

Обозначим через

$$\alpha_1, \dots, \alpha_p; \alpha_j \in \{1, \dots, n\}, j = 1, \dots, p; p < n \quad (6)$$

список номеров смежных с Γ_i граней, упорядоченный относительно заданного обхода внутренних точек грани Γ_i объекта D . Список (6) включает в себя номера смежных граней по ребрам, ограничивающим грань Γ_i . Данные ребра для грани Γ_i , отвечающие списку (6), будем обозначать через

$$r_{i, \alpha_1}, \dots, r_{i, \alpha_p} \quad (7)$$

Номерам (6) соответствуют области $D_{\alpha_j}^+$, $D_{\alpha_j}^-$ и функции f_{α_j} , описывающие эти области указанным выше образом. Каждому номеру α_j поставим в соответствие такую из областей $D_{\alpha_j}^+$, $D_{\alpha_j}^-$, $j \in \{1, \dots, p\}$, которая в окрестности точек ребра r_{i, α_j} смежности содержит внутренние точки грани Γ_i . Тогда последовательности номеров (6) смежных граней с Γ_i отвечает множество областей

$$D_{\alpha_1}^{\sigma_1}, \dots, D_{\alpha_p}^{\sigma_p} \quad (8)$$

где $\sigma_j = +$, если уравнение $f_{\alpha_j} = 0$ ориентировано положительно относительно внутренних точек грани Γ_i , т.е. если $f_{\alpha_j} > 0$ во внутренних точках грани Γ_i , принадлежащих окрестности ребра r_{i, α_j} ; $\sigma_j = -$, если $f_{\alpha_j} = 0$ ориентировано отрицательно относительно грани Γ_i (при этом $f_{\alpha_j} < 0$ во внутренних точках грани Γ_i из окрестности ребра r_{i, α_j}). Областям (8) поставим в соответствие функции

$$f_{\alpha_1}^{\sigma_1}, \dots, f_{\alpha_p}^{\sigma_p}; f_{\alpha_j}^{\sigma_j} = \begin{cases} f_{\alpha_j}, & \text{если } \sigma_j = +; \\ \bar{f}_{\alpha_j}, & \text{если } \sigma_j = -, \end{cases} \quad j = 1, \dots, p. \quad (9)$$

Здесь $\bar{f}_{\alpha_j} = -f_{\alpha_j}$ — R -отрицание над f_{α_j} ; σ_j — индекс из $D_{\alpha_j}^{\sigma_j}$.

Далее с помощью операций алгебры множеств — пересечения и объединения — из областей (8) формируется теоретико-множественное представление области $\Omega_i = \Omega_i(\cap, \cup, D_{\alpha_1}^{\sigma_1}, \dots, D_{\alpha_p}^{\sigma_p})$, содержащей точки грани Γ_i и только эти точки из плоскости $f_i = 0$. Логике построения области Ω_i формальным образом ставится в соответствие R -функция $f_{R, i}(\wedge, \vee, x_1, \dots, x_p)$. Суперпозиция R -функции $f_{R, i}$ и функций (9)

$$\varphi_i = f_{R, i}(\wedge, \vee, f_{\alpha_1}^{\sigma_1}, \dots, f_{\alpha_p}^{\sigma_p}) \quad (10)$$

является аналитическим описанием области Ω_i и представляет собой искомую функцию φ_i в (4).

К построению для грани Γ_i области Ω_i и функции φ_i , ее описывающей, не предъявляется иных требований, кроме тех,

что область Ω_i должна содержать грань Γ_i и не включать в себя остальные точки плоскости $f_i = 0$, а функция φ_i должна удовлетворять условиям (5). Поэтому при любых взаимных расположениях в пространстве R^3 грани Γ_i и смежных с ней граней $\Gamma_{s_1}, \dots, \Gamma_{s_p}$ логика формирования области Ω_i из областей (8) обуславливается конфигурацией грани Γ_i . Конфигурация грани Γ_i определяется взаимным расположением ребер (7), ограничивающих грань Γ_i и таким образом лежащих в одной плоскости. В этой связи задача автоматического построения R -функции $f_{R,i}(\wedge_i, \vee_i, x_1, \dots, x_p)$ в (10) может рассматриваться как плоская задача в R^2 для многоугольника, ограниченного ребрами (7) как сторонами, и решаться с помощью алгоритмов автоматического построения аналитических описаний многосвязных многоугольников [2—5]. Поскольку R -функции инвариантны относительно вещественных групп Ли [6] преобразований пространства, то данные построения могут выполняться по проекции грани Γ_i на одну из координатных плоскостей.

Таким образом, автоматизация построения функции φ_i в (4) для грани Γ_i заданного многогранника сводится к автоматизации решения следующих задач.

1. Формирование путем анализа заданной численной информации о многограннике упорядоченного списка (6) номеров граней, смежных с гранью Γ_i .

2. Построение проекции грани Γ_i на одну из координатных плоскостей.

3. Построение R -функции $f_{R,i}(\wedge_i, \vee_i, x_1, \dots, x_p)$ в (10) по проекции грани Γ_i на координатную плоскость с применением алгоритмов автоматического формирования аналитического описания многосвязного многоугольника.

4. Определение ориентации уравнений $f_{s_j} = 0$ плоскостей, участки которых есть смежные грани с гранью Γ_i , и формирование индексов δ_j в уравнении (9), определяющих применение R -отрицаний над функциями f_{s_j} .

5. Формирование функции φ_i в виде выражения (10) для грани Γ_i путем суперпозиции R -функции $f_{R,i}$ и функций (9).

Опишем формальные процедуры решения данных задач и общий алгоритм автоматического построения аналитического описания (2) заданного пространственно-односвязного n -гранника.

Информацию о многограннике будем задавать путем задания численной информации о каждой его грани в едином информационном массиве. В общем случае грань многогранника исследуемого класса представляет собой ограниченный многосвязный многоугольник. Поэтому информацией о грани Γ_i являются координаты ее вершин, выписанные с соблюдением принятого обхода внутренности грани как многоугольника при движении по ребрам грани, составляющим замкнутые связанные контуры. Для каждой грани обход ее вершин может быть принят свой (левый или правый),

но единый для всех связанных контуров. При этом обход всех контуров данной грани должен осуществляться с какой-либо одной (внутренней или внешней) стороны объекта в зависимости от удобства снятия информации с чертежа. В общем информационном массиве о многосвязной грани Γ_i геометрическая информация о контуре, который является внешним относительно всех других контуров, ограничивающих грань Γ_i , предшествует информации о других (внутренних относительно данного) контурах. Информация об указанных внутренних контурах в общем массиве располагается вслед за информацией о внешнем контуре и может следовать одна за другой в любом наборе. Так, численную информацию о грани Γ_2 многогранника, изображенного на рис. 1, составляет последовательность координат (x, y, z) вершин $B_1, B_5, B_6, B_2, B_1, B_7, B_8, B_9, B_7$. Для каждой грани указывается число связанных контуров, ограничивающих грань, и список количества ребер в контурах. При задании информации нумерация граней осуществляется в произвольном порядке. Задается количество граней.

Запишем уравнение $f_i = 0$ плоскости, которой принадлежит грань Γ_i , через координаты первых трех точек $B_{i,1}, B_{i,2}, B_{i,3}$ массива информации о грани Γ_i в виде

$$f_i = \begin{vmatrix} x & y & z & -1 \\ x_{i,1} & y_{i,1} & z_{i,1} & -1 \\ x_{i,2} & y_{i,2} & z_{i,2} & -1 \\ x_{i,3} & y_{i,3} & z_{i,3} & -1 \end{vmatrix} = 0. \quad (11)$$

Раскрывая определитель, входящий в уравнение (11), по верхней строке получаем уравнение плоскости общего вида $f_i = A_i x + B_i y + C_i z + D_i = 0$, в котором коэффициенты A_i, B_i, C_i, D_i выражаются через соответствующие определители третьего порядка. Вычисленные таким образом коэффициенты составляют двумерный массив, в котором i -я строка содержит коэффициенты для f_i ($i = 1, \dots, n$).

Далее для каждой грани $\Gamma_i, i \in \{1, \dots, n\}$ n -гранника формируется список (7) номеров смежных с Γ_i граней. Формирование j -го элемента списка (7) выполняется путем последовательной проверки включений

$$r_{i,j} \subseteq r_{k,c}, r_{k,c} \subseteq r_{i,j}; k \neq i, k = 1, \dots, n, \quad (12)$$

для ребра $r_{i,j}$ грани Γ_i и ребер остальных граней Γ_k . Если для ребра $r_{i,j}$ грани Γ_i и для некоторого ребра $r_{k,c}$ грани Γ_k хотя бы одно из включений (12) имеет место, индекс k запоминается в списке номеров смежных граней для Γ_i в качестве j -го элемента списка. Алгоритм выполняет построение списков номеров смежных граней для всех граней многогранника, что составляет двумерный массив номеров, в котором i -я строка содержит список номеров для грани Γ_i .

Дальнейшая работа алгоритма автоматического построения аналитического описания многогранника заключается в решении задач (3)—(5) с использованием проекций граней Γ_i ($i = 1, \dots, n$) на одну из координатных плоскостей и компиляции аналитического описания в форме (2).

В целях минимизации используемой памяти ЭВМ решения задач (2)—(5) выполняются последовательно для каждой грани Γ_i объекта.

Рассмотрим построение ортогональной проекции грани Γ_i заданного n -гранника на одну из координатных плоскостей $x = 0$ (yOz), $y = 0$ (zOx) или $z = 0$ (xOy).

Грань Γ_i задается последовательностью координат $\{x, y, z\}$ ее вершин. Поэтому построение ортогональной проекции грани Γ_i на какую-либо из координатных плоскостей осуществляется тривиальным отождествлением соответствующей пары координат в трехмерном измерении с координатами в двумерной системе координат на выбранной координатной плоскости. Следовательно, проекция точки $(x, y, z) \in R^3$ на координатные плоскости $x = 0$, $y = 0$, $z = 0$ имеет вид (y, z) (z, x), (x, y) соответственно.

Выбор координатной плоскости проецирования грани Γ_i выполняется из следующих соображений. Плоскость $f_i = 0$, участок которой есть грань Γ_i , имеет произвольное расположение в пространстве R^3 . В частности, плоскость $f_i = 0$ может иметь ортогональное или близкое к нему расположение относительно какой-либо координатной плоскости. Например, в случае многогранника (рис. 1) грань Γ_6 и плоскость $f_6 = 0$ ортогональны к координатной плоскости $z = 0$. В данном случае грань Γ_6 на плоскости $z = 0$ проецируется в отрезок прямой, а задающие ее точки — в точки, лежащие на этом отрезке. Если плоскость $f_i = 0$ расположена к координатной плоскости под углом, мало отличающимся от угла $\pi/2$, то ортогональная проекция грани Γ_i на эту плоскость имеет форму, в которой определяющие ее точки являются геометрически близкими к одной прямой. Эти обстоятельства обуславливают неустойчивость работы алгоритмов анализа геометрической информации о проекции грани Γ_i и преобразования ее в аналитическую информацию.

Для обеспечения устойчивости работы алгоритмов необходимо проецировать грань Γ_i на ту из трех координатных плоскостей, угол между которой и плоскостью $f_i = 0$ наиболее отличается от угла $\pi/2$. Вычислим косинусы углов между координатными плоскостями $x = 0$, $y = 0$, $z = 0$ и плоскостью $f_i = Ax + By + Cz + D_i = 0$: $\cos \beta_x$, $\cos \beta_y$, $\cos \beta_z$ соответственно. Рассмотрим функцию

$$K_i = \max \{ |\cos \beta_x|, |\cos \beta_y|, |\cos \beta_z| \}, \quad (13)$$

где θ совпадает с тем из индексов x, y или z при β , о модуле косинуса которого совпадает функция K_i . Функция K_i определяет выбор координатной плоскости проецирования. Построение проек-

ции грани Γ_i на выбранную координатную плоскость состоит в формировании массива двумерных координат вершин грани Γ_i по исходному массиву ее трехмерных координат (x, y, z) .

Пусть проекция грани Γ_i , $i \in \{1, \dots, n\}$ многогранника построена. Рассмотрим процессы автоматического синтеза R -функции $f_{R,i}$ в (10) по проекции грани Γ_i с использованием алгоритма автоматического построения аналитического описания произвольного многоугольника.

При построении проекции грани Γ_i на выбранную по определению в (13) критерию координатную плоскость возможны следующие случаи. Массив двумерных координат, задающих проекцию грани Γ_i , в смысле левого обхода (принятого в алгоритмах

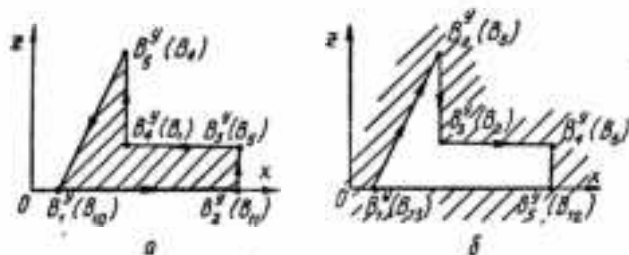


Рис. 2. Примеры ограниченной (а) и неограниченной (б) области в R^2 , определенной информацией о проекции грани

описания многоугольника) внутренних точек проекции Γ_i при семантике чтения массива слева направо определяет:

а) ограниченную в R^2 область, например проекцию на координатную плоскость $y=0$ грани Γ_6 (рис. 2, а) многогранника, изображенного на рис. 1;

б) неограниченную в R^2 область, например проекцию на плоскость $y=0$ грани Γ_7 (рис. 2, б) многогранника (рис. 1).

Распознавание данных случаев отображения грани $\Gamma_i \subset R^3$ в проекцию Γ_i в R^2 выполняется по следующим правилам. Анализируя слева направо массив информации о проекции внешнего контура грани Γ_i , среди вершин данной проекции осуществляем поиск хотя бы двух крайних точек области в R^2 , ограниченной проекцией внешнего контура Γ_i . Если таковые существуют, область в R^2 проекции грани Γ_i ограничена, в противном случае — неограничена. Алгоритм формирует признак δ , характеризующий ограниченность ($\delta = 2$) либо неограниченность ($\delta = 0$) области в R^2 , определенной информацией о проекции грани Γ_i .

Построение R -функции $f_{R,i}$ в (10) по проекции грани Γ_i выполняется с применением алгоритма аналитического описания многоугольника. Алгоритм по информации о многоугольной области проекции грани Γ_i строит R -функцию $f_{R,i}(\Lambda_1, V_1, x_1, \dots, x_n)$. В случае, когда область проекции грани Γ_i в соответствии с указанными пунктами а, б является ограниченной (признак $\delta = 2$), полученная R -функция $f_{R,i}$ — искомая. Если признак $\delta = 0$,

к построенной R -функции $f_{R,i}$ необходимо применить операцию R -отрицания (в этом случае искомой является R -функция $\bar{f}_{R,i}$, которая соответствует ограниченной области). Например, для проекции грани Γ_6 (рис. 2, а) вырабатывается признак $\delta = 2$ и строится R -функция $f_{R,6} = x_1 \wedge x_2 \wedge (x_3 \vee x_4) \wedge x_5$. Данная R -функция является искомой. В случае проекции грани Γ_7 (рис. 2, б) указанными алгоритмами формируются R -функция $f_{R,7} = x_1 \vee (x_2 \wedge x_3) \vee x_4 \vee x_6$ и признак $\delta = 0$. Поэтому искомой является R -функция $\bar{f}_{R,7}$.

Далее устанавливается ориентация относительно внутренних точек грани Γ_i уравнений $f_{s,j} = 0$ плоскостей смежных с Γ_i граней и определяются индексы δ_j в (9), указывающие на применение R -отрицаний к функциям $f_{s,j}$, $j = 1, \dots, p$.

В общем случае плоскость $f_{s,i} = 0$, которой принадлежит смежная с Γ_i грань $\Gamma_{s,i}$, разбивает множество вершин грани Γ_i на два подмножества, одно из которых включается в область $D_{s,i}^+$, а другое — в область $D_{s,i}^-$. Для установления того, какая из областей $D_{s,i}^+$, $D_{s,i}^-$ содержит в окрестности ребра $r_{i,s,i}$ внутренние точки грани Γ_i , необходимо выбрать некоторую проверочную точку T_1 на грани Γ_i , лежащую с той же стороны плоскости $f_{s,i} = 0$, что и точки грани Γ_i из окрестности ребра $r_{i,s,i}$. В качестве проверочной точки T_1 используем одну из вершин внешнего контура, ограничивающего грань Γ_i . Выбор номера точки T_1 производится на проекции грани Γ_i с учетом случаев а, б по условию: проекция точки T_1 лежит слева (справа) от линии действия вектора

$B_i^s B_{i+1}^s$ в случае ограниченной (неограниченной) области проекции Γ_i . Вектор $B_i^s B_{i+1}^s$ соответствует проекции ребра $r_{i,s,i}$, например, вектор $B_i^s B_{i+1}^s$, $i \in \{1, \dots, 5\}$ (см. рис. 2). Затем вычисляется значение функции $f_{s,i}$ в точке $T_1(x_{T_1}, y_{T_1}, z_{T_1})$. Индекс $\sigma_i = -$ в (8) и индекс $\delta_i = -$ в (9) в случае, когда $f_{s,i}(x_{T_1}, y_{T_1}, z_{T_1}) > 0$ и признак $\delta = 0$, а также если $f_{s,i}(x_{T_1}, y_{T_1}, z_{T_1}) < 0$ и $\delta = 2$. Эта информация запоминается путем сообщения знака «—» i -му элементу списка номеров смежных граней для грани Γ_i .

Дальнейшая работа алгоритма заключается в формировании суперпозиции (10) с учетом возможного R -отрицания над $f_{R,i}$ по признаку δ_i в построении описания грани Γ_i в виде (4) и последовательной компиляции аналитического описания многогранника в форме (2).

Результатом работы алгоритма автоматического построения аналитического описания n -гранника является массив коэффициентов n формообразующих плоскостей и представленное в виде единой формулы аналитическое описание. Отметим, что в развитых

макроязыках ряда проблемно-ориентированных систем [3], базирующихся на теории R -функций, такое представление описания многогранника может быть использовано в качестве автоматически построенных геометрических операторов программы на макроязыке для исследования конкретной задачи анализа или синтеза физического поля. При этом для обеспечения принадлежности описания (2) классу C^m m -раз непрерывно-дифференцируемых функций необходимо вместо системы R_3 -функций $\{x_1 \wedge x_2, x_1 \vee x_2, \bar{x}\}$ использовать систему $\{x_1 \wedge^m x_2, x_1 \vee^m x_2, \bar{x}\}$ [3] и в формуле (4) вместо $|f_i|$ взять $-f_i^2$. В задачах автоматизации формирования визуальной обстановки более эффективным по быстрдействию есть аналитическое описание n -гранника, представленное в виде n отдельных описаний граней в соответствии с [1].

Описанный алгоритм реализован на ЭВМ. В качестве одного из контрольных примеров рассматривался многогранник (рис. 1).

Список литературы: 1. Пулятин Е. П., Лискин В. М., Тоница В. С. Методы и алгоритмы формирования визуальной обстановки на экране телевизионного монитора по аналитической модели пространственной сцены. — АСУ и приборы автоматики, 1984, вып. 70, с. 9—14. 2. Лискин В. М., Тоница В. С. Применение методов геометрической логики для повышения эффективности обработки визуальной информации в задачах обучения. — Пробл. бионики, 1982, вып. 28, с. 39—46. 3. Рашев В. Л. Теория R -функций и некоторые ее приложения. — К.: Наук. думка, 1982. — 551 с. 4. Тоница В. С., Шкляров Л. И. Об одном алгоритме автоматического построения аналитических образов сложных геометрических объектов. — Вычисл. техника в машиностроении, 1974, вып. 3, с. 3—14. 5. Тоница В. С. Об автоматизации решения методом R -функций краевых задач математической физики. — В кн.: Вычислительная математика в современном научно-техническом прогрессе: Вычислительные методы в алгебре, прикладной математике, в системах обработки данных и АСУ. К., 1974, с. 644—653. 6. Серр Ж.-П. Алгебры Ли и группы Ли. — М.: Мир, 1969. — 324 с.

Поступила в редакцию 05.11.83.

УДК 681.518

А. П. ТАРАСЕНКО, В. П. МАШТАЛИР, Б. В. ХИТРОВ

ПОСЛЕДОВАТЕЛЬНАЯ НОРМАЛИЗАЦИЯ ПРОЕКТИВНЫХ
ПРЕОБРАЗОВАНИЙ

В задачах цифровой обработки визуальной информации, в частности при реставрации изображений, возникает необходимость компенсации геометрических искажений. Такие искажения являются следствием изменения взаимного расположения датчика видеoinформации и наблюдаемого объекта [1]. При этом математической моделью деформаций изображений выступает группа проективных преобразований плоскости, являющаяся гомеоморфным образом трехмерной группы движе-

Опуская несложные, но громоздкие выражения, получаем

$$\frac{\partial B_0}{\partial u} = (a_7x + a_8y + a_9) \left[\frac{\partial B}{\partial x} (\gamma_1 - x\gamma_7) - \frac{\partial B}{\partial y} (y\gamma_7 - \gamma_4) \right],$$

$$\frac{\partial B_0}{\partial v} = (a_7x + a_8y + a_9) \left[\frac{\partial B}{\partial y} (\gamma_2 - y\gamma_8) - \frac{\partial B}{\partial x} (x\gamma_8 - \gamma_2) \right].$$

Здесь γ_i — элементы обратной матрицы A , $i = \overline{1, 9}$.

Между функционалами, построенными на изображении B_0 , B , и параметрами преобразования, существует в общем случае довольно сложная нелинейная зависимость, которую можно выразить следующим образом: $f(\Phi_i(B_0), \Phi_m(B), \gamma_i) = 0$. С целью получения более простых зависимостей (типа полиномиальной относительно γ_i) рассмотрим следующую совокупность функционалов:

$$\Phi_i(B_0) = \iint_D \varphi_i(B_0(u, v)) \left(\frac{\partial B_0}{\partial u} \right)^3 du dv, \quad i = \overline{1, 3}.$$

Тогда, полагая, что $J > 0$, и осуществляя замену переменных

$$\begin{aligned} \Phi_i(B_0) &= \iint_D \varphi_i(B_0(u, v)) \left(\frac{\partial B_0}{\partial u} \right)^3 du dv = \\ &= |A|^3 \iint_D \varphi_i(B(x, y)) \left[\frac{\partial B}{\partial x} (\gamma_1 - x\gamma_7) - \frac{\partial B}{\partial y} (y\gamma_7 - \gamma_4) \right]^3 dx dy \end{aligned}$$

или, вводя обозначения $\alpha_k = |A| \gamma_k$, $k = 1, 4, 7$, получаем систему алгебраических уравнений:

$$\Phi_i(B_0) = \iint_D \varphi_i(B) \left[\frac{\partial B}{\partial x} (\alpha_1 - x\alpha_7) - \frac{\partial B}{\partial y} (y\alpha_7 - \alpha_4) \right]^3 dx dy, \quad (1)$$

$$i = \overline{1, 3}$$

Решение данной системы алгебраических уравнений можно получить, воспользовавшись методом исключения с использованием результата Сильвестра [4].

Рассуждая аналогично, можно получить линейную систему уравнений относительно параметров a_7, a_8, a_9 . Для этого рассмотрим функционалы вида

$$\begin{aligned} \Psi_i(B_0) &= \iint_D \Psi_i(B_0(u, v)) \left(\frac{\partial B_0}{\partial u} \right)^4 du dv = \\ &= \iint_D \Psi_i(B(x, y)) \left[\frac{\partial B}{\partial x} (\alpha_1 - x\alpha_7) - \frac{\partial B}{\partial y} (y\alpha_7 - \alpha_4) \right]^4 \times \\ &\quad \times (a_7x + a_8y + a_9) |A| dx dy. \end{aligned} \quad (2)$$

Подставляя один набор решений системы (1) $\alpha_1, \alpha_4, \alpha_7$ в (2), определяем соответствующий набор параметров a_7, a_8, a_9 . Для нахождения остальных параметров преобразования $a_1, a_2, a_3, a_4, a_5, a_6$ можно использовать уже нормализаторы преобразований, в которых параметры входят линейно [5]. Следовательно, мы получаем различные наборы параметров a_i , $i = \overline{1, 9}$, соответствующие различным наборам $\alpha_1, \alpha_4, \alpha_7$. Вопрос о том, какие наборы

Опуская несложные, но громоздкие выражения, получаем

$$\begin{aligned}\frac{\partial B_0}{\partial u} &= (a_7x + a_8y + a_9) \left[\frac{\partial B}{\partial x} (\gamma_1 - x\gamma_7) - \frac{\partial B}{\partial y} (y\gamma_7 - \gamma_4) \right], \\ \frac{\partial B_0}{\partial v} &= (a_7x + a_8y + a_9) \left[\frac{\partial B}{\partial y} (\gamma_3 - y\gamma_8) - \frac{\partial B}{\partial x} (x\gamma_8 - \gamma_2) \right].\end{aligned}$$

Здесь γ_i — элементы обратной матрицы A , $i = \overline{1, 9}$.

Между функционалами, построенными на изображении B_0 , B , и параметрами преобразования, существует в общем случае довольно сложная нелинейная зависимость, которую можно выразить следующим образом: $f(\Phi_i(B_0), \Phi_m(B), \gamma_i) = 0$. С целью получения более простых зависимостей (типа полиномиальной относительно γ_i) рассмотрим следующую совокупность функционалов:

$$\Phi_j(B_0) = \iint_D \varphi_j(B_0(u, v)) \left(\frac{\partial B_0}{\partial u} \right)^3 dudv, \quad j = \overline{1, 3}.$$

Тогда, полагая, что $J > 0$, и осуществляя замену переменных

$$\begin{aligned}\Phi_j(B_0) &= \iint_D \varphi_j(B_0(u, v)) \left(\frac{\partial B_0}{\partial u} \right)^3 dudv = \\ &= |A|^3 \iint_D \varphi_j(B(x, y)) \left[\frac{\partial B}{\partial x} (\gamma_1 - x\gamma_7) - \frac{\partial B}{\partial y} (y\gamma_7 - \gamma_4) \right]^3 dxdy\end{aligned}$$

или, вводя обозначения $\alpha_k = |A| \gamma_k$, $k = 1, 4, 7$, получаем систему алгебраических уравнений:

$$\Phi_j(B_0) = \iint_D \varphi_j(B) \left[\frac{\partial B}{\partial x} (\alpha_1 - x\alpha_7) - \frac{\partial B}{\partial y} (y\alpha_7 - \alpha_4) \right]^3 dxdy, \quad (1)$$

$$j = \overline{1, 3}$$

Решение данной системы алгебраических уравнений можно получить, воспользовавшись методом исключения с использованием результата Сильвестра [4].

Рассуждая аналогично, можно получить линейную систему уравнений относительно параметров a_7, a_8, a_9 . Для этого рассмотрим функционалы вида

$$\begin{aligned}\Psi_i(B_0) &= \iint_D \Psi_i(B_0(u, v)) \left(\frac{\partial B_0}{\partial u} \right)^4 dudv = \\ &= \iint_D \Psi_i(B(x, y)) \left[\frac{\partial B}{\partial x} (\alpha_1 - x\alpha_7) - \frac{\partial B}{\partial y} (y\alpha_7 - \alpha_4) \right]^4 \times \\ &\quad \times (a_7x + a_8y + a_9) |A| dxdy.\end{aligned} \quad (2)$$

Подставляя один набор решений системы (1) $\alpha_1, \alpha_4, \alpha_7$ в (2), определяем соответствующий набор параметров a_7, a_8, a_9 . Для нахождения остальных параметров преобразования $a_1, a_2, a_3, a_4, a_5, a_6$ можно использовать уже нормализаторы преобразований, в которых параметры входят линейно [5]. Следовательно, мы получаем различные наборы параметров a_i , $i = \overline{1, 9}$, соответствующие различным наборам $\alpha_1, \alpha_4, \alpha_7$. Вопрос о том, какие наборы

параметров связывают изображения $B_0(x, y)$ и $B(x, y)$, может быть решен путем использования критерия невырожденности матрицы A [6].

Таким образом, предложенная последовательная процедура нормализации плоских изображений при проективных преобразованиях позволяет, используя незначительные ограничения на функцию яркости, определять параметры компенсирующего преобразования.

Список литературы: 1. Huang T. S. Trends in digital image processing research. — In.: Advances in Digital Image Processing, Application, Implementation, 1979, p. 21—30. 2. Корня Т. В. Геометрические преобразования и прикладная многомерная геометрия. — Сб. науч. тр., 1974, вып. 307, с. 25—31. 3. Путьятин Е. П., Сергеева Н. К. Нормализация плоских изображений при проективных преобразованиях. — Оптико-когерент. информационно-измер. системы, 1977, вып. 1, с. 85—90. 4. Ван-дер-Варден. Современная алгебра. — Л.: изд. ОТГЛ, 1937. — 211 с. 5. Долженкова Т. Г., Тарасенко А. П., Горюховатский В. А. Программа определения параметров нормализации некоторых нелинейных преобразований. — РЧАП, 1980, № 5731, с. 48—51. 6. Путьятин Е. П., Хитров Б. В. Нормализация изображений, подвергнутых аффинной группе преобразований. — РЧАП, 1981, № 5926, с. 37—42.

Поступила в редколлегию 12.01.84.

УДК 519.653

В. Ф. ШОСТАК, д-р техн. наук, С. В. КОТЛЯРЕВСКИЙ

ЗАДАЧА СИНТЕЗА ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДОПУСТИМЫХ ПОГРЕШНОСТЕЙ ИТЕРАЦИОННЫХ АЛГОРИТМОВ КООРДИНАЦИИ ОДНОГО КЛАССА

Оперативная координация управляемых подсистем — одна из наиболее важных функций иерархических систем управления сложными технологическими комплексами непрерывного типа. Каждая подсистема рассматриваемого комплекса в установившемся режиме описывается непрерывно-дифференцируемой функцией цели:

$$\dot{y}_i = \varphi_i(\bar{m}_i, \bar{x}_i, \bar{z}_i), \quad i = 1, N \quad (1)$$

непрерывно-дифференцируемой вектор-функцией выходных переменных взаимодействия подсистем

$$\bar{y}_i = \bar{y}_{ij}(\bar{m}_i, \bar{x}_i, \bar{z}_i), \quad i, j = 1, N; i \neq j \quad (2)$$

и системой ограничений в форме неравенств, которые обусловлены конструктивными и технологическими условиями функционирования подсистем

$$\bar{m}_i \in M_i, \quad \bar{x}_i \in X_i, \quad i = \overline{1, N}. \quad (3)$$

Здесь N — количество подсистем комплекса; m_i — вектор управляющих воздействий i -й подсистемы; $\bar{z}_i$ — вектор возмущающих воздействий i -й подсистемы; $\bar{x}_i$ — вектор входных переменных взаимодействия $\bar{x}_i = [\bar{x}_{il}]$, $l = \overline{1, N}$; $\bar{x}_{il}$ — вектор, учитывающий влияние l -й подсистемы на i -ю; M_i , X_i — множества допустимых значений векторов m_i и $\bar{x}_i$ соответственно.

Задача оперативной оптимизации состоит в определении

$$\begin{aligned} \max_{\bar{x}, \bar{m}} \sum_{i=1}^N \varphi_i(\bar{m}_i, \bar{x}_i, \bar{z}_i), \\ \bar{x} = \{\bar{x}_i\}, \quad i = \overline{1, N}, \\ \bar{m} = \{\bar{m}_i\}, \quad i = \overline{1, N} \end{aligned} \quad (4)$$

при прогнозируемых оценках векторов возмущений $\bar{z}_i = \bar{z}_i^*$, ограничениях (3) и ограничениях, вытекающих из условия согласования подсистем

$$\bar{y}_{ij}(\bar{m}_i, \bar{x}_i, \bar{z}_i) - \bar{x}_{ij} = 0, \quad i, j = \overline{1, N}; \quad i \neq j. \quad (5)$$

Чтобы преодолеть трудности решения рассматриваемой задачи, обычно используют иерархическую двухуровневую структуру подзадач $[1] \quad {}^{(2)}R = \{{}^{(2)}D, [{}^{(1)}R_i^{(k)}], i = \overline{1, N}\}$, которая включает на нижнем уровне подзадачи оптимизации подсистем $\{{}^{(1)}R_i^{(k)}\}$, состоящие в определении

$$\max_{\bar{x}_i} \varphi_i(\bar{m}_i, \bar{x}_i^{(k)}, \bar{z}_i^*) \quad (6)$$

при условиях $\bar{y}_{ij}(\bar{m}_i, \bar{x}_i^{(k)}, \bar{z}_i^*) - \bar{x}_{ij}^{(k)} = 0$, $m_i \in M_i$ где $\bar{x}_i^{(k)}$, $\bar{x}_{ij}^{(k)}$ — заданные значения векторов переменных взаимодействия на k -й итерации алгоритма координации, и подзадачи координации ${}^{(2)}D$ на верхнем уровне. Подзадача координации состоит в определении таких значений взаимодействий $\bar{x}_i^*$ (координирующих воздействий), при которых решения подзадач $\{{}^{(1)}R_i^{(k)}\}$ являются одновременно решением общей задачи (1) — (3).

Для решения подзадачи координации целесообразно использовать алгоритм [1, 2], реализующий принцип прогнозирования взаимодействий и обладающий рядом известных преимуществ:

$$\bar{x}_{ij}^{(k+1)} = \bar{x}_{ij}^{(k)} + \gamma_{ij}^{(k)} \operatorname{sign} \left(\frac{\partial \bar{F}^{(k)}}{\partial x_{ij}} \right), \quad \bar{x}_{ij}^{(k+1)} = p(\bar{x}_{ij}^{(k+1)}). \quad (9)$$

Здесь k — номер итерации; $\gamma_{ij}^{(k)}$ — диагональная матрица шагов по координируемым переменным; p — оператор проектирования оценок координируемых переменных на множество допустимых значений; $\frac{\partial \bar{F}^{(k)}}{\partial x_{ij}}$ — вектор частных производных функции Лагранжа (10), соот-

ветствующий точке решения подзадачи нижнего уровня $\{^{(1)}R_l^{(k)}\}$ на k -й итерации,

$$F(\bar{m}, \bar{x}, \bar{\lambda}) = \sum_{i=1}^N \varphi_i(\bar{m}_i, \bar{x}_i^{(k)}, \bar{z}_i) + \sum_{i=1}^N \sum_{j=1}^N \bar{\lambda}_{ij} [\bar{y}_{ij} - \bar{x}_{ij}]. \quad (10)$$

В алгоритме (9) можно выделить следующие типы погрешностей: вычисления экстремумов в подзадачах $\{^{(1)}R_l^{(k)}\}$; согласования подсистем удовлетворения условий (7); вычисления множителей Лагранжа; вычисления частных производных по координируемым переменным; программной реализации алгоритма. В работе [2] показано, что устойчивость алгоритма координации определяется, главным образом, погрешностями согласования подсистем и что существуют такие конечные значения допустимых погрешностей согласования, при которых алгоритм координации устойчив к погрешностям, не превосходящим эти значения.

В качестве сходимости алгоритма координации [2] можно использовать значения косинусов углов $\alpha_l^{(k)}$, $k = \overline{1, K}$ (K — число итераций алгоритма (9), необходимое для решения подзадачи координации) между точными векторами $\frac{\partial F^{(k)}}{\partial x_{ij}}$ при идеальном выполнении условий (7) и их оценками $\frac{\hat{\partial F}^{(k)}}{\partial x_{ij}}$ при конечных значениях погрешностей согласования $\bar{y}_{ij}^{(k)} = \bar{y}_{ij}^{(k)} - \bar{x}_{ij}^{(k)}$. Алгоритм координации сходится, если выполняется неравенство

$$\cos \alpha_l^{(k)} = \frac{\left\langle \frac{\hat{\partial F}^{(k)}}{\partial x_i}, \frac{\hat{\partial F}^{(k)}}{\partial x_j} \right\rangle}{\left\| \frac{\hat{\partial F}^{(k)}}{\partial x_i} \right\| \left\| \frac{\hat{\partial F}^{(k)}}{\partial x_j} \right\|} > \beta_l, \quad \beta \in (0, 1]. \quad (11)$$

Предполагая, что погрешности согласования переменных взаимодействия подсистем $\{\bar{y}_{ij}^{(k)}\}$ на разных итерациях алгоритма (9) могут иметь разные значения, введем понятие последовательности векторов допустимых погрешностей алгоритма (9) $\{\bar{y}_i^{(k)}\}$, $k = \overline{1, K}$ (где $\bar{y}_i^{(k)} = \{\bar{y}_{ij}^{(k)}\}$, $i, j = \overline{1, N}$; $i \neq j$), при которых из

$$|y_{ij}(\bar{m}_i, \bar{x}_i^{(k)}, \bar{z}_i) - x_{ij}^{(k)}| < \bar{y}_{ij}^{(k)}, \quad (12)$$

$$i = \overline{1, n_{ij}}$$

следует выполнение условия (11).

Здесь $\hat{m}_i^{(k)}$ — приближенное решение подзадачи $^{(1)}R_l^{(k)}$ на k -й итерации алгоритма (9), удовлетворяющее условию (12) вместо условия (7); n_{ij} — размерность вектора переменных взаимодействия между i -й и j -й подсистемами.

Выбор последовательности допустимых погрешностей оказывает большое влияние на точность решения и во многом определяет вычислительные затраты на решение общей задачи с помощью двухуровневого алгоритма. Вычислительные затраты на решение общей задачи представляют собой сумму затрат S_D на решение подзадачи координации $^{(2)}D$ и затрат $S_i^{(k)}$ на решение подзадачи $\{^{(1)}R_i^{(k)}\}$ нижнего уровня на всех итерациях алгоритма координации:

$$S = S_D + \sum_{i=1}^N S_i^{(K)} = \sum_{k=1}^{K-1} (S_c^{(k)} + \sum S_i^{(k)}) + S_c^{(K)} + S_i^{(K)}, \quad (13)$$

где S — вычислительные затраты на решение общей задачи $^{(2)}R$; S_D — вычислительные затраты на решение подзадачи координации; $S_c^{(k)}$ — затраты на k -й итерации алгоритма координации на вычисление множителей Лагранжа, частных производных, значений шагов и координируемых переменных; $S_i^{(k)}$ — вычислительные затраты на решение подзадач $\{^{(1)}R_i^{(k)}\}$ на k -й итерации алгоритма координации.

Выбор последовательности допустимых погрешностей $\{\bar{\eta}_i^{(k)}\}$, $k = \overline{1, K}$ определяется двумя основными факторами: влиянием погрешностей на вычислительные затраты $S_i^{(k)}$ для решения подзадач $\{^{(1)}R_i^{(k)}\}$ на промежуточных итерациях; влиянием погрешностей на количество итераций алгоритма координации (9), необходимых для решения задачи. Указанные факторы отражают противоречивое влияние этих погрешностей на общие вычислительные затраты, которое проявляется, с одной стороны, в том, что с уменьшением $\{\bar{\eta}_i^{(k)}\}$ вычислительные затраты $S_i^{(k)}$ на решение подзадач $\{^{(1)}R_i^{(k)}\}$ монотонно возрастают. С другой стороны, уменьшение $\{\bar{\eta}_i^{(k)}\}$ приводит к уменьшению угла $\alpha_i^{(k)}$, определяемого выражением (10). Это повышает точность определения координируемых переменных взаимодействия на каждой итерации и уменьшает в конечном счете общее количество итераций. Таким образом, количество итераций алгоритма координации K и вычислительные затраты $S_i^{(k)}$ представляют собой некоторые функции последовательности допустимых погрешностей, т. е. $K = K(\{\bar{\eta}_i^{(k)}\})$, $S_i^{(k)} = S_i^{(k)}(\bar{\eta}_i^{(k)})$. Здесь $\bar{\eta}_i^{(k)} = \{\bar{\eta}_i^{(k)}\}$, $i = \overline{1, N}$.

Для уменьшения вычислительной сложности алгоритма (9) необходимо предварительно решить задачу синтеза последовательности допустимых погрешностей $\{\bar{\eta}_i^{(k)}\}$, которая обеспечивает минимум вычислительных затрат:

$$\min S = \min \left[\sum_{k=1}^{K(\bar{\eta})-1} (S_c^{(k)} + \sum_{i=1}^N S_i^{(k)}(\bar{\eta}_i^{(k)})) + S_c^{(K)} + \sum_{i=1}^N S_i^{(K)}(\bar{\eta}_i^{(K)}) \right], \quad (14)$$

где $\bar{\varepsilon}_{m_i}$ — точность решения подзадач $\{^{(1)}R_i^{(k)}\}$, определяемая требованиями к точности конечного решения; $\bar{\eta}$ — последовательность допустимых погрешностей

$$\bar{\eta} = \{\bar{\eta}^{(k)}\}, \quad k = \overline{1, K-1}.$$

Последовательность $\bar{\eta}_0$, доставляющая минимум функции (14), является оптимальной по критерию минимума вычислительных затрат на решение задач (1) — (3) с помощью алгоритма координации (9).

Из изложенного выше следует, что вычислительная сложность алгоритма координации (9) и точность получаемого решения определяются совокупностью различных типов погрешностей. В связи с этим представляет интерес сравнительная оценка влияния на вычислительную сложность алгоритма и точность решения различных типов погрешностей, в частности допустимых погрешностей согласования подсистем $\{\bar{\eta}_i^{(k)}\}$ и погрешностей $\{\bar{\varepsilon}_{m_i}^{(k)}\}$, определяющих точность решения подзадач $\{^{(1)}R_i^{(k)}\}$:

$$\|\bar{m}_i^{(k)} - \hat{m}_i^{(k)}\| < \|\bar{\varepsilon}_{m_i}^{(k)}\|. \quad (15)$$

Здесь $\bar{m}_i^{(k)}$, $\hat{m}_i^{(k)}$ — соответственно точное и приближенное решения подзадачи $^{(1)}R_i^{(k)}$ на k -й итерации алгоритма (9). Подзадачи $\{^{(1)}R_i^{(k)}\}$ с учетом погрешностей $\bar{\varepsilon}_{m_i}^{(k)}$ и $\bar{\eta}_i^{(k)}$ можно записать в следующем виде:

$$\max_{\bar{m}_i} \varphi_i(\bar{m}_i, \bar{x}_i^{(k)}, \bar{z}_i) \quad (16)$$

при условии

$$\bar{m}_i \in M_i \cap \hat{M}_i(\bar{\eta}_i^{(k)}), \quad (17)$$

где $\hat{M}_i(\bar{\eta}_i^{(k)})$ — множество возможных управлений $\bar{m}_i$, удовлетворяющих условию допустимой точности согласования переменных взаимодействия подсистем

$$\hat{M}_i(\bar{\eta}_i^{(k)}) = \{\bar{m}_i \mid \|\bar{y}_{ij}(\bar{m}_i, \bar{x}_i^{(k)}, \bar{z}_i) - x_{ij}^{(k)}\| < \|\bar{\eta}_{ij}^{(k)}\|\}. \quad (18)$$

Исходя из выражений (16), (17), легко доказать следующее утверждение: последовательность $\bar{\varepsilon}_{m_i} = \{\bar{\varepsilon}_{m_i}^{(k)}\}$, $k = \overline{1, K}$, $i = \overline{1, N}$ однозначно характеризует вычислительную сложность алгоритма координации (9) только в том случае, если приближенные решения $\hat{m}_i^{(k)}$ подзадач $\{^{(1)}R_i^{(k)}\}$ удовлетворяют условию (17).

Справедливость данного утверждения следует из того, что при $\bar{m}_i^{(k)} \notin M_i \cap \hat{M}_i(\bar{\eta}_i^{(k)})$ и $\hat{m}_i^{(k)} \in M_i$ оценки погрешностей $\{\bar{\varepsilon}_{m_i}^{(k)}\}$ не учитыва-

ют влияние согласования подсистем $\{\eta_i^{(k)}\}$ на показатель (11) и, следовательно, на вычислительную сложность алгоритма (9).

Другая важная особенность последовательности $\{\bar{\epsilon}_m^{(k)}\}$, $k = \overline{1, K}$ заключается в том, что точность решения общей задачи зависит только от последнего числа последовательности $\bar{\epsilon}_m^{(k)} = \{\bar{\epsilon}_{m_i}^{(k)}\}$, $i = \overline{1, N}$.

На основании изложенного выше можно предложить подход к решению задачи синтеза последовательностей, оптимальных по критерию вычислительной сложности.

1. Выбирается множество Q возможных зависимостей между $\bar{\epsilon}_{m_i}^{(k)}$ и $\bar{\eta}_i^{(k)}$, $i = \overline{1, N}$

$$\bar{\epsilon}_{m_i}^{(k)} = \psi_{iq}(\bar{\eta}_i^{(k)}), \quad q \in Q. \quad (19)$$

2. Определяется оптимальная последовательность $\{\bar{\eta}_i^{(k)}\}$, $k = \overline{1, K-1}$ в результате минимизации оценки вычислительной сложности S на множестве $Q \times \bar{v}^{(1)} \times \bar{v}^{(2)} \times \dots \times \bar{v}^{(K-1)}$, где $\bar{v}^{(k)}$ — пространства допустимых значений погрешностей $\bar{\eta}_i^{(k)} = \{\bar{\eta}_i^{(k)}\}$, $i = \overline{1, N}$.

Строгое решение описанной задачи получить чрезвычайно трудно, поэтому предложенный подход имеет методологическое значение и позволяет построить рациональную схему выбора допустимых вычислительных погрешностей с целью уменьшить вычислительную сложность алгоритма решения задачи (1) — (3):

1) погрешности $\{\bar{\eta}_i^{(k)}\}$ выбираются последовательно на каждой из промежуточных итераций исходя из условия (11);

2) для выбора погрешностей $\{\bar{\epsilon}_{m_i}^{(k)}\}$ на промежуточных итерациях выражение (19) можно использовать в следующем виде: $\|\bar{\epsilon}_{m_i}^{(k)}\| \approx \|\bar{\eta}_i^{(k)}\|$, $i = \overline{1, N}$; $k = \overline{1, K-1}$;

3) на последней итерации выбираются $\{\bar{\epsilon}_{m_i}^{(K)}\}$ и $\{\bar{\eta}_i^{(K)}\}$, исходя из заданной точности решения общей задачи (1) — (3).

Экспериментальная проверка изложенных результатов проведения на примере для двух вариантов выбора погрешностей согласования: $\min(\varphi_1 + \varphi_2)$; $\varphi_1 = (x_{21} - 1)^2 + m_{11}^2 + (m_{12} - 2)^2$; $y_{12} = x_{21} + 2m_{11} - 0,4m_{12}x_{21} + 0,3$; $\varphi_2 = (x_{12} - 0,5)^2 + m_{21}^2 + (m_{22} - 4)^2$; $y_{21} = 2x_{12} - 2m_{21} + 0,6m_{22}x_{12} - 1,2$.

В первом варианте на всех итерациях погрешности $\{\bar{\eta}_i^{(k)}\}$ согласования были выбраны постоянными исходя из требуемой точности решения задачи ${}^{(2)}R$. Во втором варианте погрешности согласования выбирались на основе предложенного подхода. Анализ результатов решения, представленных в таблице, позволяет сделать вывод о целесообразности использовать предложенный подход выбора допустимых погрешностей согласования подсистем и точностей решения подзадач первого уровня. Как видно из таблицы, на основе предложенного подхода получено более точное решение

Вариант	Значения переменных					
	x_{12}	x_{21}	m_{11}	m_{12}	m_{21}	m_{22}
1	0,49998	1,00114	0,00331	2,01737	0,00040	4,00651
2	0,49998	1,00114	-0,00079	2,00022	-0,00070	3,99941

Вариант	Значения функций цели			Количество итераций решения подзадач		
	φ_1	φ_2	$\varphi_1 + \varphi_2$	$(2)D$	$(1)R_1$	$(1)R_2$
1	$0,85 \cdot 10^{-3}$	$0,86 \cdot 10^{-4}$	$0,44 \cdot 10^{-3}$	24	538	619
2	$0,13 \cdot 10^{-5}$	$0,50 \cdot 10^{-6}$	$0,18 \cdot 10^{-5}$	24	320	353

общей задачи при одном и том же числе итераций алгоритмов координации и значительно меньших вычислительных затратах на решение подзадач нижнего уровня иерархии.

Список литературы: 1. Шостак В. Ф. Оптимизация сложных объектов в автоматизированных системах управления с иерархической структурой. — Децентрализ. методы управ., 1972, № 5, с. 141—148. 2. Шостак В. Ф. Устойчивость одного алгоритма оперативной координации подсистем непрерывных технологических комплексов. — АСУ и приборы автоматики, 1983, вып. 67, с. 3—11. 3. Полак Э. Численные методы оптимизации. — М.: Мир, 1974. — 376 с.

Поступила в редакцию 25.01.84

УДК 62—505

Е. Г. КУНИК, канд. техн. наук, В. Н. ХРАМЦОВ

ОБ АЛГОРИТМАХ ВЫДЕЛЕНИЯ ПАРЕТО-ОПТИМАЛЬНЫХ РЕШЕНИЙ В ЗАДАЧАХ ВЕКТОРНОЙ ОПТИМИЗАЦИИ

При оптимизации с учетом нескольких противоречивых критериев качества возникает вопрос о принадлежности допустимого решения X^0 области согласия X^* либо области компромиссов X^c . В последнем случае такое решение называют парето-оптимальным [1]. Рассмотрим два возможных подхода к задаче выделения парето-оптимальных решений.

Пусть в некоторой области Ω евклидова пространства E_n минимизируется m функций качества $f_1, f_2, \dots, f_m; f_i \in C^1(\Omega)$, $i = \overline{1, m}$. Тогда согласно [2] для любой внутренней точки X^0 области Ω справедливо

$$X^0 \in X^c \Leftrightarrow \exists \lambda_j (j = \overline{1, m}): \sum_{j=1}^m \lambda_j \nabla f_j(X^0) = 0; \\ \sum_{j=1}^m \lambda_j \neq 0, \lambda_j \geq 0 \quad (j = \overline{1, m}). \quad (1)$$

На этом утверждении и основан первый подход к задаче. Очевидно, что (1) эквивалентно существованию нетривиального положительно определенного решения системы однородных линейных уравнений:

$$\begin{aligned} b_{11}\lambda_1 + b_{12}\lambda_2 + \dots + b_{1m}\lambda_m &= 0; \\ b_{21}\lambda_1 + b_{22}\lambda_2 + \dots + b_{2m}\lambda_m &= 0; \\ &\vdots \\ b_{n1}\lambda_1 + b_{n2}\lambda_2 + \dots + b_{nm}\lambda_m &= 0. \end{aligned} \quad (2)$$

Здесь

$$b_{ij} = \frac{\partial f_i}{\partial x_j} \Big|_{x=x^*}, \quad X = (x_1, x_2, \dots, x_n).$$

Интерес представляет случай, когда ранг системы (2) меньше m , иначе $X^0 \in X^*$.

Выделив среди переменных $\lambda_1, \lambda_2, \dots, \lambda_m$ k линейно-независимых, приходим к системе

$$\begin{aligned}\lambda_1 &= a_{1,r+1}\lambda_{r+1} + \dots + a_{1,m}\lambda_m \\ \lambda_2 &= a_{2,r+1}\lambda_{r+1} + \dots + a_{2,m}\lambda_m \\ &\vdots \\ \lambda_r &= a_{r,r+1}\lambda_{r+1} + \dots + a_{r,m}\lambda_m\end{aligned}\quad (3)$$

где $r = m - k$. Наличие такого набора базисных переменных $\lambda_{r+1}, \dots, \lambda_m$, для которого выполняются условия

$$\lambda_l \geq 0, \quad l = \overline{1, m}, \quad (4)$$

$$\sum_{i=1}^n \lambda_i \neq 0, \quad (5)$$

и свидетельствует о существовании искомого решения системы (2).

Потребовав выполнения (5), а также условия $\lambda_j > 0, j = \overline{r+1, m}$, сведем задачу к минимизации кусочно-линейной функции:

$$F(\lambda_1, \lambda_2, \dots, \lambda_m) = \sum_{i=1}^m (|\lambda_i| - \lambda_i) = F(\lambda_{r+1}, \dots, \lambda_m) = \\ = \sum_{j=1}^r \left(\left| \sum_{i=r+1}^m a_{ij} \lambda_j \right| - \sum_{i=r+1}^m a_{ij} \lambda_j \right). \quad (6)$$

Вид поверхности функции (6) позволяет так организовать процесс оптимизации, чтобы исключить попадание в начало координат.

Задавшись начальной точкой, удовлетворяющей сформулированным ранее условиям, осуществляем последовательный спуск вдоль тех координат, по которым

$$\frac{\partial F}{\partial \lambda_k} = \sum_{i=1}^r (\text{sign}(\lambda_i) a_{ik} - a_{ik}) < 0.$$

Поиск заканчивается либо при достижении значения $F = 0$, т. е. получено искомое решение системы (2), либо когда $F \neq 0$, $\frac{\partial F}{\partial \lambda_k} > 0$,

$k = \overline{r+1, m}$. В этом случае система (2) не имеет нетривиальных положительно определенных решений, что и свидетельствует о принадлежности точки X^0 области согласия. Алгоритм прост, точен и не требует больших затрат машинной памяти.

Второй подход основан на следующем утверждении [3]. При условиях, сформулированных ранее, точка X^0 будет принадлежать области компромиссов, если система неравенств

$$AH > 0, \quad (7)$$

где $H = (h_1, h_2, \dots, h_n)^T$ — неизвестный вектор из E_n ; A — матрица размером $m \times n$ с элементами $a_{ij} = \left. \frac{\partial f_i}{\partial x_j} \right|_{X=X^0}$, не имеет нетривиальных решений. Если же система (7) имеет нетривиальные решения, то точка X^0 принадлежит области согласия. Процедура, определяющая разрешимость системы (7), реализуется следующим алгоритмом.

Для всех столбцов матрицы A подсчитываются: t_{1j} — число положительных элементов в j -м столбце; t_{2j} , t_{3j} — соответственно число отрицательных и нулевых элементов в j -м столбце. Затем определяем j_0 : $v_i = \min_{i=1, n} v_i$, где $v_i = t_{1j} t_{2j} + t_{3j}$. Разрешая каждое из неравенств системы (7) относительно h_{j_0} , получаем систему неравенств

$$h_{j_0} \sim L_i(h_1, \dots, h_n \setminus h_{j_0}), \quad i = \overline{1, m}. \quad (8)$$

Здесь $\sim$ означает отношение $>$ либо $<$, в зависимости от знака a_{ij_0} ; $L_i(h_1, \dots, h_n \setminus h_{j_0})$ — суть линейные комбинации переменных $h_1, \dots, h_n$, исключая h_{j_0} . Неравенства системы (7), в которых $a_{ij_0} = 0$, опускаем.

Неравенствами 1-го типа назовем те неравенства системы (8), в которых стоит знак $>$; неравенства же со знаком $<$ — неравенствами 2-го типа. Если в системе (8) нет неравенств разных типов или (что то же самое) $t_{1j_0} = 0$, либо $t_{2j_0} = 0$, то $X^0 \in X^*$. В противном случае составляем комбинации каждого неравенства 1-го типа с каждым неравенством 2-го типа вида

$$L_k(h_1, \dots, h_n \setminus h_{j_0}) < L_s(h_1, \dots, h_n \setminus h_{j_0}), \quad (9)$$

где $k = \overline{1, t_{2j_0}}$, $s = \overline{1, t_{1j_0}}$.

От системы (9) переходим к

$$L_i(h_1, \dots, h_n \setminus h_i) > 0. \quad (10)$$

Система (10) аналогична системе (7), но содержит одним неизвестным меньше. Если в (10) появятся неравенства вида $0 > 0$, то $X^0 \in X^c$. В противном случае возвращаемся к началу алгоритма и так повторяем до тех пор, пока не останется одно неизвестное. Система (7) примет вид

$$b_i h_i > 0, \quad i = \overline{1, p}, \quad 1 \leq i \leq n. \quad (11)$$

Если все b_i одного знака, то $X^0 \in X^*$, в противном случае — $X^0 \in X^c$.

Оба подхода просты в реализации, однако первый представляется более предпочтительным, так как при определенных условиях [3] значение p во втором алгоритме может быть велико.

Соображения простоты и наглядности определили выбор следующего тестового примера. В пространстве E_3 минимизировались четыре функции качества:

$$\begin{aligned} f_1 &= x_1^2 + (x_2 - 7)^2 + x_3^2, \quad f_2 = (x_1 - 8)^2 + (x_2 - 8)^2 + x_3^2, \\ f_3 &= (x_1 - 6)^2 + (x_2 - 3)^2 + x_3^2, \quad f_4 = (x_1 - 5)^2 + (x_2 - 6)^2 + \\ &\quad + (x_3 - 8)^2. \end{aligned}$$

Точки минимумов функции соответственно $X_1^* = (0, 7, 0)$, $X_2^* = (8, 8, 0)$, $X_3^* = (6, 3, 0)$, $X_4^* = (5, 6, 8)$ определили область компромиссов, представляющую собой пирамиду с вершинами в этих точках. Для всех узлов целочисленной решетки куба $0 \leq x_i \leq 10$; $i = 1, 2, 3$ обоими методами было проверено условие принадлежности области компромиссов. Обе программы дали один и тот же результат, полностью совпадающий с теоретическим.

Для граничных точек области Ω вопрос о принадлежности области компромиссов принципиально не отличается от рассмотренных выше [2, 3] и также сводится к разрешимости систем линейных уравнений и неравенств, аналогичных (2) и (7).

Список литературы: 1. Подиновский В. В., Носин В. Д. Парето-оптимальные решения многокритериальных задач. — М.: Наука, 1982. — 256 с. 2. Кулик Е. Г. Методы выделения области компромиссов в задачах векторной оптимизации. — АСУ и приборы автоматики, 1980, вып. 53, с. 52—62. 3. Кулик Е. Г., Житомирский М. Я. Об одном способе распознавания области компромиссов в задачах векторной оптимизации. — АСУ и приборы автоматики, 1983, вып. 69, с. 12—17. 4. Курош А. Г. Курс высшей алгебры. — М.: Наука, 1975. — 432 с.

Поступила в редколлегию 29.01.84.

ИССЛЕДОВАНИЕ И РАЗРАБОТКА МОДИФИЦИРОВАННЫХ
АДАПТИВНЫХ АЛГОРИТМОВ ИДЕНТИФИКАЦИИ

Для создания высокоэффективных систем автоматического управления важное значение имеет построение математических моделей управляемых объектов. Однако в тех случаях, когда управляемый объект характеризуется наличием неполной априорной информации, непредвиденным изменением в широких пределах внешних воздействий и динамических характеристик в условиях нормального функционирования, возникает задача создания адаптивных алгоритмов идентификации. Причем алгоритмы идентифицирующего устройства, находящегося в контуре управления, должны быть построены таким образом, чтобы для получаемых оценок вектора неизвестных параметров модель обладала прогнозирующими свойствами.

В [1—3] рассмотрены вопросы построения адаптивных алгоритмов идентификации. В [2] отмечено, что непрерывные адаптивные алгоритмы идентификации могут быть получены строго оптимальными в отличие от дискретных.

Рассмотрим непрерывный адаптивный алгоритм идентификации

$$\frac{dc(t)}{dt} = -\gamma(t) \nabla_c Q[\varphi(t), c(t)] \quad (1)$$

при

$$Q[\varphi(t), c(t)] = \frac{1}{2} M \|y(t) - c^T(t) \varphi(t)\|^2, \quad (2)$$

где $c(t) = [c_1(t), c_2(t), \dots, c_m(t)]^T$, $\varphi(t) = [\varphi_1(t), \varphi_2(t), \dots, \varphi_n(t)]^T$. Выражение (1) примет вид

$$\frac{dc(t)}{dt} = -\gamma(t) [y(t) - c^T(t) \varphi(t)] \varphi(t). \quad (3)$$

Здесь $c(t_0) = c_0$.

Численное решение уравнения (3) с использованием явной конечно-разностной схемы аппроксимации соответствует адаптивному алгоритму идентификации [1]

$$c_n = c_{n-1} - \gamma_n (\varphi_n - c_{n-1}^T \varphi_n) \varphi_n; \quad (4)$$

γ_n выбирается в виде $\gamma_n = \frac{q_n}{\|\varphi_n\|^2}$.

Рассмотрим случай [4] применения неявной конечно-разностной схемы аппроксимации, соответствующей устойчивому решению дифференциальных уравнений к длине шага

$$c_n = c_{n-1} - \gamma_n (\varphi_n - c_n^T \varphi_n) \varphi_n. \quad (5)$$

Перепишем уравнение (5) в виде

$$(I + \gamma_n \varphi_n \varphi_n^T) c_n = c_{n-1} + q_n y_n \varphi_n, \quad (6)$$

где I — единичная матрица размерности $m \times m$. Согласно [1] введем вектор ошибки определения оценки вектора параметров

$$\theta_n = h - c_n; \quad (7)$$

h — вектор параметров объекта.

Тогда (5) запишем в виде

$$\theta_n = \theta_{n-1} - \gamma_n \theta_{n-1}^T \varphi_n \varphi_n^T \quad (8)$$

или

$$\theta_{n-1} = \theta_n + \gamma_n \theta_n^T \varphi_n \varphi_n^T. \quad (9)$$

Умножая (9) скалярно на θ_{n-1} , получаем

$$\theta_{n-1}^T \theta_{n-1} = \theta_n^T \theta_n + 2\gamma_n \theta_n^T \varphi_n \theta_n^T \varphi_n + \gamma_n^2 (\theta_n^T \varphi_n)^2 \varphi_n^T \varphi_n. \quad (10)$$

Запишем условие сходимости алгоритма (5) с учетом (10):

$$Q_n = \theta_{n-1}^T \theta_{n-1} - \theta_n^T \theta_n = 2\gamma_n (\theta_n^T \varphi_n)^2 + \gamma_n^2 (\theta_n^T \varphi_n)^2 \varphi_n^T \varphi_n > 0. \quad (11)$$

При выборе $\gamma_n = \frac{q_n}{\|\varphi_n\|^2}$ из (11) получим

$$Q_n = \frac{(\theta_n^T \varphi_n)^2}{\|\varphi_n\|^4} (2q_n + q_n^2) > 0. \quad (12)$$

Условие (12) будет выполнено, если q_n удовлетворяет условиям $q_n \in]-\infty, -2[\cup]0, +\infty[$. Используя матричное тождество

$$(A + BCB)^{-1} = A^{-1} - A^{-1}B(C^{-1} + B^T A^{-1}B)^{-1}B^T A^{-1}, \quad (13)$$

из (6) получаем

$$c_n = \left(I - \frac{\varphi_n \varphi_n^T}{\|\varphi_n\|^2 + \frac{1}{\gamma_n}} \right) (c_{n-1} + \gamma_n y_n \varphi_n). \quad (14)$$

С учетом выбора γ_n (14) запишем в виде

$$c_n = \left(I - \frac{\varphi_n \varphi_n^T q_n}{(q_n + 1) \|\varphi_n\|^2} \right) \left(c_{n-1} + \frac{q_n y_n \varphi_n}{\|\varphi_n\|^2} \right). \quad (15)$$

Перемножив выражения, стоящие в скобках, из (15) получим

$$\begin{aligned} c_n = c_{n-1} - \frac{q_n}{q_n + 1} \frac{c_{n-1}^T \varphi_n}{\|\varphi_n\|^2} \varphi_n + \frac{q_n}{q_n + 1} \frac{y_n \varphi_n}{\|\varphi_n\|^2} = c_{n-1} + \\ + \frac{q_n}{q_n + 1} \frac{1}{\|\varphi_n\|^2} (y_n - c_{n-1}^T \varphi_n) \varphi_n. \end{aligned} \quad (16)$$

Из (16) следует, что в режиме обучения $q_n \in]+\infty, 10]$, в режиме фильтрации $q_n \rightarrow 0$ и выбирается таким образом, что

$$q_n/q_n + 1 \sim 1/n,$$

где n — числа натурального ряда.

Рассмотрим согласно [3] непрерывный вариант адаптивного алгоритма

$$\frac{dc(t)}{dt} = \frac{q}{\varphi^T(t) \varphi(t)} \left[\frac{dy(t)}{dt} - c^T(t) \frac{d\varphi(t)}{dt} \right] \varphi(t). \quad (17)$$

Тогда для неявной конечно-разностной схемы аппроксимации получим

$$c_n = c_{n-1} + \frac{q_n}{\|\varphi_0\|^2} [y_n - y_{n-1} - c_{n-1}^T (\varphi_n - \varphi_{n-1})] \varphi_n. \quad (18)$$

Перепишем уравнение (11) в виде

$$\left[I + \frac{(\varphi_n - \varphi_{n-1}) \varphi_n^T}{\|\varphi_n\|^2} \right] c_n = c_{n-1} + \frac{y_n - y_{n-1}}{\|\varphi_0\|^2} \varphi_n. \quad (19)$$

Используя тождество (13), из (19) получаем

$$\begin{aligned} c_n &= \left[I - \frac{\varphi_n (\varphi_n - \varphi_{n-1})^T}{\|\varphi_n\|^2 + \varphi_n^T (\varphi_n - \varphi_{n-1})} \right] \left[c_{n-1} + \frac{q_n (y_n - y_{n-1})}{\|\varphi_n\|^2} \varphi_n \right] = \\ &= c_{n-1} + \frac{q_n}{(q_n + 1) \|\varphi_n\|^2 - q_n \varphi_n^T \varphi_{n-1}} [(y_n - y_{n-1}) - c_{n-1}^T (\varphi_n - \varphi_{n-1})] \varphi_n. \end{aligned} \quad (20)$$

Полученный алгоритм (20) лучше использовать для нестационарных объектов, так как он обладает лучшими прогнозирующими свойствами к изменению параметров.

В качестве примера была рассмотрена модель

$$y = a_0 + a_1 t + a_2 t^2 + a_3 t^3, \quad (21)$$

где вектор параметров объекта был задан в виде $a(t) = \{\sin t, \sin 2t, \sin 3t, \sin 4t\}$.

Следует отметить, что в адаптивных алгоритмах (16) и (20) начальное приближение вектора параметров c_0 следует задавать в виде

$$c_0 = \frac{y_0 \varphi_0}{\|\varphi_0\|^2}, \quad (22)$$

если отсутствует априорная информация об их значениях. Формула (22) следует из требования выполнения условия $y_0 = c_0^T \varphi_0$.

В случае применения к алгоритму (17) явной конечно-разностной аппроксимации, мы приходим к алгоритму (16). При моделировании на ЭВМ рассмотренных выше алгоритмов $\|n - c_n\|^2$ для алгоритма (20) меньше, чем для алгоритма (16).

Список литературы: 1. Райбман Н. С., Чадков В. М. Построение моделей процессов производства. — М.: Энергия, 1965. — 376 с. 2. Цыпкин Я. З. Адаптация и обучение в автоматических системах. — М.: Наука, 1968. — 400 с. 3. Цыпкин Я. З. Основы теории обучающихся систем. — М.: Наука, 1970. — 251 с. 4. Норенков И. П., Маничев В. Б. Системы автоматизированного проектирования электронной и вычислительной аппаратуры. — М.: Высш. шк., 1983. — 272 с.

Поступила в редакцию 05.11.83.

К ВОПРОСУ РЕШЕНИЯ БИЛИНЕЙНЫХ МОДЕЛЕЙ
СЕТЕВЫХ СИСТЕМ

Проектирование новых и исследование существующих систем, для которых выполняются первый и второй постулаты сетей (аналоги законов Кирхгофа), связано с решением систем билинейных уравнений при двусторонних ограничениях на переменные. Топологической моделью сетевой системы является связанный ориентированный мультиграф $G(V, E)$, множеству узлов V которого соответствуют математические модели элементов в виде билинейных уравнений, а множество дуг E отображает связь между переменными модели. Задача состоит в нахождении решения (или наилучшего по заданному критерию приближения к решению) системы уравнений

$$f_{pj} = \sum_{i \in I_p} r_{pi} x_i y_{ij} = 0, \quad p \in V, \quad j = \overline{0, n} \quad (1)$$

при условии $[\bar{x}, \bar{y}] = \bar{z} \in D = D_x \cup D_y$, $D_x = \{\bar{x}: x_i^{\min} < x_i < x_i^{\max}, i = \overline{1, s}\}$, $D_y = \{\bar{y}: y_{ij} = 1, y_{ij}^{\min} < y_{ij} < y_{ij}^{\max}, i = \overline{1, s}, j = \overline{1, n}\}$ (2).

Здесь коэффициенты r_{pi} равны 1 или -1 соответственно для входных и выходных дуг p -го узла; I_p — множество номеров дуг, инцидентных p -му узлу; $s = \dim \bar{x}$; n — число типов билинейных уравнений.

Рассмотрим решение задачи (1), (2) в следующей постановке:

$$\min_{\bar{z} \in D} \left\{ F = \sum_{p \in V} \sum_{j=0}^n \alpha_{pj} f_{pj}^2 \right\}, \quad (3)$$

где α_{pj} — положительные весовые коэффициенты.

Поскольку функция F невыпукла, то одним из ключевых является вопрос об ее одноэкстремальности в области D .

Утверждение 1. Любой локальный минимум невыпуклой функции F , заданной на гиперпараллелепипеде D , одновременно является и глобальным.

Для доказательства утверждения задачу (3) сформулируем в эквивалентной постановке:

$$\sum_{p \in V} \sum_{j=0}^n f_{pj}^2 \rightarrow \min; \quad (4)$$

$$\alpha_{pj} \sum_{i \in I_p} r_{pi} x_i y_{ij} - f_{pj} = 0, \quad p \in V, \quad j = \overline{0, n}, \quad f_{pj} \in R^1, \quad \bar{x} \in D_x, \quad \bar{y} \in D_y$$

Здесь R^1 — вещественная ось.

В результате замены переменных $x_{ij} = h_{ij}$ задача (4) приводится к задаче выпуклого: квадратичного программирования:

$$\sum_{p \in V} \sum_{i=0}^n f_{pi}^2 \rightarrow \min;$$

$$x_{pi} \sum_{i \in I_p} r_{pi} h_{ij} - f_{pi} = 0, \quad p \in V, \quad j = \overline{0, n}, \quad (5)$$

$$h_{ij} - x_{ij}^{\min} > 0; \quad -h_{ij} + x_{ij}^{\max} > 0, \quad \bar{x} \in D_x, \quad h_{ij}, f_{pi} \in R^1.$$

Из эквивалентности задач (3), (5) и единственности (по значению целевой функции) решения (5) следует справедливость утверждения.

Задачу оптимизации выпуклой функции при невыпуклых билинейных ограничениях, являющуюся одноэкстремальной, естественно назвать задачей обобщенного выпуклого программирования, в частности обобщенного квадратичного программирования, если минимизируемым критерием является выпуклая квадратичная функция.

Приведенное доказательство утверждения 1 одновременно указывает на один из возможных методов решения системы билинейных уравнений в виде задачи выпуклого квадратичного программирования (5). При этом, однако, надо иметь в виду, что такой подход связан со значительным повышением размерности задачи, поскольку переход от задачи (3) к (5) приводит к увеличению числа переменных на $|V|(n+1)$ и появлению многочисленных функциональных ограничений в виде неравенств. Более целесообразным является подход, основанный на поочередном фиксировании векторов $\bar{y}$, $\bar{x}$ в задаче (3) и решении последовательности соответствующих задач выпуклого квадратичного программирования [1].

Введем множество всех решений задачи (3)

$$Z^* = \text{Arg min}_{\bar{z} \in D} F(\bar{z})$$

и представим вектор переменных $\bar{z} \in Z^*$ следующим образом:

$$\bar{z} = \{\bar{z}_1, \bar{z}_2\}, \quad \bar{z}_1 = \{z_i : \left| \frac{\partial F}{\partial z_i} \right| > 0\}, \quad \bar{z}_2 = \{z_i : \frac{\partial F}{\partial z_i} = 0\}.$$

Утверждение 2. В любой точке множества Z^* вектор граничных переменных $\bar{z}_1$ определяется однозначно.

Доказательство. Граничность координат вектора $\bar{z}_1$ следует из необходимых, а в нашем случае и достаточных, условий оптимальности решения задачи (3)

$$\frac{\partial F}{\partial z_i} > 0, \quad z_i = z_i^{\min}; \quad \frac{\partial F}{\partial z_i} < 0, \quad z_i = z_i^{\max}; \quad \frac{\partial F}{\partial z_i} = 0, \\ z_i^{\min} < z_i < z_i^{\max}. \quad (6)$$

Если множество Z^* состоит из одной точки, то единственность вектора $\bar{z}_1$ очевидна. Пусть мощность множества Z^* больше еди-

нишы. Доказательство проведем от противного, т.е. предположим, что двум различным точкам $\bar{z}^{(1)}$ и $\bar{z}^{(2)}$ множества Z^* соответствуют векторы $\bar{z}_1^{(1)}$ и $\bar{z}_1^{(2)}$, отличающиеся как составом переменных, так и их количеством. Введем множества $L_j = M_j \setminus M_1 \cap M_2$, $j = 1, 2$, где M_1, M_2 — множества индексов переменных, образующих соответственно векторы $\bar{z}_1^{(1)}, \bar{z}_1^{(2)}$, и определим следующие области: $D_j = \{\bar{z} : z_i^{\min} < z_i < z_i^{\max}, i \notin L_j; z_i^{\min} + \Delta_i < z_i < z_i^{\max} + \Delta_i, i \in L_j\}$, $j = 1, 2$. Здесь $\Delta_i = \delta |1 - 2 \operatorname{sign}(z_i - z_i^{\max})|$; δ — малое положительное число.

В точке $\bar{z}^{(1)} \in D_1$ для переменных $z_i, i \in L_1$ условия оптимальности (6) уже не будут выполняться и, следовательно, в этой области найдется точка $\bar{z} \in D_1$ такая, что $F(\bar{z}) < F(\bar{z}^{(1)})$. С другой стороны, поскольку в точке $\bar{z}^{(2)} \in D_1$ условия оптимальности не нарушаются, то $F(\bar{z}^{(2)}) = F(\bar{z}^{(1)})$. Таким образом, в области D_1 должны существовать неэквивалентные локальные минимумы. Но согласно утверждению 1 это невозможно, поэтому $L_1 = \emptyset$.

Аналогично, в точке $\bar{z}^{(2)} \in D_2$ для переменных $z_i, i \in L_2$ условия оптимальности не будут выполняться и, следовательно, найдется точка $\bar{z} \in D_2$ такая, что $F(\bar{z}) < F(\bar{z}^{(2)})$. Но поскольку в точке $\bar{z}^{(1)} \in D_2$ условия (6) не нарушаются, то $F(\bar{z}^{(1)}) = F(\bar{z}^{(2)})$. Отсюда, как в предыдущем случае, вытекает, что $L_2 = \emptyset$. Из определения множеств L_1 и L_2 получаем соотношения $M_1 \setminus M_1 \cap M_2 = \emptyset$, $M_2 \setminus M_1 \cap M_2 = \emptyset$, откуда следует равенство $M_1 = M_2$, доказывающее справедливость утверждения 2.

Рассмотрим систему уравнений, определяющую необходимые и достаточные условия оптимальности решения задачи (3) по координатам вектора $\bar{z}_1$

$$\frac{\partial F}{\partial z_i} = 0. \quad (7)$$

Для некоторого $\bar{z}^* \in Z^*$ и $\delta > 0$ определим множество $Z_\delta = \{\bar{z} : \bar{z} \in Z^*, \|\bar{z} - \bar{z}^*\| < \delta\}$. Предположим, что ранг матрицы Якоби системы (7) в области Z_δ остается постоянным.

Утверждение 3. *Размерность множества решений задачи (3) в области Z_δ равна дефекту матрицы Якоби системы уравнений (7) в этой области.*

Доказательство. В области Z^* вектор $\bar{z}_1$ определяется однозначно ($\bar{z}_1 = \bar{z}_1^*$), поэтому для любого $\bar{z} \in Z^*$ справедливо равенство $F(\bar{z}_1, \bar{z}_2) = F(\bar{z}_1^*, \bar{z}_2)$. Следовательно, вопрос о размерности множества решений задачи (3) в области Z_δ сводится к исследованию размерности множества решений системы (7) в этой же области. Поскольку функция $F(\bar{z})$ является непрерывно дифференцируемой в области D , то размерность множества решений задачи (3) в области Z_δ будет определяться дефектом матрицы Якоби.

Если среди уравнений системы (7) имеются линейно зависящие, то в этом случае заведомо дефект матрицы будет отличен от нуля и вектор $\bar{z}_2$ определяется неоднозначно. Такая ситуация возникает, например, когда две смежные переменные y_{mj} , y_{kj} , являющиеся компонентами вектора $\bar{z}_2$, инцидентны одним и тем же узлам p и g графа $G(V, E)$. Действительно, для переменных y_{mj} и y_{kj} уравнения (7) будут иметь вид

$$\begin{aligned}\frac{\partial F}{\partial y_{mj}} &= 2[a_p f_p f_{pm} x_m + a_g f_g f_{gm} x_m] = 0, \\ \frac{\partial F}{\partial y_{kj}} &= 2[a_p f_p f_{pk} x_k + a_g f_g f_{gk} x_k] = 0.\end{aligned}$$

Поскольку для смежных переменных $r_{pm} r_{gm} = -1$, $r_{pk} r_{gk} = -1$, нетрудно заметить, что после преобразований получаются два одинаковых уравнения.

Легко проверить, что одинаковые уравнения в системе (7) появляются и в случае, когда несмежные переменные y_{ij} , y_{kl} , $i, k \in I_p$ вектора $\bar{z}_2$ входят только в одно из билинейных уравнений (1).

Утверждение 4. В любой точке множества Z^* вектор невязок $\bar{\Phi} = \{f_{pj}\}$, $p \in V$, $j = \bar{0}, n$ уравнений (1) определяется однозначно.

Справедливость утверждения непосредственно вытекает из единственности решения задачи (5) по координатам f_{pj} , $p \in V$, $j = \bar{0}, n$, характеризующим невязки соответствующих уравнений.

На практике нахождение решения системы уравнений (1) с нулевым значением функции невязки $F(\bar{z})$ часто связано с коррекцией первоначально заданной области D в пределах, допустимых по технологическим условиям.

Введем расширенные области определения варьируемых переменных $D(k)$, получающиеся путем изменения граничных значений координат вектора $\bar{z}_1$ на k -м шаге коррекции области D :

$$D(0) = D, D(k+1) = D(k) \cup \left(\bigcup_{i \in \Gamma_k} D_k^{(i)} \right), k = 0, 1, 2, \dots$$

$$\begin{aligned}D_k^{(i)} = \{z_i : z_i^{\min} - \delta_k^{(i)} [1 - \operatorname{sign}(z_i - z_i^{\min}(k))] < z_i < z_i^{\max} + \\ + \delta_k^{(i)} [1 - \operatorname{sign}(z_i^{\max}(k) - z_i)]\}, i \in \Gamma_k,\end{aligned}\quad (8)$$

где Γ_k — множество индексов переменных, составляющих вектор граничных переменных $\bar{z}_1$ в области $D(k)$; $z_i^{\min}(k)$, $z_i^{\max}(k)$ — граничные значения переменной в области $D(k)$.

Так как $D(k) \subseteq D(k+1)$ то, очевидно, справедливо соотношение

$$\min \{F(\bar{z}) : \bar{z} \in D(k+1)\} \leq \min \{F(\bar{z}) : \bar{z} \in D(k)\}, \quad (9)$$

причем знак равенства в (9) возможен лишь в том случае, если $\Gamma_k = \emptyset$ (или, что то же самое, $D(k+1) = D(k)$). Равенство нулю

множества Γ_k означает, что в области $D(k)$ существует решение системы (1) с нулевым значением функции $F(\bar{z})$.

Рассмотрим вопрос о поведении минимального значения функции невязки при коррекции границы по переменной z_i , $i \in \Gamma$. Для простоты записи индекс k , характеризующий номер шага коррекции, будем опускать. Обозначим $F^*(\bar{z}^{(i)}) = \min \{F(\bar{z}) : \bar{z} \in D \cup D^{(i)}(\bar{z}^{(i)})\}$, где $D^{(i)}(\bar{z}^{(i)})$ определяется из выражения (8). При изменении допустимой области за счет расширения интервала варьирования переменной z_i , $i \in \Gamma$ оптимальное значение функции невязки $F^*(\bar{z}^{(i)})$ будет монотонно уменьшаться по крайней мере до тех пор, пока производная $\frac{\partial F}{\partial z_i}$ не станет равной нулю.

Уменьшение функции F^* будет происходить за счет изменения переменной z_i , а также в результате вызванного этим изменением нарушения условий оптимальности решения (б) по другим переменным. Как только соответствующее ограничение для переменной z_i станет пассивным в области Z^* , дальнейшее увеличение параметра $\bar{z}^{(i)}$ уже не будет влиять на минимальное значение функции невязки в области $D^{(i)}(\bar{z}^{(i)})$.

Фиксируя все переменные, кроме z_i , функцию невязки запишем в следующем виде:

$$F = a(z_i - b)^2 + c, \quad (10)$$

где $a > 0$, b, c — коэффициенты, зависящие от фиксированных значений координат вектора $\bar{z}$.

Такое представление дает возможность определить максимальную величину коррекции границы по переменной z_i и оценить при этом степень уменьшения функции невязки. Действительно, производная $\frac{\partial F}{\partial z_i}$ обращается в нуль при $z_i = b$, поэтому величину коррекции граничного значения z_i^p по этой переменной целесообразно выбирать в интервале $[0, |z_i^p - b|]$ ($\bar{z}^{(i)} = 0$ соответствует случаю, когда по технологическим условиям значение z_i^p изменять нельзя). Нижняя оценка уменьшения функции $F^*(\bar{z}^{(i)})$ в расширенной области равна

$$\Delta F_n(\bar{z}^{(i)}) = a(\bar{z}^{(i)})^2, \quad \bar{z}^{(i)} \in [0, |z_i^p - b|]. \quad (11)$$

Разработанный в [1] для решения задачи (3) метод покоординатного спуска с аналитическим нахождением минимума по каждой переменной основан на представлении функции невязки в форме (10) и поэтому может быть одновременно использован для коррекции границ допустимой области. Практически коррекцию можно организовать следующим образом. В заданной области D решается задача (3) и при выполнении на n -й итерации критерия

$$\frac{F^{(n-1)} - F^{(n)}}{F^{(n)}} < \varepsilon$$

для переменных z_i , $i \in \Gamma$ вычисляются оценки $\bar{z}_{\max}^{(i)} = |z_i^p - b|$.

В зависимости от технологических условий, накладываемых на переменные z_i определяются величины коррекции граничных значений $\delta^{(i)}$ по формуле $\delta^{(i)} = \lambda_i \delta_{\max}^{(i)}$, $\lambda_i \in [0, 1]$.

Граничные значения z_i^p изменяются на $\delta^{(i)}$, в новой области решается задача (3) и т. д., до тех пор, пока не будет найдено решение с нулевым значением функции невязки.

Если для решения задачи (3) используется метод, не учитывающий структурные особенности функции F , то $\delta^{(i)}$ можно определять, например, из соотношения $\delta^{(i)} = \lambda_i z_i^p$, $i \in \Gamma$.

Полученные результаты использовались при совершенствовании методического обеспечения диалоговой системы расчета балансовых уравнений сложных технологических комплексов [2].

Список литературы: 1. Романов П. Г. Декомпозиционные методы решения блочных моделей сетевых систем.— АСУ и приборы автоматизации, 1984, вып. 72, с. 34—41. 2. Романов П. Г. Диалоговая система расчета балансовых уравнений сложных технологических комплексов.— АСУ и приборы автоматизации, 1984, вып. 71, с. 17—23.

Поступила в редакцию 12.01.84.

УДК 681.5.015.42

А. Д. ТЕВЯШЕВ, канд. техн. наук, Н. В. ФЕДОРОВ, канд. техн. наук, С. И. КОЗЫРЕНКО

О ЛИНЕАРИЗАЦИИ И РЕШЕНИИ УРАВНЕНИЙ МОДЕЛИ УСТАНОВИВШЕГОСЯ ПОТОКОРАСПРЕДЕЛЕНИЯ В ИНЖЕНЕРНЫХ СЕТЯХ

При планировании режимов функционирования инженерных сетей известно, что исходные данные могут изменяться в некотором диапазоне значений. Поэтому важно определять величину возможных отклонений расчетных значений от плановых при изменении или неточном задании исходных данных. Для этой цели расчетные величины представим как функции исходных данных и разложим эти функции в ряд Тейлора, ограничившись линейными членами. Тогда изменение расчетной величины можно рассматривать как линейную функцию случайных величин — изменений исходных данных. Определение математического ожидания и дисперсии линейных функций нескольких случайных величин можно проводить по заданным числовым характеристикам аргументов, не обращая внимания на законы распределения.

Математическая модель установившегося потокораспределения в инженерных сетях (МУПР). Сеть представляет собой систему взаимодействия большого количества двухполюсных элементов двух типов — активных (компрессорные или насосные станции) и пассивных (участки трубопроводов). Входами сети являются все ее узловые вершины, через которые в сеть поступает целевой продукт, а ее выходами — все узловые вер-

шины, через которые осуществляется отбор целевого продукта. Общий поток целевого продукта, подаваемый в сеть, равен суммарному потоку, потребляемому из сети. Для представления сети в виде линейного связного графа реальная сеть дополняется нулевой вершиной и фиктивными участками, соединяющими входы и выходы сети с нулевой точкой. Выберем дерево графа сети так, чтобы его ветвями стали только реальные участки сети и фиктивная дуга, в которой не задан расход целевого продукта (присвоим этой дуге номер 1). Тогда математическую модель установившегося потокораспределения в инженерной сети можно описать системой уравнений

$$\psi_i(P_{in}, P_{ik}, q_i, \bar{\beta}_i) = 0, \quad i \in M, \quad (1)$$

$$\dot{q}_i = \sum_{r \in E_i} b_{ir} q_r, \quad i \in E_1, \quad (2)$$

где M — множество индексов реальных дуг; E_1 , E_2 — множества индексов ветвей и хорд графа сети; q_i — расход целевого продукта по i -й дуге; P_{in} , P_{ik} — давление в начале и в конце i -й дуги; $\bar{\beta}_i$ — вектор параметров, характеризующих i -ю дугу; b_{ir} — элемент цикломатической матрицы.

Для пассивных элементов уравнения (1) могут быть представлены в виде

$$P_{ik}^* - P_{in}^* - \beta_{pi} q_i^{p_i} \operatorname{sign} q_i, \quad (3)$$

а для активных:

$$\beta_{iai} P_{ik}^* - P_{in}^* - \beta_{2ai} (q_i - \beta_{3ai} P_{in})^{n_i} \operatorname{sign} (q_i - \beta_{3ai} P_{in}) = 0. \quad (4)$$

МУПР считается заданной, если заданы граничные условия функционирования сети, т. е. если для каждого входа и выхода сети заданы либо давление, либо расход по соответствующей фиктивной дуге.

Постановка задачи. Заданы значения

$$P_{1i}^0, P_{ik}^0 (i \in L_{22}), P_{in}^0 (i \in K_{22}), q_i^0 (i \in L_{21} \cup K_{21}), \\ \beta_{pi}^0 (i \in M_{11} \cup M_{21}), \beta_{iai}^0, \beta_{2ai}^0, \beta_{3ai}^0 (i \in M_{12} \cup M_{22}),$$

где

$$P_i^0 = \begin{cases} P_{ik}^0, & \text{если ветвь } i \text{ соответствует входу сети;} \\ P_{in}^0, & \text{если ветвь } i \text{ соответствует выходу сети;} \end{cases}$$

множества индексов: M_{11} , M_{12} — реальных ветвей, соответствующих пассивным и активным элементам; M_{21} , M_{22} — реальных хорд, соответствующих пассивным и активным элементам; L_{21} — хорд, соответствующих входам сети с заданным расходом; L_{22} — хорд, соответствующих выходам с заданным давлением; K_{21} — хорд, соответствующих входам с заданным расходом; K_{22} — хорд, соответствующих выходам с заданным давлением.

Истинные значения давлений и расходов на входах и выходах сети, а также значения параметров пассивных и активных элементов на планируемый интервал времени могут быть представлены в виде

$$P_1 = P_1^0 + \Delta P_1, \quad (5)$$

$$P_{ik} = P_{ik}^0 + \Delta P_{ik}, \quad i \in L_{22}, \quad (6)$$

$$P_{in} = P_{in}^0 + \Delta P_{in}, \quad i \in K_{22}, \quad (7)$$

$$q_i = q_i^0 + \Delta q_i, \quad i \in L_{21} \cup K_{21}, \quad (8)$$

$$\beta_{pi} = \beta_{pi}^0 + \Delta \beta_{pi}, \quad i \in M_{11} \cup M_{21}, \quad (9)$$

$$\beta_{1ai} = \beta_{1ai}^0 + \Delta \beta_{1ai}, \quad \beta_{2ai} = \beta_{2ai}^0 + \Delta \beta_{2ai}, \quad (10)$$

$$\beta_{3ai} = \beta_{3ai}^0 + \Delta \beta_{3ai}, \quad i \in M_{12} \cup M_{22},$$

где ΔP_1 , ΔP_{ik} , ΔP_{in} , Δq_i , $\Delta \beta_{pi}$, $\Delta \beta_{1ai}$, $\Delta \beta_{2ai}$, $\Delta \beta_{3ai}$ — случайные величины с математическим ожиданием, равным нулю, и дисперсией $\sigma_{P_1}^2$, $\sigma_{P_{ik}}^2$, $\sigma_{P_{in}}^2$, $\sigma_{q_i}^2$, $\sigma_{\beta_{pi}}^2$, $\sigma_{\beta_{1ai}}^2$, $\sigma_{\beta_{2ai}}^2$, $\sigma_{\beta_{3ai}}^2$; P_1^0 , P_{ik}^0 , P_{in}^0 , q_i^0 , β_{pi}^0 , β_{1ai}^0 , β_{2ai}^0 , β_{3ai}^0 — планируемые значения исходных данных.

Необходимо определить значения P_{ik}^0 ($i \in L_{21}$), P_{in}^0 ($i \in K_{21}$), q_i^0 ($i \in L_{22} \cup K_{22}$), β_{pi}^0 , соответствующие плановым значениям исходных данных, а также дисперсии случайных величин ΔP_{ik} , ΔP_{in} , Δq_i , $\Delta \beta_{pi}$, таких, что

$$q_i = q_i^0 + \Delta q_i; \quad (11)$$

$$q_i = q_i^0 + \Delta q_i, \quad i \in L_{22} \cup K_{22}; \quad (12)$$

$$P_{ik} = P_{ik}^0 + \Delta P_{ik}, \quad i \in L_{21}; \quad (13)$$

$$P_{in} = P_{in}^0 + \Delta P_{in}, \quad i \in K_{21}. \quad (14)$$

Алгоритм решения. Из системы уравнений (1), (2), используя свойства равенств (3), (4), можно выделить следующую независимо решаемую подсистему уравнений:

$$\sum_{i \in M_{11}} b_{1ri} \beta_{pi} q_i^{pi} \text{sign } q_i + \sum_{i \in M_{12}} b_{1ri} [(\beta_{1ai} - 1) P_{in}^* + \beta_{2ai} (q_i - \beta_{3ai} P_{in})^{ai} \text{sign } (q_i - \beta_{3ai} P_{in})] + \beta_{3ai} q_i^{ai} \text{sign } q_i = 0, \quad r \in M_{21}; \quad (15)$$

$$\sum_{i \in M_{11}} b_{1ri} \beta_{pi} q_i^{pi} \text{sign } q_i + \sum_{i \in M_{12}} b_{1ri} [-(\beta_{1ai} - 1) P_{in}^* + \beta_{2ai} (q_i - \beta_{3ai} P_{in})^{ai} \text{sign } (q_i - \beta_{3ai} P_{in})] - (\beta_{1ai} - 1) P_{in}^* + \beta_{2ai} (q_r - \beta_{3ai} P_{in})^{ai} \text{sign } (q_r - \beta_{3ai} P_{in}) = 0, \quad r \in M_{22}; \quad (16)$$

$$b_{1ri} P_{in}^* + \sum_{i \in M_{11}} b_{1ri} \beta_{pi} q_i^{pi} \text{sign } q_i + \sum_{i \in M_{12}} b_{1ri} [-(\beta_{1ai} - 1) P_{in}^* + \beta_{2ai} (q_i - \beta_{3ai} P_{in})^{ai} \text{sign } (q_i - \beta_{3ai} P_{in})] - P_{in}^* = 0, \quad r \in L_{22}; \quad (17)$$

$$b_{1r1}P_1^* + \sum_{i \in M_{11}} b_{1ri}\beta_{pi}q_i^{pi} \operatorname{sign} q_i + \sum_{i \in M_{12}} b_{1ri}[-(\beta_{1ai} - 1)P_{in}^* + \beta_{2ai}(q_i - \beta_{2ai}P_{in})^{ai} \operatorname{sign}(q_i - \beta_{2ai}P_{in})] + P_{rn}^* = 0, \quad r \in K_{22}, \quad (18)$$

значения, $i \in M_{11} \cup M_{12}$ выражаются через q_r ($r \in E_2$) по формуле (2), а P_{in} ($i \in M_{12} \cup M_{22}$) являются функциями величин P_i и q_r ($r \in E_2$) [1]. Подставив в систему (15) — (18) исходные данные и решив ее одним из методов, описанных в [1, 2], мы найдем q_r ($r \in M_{21} \cup M_{22} \cup L_{22} \cup K_{22}$). Значения P_{rk}^0 ($r \in L_{21}$), P_{rn}^0 ($r \in K_{21}$), q_i^0 ($i \in L_{22} \cup K_{22}$) определяются из следующих формул подстановкой найденных значений q_r^0 :

$$P_{rk} = \{b_{1r1}P_1^* + \sum_{i \in M_{11}} b_{1ri}\beta_{pi}q_i^{pi} \operatorname{sign} q_i + \sum_{i \in M_{12}} b_{1ri}[-(\beta_{1ai} - 1)P_{in}^* + \beta_{2ai}(q_i - \beta_{2ai}P_{in})^{ai} \operatorname{sign}(q_i - \beta_{2ai}P_{in})]\}^{1/2}, \quad r \in L_{21}, \quad (19)$$

$$P_{rk} = \{-b_{1r1}P_1^* - \sum_{i \in M_{11}} b_{1ri}\beta_{pi}q_i^{pi} \operatorname{sign} q_i - \sum_{i \in M_{12}} b_{1ri}[-(\beta_{1ai} - 1)P_{in}^* + \beta_{2ai}(q_i - \beta_{2ai}P_{in})^{ai} \operatorname{sign}(q_i - \beta_{2ai}P_{in})]\}^{1/2}, \quad r \in K_{21}, \quad (20)$$

$$q_i = \sum_{r \in E_2} b_{1ri}q_r, \quad i \in L_{22} \cup K_{22}, \quad i = 1. \quad (21)$$

С другой стороны, система уравнений (15) — (18) определяет q_r ($r \in M_{21} \cup M_{22} \cup L_{22} \cup K_{22}$) как неявные функции аргументов P_1 , P_{rk} ($r \in L_{22}$), P_{rn} ($r \in K_{22}$), q_i ($i \in L_{21} \cup K_{21}$), β_{pi} ($i \in M_{11} \cup M_{21}$), β_{jai} ($j = 1, 2, 3; i \in M_{11} \cup M_{21}$), что дает возможность вычислить производные q_r ($r \in M_{21} \cup M_{22} \cup L_{22} \cup K_{22}$) по всем указанным переменным в точке с координатами, равными плановым значениям исходных данных [3].

Учитывая это обстоятельство, разложим в ряд Тейлора в этой точке функции (19) — (21), ограничившись линейными элементами и пренебрегая остатком. Тогда, например, можем записать

$$\begin{aligned} \Delta P_{rk} = & \left(\frac{\partial f_r}{\partial P_1} \right)_{P_1=P_1^0} \Delta P_1 + \sum_{i \in L_{22}} \left(\frac{\partial f_r}{\partial P_{rk}} \right)_{P_{rk}=P_{rk}^0} \Delta P_{rk} + \sum_{i \in K_{22}} \left(\frac{\partial f_r}{\partial P_{rn}} \right)_{P_{rn}=P_{rn}^0} \Delta P_{rn} + \\ & + \sum_{i \in L_{21} \cup K_{21}} \left(\frac{\partial f_r}{\partial q_i} \right)_{q_i=q_i^0} \Delta q_i + \sum_{i \in M_{11} \cup M_{21}} \left(\frac{\partial f_r}{\partial \beta_{pi}} \right)_{\beta_{pi}=\beta_{pi}^0} \Delta \beta_{pi} + \\ & + \sum_{i \in M_{12} \cup M_{22}} \sum_{j=1,2,3} \left(\frac{\partial f_r}{\partial \beta_{jai}} \right)_{\beta_{jai}=\beta_{jai}^0} \Delta \beta_{jai}, \quad r \in L_{21}, \end{aligned} \quad (22)$$

где f_r обозначена левая часть равенства (19).

Таким образом, ΔP_{rk} можно рассматривать как линейную функцию случайных величин. Считая для простоты записи (что

не уменьшает общности) все величины независимыми, можем записать соотношение для $\sigma_{P_{rk}}^2$, $r \in L_{21}$:

$$\begin{aligned} \sigma_{P_{rk}}^2 = & \left(\frac{\partial f_r}{\partial P_1} \right)_{P_1=P_0}^2 \sigma_{P_1}^2 + \sum_{i \in L_{22}} \left(\frac{\partial f_r}{\partial P_{rk}} \right)_{P_{rk}=P_{rk}^0}^2 \sigma_{P_{rk}}^2 + \\ & + \sum_{i \in K_{21}} \left(\frac{\partial f_r}{\partial P_{rk}} \right)_{P_{rk}=P_{rk}^0}^2 \sigma_{P_{rk}}^2 + \sum_{i \in L_{21} \cup K_{21}} \left(\frac{\partial f_r}{\partial q_i} \right)_{q_i=q_i^0}^2 \sigma_{q_i}^2 + \\ & + \sum_{i \in M_{11} \cup M_{22}} \sum_{j=1, 2, 3} \left(\frac{\partial f_r}{\partial p_{jai}} \right)_{p_{jai}=p_{jai}^0}^2 \sigma_{p_{jai}}^2, \quad r \in L_{21}. \end{aligned} \quad (23)$$

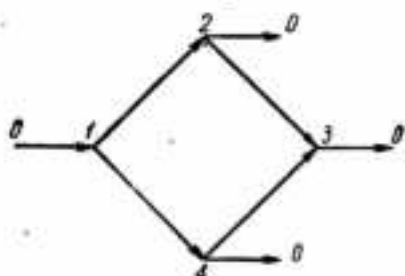
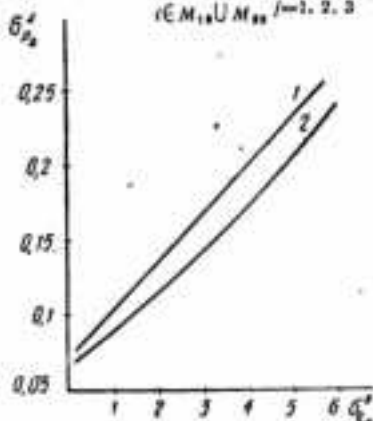


Рис. 1. Зависимость дисперсий давлений от дисперсий расходов для линейризованной (а) и нелинейной (б) модели: $\sigma_{q_2}^2 = 0,3$; $\sigma_{q_1}^2 = 0,3$ соответственно

Рис. 2. Граф сети

Аналогично получаются формулы для $\sigma_{P_{rk}}^2$ ($r \in K_{21}$), $\sigma_{q_i}^2$ ($i \in L_{22} \cup K_{22}$).

Пример. На рис. 1 приведены результаты исследований величины смещения между оценками дисперсий давлений на выходах сети в зависимости от величины дисперсий расходов, полученные по линейризованной модели (в соответствии с изложенной методикой) и нелинейной модели с помощью метода Монте-Карло для сети, граф которой представлен на рис. 2. Как видно из графиков зависимостей, точность расчета с помощью линейризованной модели удовлетворительна.

Приведенный алгоритм, несмотря на кажущуюся громоздкость, достаточно прост в реализации. Использование программ, реализующих данный алгоритм, необходимо при решении различных задач, связанных с определением реакции сети на изменение граничных условий ее функционирования или изменение параметров ее элементов.

Список литературы: 1. Тевяшев А. Д., Федоров Н. В. Гидравлический расчет одного класса инженерных сетей с активными элементами. — АСУ и приборы автоматики, 1980, вып. 55, с. 11—16. 2. Тевяшев А. Д., Федоров Н. В. Эффективный алгоритм гидравлического расчета инженерных сетей. — АСУ и приборы автоматики, 1980, вып. 56, с. 4—10. 3. Фихтенгольц Г. М. Курс дифференциального и интегрального исчисления. — М.: Наука, 1969. — Т.1. 600 с.

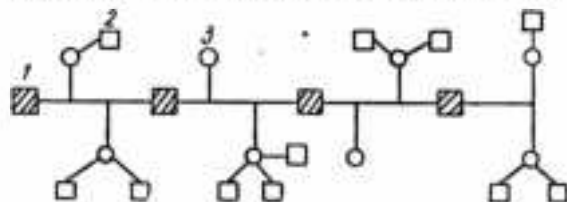
Поступила в редакцию 10.05.83.

УДК 681.3

М. Я. ДАНИЛКИС, Ю. М. ЛУБЕНСКИЙ

ОПТИМАЛЬНОЕ РАСПРЕДЕЛЕНИЕ РЕСУРСОВ В ГРУППОВОЙ СИСТЕМЕ ВОДОСНАБЖЕНИЯ

Одной из основных задач автоматизированных систем управления (АСУ) групповыми системами водоснабжения (ГСВ) является обеспечение всех потребителей водой с минимальными эксплуатационными затратами. При этом в АСУ должна решаться в реальном масштабе времени задача планирования оптимальных режимов работы насосных станций для обеспечения прогнозируемых потребностей в воде. В результате решения этой задачи находят оптимальное распределение нагрузок меж-



Типовая структура ГСВ:

1 — насосная станция ГСВ; 2 — местные насосные станции; 3 — потребители системы

ду насосными станциями системы, что позволяет выявить резервы экономии электроэнергии и воды.

Структура ГСВ приведена на рисунке. Имеется M работающих последовательно насосных станций ГСВ. Каждая μ -я насосная станция ГСВ питает N_μ потребителей. Каждый j -й потребитель μ -й станции ГСВ питается дополнительно от $R_{\mu j}$ местных насосных станций. Известны также функции эксплуатационных затрат $g(x)$ на подачу x количества воды каждому потребителю от питающих его насосных станций. Необходимо определить значения $x_{\mu j}$ и $x_{\mu j}$ подачи воды j -му потребителю μ -й насосной станции соответственно от i -й местной насосной станции и μ -й станции ГСВ, доставляющие минимум сумме эксплуатационных затрат системы.

Сформулированную выше задачу можно записать так

$$\sum_{p=1}^M \sum_{j=1}^{N_p} \left[\sum_{i=1}^{R_{pj}} g_{pj,i}(x_{pj,i}) + g_{pj}(x_{pj}) \right] \rightarrow \min; \quad (1)$$

$$\sum_{i=1}^{R_{pj}} x_{pj,i} + x_{pj} = G_{pj}; \quad (2)$$

$$\sum_{i=1}^{N_p} x_{pj} + x_{p+1} = x_p; \quad (3)$$

$$x_p \leq A_p; \quad (4)$$

$$x_{pj} \leq B_{pj}; \quad (5)$$

$$x_{pj,i} \leq C_{pj,i}; \quad (6)$$

где $p = 1 \dots M$, $j = 1 \dots N_p$, $i = 1 \dots R_{pj}$.

Ограничение (2) определяет необходимость подачи воды каждому потребителю в требуемом (прогнозируемом) количестве G_{pj} . Ограничение (3) выражает уравнения баланса в системе. Выражения (4) — (6) отражают ограничения по производительности насосных станций и пропускной способности системы.

При аппроксимации функций $g(x)$ линейными зависимостями поставленная задача становится задачей линейного программирования транспортного типа, для решения которой разработаны эффективные алгоритмы.

Как известно, решением задачи линейного программирования является одна из вершин выпуклого многогранника, определяемого условиями (2) — (6) [1]. Однако в случае предельно больших или предельно малых расходов (как этого требует решение задачи с линейными моделями) линейность затрат может существенно нарушаться. Кроме того, важнейшим экономическим показателем является рациональное использование природных ресурсов. Работа станций на максимальных нагрузках зачастую приводит к досрочной выработке и истощению местных ресурсов.

Задачи распределения, к которым относится и данная, с нелинейной целевой функцией при наличии ограничений поддаются численному решению с использованием методов динамического программирования [2]. Эти методы хорошо справляются с проблемой поиска глобального экстремума, с табличной формой задания целевых функций. Эта форма наиболее удобна и точна для задания и корректировки функций при построении их в процессе функционирования АСУ. Однако метод динамического программирования при значительном увеличении количества распределяемых ресурсов становится практически неосуществим. В рассматриваемой задаче значения G_{pj} выступают в качестве распределяемых ресурсов. Количество их даже для простейшей структуры ГСВ может быть значительным.

Покажем, что блочная структура ограничений (2) позволяет для систем данного типа модифицировать схему решения задачи распределения с использованием принципов динамического программирования. Следуя общим методам динамического программирования, введем в рассмотрение набор функций

$$f_n(x_n) = \min_{x_{n/j}, x_{n/l}, x_n \in D} \sum_{j=1}^M \sum_{l=1}^{N_j} \left[\sum_{i=1}^{R_{njl}} g_{njl}(x_{njl}) + g_{nl}(x_{nl}) \right], \quad (7)$$

где D — допустимая область изменения переменных, определяемая условиями (2) — (6); $\mu = n \dots M$ и $x_n \in D$.

Тогда из (7) в силу структуры ограничений (2) — (6) можно получить рекуррентное соотношение:

$$\begin{aligned} f_n(x_n) &= \min_D \left\{ \sum_{j=1}^{N_n} \left[\sum_{l=1}^{R_{njl}} g_{njl}(x_{njl}) + g_{nl}(x_{nl}) \right] + \right. \\ &\quad \left. + \sum_{\mu=n+1}^M \sum_{l=1}^{N_\mu} \left[\sum_{i=1}^{R_{\mu lj}} g_{\mu lj}(x_{\mu lj}) + g_{\mu l}(x_{\mu l}) \right] \right\} = \\ &= \min_{\substack{x_{n/j}, x_{n/l}, x_{n+1} \\ \sum x_{njl} + x_{n+1} = x_n \\ \sum x_{njl} + x_{n+1} = G_{n/l}}} \left\{ \sum_{j=1}^{N_n} \left[\sum_{l=1}^{R_{njl}} g_{njl}(x_{njl}) + g_{nl}(x_{nl}) \right] + f_{n+1}(x_{n+1}) \right\} = \\ &= \min_{\substack{x_{n/j}, x_{n+1} \\ \sum x_{njl} + x_{n+1} = x_n}} \min_{\substack{x_{n/l} \\ \sum x_{njl} = G_{n/l} - x_{n+1}}} \left\{ \sum_{j=1}^{N_n} \left[\sum_{l=1}^{R_{njl}} g_{njl}(x_{njl}) + \right. \right. \\ &\quad \left. \left. + g_{nl}(x_{nl}) \right] + f_{n+1}(x_{n+1}) \right\}. \quad (8) \end{aligned}$$

Внутренний минимум в выражении (8) берется по переменным, удовлетворяющим равенству

$$\sum_{l=1}^{R_{njl}} x_{njl} = G_{n/l} - x_{n+1}. \quad (9)$$

Количество равенств (9) равно количеству потребителей на n -м подъеме. Функции $f_{n+1}(x_{n+1})$ и $g_{nl}(x_{nl})$ не зависят от переменных x_{njl} . Матрица коэффициентов при неизвестных x_{njl} в (9) имеет блочную структуру, где каждый блок есть вектор-строка из R_{njl} единиц. Это обстоятельство позволяет знак $\sum_{l=1}^{R_{njl}}$ и знак $\min$ в выражении (8) поменять местами. Таким образом, выражение (8) примет вид

$$f_n(x_n) = \min_{x_{n/j}, x_{n+1}} \left\{ \sum_{j=1}^{N_n} \left[\min_{x_{n/l}} \sum_{l=1}^{R_{njl}} g_{njl}(x_{njl}) + g_{nl}(x_{nl}) \right] + f_{n+1}(x_{n+1}) \right\}. \quad (10)$$

Введем в рассмотрение набор функций двух переменных

$$f_{\mu j}(G_{\mu j}, x_{\mu j}) = \min_{x_{\mu j}} \sum_{i=1}^{R_{\mu j}} g_{\mu ji}(x_{\mu ji}) + g_{\mu j}(x_{\mu j}), \quad \sum_i x_{\mu ji} = G_{\mu j} - x_{\mu j}. \quad (11)$$

Определение первого слагаемого этой суммы есть задача оптимального распределения одного ресурса $G_{\mu j} - x_{\mu j}$ между местными источниками j -го потребителя. Для решения этой задачи воспользуемся стандартной процедурой решения задачи одномерного распределения методом динамического программирования [2]. Для этого введем функцию

$$S_{\mu j}^k(G) = \min_{x_{\mu j}, \sum_i x_{\mu ji} = G} \sum_{i=1}^{R_{\mu j}} g_{\mu ji}(x_{\mu ji}). \quad (12)$$

Рекуррентное соотношение (12) позволяет определить значение функции $S_{\mu j}(G)$:

$$S_{\mu j}^k(G) = \min_{x_{\mu jk}} [g_{\mu jk}(x_{\mu jk}) + S_{\mu j}^{k-1}(G - x_{\mu jk})], \quad (13)$$

где k принимает значения от 1 до $R_{\mu j}$ и $S_{\mu j}^0(G) = 0$.

Определив функции $S_{\mu j}(G)$ и заменив G на $G_{\mu j} - x_{\mu j}$, получаем набор функций двух переменных:

$$f_{\mu j}(G_{\mu j}, x_{\mu j}) = S_{\mu j}(G_{\mu j} - x_{\mu j}) + g_{\mu j}(x_{\mu j}). \quad (14)$$

Выражение (8) теперь примет вид

$$f_n(x_n) = \min_{\substack{x_{\mu j}, x_{n+1} \\ \sum_i x_{\mu ji} + x_{n+1} = x_n}} \left[\sum_{i=1}^{N_n} f_{\mu j}(G_{\mu j}, x_{\mu j}) + f_{n+1}(x_{n+1}) \right]. \quad (15)$$

Для заданного набора значений $G_{\mu j}$ формула (15) представляет собой задачу распределения ограниченного ресурса x_n . Как и функцию $S_{\mu j}(G)$, это распределение можно получить, используя стандартную процедуру динамического программирования (13).

Выражение (15) позволяет рекуррентно осуществлять поиск оптимального решения исходной задачи.

Результаты проведенного анализа задачи позволяют предложить алгоритм, сводящий многомерную задачу распределения к последовательности одномерных задач оптимизации. При этом общий алгоритм решения задачи можно разделить на два основных этапа.

Первый этап сводится к определению набора функций $f_{\mu j}(G_{\mu j}, x_{\mu j})$ в соответствии с выражением (11). Эти функции определяются для всего допустимого диапазона изменения величин $G_{\mu j}$ и $x_{\mu j}$ для каждого потребителя системы. Этот этап можно осуществлять один раз (например, на стадии проектирования АСУ ГСВ). Полученные функции для всех потребителей системы служат исходной информацией для второго этапа решения задачи, который выпол-

няется в процессе функционирования АСУ в реальном масштабе времени.

Второй этап решения задачи проводится при получении численных значений о требуемой подаче воды потребителями системы. Эти значения подставляются в функции $f_{rj}(G_{rj}, x_{rj})$, превращая их в функции одной переменной x_{rj} . Методом динамического программирования по формуле (15) осуществляется решение исходной задачи на заданном наборе значений G_{rj} .

Такое функциональное подразделение этапов решения задачи позволяет наиболее трудоемкую часть решения вынести на стадию проектирования, что имеет большое значение при необходимости оперативного решения задачи в автоматизированной системе управления, работающей в режиме реального времени.

Необходимость второго этапа решения задачи вызвана наличием ограничений (4). В системах, где ограничение (4) не является существенным, т.е. оптимальное решение лежит строго внутри области $x_r < A_r$, общая задача распределения автоматически распадается на самостоятельные задачи каждого потребителя системы. В этом случае после первого этапа решения отыскиваются x_{rj}^0 , доставляющие минимум функции $f_{rj}(G_{rj}, x_{rj})$ и являющиеся решением исходной задачи. Найденные значения x_{rj} позволяют определить оптимальные значения всех остальных переменных системы.

Список литературы: 1. Гасс С., Линейное программирование. — М.: Физматгиз, 1961. — 303 с. 2. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. — М.: Наука, 1965. — 460 с.

Поступила в редколлегию 11.01.82.

УДК 621.372

И. В. НЕДИН, Б. Н. СЕРДЮК

ИНФОРМАЦИОННО-АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИТУАЦИОННОГО МОДЕЛИРОВАНИЯ ЭКСПЛУАТАЦИОННЫХ РЕЖИМОВ РАБОТЫ ЭЛЕКТРОСЕТЕЙ

Постановка задачи. При решении задач управления эксплуатационными режимами работы распределительных электросетей требуется оценивать фактический уровень их технико-экономических показателей. Обязательным условием выполнения такой оценки является учет эксплуатационной динамики схемы электросети, реализуемый посредством моделирования процесса функционирования сети в течение расчетного периода T_p [1]. Моделирование предусматривает последовательный анализ образующихся в течение T_p i -х ситуаций, отличающихся

продолжительностью t_i , конфигурацией и параметрами схемы сети. При ситуационном моделировании расширяется состав используемой информации, возрастает объем вычислений и, следовательно, повышаются требования к организации размещения и хранения данных, к алгоритмам их переработки. Поэтому в статье рассмотрены вопросы организации информационно-программного обеспечения при решении задач оценки фактических технико-экономических результатов (ТЭР) оптимизации эксплуатационных режимов и схем городских электросетей по критериям экономичности и надежности электроснабжения потребителей.

Математическое содержание задачи. Состав используемой информации и организация ее переработки определяются показателями P , являющимися критериями оценки эффективности оптимизации (потери электроэнергии $\Delta\mathcal{E}$ и суммарный ожидаемый недоотпуск электроэнергии ΔW_n), и методикой моделирования. Решение задачи базируется на применении ситуационной модели (СМ), которая характеризует поведение объекта исследования в течение T_p с точки зрения его структурно-топологических свойств, влияющих на $\Delta\mathcal{E}$ и ΔW_n . СМ представляет последовательность состояний схемы сети продолжительностью t_i в течение T_p [1], каждое из которых определяется сочетанием отключенных в i -м интервале T_p коммутационных аппаратов (КА) или участков (У) схемы (рис. 1, а). Результирующие фактические и расчетные значения $\Delta\mathcal{E}$ и ΔW_n за период T_p определяются по формулам

$$\Delta\mathcal{E} = \sum_{i=1}^N \Delta\mathcal{E}_i^f; \Delta W_n^f = \sum_{i=1}^N \Delta W_{ni}^f; \Delta W_n^p = \frac{1}{T_p} \sum_{i=1}^N \Delta W_{ni}^p t_i,$$

где $\Delta\mathcal{E}_i^f$, ΔW_{ni}^f , ΔW_{ni}^p — ситуационные фактические и расчетные значения ТЭР. Аналогично оцениваются фактические значения показателей надежности схемы сети относительно ее узлов

$$\omega_{zs} = \frac{1}{T_p} \sum_{i=1}^N \omega_{zi} t_i; \tau_{zs} = \frac{1}{T_p} \sum_{i=1}^N \tau_{zi} t_i.$$

Здесь ω_{zi} , τ_{zi} — ситуационные значения параметра потока отказов и времени восстановления электроснабжения схемы относительно s -го узла нагрузки (УН) схемы. В свою очередь ситуационные значения перечисленных показателей являются функцией множества параметров сети и ее элементов:

$$\Delta W_{ni}^f = f_1(P_s, \omega_{si}^f, \tau_{si}^f); \Delta W_{ni}^p = f_2(K_i, P_s, \omega_i, \tau_i, \nu_i, \theta_i, PO_s); \\ \Delta\mathcal{E}_i = f_3(K_i, P_s, r_i, P_i^{\text{ном}}),$$

где K_i — конфигурация схемы сети; ω_{si}^f , τ_{si}^f — фактические частота и длительность отключений s -х УН; P_s — нагрузка в s -м УН; ω_i , τ_i , ν_i , θ_i — соответственно параметр потока отказов, среднее

время послеаварийного ремонта, параметр потока преднамеренных отключений и среднее время обслуживания, задаваемые с учетом сезонной неравномерности; PO_s — суточное распределение отключений УН, задаваемое с учетом сезонного характера; r_j и $P_j^{дон}$ — активные сопротивления и пропускные способности j -х ветвей схемы.

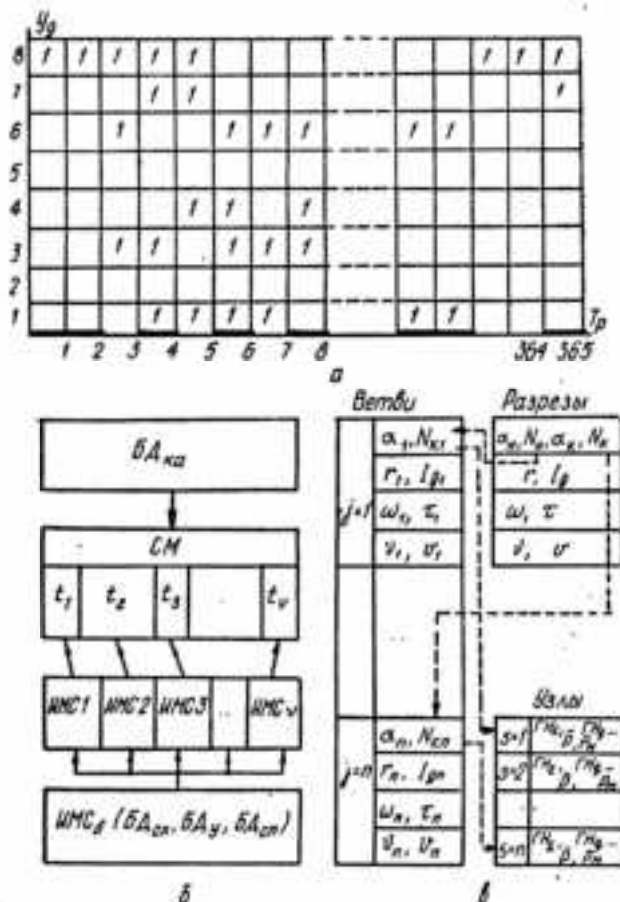


Рис. 1. Организация размещения и преобразования основных видов информации

Математические модели для оценки перечисленных показателей приводятся в [1—3].

Информационное обеспечение и алгоритмическая реализация моделирования. Реализация ситуационного моделирования при решении оценочных и оптимизационных задач связана с необходимостью анализа значительного количества ситуаций в системе, включающей сотни и тысячи элементов. Указанное

обстоятельство обуславливает жесткие требования к быстродействию соответствующих алгоритмов и, следовательно, к организации информации, используемой при расчете, который должен выполняться для периода любой заданной продолжительности T_p при различной локализации района сети, для которого производится расчет. Этим обусловлено разделение используемой информации по ряду условий и признаков, ускоряющих ее поиск и переработку.

По своему назначению информация распределена в базах данных $БД_{сх}$, $БД_{уз}$ и $БД_{сн}$. Содержимое каждой базы разделено по классам рабочего напряжения и пунктам (подстанциям), являющимся источниками питания (ИП) для сети соответствующего класса напряжения. [4]. Содержимое $БД_{сх}$ включает данные о времени и продолжительности отключений в течение T_p g -х участков ($У_g$) схемы и является источником информации для построения СМ, как показано на рис. 1, б. $БД_{сх}$, $БД_{уз}$, $БД_{сн}$ представляют содержимое базовой информационной модели сети (ИМС_б), которая преобразуется, как показано на рис. 1, б, в ситуационные ИМС_с, служащие информационной основой для расчета ситуационных значений $П$. В $БД_{сх}$ хранятся данные о конфигурации схемы и ее параметрах — r_j , $P_j^{ном}$, ω_j , τ_j , ν_j , θ_j . Показатели надежности заданы своими среднегодовыми значениями. В $БД_{сх}$ указано также состояние КА участков сети, соответствующее ее нормальной схеме.

В $БД_{уз}$ сосредоточена информация о нагрузке в s -х УН, включающая сведения о составе потребителей и их графиках нагрузки (ГН), а содержимое $БД_{сн}$ представляет данные о типовых суточных ГН с дифференциацией по сезонам, а также зависимости вероятностей отключений УН от времени суток (распределения отключений — PO_s) и характеристики сезонного изменения ω_j , τ_j , ν_j , θ_j . В $БД_{сн}$ могут находиться сведения о достоверности и точности PO , ω , τ , ν , θ , являющихся результатом обработки ретроспективных данных для конкретной электросети. ГН могут быть получены как посредством обработки измерений, так и из литературных источников или служебных материалов. Связь $БД_{уз}$ и $БД_{сн}$ обеспечивается перечислением в информации об s -х узлах типов μ_s потребителей, которым соответствуют сосредоточенные в $БД_{сн}$ типовые ГН.

Ситуационная модель представляется двоичной (при реализации на ЭВМ 2-го поколения) или битовой матрицей (рис. 1, а). Строки матрицы представляют T_p , разделенный на k равных f -х интервалов. Значение интервала T_p принимается в зависимости от решаемой задачи. Для оценки фактической эффективности оптимизации, выполняемой с заблаговременностью в 1 год, продолжительность T_p равна 365 суткам, а $f_j = 1$ суткам. Число столбцов матрицы равно k . Каждый ряд или бит столбца матрицы соответствует порядковому но-

меру одного из m g -х участков U_g отключавшихся (или U_g , отключение которых планируется) в течение T_p . Порядок матрицы равен $k \times m$.

При построении СМ упорядочивается нумерация U_g , отключавшихся в течение T_p . Затем поочередно для каждого g -го U просматриваются все j -е столбцы матрицы и записываются 1 в g -е разряды тех столбцов, номера j которых соответствуют упорядоченным номерам U_g . Каждая i -я ситуация может включать более одного столбца матрицы. Поэтому при определении границ ситуаций, образованных при одновременном отключении множества U в нахождении t_i , производится попарное сравнение столбцов, рассматриваемых слева направо. По количеству совпадений определяется t_i и затем расшифровываются наименования участков, отключенных в i -й ситуации. Затем производится преобразование содержимого ИМС₀ в ИМС _{i} с учетом i -го сочетания отключенных участков ΣU_{ig} .

В ИМС _{i} хранится расчетная схема сети, подготовленная для расчета и оптимизации. Расчетная схема состоит из дерева сети, ориентированного относительно выбранных ИП и УН при помощи α_j -вторых адресных отображений (ВАО), и массива точек деления (или разрезов), соответствующих отключенным U_g в данной ситуации (рис. 1, в). В информации о расчетной схеме объединены массивы α_j наименований конечных пунктов j -х участков N_{kj} , служащих для связи с массивами данных об УН, в которых сосредоточены сведения о нагрузке, а также массивы r_j , $J_{допj}$, ω_j , τ_j , ν_j , θ_j . Численные значения показателей надежности в этих массивах скорректированы с учетом сезонного интервала, в пределы которого входит i -я ситуация. Массивы данных о разрезах используются при оптимизации, после выполнения которой их содержимое изменяется вследствие изменения конфигурации схемы. Поэтому в этих массивах содержатся ВАО (α_n , α_k), связывающие разрез с участками дерева, а также данные, характеризующие участок U_g , соответствующий разрезу. В число этих данных входят характеристики КА, наличие противоаварийной автоматики или АВР, признаки, указывающие на допустимость перемещения разреза.

В массивах информации об узлах сосредоточены типы μ_g суточных ГН g -х потребителей, сконцентрированных в УН _{g} , суммарные ГН _{g} , используемые при учете суточной неравномерности отключений УН для расчета ΔW_n , а также некоторые укрупненные характеристики нагрузки — P_n , $P_{\max n}$. Отбор ГН производится с учетом сезонности i -й ситуации из БД_{см} по μ_g , являющимся адресными отображениями графиков.

Такая организация содержимого расчетной ИМС обеспечивает минимальные затраты времени при выполнении расчетов и оптимизации с использованием наиболее эффективных из применяемых в настоящее время для этой цели алгоритмов.

Алгоритм реализации ситуационного моделирования, ориентированный на использование информационной системы, приведен на рис. 2.

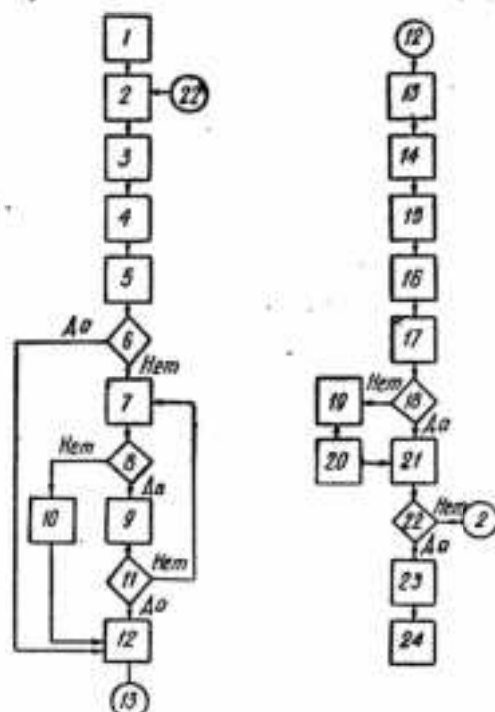


Рис. 2. Алгоритмы реализации ситуационного моделирования при оценке фактических ТЭР: 1 — задание начальных значений $t_i = 0$, $i = 0$; 2 — изменение текущего индекса интервала T_p ; 3 — отбор i -го столбца V_{gi} ; 4 — $i = i + 1$ (изменение индекса последующего интервала T_p); 5 — $i = i + 1$; 6 — выполняется ли условие $i > n$?; 7 — отбор i -го столбца V_{gi} ; 8 — выполняется ли равенство $V_{gi} = Y_{gi}/t_i$; 9 — $t_i = t_i + 1$ (увеличение длительности i -й ситуации); 10 — $i = i$; 11 — $i = i + 1$; 12 — запоминание t_i ; 13 — расшифровка наименований отключенных участков по V_{gi} ; 14 — формирование расчетной схемы для i -й ситуации с определением параметров ветвей, скорректированных с учетом сезонного интервала T_p , включающего i -ю ситуацию; 15 — формирование маски YH_p ; 16 — подготовка списков конечных пунктов участков и списков узлов; 17 — расчет ТЭР для i -й ситуации; 18 — оптимизировать ли схему, соответствующую i -й ситуации?; 19 — оптимизация по заданному критерию; 20 — расчет послеоптимизационных результатов; 21 — запоминание до- и послеоптимизационных результатов оценки ситуационных ТЭР; 22 — последний ли интервал T_p рассмотрен?; 23 — расчет результирующих ТЭР с учетом и без учета ситуационной оптимизации; 24 — сравнение фактических результатов расчета с ожидаемыми в начале T_p и принятие решения.

Реализация информационно-алгоритмического обеспечения и результаты применения. Описанная система представления информации и алгоритм реализованы в задаче оценки фактической эффективности оптимизации эксплуатационных режимов электросетей на ЭВМ 2-го поколения на базе автоматизированного банка данных, описанного в [4]. В таком варианте система определения ТЭР оптимизации (СТЭР) обеспечивает координацию работы функциональных программных модулей и адаптационных программ в соответствии с последовательностью моделируемых ситуаций. В настоящее время система реализована на ЭВМ 3-го поколения. Программное обеспечение системы оценки ТЭР составлено на языке ПЛ-1. Продолжительность расчета фактических ТЭР для сети, схема которой содержит 1000 узлов, при однократном моделировании ситуаций СМ (для $V = 250$) составляет около 10 мин.

При помощи СТЭР в одной из энергосистем был проведен анализ фактических результатов оптимизации, выполнявшийся с заблаговременно-

стью, равной одному году. Полученные результаты показали, что учет эксплуатационной динамики схемы электросети при $T_p = 1$ год вызывает значительные отклонения (около 30—70 %) ТЭР от их ожидаемого в начале T_p уровня в худшую сторону. Выполнены исследования, подтверждающие целесообразность использования информационно-вычислительной системы в качестве советчика диспетчера при выборе временных схем (например, ремонтных).

В заключение следует отметить, что описанное информационно-алгоритмическое обеспечение может быть положено в основу универсальных моделирующих систем, предназначенных для перспективных оптимизационных расчетов электросетей в детерминированной и вероятностной форме при любой заблаговременности, возможной при их эксплуатации.

Список литературы: 1. Недин И. В. Применение ситуационного моделирования при исследовании и оценке эксплуатационной надежности сложных систем электроснабжения. — Метод. вопр. исслед. надежности больших систем энергетики, 1979, вып. 18, с. 77—84. 2. Зорин В. В., Тисленко В. В. Особенности расчета показателей надежности схем электросетей. — Изв. вузов СССР. Энергетика, 1973, № 6, с. 15—21. 3. Недин И. В. Методы и модели учета нестационарности при оценке надежности питания узлов системы электроснабжения. — Метод. вопр. исслед. надежности больших систем энергетики, 1976, вып. 12, с. 89—99. 4. Крушельницкий А. Э., Мусатов О. В., Недин И. В. Автоматизированный банк данных для эксплуатационных расчетов системы электроснабжения. — Механизация и автоматизация упр., 1979, № 2, с. 21—24

Поступила в редакцию 02.11.83.

УДК 628.12+681.513.5

В. М. ТАУБЕ, В. А. ВАЙСФЕЛЬД

УСЛОВИЯ СУЩЕСТВОВАНИЯ ДОПУСТИМОГО ДИСКРЕТНОГО УПРАВЛЕНИЯ НАСОСНОЙ СТАНЦИЕЙ РАБОТАЮЩЕЙ НА РЕЗЕРВУАРАХ

Описание и математическая модель системы. Одной из важнейших народнохозяйственных задач является дальнейшее совершенствование работы водопроводных сооружений и агрегатов путем достижения оптимального режима их работы. Большинство городских систем водоснабжения имеют в своем составе насосные станции, работающие на резервуары. В этих системах осуществляется дискретное управление системой в зависимости от уровня заполнения резервуаров.

Существование допустимых управлений в таких системах зависит от правильного выбора параметров системы. В связи с этим необходимо найти такие условия на параметры системы,

которые обеспечивают существование хотя бы одного допустимого управления. Такие условия окажутся полезными как при проектировании новых систем, так и при анализе и построении оптимального управления существующими системами.

Рассмотрим систему, состоящую из насосной станции и некоторого числа резервуаров, заполняемых этой станцией. Вода из резервуаров поступает к потребителям, причем резервуары между собой не соединены и на конце каждого выходного водовода насосной станции находится только один резервуар.

Такая система является динамической многосвязной системой. Для построения математической модели системы введем следующие обозначения: $Q(t)$ — подача насосной станции в момент t , m — количество резервуаров в системе, $V_i(t)$ — объем воды в i -м резервуаре, $1 \leq i \leq m$, $Q_i(t)$ — подача воды в i -й резервуар, $P_i(t)$ — потребление воды из i -го резервуара в момент t .

Функции $P_i(t)$ считаются известными на некотором отрезке времени $[0, T]$. Они могут быть заданы исходя из априорного анализа потребителей или быть получены в результате прогноза. Полагаем, что давления на входах в резервуары не меняются при изменении времени на отрезке $[0, T]$. Тогда всегда найдутся такие константы α_i , что

$$Q_i(t) = \alpha_i Q(t) \quad (1 \leq i \leq m), \quad 0 < \alpha_i < 1, \quad \sum_{i=1}^m \alpha_i = 1. \quad (1)$$

Математическая модель системы [2] записывается в виде

$$\frac{dV_i(t)}{dt} = \alpha_i Q(t) - P_i(t), \quad V_i(0) = V_i^0, \quad (1 \leq i \leq m), \quad (2)$$

где V_i^0 — объем воды в i -м резервуаре в начальный момент времени.

Введем векторные обозначения:

$$\vec{V}(t) = \{V_1(t), \dots, V_m(t)\}; \quad \vec{\alpha} = \{\alpha_1, \dots, \alpha_m\}; \\ \vec{P}(t) = \{P_1(t), \dots, P_m(t)\}; \quad \vec{V}^0 = \{V_1^0, \dots, V_m^0\}.$$

Уравнения (2) можно переписать в векторном виде:

$$\frac{d\vec{V}(t)}{dt} = \vec{\alpha} Q(t) - \vec{P}(t), \quad \vec{V}(0) = \vec{V}^0. \quad (3)$$

Уравнение (3) позволяет находить объем воды в резервуарах в любой момент времени t , зная объем воды в начальный момент времени, подачу насосной станции и потребление воды из резервуаров.

Постановка задачи. На насосной станции всегда имеется возможность управлять подачей $Q(t)$ в некоторых пределах $Q^{\min}$ и $Q^{\max}$:

$$Q^{\min} \leq Q(t) \leq Q^{\max}. \quad (4)$$

При этом характер управления может быть различным: непрерывным, когда работает постоянная комбинация насосов и допускается лишь регулирование числа оборотов двигателей насосных агрегатов или дросселирование: $Q(t)$ — непрерывная функция; дискретным, когда допускаются включения и отключения насосных агрегатов и отсутствует возможность непрерывного регулирования; $Q(t)$ — кусочно-постоянная функция; смешанным, когда допустимо как непрерывное, так и дискретное управление: $Q(t)$ — кусочно-непрерывная функция.

Управление подачей насосной станции производится с целью поддержания объемов воды в резервуарах в допустимых пределах $V_i^{\min}, V_i^{\max}$:

$$V_i^{\min} < V_i(t) < V_i^{\max}, \quad 0 < t < T. \quad (5)$$

Управление $Q(t)$ будем называть допустимым на отрезке времени $[0, T]$, если для объемов воды в резервуарах $V_i(t)$, полученных в результате решения системы уравнений (3), выполняются неравенства (5).

Получим необходимые и достаточные условия существования допустимых управлений в системе, описываемой уравнениями (3) при ограничениях (4) и (5). Для этого пределаем ряд преобразований.

Вводя обозначения

$$u(t) = \int_0^t Q(s) ds, \quad y_i(t) = \int_0^t P_i(s) ds \quad (6)$$

и интегрируя уравнения (3), получаем

$$V_i(t) = \alpha_i u(t) - y_i(t) + V_i^n, \quad V_i^{\min} < V_i(t) < V_i^{\max}. \quad (7)$$

Условие (4) запишем в виде

$$Q^{\min} < u'(t) < Q^{\max}. \quad (8)$$

Для удобства записи заменим $Q^{\min}$ и $Q^{\max}$ соответственно на $u^{\min}$ и $u^{\max}$.

Тогда (7) и (8) можно записать так:

$$V_i^{\min} < \alpha_i u(t) - y_i(t) + V_i^n < V_i^{\max}, \quad u^{\min} < u'(t) < u^{\max}. \quad (9)$$

Преобразуем полученные неравенства

$$\frac{y_i(t) + V_i^{\min} - V_i^n}{\alpha_i} < u(t) < \frac{y_i(t) + V_i^{\max} - V_i^n}{\alpha_i}, \quad u^{\min} < u'(t) < u^{\max}. \quad (10)$$

Определим функции

$$b(t) = \min_{1 \leq i \leq m} \frac{y_i(t) + V_i^{\max} - V_i^n}{\alpha_i}, \quad a(t) = \max_{1 \leq i \leq m} \frac{y_i(t) + V_i^{\min} - V_i^n}{\alpha_i}. \quad (11)$$

Очевидно, что выполнение неравенств (10) эквивалентно выполнению следующих неравенств:

$$a(t) < u(t) < b(t), \quad u^{\min} < u'(t) < u^{\max}. \quad (12)$$

Итак, задача существования допустимых управлений сводится к следующей математической задаче: каким ограничениям должны удовлетворять функции $a(t)$ и $b(t)$ и константы $u^{\min}$ и $u^{\max}$, чтобы неравенства (12) были совместны, где $a(t)$ и $b(t)$ — монотонно возрастающие функции, а $u^{\min}$ и $u^{\max}$ — положительные константы.

Монотонность функций $a(t)$ и $b(t)$ следует из монотонности функций $y_i(t)$. Функции $a(t)$ и $b(t)$ определяются параметрами системы и поэтому условия существования допустимых управлений можно давать как ограничения на эти функции.

Заметим, что из второго двойного неравенства в (12) следует

$$u^{\min}t < u(t) < u^{\max}t. \quad (13)$$

Сопоставив их с первым двойным неравенством (12), получим первую группу необходимых условий:

$$u^{\max}t - a(t) > 0, \quad u^{\min}t - b(t) < 0. \quad (14)$$

Кроме того, для любого числа h и момента времени t , таких, что $h > 0$, $0 < t < T$, $0 < t+h < T$, выполняются неравенства

$$u^{\min}h < u(t+h) - u(t) = \int_t^{t+h} u'(s) ds < u^{\max}h. \quad (15)$$

Причем

$$\begin{aligned} u(t+h) < b(t+h) \\ u(t) > a(t) \end{aligned} \Rightarrow u^{\min}h < b(t+h) - a(t). \quad (16)$$

И аналогично:

$$\begin{aligned} u(t+h) > a(t+h) \\ u(t) < b(t) \end{aligned} \Rightarrow u^{\max}h > a(t+h) - b(t). \quad (17)$$

Условия (16) и (17) составляют вторую группу необходимых условий.

Запишем неравенства (14), (16), (17) вместе:

$$\begin{aligned} 1^\circ u^{\max}t - a(t) > 0; \quad 2^\circ u^{\min}t - b(t) < 0; \\ 3^\circ b(t+h) - a(t) > u^{\min}h; \quad 4^\circ a(t+h) - b(t) < u^{\max}h. \end{aligned} \quad (18)$$

Оказывается, что условия (18) являются достаточными для существования допустимого кусочно-непрерывного управления.

Если еще выполнено условие

$$\min_{0 \leq t \leq T} [b(t) - a(t)] = \delta > 0, \quad (19)$$

то всегда существует дискретное управление, причем $Q(t)$ будет иметь конечное число разрывов.

Последнее утверждение будет доказано ниже.

Достаточность условий (18) и (19) для существования допустимых дискретных управлений. Будем считать, что условия (18) и (19) выполнены. Докажем, что в этом случае можно построить

допустимое управление $u(t)$, производная которого является кусочно-постоянной функцией с конечным числом точек разрыва и принимает два значения $u^{\min}$ и $u^{\max}$.

Пусть $u_m(t)$ — кусочно-постоянная функция, имеющая m разрывов и принимающая два значения $u^{\min}$ и $u^{\max}$. Если $\tau_1, \tau_2, \dots, \tau_m$ — точки ее разрыва и $u(0) = u^{\max}$, то $u_m(t) = \int_0^t u_m(t) dt$ представим в виде:

если m — нечетно, $m = 2l + 1$

$$u_m(t) \begin{cases} u^{\max} t, & 0 \leq t < \tau_1, \\ u^{\min} t + (u^{\max} - u^{\min}) \sum_{i=1}^{2k-1} (-1)^{i+1} \tau_i, & \tau_{2k-1} \leq t < \tau_{2k}, \\ u^{\max} t + (u^{\max} - u^{\min}) \sum_{i=1}^{2k} (-1)^{i+1} \tau_i, & \tau_{2k} \leq t < \tau_{2k+1}, \\ u^{\min} t + (u^{\max} - u^{\min}) \sum_{i=1}^m (-1)^{i+1} \tau_i, & \tau_m \leq t < T; \end{cases} \quad (20)$$

если m — четно, $m = 2l$

$$u_m(t) \begin{cases} u^{\max} t, & 0 \leq t < \tau_1, \\ u^{\min} t + (u^{\max} - u^{\min}) \sum_{i=1}^{2k-1} (-1)^{i+1} \tau_i, & \tau_{2k-1} \leq t < \tau_{2k}, \\ u^{\max} t + (u^{\max} - u^{\min}) \sum_{i=1}^{2k} (-1)^{i+1} \tau_i, & \tau_{2k} \leq t < \tau_{2k+1}, \\ u^{\max} t + (u^{\max} - u^{\min}) \sum_{i=1}^m (-1)^{i+1} \tau_i, & \tau_m \leq t < T. \end{cases} \quad (21)$$

Допустимое управление $u_m(t)$ построим индуктивно. Если выполнено неравенство $u_0(t) = u^{\max} t \leq b(t)$, то в силу 1° (18) управление $u_0(t)$ уже будет допустимым.

Поэтому будем предполагать, что существует наименьший момент времени θ_1 , после которого неравенство $u_0(t) \leq b(t)$ нарушается. Тогда должно выполняться

$$u^{\max} \theta_1 = b(\theta_1). \quad (22)$$

Найдем момент перехода τ_1 с $u^{\max}$ на $u^{\min}$, такой чтобы на наибольшем отрезке выполнялись неравенства

$$a(t) \leq u^{\max} t \leq b(t) \quad 0 \leq t \leq \tau_1;$$

$$a(t) \leq u^{\min} t + (u^{\max} - u^{\min}) \tau_1 \leq b(t) \quad \tau_1 \leq t \leq \tau. \quad (23)$$

Из первого неравенства (23) следует, что

$$0 \leq \tau_1 \leq \theta_1. \quad (24)$$

Преобразуя второе неравенство (23), получаем

$$\frac{a(t) - u^{\min}_t}{u^{\max} - u^{\min}_t} < \tau_1 < \frac{b(t) - u^{\min}_t}{u^{\max} - u^{\min}_t}. \quad (25)$$

Если $0 < t < \theta_1$, то

$$a(t) < u^{\max}_t < b(t). \quad (26)$$

Отсюда следует, что при $\tau_1 < t < \theta_1$

$$\frac{b(t) - u^{\min}_t}{u^{\max} - u^{\min}_t} > \frac{(u^{\max} - u^{\min})t}{u^{\max} - u^{\min}_t} = t > \tau_1 \quad (27)$$

и

$$\frac{a(t) - u^{\min}_t}{u^{\max} - u^{\min}_t} < t < \theta_1 \Rightarrow \max_{0 < t < \theta_1} \frac{a(t) - u^{\min}_t}{u^{\max} - u^{\min}_t} < \theta_1. \quad (28)$$

Из неравенств (27) и (28) следует, что при

$$\max_{0 < t < \theta_1} \frac{a(t) - u^{\min}_t}{u^{\max} - u^{\min}_t} < \tau_1 < \theta_1 \quad (29)$$

неравенство (25) выполняется на отрезке (τ_1, θ_1) .

Положим

$$\tau_1 = \min_{\theta_1 < t < \tau} \frac{b(t) - u^{\min}_t}{u^{\max} - u^{\min}_t}. \quad (30)$$

Докажем, что момент τ_1 , определенный в (30), удовлетворяет неравенствам (29).

Поскольку

$$\frac{b(\theta_1) - u^{\min}_{\theta_1}}{u^{\max} - u^{\min}_t} = \frac{(u^{\max} - u^{\min})\theta_1}{u^{\max} - u^{\min}_t} = \theta_1,$$

то $\tau_1 < \theta_1$.

Пусть $\tilde{t}_1$ — любой момент времени, в который достигается максимум в (29), а $\hat{t}_1$ — наибольший момент, в который достигается минимум в (30).

Левое неравенство в (29) означает, что

$$\frac{a(\tilde{t}_1) - u^{\min}_{\tilde{t}_1}}{u^{\max} - u^{\min}_t} < \frac{b(\hat{t}_1) - u^{\min}_{\hat{t}_1}}{u^{\max} - u^{\min}_t}. \quad (31)$$

Формула (31) эквивалентна неравенству

$$b(\hat{t}_1) - a(\tilde{t}_1) > u^{\min}(\hat{t}_1 - \tilde{t}_1), \quad (32)$$

которое выполняется в силу 3° (18).

Итак, для момента τ_1 , определенного по формуле (30), выполняются неравенства (29), а значит, и (25) на отрезке $[\tau_1, \theta_1]$.

Докажем, что (25) выполняется и при $\theta_1 < t < \hat{t}_1$. В самом деле, правое неравенство (25) выполняется в силу определения τ_1 , вообще при $\tau_1 < t < T$, а левое

$$\tau_1 = \frac{b(\hat{t}_1) - u^{\min} \hat{t}_1}{u^{\max} - u^{\min}} > \frac{a(t) - u^{\min} t}{u^{\max} - u^{\min}} \text{ при } t < \hat{t}_1 \quad (33)$$

выполняется в силу условия 3° (18).

Если левое неравенство (25) выполняется на отрезке (τ_1, T) , то допустимое управление $u_1(t)$ уже построено. В противном случае существует наименьший момент $\theta_2 > \hat{t}_1$, после которого левое неравенство (25) нарушается. Тогда, очевидно, должно выполняться

$$u^{\min} \theta_2 + (u^{\max} - u^{\min}) \tau_1 = a(\theta_2). \quad (34)$$

Проведем индуктивный переход с нечетного номера на четный (с четного на нечетный проводится аналогично).

Пусть уже построено управление $u_{2k-1}(t)$ и $\tau_1, \tau_2, \dots, \tau_{2k-1}$ точки разрыва ее производной. Тогда должно выполняться

$$a(t) < u_{2k-1}(t) < b(t) \text{ при } 0 < t < \theta_{2k}, \quad (35)$$

где θ_{2k} — наименьший момент нарушения левого неравенства в (35). Причем в силу (20)

$$u^{\min} \theta_{2k} + (u^{\max} - u^{\min}) \sum_{i=1}^{2k-1} (-1)^{i+1} \tau_i = a(\theta_{2k}). \quad (36)$$

Определим

$$\tau_{2k} = \min_{\theta_{2k} < t < T} \left\{ \sum_{i=1}^{2k-1} (-1)^{i+1} \tau_i + \frac{u^{\max} t - a(t)}{u^{\max} - u^{\min}} \right\}. \quad (37)$$

Докажем, что

$$\tau_{2k-1} < \tau_{2k} < \theta_{2k}. \quad (38)$$

В самом деле

$$\begin{aligned} \tau_{2k} &< \sum_{i=1}^{2k-1} \tau_i + \frac{u^{\max} \theta_{2k} - a(\theta_{2k})}{u^{\max} - u^{\min}} = \theta_{2k}; \\ \tau_{2k} &> \sum_{i=1}^{2k-1} (-1)^{i+1} \tau_i + \min_{\theta_{2k-2} < t < T} \frac{u^{\max} t - a(t)}{u^{\max} - u^{\min}} = \\ &= \tau_{2k-1} + \sum_{i=1}^{2k-2} (-1)^{i+1} \tau_i > \tau_{2k-1}. \end{aligned} \quad (39)$$

Построим управление $u_{2k}(t)$ по моментам $\tau_1, \tau_2, \dots, \tau_{2k-1}, \tau_{2k}$.

Докажем, что на отрезке $[0, \hat{t}_{2k}]$ управление $u_{2k}(t)$ допустимо ($\hat{t}_{2k}$ — наибольший момент, при котором достигается минимум в (37)).

Для этого достаточно доказать неравенство

$$a(t) < u^{\max}t + (u^{\max} - u^{\min}) \sum_{i=1}^{2k} (-1)^{i+1} \tau_i < b(t) \quad (40)$$

при $\tau_{2k} < t < \bar{t}_{2k}$.

Неравенство (40) эквивалентно:

$$\sum_{i=1}^{2k-1} (-1)^{i+1} \tau_i + \frac{u^{\max}t - b(t)}{u^{\max} - u^{\min}} < \tau_{2k} < \sum_{i=1}^{2k-1} (-1)^{i+1} \tau_i + \frac{u^{\max}t - a(t)}{u^{\max} - u^{\min}}. \quad (41)$$

Поскольку при $\tau_{2k} < t < \bar{t}_{2k}$ в силу (35)

$$u^{\min}t + (u^{\max} - u^{\min}) \sum_{i=1}^{2k-1} (-1)^{i+1} \tau_i > a(t),$$

то

$$\frac{u^{\max}t - a(t)}{u^{\max} - u^{\min}} + \sum_{i=1}^{2k-1} (-1)^{i+1} \tau_i > t > \tau_{2k}. \quad (42)$$

Значит, правое неравенство (41) выполняется на отрезке $(\tau_{2k}, \bar{t}_{2k})$ и в силу определения момента τ_{2k} выполняется и на отрезке $(\bar{t}_{2k}, T)$, следовательно, оно выполняется на отрезке (τ_{2k}, T) .

Левое неравенство (41) следует из

$$\frac{u^{\max}t - b(t)}{u^{\max} - u^{\min}} < \frac{u^{\max}\bar{t}_{2k} - b(\bar{t}_{2k})}{u^{\max} - u^{\min}}, \quad (43)$$

которое выполняется при $t < \bar{t}_{2k}$ в силу условия 4^а (18).

Если левое неравенство (41) выполняется на отрезке $[\tau_{2k}, T]$, то допустимое управление $u_{2k}(t)$ уже построено. В противном случае существует наименьший момент $\bar{t}_{2k+1} > \tau_{2k}$, после которого неравенство нарушается.

При этом, очевидно, должно выполняться

$$u^{\max}\bar{t}_{2k+1} + (u^{\max} - u^{\min}) \sum_{i=1}^{2k-1} (-1)^{i+1} \tau_i = b(\bar{t}_{2k+1}). \quad (44)$$

Итак, описанным выше способом строится последовательность управлений $u_m(t)$, которые являются допустимыми на отрезках $[0, \bar{t}_{m+1}]$.

Докажем, что если выполнено условие (19), то найдется номер m , такой, что управление $u_m(t)$ будет допустимым на отрезке $[0, T]$. Для этого достаточно доказать ограниченность снизу $\bar{t}_{m+1} - \bar{t}_m$ константой, не зависящей от m .

Вычтем из формулы (44) равенство (36):

$$u^{\max}\bar{t}_{2k+1} - u^{\min}\bar{t}_{2k} - (u^{\max} - u^{\min}) \tau_{2k} = b(\bar{t}_{2k+1}) - a(\bar{t}_{2k});$$

$$u^{\max}(\bar{t}_{2k+1} - \bar{t}_{2k}) + (u^{\max} - u^{\min})(\bar{t}_{2k} - \tau_{2k}) = b(\bar{t}_{2k+1}) - a(\bar{t}_{2k}).$$

Но из (37) следует, что

$$(u^{\max} - u^{\min}) (\theta_{2k} - \tau_{2k}) = -u^{\max} (\hat{t}_{2k} - \theta_{2k}) + a (\hat{t}_{2k}) - a (\theta_{2k}). \quad (45)$$

Подставив (45) в (44), получим

$$u^{\max} (\theta_{2k+1} - \theta_{2k}) = b (\theta_{2k+1}) - a (\hat{t}_{2k}) + u^{\max} (\hat{t}_{2k} - \theta_{2k}). \quad (46)$$

Но поскольку $\theta_{2k} < \hat{t}_{2k} < \theta_{2k+1}$, то

$$\theta_{2k+1} - \theta_{2k} > \frac{b (\theta_{2k+1}) - a (\theta_{2k+1})}{u^{\max}} > \frac{b}{u^{\max}}, \quad (47)$$

что и требовалось доказать.

Итак, в любой системе «насосная станция — резервуары» можно легко проверить наличие такого дискретного управляющего воздействия (подачи насосной станции), которое при заданных потреблении из резервуаров обеспечивает поддержание уровней воды в резервуарах в технологических пределах.

Такой анализ особенно полезно проводить при решении задачи оптимального управления рассмотренной системы.

Список литературы: 1. Турк В. И., Минаев Л. В., Карелин В. Я. Насосы и насосные станции. — М.: Стройиздат, 1976. — 304 с. 2. Теория систем в приложении к проблемам окружающей среды: Пер. с ит./ Под ред. Серджо Ринальди. — К.: Вища шк. Головное изд-во, 1981. — 264 с.

Поступила в редакцию 05.11.83.

УДК 51:330.115

А. В. ПАНИШЕВ, канд. техн. наук, А. М. ДАНИЛЬЧЕНКО

ОБ ОДНОЙ ЗАДАЧЕ УПРАВЛЕНИЯ ТРАНСПОРТНЫМ ПРОЦЕССОМ

Расширим возможности практического использования минимаксной модели теории расписаний, рассмотренной в [1], при составлении оперативных планов грузовых автомобильных перевозок.

Анализируется ситуация, возникающая в процессе последовательного обслуживания погрузочным механизмом автомобилей, выполняющих маятниковые маршруты [2]. Сокращение времени выполнения заданного объема перевозок обеспечивается устранением простоев в пункте погрузки. Составление непрерывного графика работы автомобилей и погрузочного механизма является сложной задачей, решаемой с помощью эвристических методов. Для ограниченного количества транспортных средств она зачастую не имеет решения. В этом случае темпы отправления автомобилей в рейсы могут быть увеличены за

счет сокращения суммарного времени простоя погрузочного механизма. Рассмотрим простейшую оптимизационную задачу организации перевозочного процесса.

1. Предположим, что для выполнения заданного объема работ выделено n автомобилей, поочередно обслуживаемых погрузочным механизмом. На погрузку j -го автомобиля требуется время τ_j , $j = \overline{1, n}$, по истечении которого начинается рейс длительности β к единственному потребителю. Пусть каждый автомобиль выполняет $m > 2$ рейсов.

Требуется найти такую последовательность моментов отправления автомобилей в рейсы, которая бы минимизировала время перевозочного процесса T .

Заметим, что организация непрерывной работы каждого автомобиля и погрузочного механизма возможна здесь в единственном случае. Все рассматриваемые ресурсы работают без простоев, если значения β , n , τ_j таковы, что

$$\sum_{j=1}^n \tau_j - \tau_1 = \beta, \quad j = \overline{1, n}.$$

Очевидно, $\beta = (n - 1) \tau_1$, и для произвольных τ_j простои неизбежны.

При различных значениях τ_j , $j = \overline{1, n}$ и одинаковых длительностях рейсов β минимальное время выполнения процесса обеспечивается применением алгоритма последовательного обслуживания (первый поступивший на пункт погрузки автомобиль обслуживается первым) [3]. Тогда

$$T = (m - 1) \left[\tau_1 + \max \left(\beta, \sum_{j=1}^n \tau_j - \tau_1 \right) \right] + \sum_{j=1}^n \tau_j + \beta.$$

Отсюда следует, что первым обслуживается автомобиль с минимальным временем погрузки $\tau_{\min} = \min (\tau_j | 1 \leq j \leq n)$, а порядок, в котором оставшиеся $n - 1$ автомобилей отправляются в рейс, безразличен. Решение задачи заключается в поиске наименьшего элемента в массиве и требует не более n элементарных действий.

Сформулируем задачу, близкую к рассмотренной.

2. Пусть требуется τ_j единиц времени на обслуживание погрузочным механизмом автомобиля j , совершающего рейс длительности β_j , $j = \overline{1, n}$. Каждый автомобиль выполняет m одинаковых рейсов. Задача состоит в определении такой очередности всех загрузок, которой бы соответствовало минимальное время выполнения заданного объема работ.

В дальнейшем будет показано, что достаточно рассмотреть случай, когда последовательность первых n загрузок совпадает с $m - 1$ остальными последовательностями. Общее время выполнения перевозок T складывается из трех значений T_1 , T_2 , T_3 : $T_1 = (m - 1) (\tau_1 + \tau_2 + \dots + \tau_n)$. Через T_2 обозначим время

простая погрузочного механизма до начала обслуживания первого автомобиля, отправляющегося в $m-1$ -й рейс:

$$T_2 = (m-2) \sum_{j=1}^n \max \left(0, \beta_j - \sum_{i=1}^n \gamma_i + \gamma_j \right).$$

Промежуток времени с момента завершения первым автомобилем $m-1$ -го рейса до момента окончания всех работ равен

$$\begin{aligned} T_3 &= \max_{1 \leq l \leq n} \left[\sum_{i=1}^l \gamma_i + \beta_l + \sum_{j=1}^l \max \left(0, \beta_j - \sum_{i=1}^n \gamma_i + \gamma_j \right) \right] = \\ &= \max_{1 \leq l \leq n} \left\{ \sum_{i=1}^l \left[\max \left(0, \beta_i - \sum_{i=1}^n \gamma_i + \gamma_i \right) + \gamma_i \right] + \beta_l \right\}. \end{aligned}$$

Так как $T_1 = \text{const}$, $T_2 = \text{const}$, то рассматриваемая задача состоит в нахождении минимального значения для длительности T_3 , зависящей от того, в какой последовательности $\pi = (i_1, i_2, \dots, i_n)$ выполняются загрузки. Вводя обозначения

$$\varphi_{ij} = \max \left(0, \beta_j - \sum_{i=1}^n \gamma_i + \gamma_i \right) + \gamma_i, \quad \beta_{i_l} = \beta_{i_l},$$

получаем

$$T_3(\pi) = \max_{1 \leq l \leq n} \left(\sum_{j=1}^l \varphi_{ij} + \beta_{i_l} \right). \quad (1)$$

Как показано в [4], решение задачи минимизации (1) при $\beta_{i_l} > 0$, $l = \overline{1, n}$, $\varphi_{ij} \geq 0$ заключается в упорядочении по невозрастанию значений β_{i_l} , $j = \overline{1, n}$.

Последовательность π , оптимизирующая функцию (1), доставляет минимум функции

$$\begin{aligned} T(\pi) = T_1 + T_2 + T_3(\pi) &= \max_{1 \leq k_1 < \dots < k_m \leq n} \left(\sum_{i=1}^{k_1} \gamma_{i_1} + \beta_{i_{k_1}} + \sum_{i=k_1+1}^{k_2} \gamma_{i_2} + \dots + \right. \\ &\quad \left. + \sum_{i=k_{m-1}+1}^{k_m} \gamma_{i_l} + \beta_{i_{k_m}} \right), \end{aligned} \quad (2)$$

где $\pi = (i_1, i_2, \dots, i_n)$ рассматривается как перестановка n компонент $(\gamma_1, \beta_1, \gamma_1, \beta_1, \dots, \gamma_l, \beta_l)$. В компоненте j , $j = \overline{1, n}$ каждое из двух значений γ_j, β_j повторяется m раз. Используя результаты, содержащиеся в [1], можно сделать следующий вывод: выражение (2) является целевой функцией одномаршрутной задачи Джонсона для $m+1$ машин, $m = 1, 2, \dots$, где длительности прохождения детали j на первых m машинах равны, причем $t_{jk} = \gamma_j + \beta_j$, $k = \overline{1, m}$, $j = \overline{1, n}$, а на обработку этой же детали на машине $m+1$ требуется время $t_{j,m+1} = \beta_j \leq t_{jk}$.

Мы установили, что частный случай $m \times n$ задачи Беллмана — Джонсона и рассматриваемая в данной работе задача организации перевозочного процесса эквивалентны [4]. С позиций теории расписаний построение оптимального графика работы погрузочного средства и всех транспортных ресурсов проводится в классе P перестановочных расписаний π . Таким образом, получен следующий результат.

Утверждение. Пусть в $m \times n$ задаче Беллмана — Джонсона для каждой детали i , $i = \overline{1, n}$, длительности обработки на первых $m-1$ машинах равны и не меньше, чем время обработки на машине m . Последовательность деталей, упорядоченных по неубыванию времен прохождения на m -й машине, задает оптимальное расписание задачи.

Обратимся к примеру со следующими числовыми данными: $\beta_i = 5, 2, 2$; $\gamma_i = 1$; $m = 3$.

Определим $T_1 = 6$; $T_2 = \max(0,5 - 3 + 1) + \max(0,2 - 3 + 1) = 3$; $T_3(\pi^*) = \max[\max(0,5 - 3 + 1) + 1 + 5, \max(0,5 - 3 + 1) + \max(0,2 - 3 + 1) + 1 + 1 + 2, \max(0,5 - 3 + 1) + \max(0,2 - 3 + 1) + \max(0,2 - 3 + 1) + 1 + 1 + 1 + 2] = 9$. Получаем $T(\pi^*) = 18$.

Список литературы: 1. Панишев А. В., Данильченко А. М. Об одном классе задач теории расписаний. — АСУ и приборы автоматики, 1983, вып. 68, с. 31—36. 2. Житков В. А., Ким К. В. Методы оперативного планирования грузовых автомобильных перевозок. — М.: Транспорт, 1982. — 183 с. 3. Зелькович М., Шоу А., Гекмон Дж. Принципы разработки программного обеспечения. — М.: Мир, 1982. — 366 с. 4. Танаев В. С., Шкурба В. В. Введение в теорию расписаний. — М: Наука, 1975. — 266 с.

Поступила в редакцию 07.03.83.

УДК 519.713

И. Г. ФИЛИППЕНКО, канд. техн. наук, В. В. КАРАСЮК,
В. С. МЕРКУЛОВ

**ЭЛЕМЕНТЫ АВТОМАТИЗАЦИИ ЭКСПЕРИМЕНТА ПРИ
ИМИТАЦИОННОМ ИССЛЕДОВАНИИ ОДНОРОДНОЙ
ДИСКРЕТНОЙ СТРУКТУРЫ**

Разработка новых принципов структурной и функциональной организации средств вычислительной техники все в большей степени ориентируется на однородные микропроцессорные вычислительные структуры (ОВС). Благодаря свойству постепенной деградации, ОВС присущи высокие показатели готовности к работе, стабильная надежность функционирования и высокое реальное быстродействие [1].

При создании ОВС наиболее сложным является организация взаимодействия отдельных вычислителей и распределение решаемой задачи по элементам структуры. Такие проблемы не существуют в живой природе, хотя там часто единая задача решается множеством однотипных элементов. Эти свойства биологических информационных устройств привлекают внимание разработчиков технических средств, поэтому в настоящее время так актуальна задача исследования систем, построенных на нейронных принципах. Особенности этих систем следующие: вероятностный характер взаимодействия элементов; прием, преобразование и передача информации при прохождении сигнала по «биологическому каналу связи» осуществляется путем последовательного взаимодействия большого количества элементов; элементы варьируют своими мембранными потенциалами, за счет которых изменяется порог возбудимости, в зависимости от «алгоритма» обработки информации [2].

С учетом перечисленных свойств можно построить модель элементарного нейроподобного вычислителя [3]. Из набора таких моделей вычислителей можно создать модель ОВС. Для исследования и определения параметров модели ОВС создан имитационный программный комплекс. При статистическом исследовании возник ряд трудностей, которые решены путем введения в комплекс некоторых подсистем и использования определенных режимов работы.

С точки зрения организации машинных экспериментов моделирующий алгоритм (машинная модель) рассматривается в виде «черного ящика», имеющего входы $\bar{X} = \{\bar{X}_i\}$, $i = \overline{1, n}$ и выходы $\bar{Y} = \{\bar{Y}_i\}$.

Так как цель моделирования — оптимизация некоторого критерия качества, то значения этого критерия целесообразно выбрать в качестве выходных, и план эксперимента будет заключаться в выборе оптимизирующей последовательности $\bar{X}_1, \bar{X}_2, \dots$ в соответствии с определенным методом (например градиентного поиска).

Традиционная схема исследования может удовлетворить потребностям изучения этой структуры только путем перебора конечного числа вариантов в пределах некоторого множества. Сложность структуры модели определяется числом факторов, описывающих размерность вектора $\bar{X}_i = \{X_{i1}, X_{i2}, X_{ik}\}$, и количеством уровней для каждого фактора. В рассматриваемой модели количество факторов зависит от реализуемой функции и может изменяться от 8 до 60. Каждый фактор может иметь до пяти значений уровней, распределенных по логарифмической шкале.

Это приводит к необходимости использовать градиентные методы и вычислять отношение $\ast (\bar{Y}_i, \bar{Y}_{i-1})/\mu (\bar{X}_i, \bar{X}_{i-1})$ по каждой реализации исследуемого вероятностного процесса, которое влияет

на выбор направления перемещения в факторном пространстве, обеспечивая тем самым оптимальное «сканирование» многофакторного пространства.

На подготовительном этапе планирования эксперимента основная трудность состоит в задании критерия качества $F_k = F_k(\bar{X}_i, \epsilon, T)$, где ϵ — точность выполнения преобразования информации ОВС; T — время выполнения преобразования; $\bar{X}_i$ — вектор значений входных параметров. Практически F_k вводится в символическом виде, в системе существует ряд процедур трансляции исходного выражения в машинное представление и вычисления значения критерия $F_k(\bar{X}_i, \epsilon, T)$ при i -й реализации процесса. В точке экстремума качества, для которой выполняется условие $F_k(\bar{X}_i, \epsilon, T) = \min \{F_k\}$, производится многократное моделирование для статистической устойчивости результатов.

Таким образом, планирование машинных экспериментов позволяет искать оптимальные значения уровней факторов по заданному критерию качества и осуществлять многократное моделирование в точке оптимума для обеспечения статистической устойчивости результатов.

В целях управления ходом единичного эксперимента применяется автоматизированное управление. Необходимость такого управления вызвана специфическими особенностями исследования и проектирования нейрореподобных однородных вычислительных структур, в частности, невысокой достоверностью исходных данных, наличием трудноформализуемых факторов и ограничений, множественным видом критерия качества и др. Все эти факторы неопределенности позволяют отнести задачу синтеза ОВС к классу слабо структуризованных. Интерактивный режим исследования позволяет объединить мощные математические методы и алгоритмы оптимизации с опытом и неформализованными знаниями исследователя.

Диалоговый режим дает возможность активно вмешиваться в процесс моделирования, осуществлять контроль за ходом выполнения процедур комплекса. Возможен автономный и пошаговый режимы работы, с индикацией процесса моделирования по шагам. Предусмотрен отладочный режим, который может быть подключен в любой момент модельного времени и использован для отладки дополнительно вводимых в комплекс процедур.

Принцип работы диалога заключается в следующем: первоначально задается шаг моделирования $t_{\text{ш}}$ и значение модельного времени выхода системы на связь с исследователем — $t_{\text{м}}$. До модельного времени $t_{\text{м}}$ система работает автономно, выдавая накопленную статистическую информацию с интервалом $t_{\text{ш}}$. В момент $t_{\text{м}}$ исследователь может установить новые значения $t_{\text{ш}}$ и $t_{\text{м}}$. Здесь же может быть установлен, изменен или отменен любой режим работы системы.

Работа в диалоговом режиме подсистемы статистической обработки сопровождается затратами машинного времени (до 18 % общего времени моделирования). Время моделирования (рис. 1) при постоянной величине шага t_w : $t_m = t_{\text{мн}} + t_{\text{д}} + \Delta t$, при условии $t_{\text{мн}} + \Delta t = n t_w$. Здесь $t_{\text{мн}}$ — затраты машинного времени на моделирование развития процесса; $t_{\text{д}}$ — время для выполнения процедур статистической обработки и диалога; Δt — время работы модели после достижения цели единичного эксперимента;

$$t_{\text{д}} = \frac{t_{\text{м}}}{t_w} t_{\text{отв}},$$

где t_w — величина шага; $t_{\text{отв}}$ — время формирования ответа.

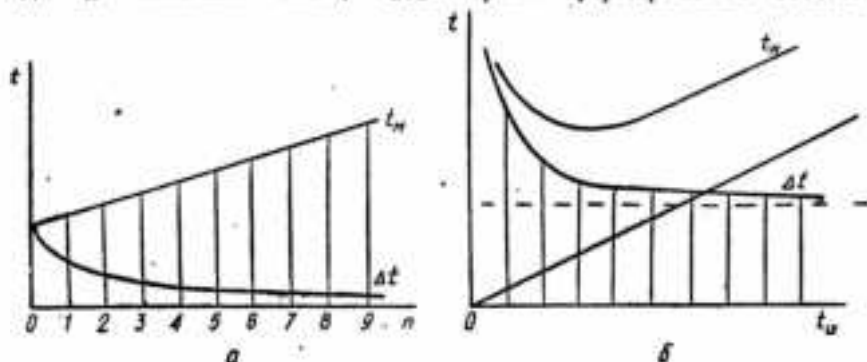


Рис. 1. Зависимости времени моделирования от количества проверок (а) и шага проверки (б)

Тогда

$$t_m = \frac{t_{\text{м}}}{t_w} t_{\text{отв}} + t_{\text{мн}} + \Delta t,$$

откуда

$$t_m = \frac{t_w t_{\text{мн}} + \Delta t t_w}{t_w - t_{\text{отв}}} = \frac{t_{\text{мн}} + \Delta t}{1 - \frac{t_{\text{отв}}}{t_w}}.$$

Таким образом, диалоговый режим позволяет рационально организовать процесс исследования ОВС. В то же время существует такое $t_w \in (t_{\text{отв}})$, при котором $t_m \rightarrow \min$.

Кроме диалогового режима, система может осуществлять моделирование и в автоматическом режиме. Наблюдаемые выходные переменные $\bar{Y}_i$ к своему установившемуся значению $\bar{Y}_*$ приближаются асимптотически, поэтому для единичного эксперимента определение момента, модельного времени выхода в состояние, описываемое $\bar{Y}_*$, является решающим для прекращения эксперимента и перехода к новой точке факторного пространства.

Таким образом, момент прекращения моделирования можно определить по распределению выходных переменных и задать логическое условие UZ : $UZ = \bar{R}$, где $\bar{R} = (r_1, r_2, \dots, r_n)$ — вектор распределения состояния элементов; n — количество подмножеств в ОВС.

Пользователю удобно условие UZ записывать в виде логического выражения, в котором присутствуют mnemonic обозначения состояний элементов ОВС. Например, при реализации операции «размножение» (рис. 2) $UZ = _GSV01 = 000 \&$

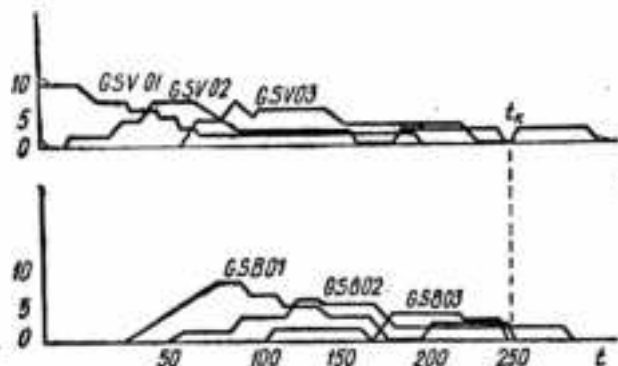


Рис. 2. Распределение состояний элементов подмножеств при реализации операции «размножение»

— $GSB01 = 000$ вызовет автоматическое прекращение эксперимента в момент модельного времени $t_m = 250$, когда установится в системе заданное распределение состояния элементов. Здесь $GSV01, GSV02, GSV03$ — распределения элементов 1, 2, 3 подмножеств в состоянии GSV во времени; $GSB01, GSB02, GSB03$ — распределения элементов 1, 2 и 3 подмножеств в состоянии GSB . Основным недостатком применения такого подхода является сложность процедуры трансляции и проверки истинности UZ . К тому же такие проверки выполняются в процессе моделирования многократно, через заданные интервалы модельного времени.

Таким образом, проверка UZ при каждом обращении к процедурам статистической обработки позволяет своевременно обнаружить момент прекращения моделирования и избежать лишнего прогона модели.

В прототипе рассматриваемой структуры в силу ее биологических принципов обмена информацией возможны ситуации, когда посланные сигналы не могут быть приняты (игнорируются). При обычном подходе необходимо, чтобы были сформированы атрибуты сигналов и их указатели занесены в соответствующие списки. После этого проверяется условие их

приема. В случае, когда оно не выполняется, сигналы игнорируются. Очевидна неоптимальность такого подхода.

На основании закона больших чисел можно предположить, что распределение моментов появления сигналов во времени подчинено равномерному закону и в некоторый момент модельного времени t_m это распределение может быть задано принудительно. Таким образом, последовательность действий по обработке сигналов заключена в следующем.

1. Определить возможность игнорирования сигналов определенного типа.

2. Если сигналы могут игнорироваться, то определить количество игнорируемых сигналов:

$$RR = \left[\frac{(TSC - TS - M(TK))(K - 1)}{M(TK)} \right],$$

где TSC — модельное время сигнала, который может изменить состояние системы; TS — модельное время первого игнорируемого сигнала; $M(TK)$ — математическое ожидание периода следования сигналов типа TK ; K — количество «перемещаемых» сигналов.

3. Если количество игнорируемых сигналов превышает заданное оптимальное число, то выполняются действия по игнорированию сигналов, т. е. формируются атрибуты сигналов со значениями времен

$$TS(I) = TSC - M(TK) + \left[\frac{M(TK)}{K} (I - 1) \right].$$

Здесь I — порядковый номер «перемещаемого» сигнала.

Таким образом, применение этого подхода дает возможность не моделировать большое количество особых состояний. Подключение подсистемы «адаптации» может производиться оператором в любой момент модельного времени (также как и выключение ее). Эта подсистема позволяет поднять производительность имитационного комплекса и сократить непроизводительные затраты машинного времени.

Все рассмотренные особенности реализованы в имитационном программном комплексе для моделирования однородных структур и позволили автоматизировать исследование ОВС.

Список литературы: 1. Евреинов Э. Н. Однородные вычислительные системы, структуры и среды. — М.: Радио и связь, 1981. — 208 с. 2. Социхо В. П. Электрические модели нейронов. — М.: Энергия, 1965. — 88 с. 3. Филиппенко Н. Г. Элемент высоконадежной вычислительной нейрореподобной проблемно-ориентированной структуры. — В кн.: Тез. докл. всесоюз. семинара «Специализированные процессоры параллельного действия для решения краевых задач». Рига, 1981, с. 105—106.

Поступила в редакцию 11.03.83.

**ФОРМИРОВАНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ ЗАПРОСОВ
К БОЛЬШОЙ БАЗЕ ДАННЫХ**

Прогресс в области проектирования средств, реализующих современную технологию обработки данных, обеспечивает появление достаточно совершенных систем управления базами данных, удовлетворяющих контингенту пользователей. Это приводит к резкому увеличению объемов данных, размещаемых на томах памяти, и образованию большой базы данных (ББД).

Так как внешняя память имеет число дисководов, существенно меньшее числа томов памяти, то в процессе обслуживания пакета запросов требуется осуществлять замену (переустановку) томов памяти, часто приводящую к значительным затратам времени. Поэтому естественной является необходимость в формировании такой последовательности обслуживания запросов ББД, которая обеспечивает минимальные потери времени на перестановку томов памяти.

Предположим, ББД размещена на m томах, а ЭВМ имеет d дисководов, причем $m \gg d$. Любой запрос обращается к томам, количество которых не больше количества дисководов. Задана матрица M обращений запросов к томам ББД. Строки матрицы M соответствуют запросам, а столбцы — томам ББД. На пересечении i -й строки $i = \overline{1, n}$, где n — количество запросов в очереди, и j -го столбца; $j = \overline{1, m}$, стоит 1, если данные для запроса q_i размещены на томе v_j , 0 — в противном случае. Обозначим через $|S_i|$ количество томов памяти, необходимых запросу q_i , что соответствует количеству единиц в i -й строке матрицы M .

Поскольку в пакете имеются запросы, обращающиеся к томам, количество которых меньше количества дисководов, то имеет смысл во время выполнения таких запросов подготавливаться к обслуживанию следующих запросов путем замены томов на свободных дисководах. В результате такого совмещения времени переустановки томов с временем обслуживания запроса суммарное время реализации пакета запросов уменьшится. Однако и в этом случае потери времени из-за простоев ЭВМ, связанных с заменой томов при обслуживании произвольной очереди запросов, значительны.

Рассмотрим составляющие затрат времени на переустановку томов на дисководах. Время снятия тома $t_{сн}$ состоит из времени торможения t_t и времени доставки тома от ЭВМ к месту складирования $t_{д1}$. Время установки $t_{уст}$ содержит время доставки $t_{д2}$ от места складирования до ЭВМ и время раз-

гона t_p [1]. Ряд других факторов, например время реакции системы на команды перевода томов в оперативный или автономный режим [1], время реакции операторов на указания системы снять или установить том, из-за их незначительного влияния на время снятия и установки не будем принимать во внимание. Определим время замены $t_{зм}$ томов как сумму времен снятия и установки, т. е. $t_{зм} = t_{сн} + t_{уст}$.

Остановку или разгон нескольких томов можно выполнить практически одновременно. Время доставки зависит от количества операторов на ЭВМ и количества снимаемых и устанавливаемых томов.

Количество дисководов, свободных от запроса q_i при его выполнении, определим как $d_{св,i} = d - |S_i|$. В $d_{св,i}$ включаются как незанятые дисководы, так и те, на которых расположены тома памяти, не нужные запросу q_i .

При переходе в процессе обработки от запроса q_i к запросу q_{i+1} необходимо на дисководах заменить V_{i+1} томов, $V_{i+1} = |S_{i+1}| - |S_i \cap S_{i+1}|$.

В общем случае значение V_{i+1} содержит две составляющие: V_{i+1}^1 , V_{i+1}^2 — количество томов, которые можно и которые нельзя заменить на свободных от запроса q_i дисководах. Заметим, что V_{i+1}^1 существует, если $d_{св,i} > 0$, а V_{i+1}^2 — если $V_{i+1} > d_{св,i}$.

Таким образом, справедливы следующие соотношения:

$$V_{i+1}^1 = \begin{cases} d_{св,i} & \text{если } V_{i+1} > d_{св,i} \\ V_{i+1} & \text{если } V_{i+1} \leq d_{св,i} \end{cases}$$

$$V_{i+1}^2 = \begin{cases} V_{i+1} - d_{св,i} & \text{если } V_{i+1} > d_{св,i} \\ 0 & \text{если } V_{i+1} \leq d_{св,i} \end{cases}$$

Время, необходимое на замену V_{i+1}^1 томов, определим из соотношения $t_{i+1}^1 = V_{i+1}^1 (t_{дс} + t_{дз}) + t_r + t_p$. Аналогично определим затраты времени на замену V_{i+1}^2 томов:

$$t_{i+1}^2 = V_{i+1}^2 (t_{дс} + t_{дз}) + t_r + t_p.$$

Затраты времени можно совместить с временем обслуживания $t_{об,i}$ запроса q_i полностью, если $t_{i+1}^1 \leq t_{об,i}$ и частично, если $t_{i+1}^1 > t_{об,i}$. Следовательно, потери времени на замену V_{i+1}^1 томов составят

$$\tilde{t}_{i+1}^1 = \begin{cases} t_{i+1}^1 - t_{об,i} & \text{если } t_{i+1}^1 > t_{об,i} \\ 0 & \text{если } t_{i+1}^1 \leq t_{об,i} \end{cases} \quad (1)$$

Потери времени, связанные с заменой V_{i+1}^2 томов, совпадают с затратами времени на замену V_{i+1}^2 томов, т. е. $\tilde{t}_{i+1}^2 = t_{i+1}^2$.

Если время обслуживания запроса q_i превышает время, необходимое на замену томов, т. е. $t_{об_i} > t_{i+1}^1$, то, согласно (1), $\tilde{t}_{i+1}^1 = 0$. В результате суммарные потери времени на замену томов при переходе от запроса q_i к запросу q_{i+1} определяются следующим образом:

$$\tilde{t}_{i+1} = \tilde{t}_{i+1}^2 = V_{i+1}^2 (t_{1i} + t_{2i}) + t_{\tau} + t_p.$$

Если время обслуживания запроса q_i значительно меньше времени, необходимого на замену томов, то $\tilde{t}_{i+1}^1 \approx t_{i+1}^1$. Тогда суммарные затраты времени на замену томов равны

$$\tilde{t}_{i+1} = \tilde{t}_{i+1}^1 + \tilde{t}_{i+1}^2 = t_{i+1}^1 + \tilde{t}_{i+1}^2 = V_{i+1} (t_{1i} + t_{2i}) + t_{\tau} + t_{\tau} + t_p + t_p.$$

Руководствуясь тем, что торможение и разгон нескольких томов можно осуществлять практически одновременно, данное выражение представим в виде $\tilde{t}_{i+1} = V_{i+1} (t_{1i} + t_{2i}) + t_{\tau} + t_p$.

Подводя итог проведенным рассуждениям, запишем

$$\tilde{t}_{i+1} = \begin{cases} V_{i+1}^2 (t_{1i} + t_{2i}) + t_{\tau} + t_p, & \text{если } t_{i+1}^1 \leq t_{об_i}, \\ V_{i+1} (t_{1i} + t_{2i}) + t_{\tau} + t_p - t_{об_i}, & \text{если } t_{i+1}^1 > t_{об_i}. \end{cases} \quad (2)$$

Соотношения (2) записаны исходя из того, что ЭВМ обслуживает один оператор. При наличии $\bar{P}$ операторов формула (2) преобразуется следующим образом:

$$\tilde{t}_{i+1} = \begin{cases} \left[\frac{V_{i+1}^2}{\bar{P}} \right] (t_{1i} + t_{2i}) + t_{\tau} + t_p, & \text{если } t_{i+1}^1 \leq t_{об_i}, \\ \left[\frac{V_{i+1}}{\bar{P}} \right] (t_{1i} + t_{2i}) + t_{\tau} + t_p - t_{об_i}, & \text{если } t_{i+1}^1 > t_{об_i}. \end{cases}$$

Простой ЭВМ, вызванные переустановкой томов, при реализации пакета запросов равны $\tilde{T} = \sum_{i=1}^n \tilde{t}_{i+1}$.

Введем показатель качества K организации работы большой базы данных с точки зрения минимизации простоев ЭВМ из-за переустановок томов как отношение суммарного времени потерь на переустановки $\tilde{T}$ к полезному времени T реализации пакета, определяемого как сумма времен обработки запросов, т. е.

$$K = \frac{\tilde{T}}{T}.$$

Чем меньше K , тем лучше организовано обслуживание запросов ББД. Уменьшить значение K можно путем уменьшения $\tilde{T}$; $\tilde{T}$

зависит от очередности выполнения запросов. В связи с этим цель настоящей работы заключается в нахождении на множестве Π перестановок запросов такой перестановки, которая обеспечивает минимум $\tilde{T}$.

Анализ соотношений (2) показывает, что уменьшение $\tilde{t}_{i+1}$ и, следовательно, уменьшение $\tilde{T}$ можно обеспечить путем минимизации V_{i+1} . Другими словами, минимум $\tilde{T}$ достигается, если при переходе от запроса к запросу для всего пакета запросов требуется заменять минимально возможное количество томов.

Эталоном такой очередности обслуживания запросов, обеспечивающим минимальное «разнообразие» томов памяти при переходе от запроса к запросу, может служить двоично-отраженный код Грея. Специфика этих кодов состоит в том, что последовательные кодовые слова различаются только в одном разряде.

Данные рассуждения служат основой предлагаемого алгоритма планирования последовательности обслуживания запросов ББД.

Алгоритм реализуется путем выполнения следующих шагов:

1. Считая любую строку матрицы M словом кода Грея, получим, руководствуясь [2], ее порядковый номер в последовательности кодовых слов, т. е. если задано кодовое слово $G_j = (g_m, g_{m-1}, \dots, g_1)$, то цифры его порядкового номера $j = (b_m, b_{m-1}, \dots, b_1)_2$, $0 \leq j < 2^m$ определяются как

$$b_k = \sum_{i=k+1}^m g_i (\text{mod } 2), \quad 0 \leq k < m. \quad (3)$$

Поясним процесс вычисления порядкового номера кодового слова. Допустим, в последовательности кодов Грея имеется слово $G_j = 1010110$. Цифры его порядкового номера в двоичной системе счисления, вычисляемые по формуле (3), равны: $b_0 = 0 + 1 + 1 + 0 + 1 + 0 + 1 (\text{mod } 2) = 0$, $b_1 = 1 + 1 + 0 + 1 + 0 + 1 (\text{mod } 2) = 0$, аналогично $b_2 = 1$, $b_3 = 0$, $b_4 = 0$, $b_5 = 1$, $b_6 = 1$, $b_7 = 0$. В результате порядковый номер $j = 011001002 = 100_{10}$. Таким образом, кодовое слово G_j в последовательности G (7) занимает строку с номером 100.

2. Произведем упорядочивание строк матрицы M в порядке неубывания их порядковых номеров. Полученная матрица M отражает последовательность обслуживания запросов, обеспечивающую значение потерь времени, связанных с переустановкой томов памяти, близкое к минимуму T .

3. Чтобы найти минимально возможное значение T для сформированной в результате выполнения первого и второго шагов алгоритма очереди запросов, определим номера томов, которые необходимо заменять в процессе обслуживания пакета запросов. Это важно, так как при переходе от запроса

q_i к запросу q_{i+1} имеет смысл снимать не все тома, которые были необходимы запросу q_i , а только то их количество, которое требуется заменить томами для запроса q_{i+1} . Следовательно, потребуется меньшее число замен томов, чем это следует из сравнительного анализа i -й и $i+1$ -й строки матрицы M .

Введем частоту f_j , $j = \overline{1, m}$ использования томов памяти запросами q_i , $i = \overline{1, n}$. Первоначально частота f_j представляет собой сумму единиц j -го столбца матрицы M . Каждое использование тома v_j запросом q_i приводит к уменьшению частоты f_j на единицу.

Таким образом, представляется естественным из нескольких претендующих на замену томов заменять те, для которых значение f_j на соответствующем этапе обслуживания пакета запросов минимальны.

Выполним сравнительный анализ потерь времени на переустановку томов памяти произвольной и сформированной предложенным алгоритмом очередей запросов на следующем примере.

Пример. Допустим, ББД содержит $m = 7$ томов, количество запросов $n = 14$, количество дисководов $d = 4$. Задана случайная очередь запросов к томам ББД в виде матрицы M (табл. 1). Чтобы не затруднять анализ матрицы M , нулевые элементы в нее не занесены.

Примем время обслуживания каждого запроса равным $t_{об} = 3$ мин, время доставки тома от ЭВМ к месту складирования $t_{дв} = 20$ с = 0,33 мин и от места складирования до ЭВМ $t_{дв} = 20$ с = 0,33 мин. Согласно эксплуатационной документации по НСМД ЕС 5061 время разгона $t_p = 66$ с = 1,1 мин, время торможения $t_t = 16$ с = 0,27 мин. Будем полагать, что ЭВМ обслуживает один оператор.

Вначале предположим, что работа вычислительной системы организована так, что не используется возможность совмещения времени замены томов на свободных при выполнении какого-либо запроса дисководах с временем обслуживания этого запроса.

Оценим потери времени, связанные с переустановкой томов в этом случае.

При переходе от запроса q_1 к запросу q_2 необходимо снять 2 тома (v_1 и v_2) и установить 3 тома (v_4 , v_5 , v_6). Потери времени, которые складываются из времени торможения двух томов, их доставки к месту складирования, доставки трех томов от места складирования к ЭВМ, времени разгона томов, следуя второму соотношению в (2) без учета $t_{об}$, равны

$$\bar{t}_2 = 0,27 + 2 \cdot 0,33 + 3 \cdot 0,33 + 1,1 = 3,02 \text{ мин.}$$

Аналогично определим потери времени на переустановку томов для всех остальных запросов: $\tilde{t}_3 = 0,27 + 0,33 + 0,33 + 1,1 = 2,03$ мин; $\tilde{t}_4 = 0,27 + 2 \cdot 0,33 + 2 \cdot 0,33 + 1,1 = 2,69$ мин; $\tilde{t}_5 = 0,27 + 3 \cdot 0,33 + 1 \cdot 0,33 + 1,1 = 2,69$ мин; $\tilde{t}_6 = 2,03$ мин; $\tilde{t}_7 = 2,69$ мин; $\tilde{t}_8 = 2,69$ мин; $\tilde{t}_9 = 2,03$ мин; $\tilde{t}_{10} = 2,69$ мин; $\tilde{t}_{11} = 2,69$ мин; $\tilde{t}_{12} = 3,35$ мин; $\tilde{t}_{13} = 2,69$ мин; $\tilde{t}_{14} = 2,69$ мин. Тогда суммарные потери времени

$$\tilde{T}_1 = \sum_{i=1}^{13} \tilde{t}_{i+1} = 33,5 \text{ мин.}$$

Определим потери времени, связанные с переустановкой томов, используя возможность совмещения времени замены томов с временем обслуживания запроса.

При переходе от запроса q_1 к запросу q_2 один из томов, необходимых запросу q_2 , например v_6 , может быть установлен во время выполнения запроса q_1 . Следовательно, после обслуживания запроса q_1 необходимо снять два тома (v_1 и v_2) и установить два тома (v_4 и v_5).

Вычислим потери времени, следуя первому соотношению в (2):

$$\tilde{t}_2 = 0,27 + 2 \cdot 0,33 + 2 \cdot 0,33 + 1,1 = 2,69 \text{ мин.}$$

Аналогично, $\tilde{t}_3 = \tilde{t}_4 = \tilde{t}_5 = \tilde{t}_6 = 2,03$ мин; $\tilde{t}_7 = 0$ мин; $\tilde{t}_8 = 2,03$ мин; $\tilde{t}_9 = \tilde{t}_{10} = 0$ мин; $\tilde{t}_{11} = \tilde{t}_{12} = \tilde{t}_{13} = 2,03$ мин; $\tilde{t}_{14} = 0$ мин. Отсюда суммарные потери времени $\tilde{T}_2 = 18,93$ мин.

Упорядочим заданную очередь запросов (табл. 1) в соответствии с разработанным алгоритмом. С этой целью по формуле (3) вычислим порядковые номера строк матрицы M в двоичной и десятичной форме (табл. 2). Упорядочим строки матрицы M в порядке неубывания их порядковых номеров. В результате получим матрицу $\tilde{M}$ (табл. 3).

Определим потери времени, анализируя матрицу $\tilde{M}$: $\tilde{t}_{13} = \tilde{t}_6 = \tilde{t}_1 = \tilde{t}_8 = \tilde{t}_{12} = \tilde{t}_{10} = \tilde{t}_9 = 0$.

При переходе к обслуживанию запроса q_2 потери времени на замену томов равны

$$\tilde{t}_2 = 0,27 \cdot 3 \cdot 0,33 + 3 \cdot 0,33 + 1,1 - 3 = 0,35 \text{ мин.}$$

Но для обслуживания запроса q_2 необходим том v_3 , который не нужен предшествующему запросу q_1 , но который остался на дисководе после выполнения запроса q_{10} . Поэтому фактические

Таблица 1

Номера запросов	Номера томов						
	7	6	5	4	3	2	1
1					1	1	1
2		1	1	1	1		
3	1	1	1				
4						1	1
5		1	1	1		1	
6					1	1	
7	1	1			1		
8				1		1	
9			1				
10				1		1	1
11		1					1
12				1	1	1	
13						1	1
14		1		1		1	

Таблица 2

Порядковый номер							
6	5	4	3	2	1	0	
0	0	0	0	0	1	0	5
0	1	0	1	0	0	0	40
1	0	1	1	1	1	1	95
0	0	0	0	0	1	0	2
1	1	0	1	1	0	0	44
0	0	0	0	1	0	0	4
1	0	0	0	1	1	1	71
0	0	0	1	1	0	0	12
0	0	1	1	1	1	1	31
0	0	1	1	0	1	1	27
0	1	1	1	1	1	0	62
0	0	1	0	1	1	1	23
0	0	0	0	0	1	0	2
0	1	1	0	0	1	1	51

Таблица 3

Номера запросов	Номера томов						
	7	6	5	4	3	2	1
4						1	1
13						1	1
6					1	1	
1					1	1	1
8				1		1	
12			1	1	1		
10			1		1	1	
9			1				
2		1	1	1	1		
5		1	1	1			1
14		1		1		1	
11		1					1
7	1	1			1		
3	1	1	1				

потери времени в этом случае $\tilde{t}_2 = 0$ мин; $\tilde{t}_5 = 2,03$ мин; $\tilde{t}_{14} = \tilde{t}_{11} = \tilde{t}_7 = \tilde{t}_3 = 0$ мин.

Суммарные потери времени при обслуживании всего пакета запросов $\tilde{T}_2 = 2,03$ мин.

Оценим качество К организации работы ББД с точки зрения минимизации простоев ЭВМ, связанных с переустановкой томов, для анализируемых случаев. В результате получим:

$$K_1 = \frac{\tilde{T}_1}{T} = \frac{33,5}{52} = 0,64; \quad K_2 = \frac{\tilde{T}_2}{T} = \frac{18,93}{52} = 0,36;$$

$$K_3 = \frac{\tilde{T}_3}{T} = \frac{2,03}{52} = 0,04,$$

где $T = 3 \cdot 14 = 52$ — полезное время реализации пакета запросов.

Анализ показателей качества свидетельствует о высокой эффективности предложенного алгоритма.

Список литературы: 1. Рейтборт Н. М. Пособие для оператора ЕС-ЭВМ. — М.: Статистика, 1979. — 192 с. 2. Рейтборт Н. М., Никитин Ю. А., Дед Н. Комбинаторные алгоритмы: Теория и практика. — М.: Мир, 1980. — 476 с.

Поступила в редколлегию 17.01.84.

**ОБ ОДНОМ ПОДХОДЕ К РАЦИОНАЛЬНОМУ РАЗМЕЩЕНИЮ
ИНФОРМАЦИИ В СИСТЕМЕ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ**

Для решения задач на ЭВМ необходимы алгоритмы (программы) и данные. С алгоритмическими языками и правилами записи алгоритмов знакомы сегодня большинство инженерно-технических работников. Однако принятого в распространенных алгоритмических языках представления данных для отображения всего многообразия применяемых структур оказалось недостаточно при таком классе задач, как информационно-поисковые и другие задачи обработки массивов данных.

Вследствие этого программное обеспечение ЭВМ развивалось в последние годы по нескольким основным направлениям. К одному из них относятся системы управления базами данных (СУБД), обеспечивающие переход от обработки данных, представленных файлами или другими стандартными типами данных в системах программирования, к работе с базами данных (БД). Архитектура СУБД обеспечивает принципиальную возможность создания массовых баз данных (статистических, нормативных, кадровых, физических экспериментов и др.) для их многочисленных применений в различных приложениях [1].

Существует три типа организации данных: внешняя, глобальная логическая организация данных и физическая [2]. Внешняя организация связана с таким представлением данных, каким его понимают прикладные программисты или конечные пользователи.

Глобальная логическая организация данных — это общая организация или концептуальная модель базы данных, на основании которой могут быть получены различные внешние организации. Такое логическое представление данных полностью независимо от физической организации. Оно описывается на языке описания данных, который является частью программных средств управления базами.

Физическая организация — это физическое представление данных и расположение их на запоминающих устройствах.

В настоящий момент принято считать три подхода к представлению баз данных: иерархический, сетевой, реляционный [2].

В иерархическом подходе используются древовидные структуры данных. Дерево представляет собой иерархию элементов, называемых узлами. На самом верхнем уровне иерархии имеется только один узел — корень. Каждый узел, кроме корня,

связан с одним узлом на более высоком уровне, называемым исходным для данного узла. Каждый элемент может быть связан с одним или несколькими элементами на более низком уровне. Они называются порожденными. Элементы, расположенные в конце ветви, т. е. не имеющие порожденных, называются листьями. Деревья применяются как для логического, так и для физического описания данных. В логическом описании они используются для определения связей между типами сегментов или типами записей, а при определении физической организации данных — для описания набора указателей и связей между элементами в индексах [3].

Если в отношении между данными порожденный элемент имеет более одного исходного элемента, то это отношение уже нельзя описать как древовидную структуру. Его описывают в виде сетевой структуры. В этом случае элемент может быть связан с любым другим элементом.

Сетевая структура подразумевает возможность установления произвольных связей между файлами БД. Совокупность файлов и связей между ними образует структуру, по своей конфигурации представляющую сеть. Межфайловые связи организуются как отношения между порождающими и порождаемыми записями, принадлежащими различным файлам [4].

В реляционных БД связи между данными представляются в форме двумерных таблиц, называемых отношением. Поскольку любая сетевая структура может быть с некоторой избыточностью разложена в совокупность древовидных структур, то и любое представление данных может быть сведено к двумерным плоским файлам. Процесс представления связей между данными в форме двумерных таблиц, выполняемый шаг за шагом для каждой связи, называется нормализацией. Различные пользователи могут выделять в БД различные наборы элементов данных и связи между ними. Следовательно, необходимо иметь возможность извлекать подмножества столбцов таблицы для одних пользователей, создавая таблицы меньшей размерности, а также объединять таблицы для других пользователей, создавая при этом таблицы большей размерности. Так как некоторые таблицы могут иметь очень много строк и столбцов, необходимо уметь выделять в них нужные подмножества. Благодаря операциям разрезания и склеивания, таблицы обладают гибкостью, которой лишено большинство древовидных и сетевых структур.

Основной недостаток реляционных БД — неэффективное использование ЭВМ. Выполнение операций соединения требует значительного времени. Если для небольших отношений эта проблема решается сравнительно легко, то для больших файлов она очень сложна. Эффективность использования операций здесь зависит от физического размещения данных, пе-

ред определением которой нужно достаточно квалифицированно установить связи между данными.

Успех реализации любой из рассмотренных СУБД зависит от способа логического и физического описания данных в конкретном применении. Использование сетевых или древовидных структур в некоторых приложениях оказывается мало эффективно, так как множество данных по своей природе не связаны между собой.

При выборе физической организации базы данных решающими факторами являются эффективность используемого метода, обеспечение малого времени ответа и минимальные затраты.

Рассматриваемая в данной статье автоматизированная информационно-поисковая система кадры — табельный учет с использованием дисплеев (АИСКТД) ориентирована на использование в системах с основным документом анкетного типа (управление кадрами, контроль исполнения приказов и т. д.). Обработка данных для решения большинства задач в этих системах выполняется с помощью последовательного просмотра (сканирования) всех записей файла в заданном порядке. Использование для этой цели продолжительных операций поиска на диске с произвольной выборкой записей нецелесообразно. Исходя из этого, в СУБД АИСКТД выбрана последовательная организация записей базового файла БД (поискового файла).

В системе предусмотрена возможность подключения дополнительных файлов. Связи между базовым и дополнительными файлами могут быть представлены в виде дерева.

Специфика обработки информации в системах с документами анкетного типа требует запоминания всей поступающей в систему информации по любому отдельному элементу каждого экземпляра документа. Например, при обработке документа «Личная карточка» по каждому работнику для элемента ПОДРАЗДЕЛЕНИЕ должны запоминаться все значения кодов структурных подразделений предприятия, в которых данный работник работал.

Для получения отчетов, отражающих динамику изменения значений элементов данных, необходима такая организация хранения данных, при которой выдерживалась бы очередность поступления (изменения) значений этих данных.

Большинство обрабатываемых информационных запросов (в системе описываемого здесь типа) обращается к последним значениям элементов данных. Следовательно, все поступающие значения одного элемента данных должны быть координированы в записи таким образом, чтобы первое обращение к этому элементу позволяло выбрать последнее (по времени поступления) значение этого элемента, второе — предыдущее, и т. д. Такое размещение информации значительно снизит время обработки большинства запросов.

При организации структуры хранения информации нужно также учитывать следующее: не по всем элементам данных требуется хранить все значения; изменения значений разных элементов могут поступать в любом порядке; изменения значений одного элемента поступают в том же порядке, в каком они фактически происходят; пользователю должна предоставляться возможность хранения значений элемента, для которого ранее такая возможность запрещалась; возможность внесения изменений для выполнения коррекции ошибочных данных.

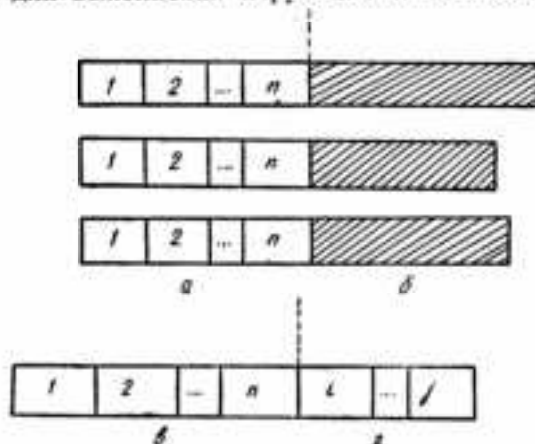


Рис. 1. Общая организация данных в записи

Одним из существенных условий конкретного приложения системы АИСКТД является требование использования такой физической организации БД, файлы которой являлись бы стандартными в ОС ЕС и информация которых могла быть доступна программам, написанным на языках высокого уровня (например PL/I).

Чтобы удовлетворить все перечисленные выше требования для физического размещения информации базовых файлов системы АИСКТД, была выбрана последовательная организация записей переменной длины со структурой (рис. 1). В общем случае запись имеет статическую (рис. 1, а) и переменную (рис. 1, б) части. В статической содержатся последние значения всех элементов (1, 2 ..., n), входящих в запись базового файла. Местоположение этих элементов в записи, а также вся характеристическая информация по ним хранится в специальном словаре определений данных. Если в системе используется файл, в котором не требуется хранение всех значений элементов, то в таком файле переменная часть в записях отсутствует. В этом случае файл определяется с записями фиксированной длины.

Переменная часть записи предназначена для накопления архивных (предыдущих) значений элементов по конкретной записи. Связи между отдельными значениями одного элемента устанавливаются с помощью специальных указателей [2], представляющих собой поле в записи, неразрывно связанное с конкретным значением элемента и указывающее на место расположения следующего значения этого элемента. Указатели представлены на рис. 1, 2 в виде стрелок, идущих от последнего значения элемента к предыдущему. В указателе используется адрес относительно начала записи. Отдельные значения одного элемента не обязательно должны располагаться непрерывно: они могут занимать различное место в переменной части записи и могут быть рассеяны среди различных значений других элементов.

Рассмотрим более подробно структуру переменной части записи, которая состоит из двух полей: поля указателей элементов статической части (рис. 1, $\sigma = 1, 2, \dots, n$ — указатели 1, 2, ..., n -го элементов) и поля архивной информации (рис. 1, $g = i, j$ — архивные значения i -го, j -го элементов). В первом поле записываются указатели последних значений всех элементов записи, которые ссылаются на свои предыдущие значения. Если элемент имеет единственное значение, то указатель содержит нулевой адрес (если элемент имеет несколько значений, то данный указатель является признаком конца списка этих значений). В поле архивной информации записываются все предыдущие значения для любых элементов записи. Вместе с самым значением записывается и указатель, адресующий предшествующее значение этого элемента (если оно есть).

Опишем алгоритмы создания, добавления и удаления отдельных значений элементов в записи. На диаграммах для обозначения конца цепи используется звездочка.

Пусть создается запись с тремя элементами A_1, B_1, C_1 . На диаграмме рис. 2 (начало поля архивной информации) показан вид создаваемой записи. В поле статической части записываются значения этих элементов. В поле указателей статической части для каждого элемента записывается указатель с нулевым адресом. Поле архивной информации отсутствует.

При добавлении новых значений элементов (B_2, A_2, B_3) для сохранения хронологической упорядоченности списка и выдерживания требования обработки первыми тех значений, которые появились последними, выполняются следующие операции (рис. 2, б). Значение элемента из статической части вместе со своим указателем из поля указателей переменной части помещается в свободное место поля архивной информации. Новое значение элемента помещается в статическую часть, а в поле его указателя помещается адрес значения из архивной части.

Для удаления значения элемента (B_1) из списка (рис. 2, а) просматриваются указатели, начиная с начала списка (с последнего значения). При этом в поле указателя предыдущего значения помещается адрес из указателя удаляемого значения, который ссылается на значение, следующее за удаляемым.

При такой организации информации в БД перед выполнением операций по ее обновлению не требуется производить операцию упорядочения порядка следования изменений между разными элементами записи.

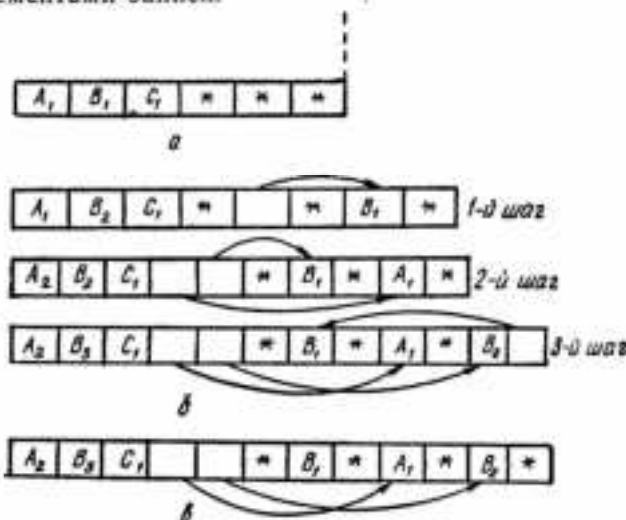


Рис. 2. Представление связей между отдельными значениями одного элемента с помощью указателей

При выполнении операций удаления отдельных значений элементов памяти, занимаемая ими, может не освобождаться (т. е. удаление выполняется не физически, а логически). При интенсивном пополнении новыми значениями элементов размер выделенной памяти под архивные значения может быть исчерпан. В этом случае возникает необходимость в освобождении памяти, занятой информацией, которая уже не нужна. Для этих целей в системе АИСКТД используются специальные алгоритмы сборки мусора [5].

Предложенный подход к организации физического хранения данных на машинных носителях информации хорошо согласуется с существующей технологией обработки данных, поэтому допускает эффективную реализацию в существующей среде. Организация файлов предложенной структуры позволяет использовать их не только программами СУБД АИСКТД, но и в программах пользователей, использующих традиционные языки программирования. При этом никакой дополнительной информации (типа хранимого в системе АИСКТД описания данных) не требуется.

Список литературы: 1. Мартин Д. База данных. — М.: Радио и связь, 1983. — 168 с. 2. Мартин Д. Организация баз данных в вычислительных системах. — М.: Мир, 1980. — 662 с. 3. Флорес Н. Структуры и управления данными. — М.: Финансы и статистика, 1982. — 319 с. 4. Система управления базами данных СЕДАН / Ю. В. Пурян, Ж. А. Михайлов, И. В. Демидов, Ю. И. Цыганков. — М.: Финансы и статистика, 1981. — 102 с. 5. Фостер Дж. Обработка списков. — М.: Мир, 1974. — 69 с.

Поступила в редколлегию 11.11.83.

УДК 681.325

А. А. БОРИСЕНКО, канд. техн. наук, С. И. ГУБАРЕВ, канд. техн. наук, Г. В. КУНО

АЛГОРИТМЫ ПОСТРОЕНИЯ КОДОВ С ПОСТОЯННЫМ ВЕСОМ НА ОСНОВЕ БИНОМИАЛЬНЫХ ЧИСЕЛ

При помехоустойчивом кодировании широко применяются коды с постоянным весом. Их положительные свойства — относительно высокая помехоустойчивость и простота технической реализации [1]. В работе [2] отмечено, что ограничивающим фактором, позволяющим в полной мере реализовать достоинства этих кодов, является отсутствие регулярных методов их построения. Решить такую задачу мы предлагаем на основе применения двоичных биномиальных чисел $A_i = (a_{i-1}, a_{i-2}, \dots, a_0)$, где $i = 1, 2, \dots, p$; i — количество разрядов биномиального числа. Эти числа могут быть получены с помощью биномиальных систем счисления с двоичным алфавитом [3].

Диапазон представления биномиальных чисел $P = C_n^k$, где k , n — целочисленные параметры системы счисления; $k > 0$, $n > k$; n — разрядность биномиальных чисел; k — контрольное число.

Биномиальные числа характеризуются следующими признаками: длина i их не должна быть больше $n-1$ и меньше $l = n-k$ разрядов; биномиальное число должно содержать или k единиц, или l нулей.

Пусть, например, заданы двоичные последовательности 0000, 00, 0111, 1000. Требуется определить, какие из указанных последовательностей при $n=6$, $k=4$ являются биномиальными числами, вычислить их диапазон и преобразовать в десятичную систему счисления. Диапазон представления биномиальных чисел для указанных n и k $P = C_n^k = C_6^4 = 15$. Последовательность 0000 не содержит единиц и содержит нулей больше l , поэтому для заданных n и k она не является биномиальным числом. Последовательности 00, 0111 биномиальны, так как в первой число нулей равно l , а во второй число единиц равно k . При этом их длина не превышает $n-1=5$ и не меньше $n-k=2$ разрядов.

Переход от биномиального числа к двоичному кодовому слову с постоянным весом осуществляется приписыванием к нему единиц, если двоичное число содержит l нулей, или нулей, если в нем содержится k единиц, до тех пор, пока длина двоичного слова не станет равной n . Например, 00 001111, 01111 011110. Правило обратного перехода состоит в вычеркивании из кодового слова с постоянным весом единиц или нулей, начиная с младшего разряда, до тех пор, пока их число не станет равным k или l . Например, 011110 01111, 001111 00.

Для решения задачи перебора кодов с постоянным весом рассмотрим алгоритм перебора биномиальных чисел, который основывается на реализации операции сложения биномиального числа с единицей до получения максимального числа $C_n^k - 1$, т. е. необходимо разработать алгоритм работы биномиального счетчика.

Этот алгоритм состоит в следующем. 1. Формируется начальное кодовое слово, состоящее из нулей. 2. В младший разряд записывается единица. 3. Если число единиц в кодовом слове меньше k , то приписываем справа нуль и переходим к пункту 2. 4. Если число единиц станет равным k , а длина — $n - 1$ разрядов, то в младший разряд, содержащий нуль, записывается единица; а в разряды справа от него, если число единиц слева не равно k , последовательно записываются нули до тех пор, пока их общее число не станет равным $n - k$. 5. Возврат к пункту 2.

Цикл повторяется до тех пор, пока k старших разрядов кодового слова не заполнятся единицами.

В табл. 1 для $n = 6$, $k = 4$ приведены все состояния биномиального счетчика, расположенные в порядке возрастания и полученные в соответствии с приведенным алгоритмом.

Таблица 1

Б/С №	Биномиаль- ное число	Б/С №	Биномиаль- ное число	Б/С №	Биномиаль- ное число	Б/С №	Биномиаль- ное число
0	00	4	01111	8	10111	12	11100
1	010	5	100	9	1100	13	11101
2	0110	6	1010	10	11010	14	1111
3	01110	7	10110	11	11011		

Перебор кодов с постоянным весом осуществляется на основе перебора биномиальных чисел, который рассмотрен выше.

Представляет интерес также задача перебора биномиальных чисел и соответственно кодов с постоянным весом в порядке убывания (алгоритм работы вычитающего счетчика). Алгоритм вычитания состоит в присвоении k значения l и инвертировании разрядов в суммирующем счетчике с новым k .

Таблица 2

Д д	Би- номи- альные числа сум- мирующего счетчика	Би- номи- альные числа сум- мирующего счетчика с $k=1$	Би- номи- альные числа вычитаю- щего счетчика	Д д	Би- номи- альные числа сум- мирующего счетчика	Би- номи- альные числа сум- мирующего счетчика с $k=1$	Би- номи- альные числа вычитаю- щего счетчика
0	00	0000	1111	8	10111	0101	1010
1	010	00010	11101	9	1100	011	100
2	0110	00011	11100	10	11010	10000	01111
3	01110	00100	11011	11	11011	10001	01110
4	01111	00101	11010	12	11100	1001	0110
5	100	0011	1100	13	11101	101	010
6	1010	01000	10111	14	1111	11	00
7	10110	01001	10110				

В табл. 2 приведены двоичные последовательности вычитающего биномиального счетчика.

На основе приведенных алгоритмов можно решить задачу генерирования кодов с постоянным весом по заданным номерам. Ее решение заключается в одновременном переборе чисел в степенной, например, двоичной и биномиальной системе счисления. При совпадении одного из перебираемых двоичных чисел с заданным происходит одновременный останов перебора двоичных и биномиальных чисел, результат генерирования будет отображен на биномиальном счетчике.

Это решение представляет интерес вследствие возможности его относительно простой схемой реализации. Недостаток его состоит в низкой скорости преобразования, что приводит к принципиальному ограничению на длину генерируемых кодовых комбинаций. Повысить скорость можно на основе приведенного ниже алгоритма перевода чисел из степенной системы счисления в биномиальную.

1. Определяется коэффициент a_{j-1} сравнением переводимого числа A_i с весом $(j-1)$ -го разряда $C_{n-1}^{k-q_j}$, где $q_j = 0$. Если $A_i \geq C_{n-1}^{k-q_j}$, то $a_{j-1} = 1$, $q_{j-1} = q_j + 1 = 1$. В противном случае $a_{j-1} = 0$, $q_{j-1} = q_j + 0 = 0$.

2. Определяется остаток $R = A_i - C_{n-1}^{k-q_j}$, если $a_{j-1} = 1$ или $R = A_i$ при $a_{j-1} = 0$.

3. Сравнивается остаток R с весом $(j-2)$ -разряда $C_{n-2}^{k-q_j-1}$. Если $R \geq C_{n-2}^{k-q_j-1}$, то $a_{j-2} = 1$, $q_{j-2} = q_{j-1} + 1$. В противном случае $a_{j-2} = 0$, $q_{j-2} = q_{j-1} + 0$.

4. Определяется коэффициент a_{j-3} , a_{j-4} , ... и т. д. по аналогии с пунктом 2,3 до тех пор, пока кодовая комбинация не станет удовлетворять условиям биномиального числа.

5. Коэффициент a_l , где $l = j-1, j-2, \dots, 0$, при котором выполняются условия биномиального числа, является коэффициентом a_0 .

Пример. Перевести число $A_1 = 2$ в биномиальную систему счисления с $n = 6$, $k = 4$. Так как $A_2 = 2 < C_5^4 = 5$, то $a_{j-1} = 0$, $q_{j-1} = q_j + 0 = 0 + 0 = 0$. Из того, что $R = A_2 = 2 > C_{6-2}^{4-0} = 1$, следует $a_{j-2} = 1$, $q_{j-2} = q_{j-1} + 1 = 0 + 1 = 1$. Определяется остаток $R_1 = R - C_{6-2}^{4-0} = 1$. Вес $(j-3)$ -го разряда равен $C_{6-3}^{4-1} = 1$. Так как $R = C_{6-3}^{4-1} = 1$, то $a_{j-3} = 1$, $q_{j-3} = q_{j-2} + 1 = 2$. Определяется новый остаток $R_2 = 1 - C_{6-3}^{4-1} = 0$. Вес $(j-4)$ -го разряда равен $C_{6-4}^{4-2} = 1$, а $a_{j-4} = 0$ ($R_2 = 0 < C_{6-4}^{4-2} = 1$). После появления коэффициента $a_{j-4} = 0$ число нулей в комбинации $a_{j-1}, a_{j-2}, a_{j-3}, a_{j-4} = 0110$ достигло двух, что соответствует определению биномиального числа.

Наличие ограничений на число единиц k и число нулей l позволяет обнаруживать ошибки в биномиальных числах. Для этого необходимо дополнить кодовые комбинации их нулями до длины $n - 1$. В табл. 3 для $n = 6$, $l = 4$, $k = 2$ приведены биномиальные комбинации с одинаковым числом разрядов. Ошибка будет обнаруживаться в случае, когда число единиц в комбинации будет больше двух, например комбинация 00111 является ошибочной. Ошибка будет также и в случае, если число нулей перед последней единицей больше или равно $n - k$.

Таблица 3

n 6	Равномерные биномиаль- ные комби- нации	n 5	Равномерные биномиаль- ные комби- нации	n 4	Равномерные биномиаль- ные комби- нации	n 3	Равномерные биномиаль- ные комби- нации
0	00000	4	00101	8	01010	12	10010
1	00010	5	00110	9	01100	13	10100
2	00011	6	01000	10	10000	14	11000
3	00100	7	01001	11	10001		

Из табл. 3 видно, что имеется 17 запрещенных комбинаций, из них одна (00001) обнаруживается на основе второго условия, остальные — с помощью первого. Таким образом, данный код эффективен для ошибок, носящих асимметричный характер типа $0 \rightarrow 1$. В случае другого характера ошибок, вплоть до симметричного, степень асимметрии регулируется изменением k .

На основе предложенного алгоритма перебора биномиальных чисел был реализован помехоустойчивый счетчик с асимметричным характером исправляемых ошибок.

Таким образом, предложенные алгоритмы, использующие биномиальные числа, решают задачи перебора и генерирования кодов с постоянным весом практически необходимой мощности и позволяют разрабатывать помехоустойчивые суммирующие и вычитающие счетчики.

Список литературы: 1. Березюк Н. Г., Андрущенко А. Г., Мощницкий С. С. Кодирование информации.—Х.: Вища шк. Изд-во при Харьк. ун-те, 1978.—252 с. 2. Цымбал В. П. Теория информации и кодирование.—К.: Вища шк., Головное изд-во, 1977.—287 с. 3. Система с биномиальным основанием и двоичным алфавитом / А. А. Борисенко; ВНИИТИ. М., 1982, № 909—88. 20 с.

Поступила в редколлегию 12.12.83

УДК 681.142.2

В. М. ДОБРЯНСКИЙ, С. Л. ДОНИНА, Н. П. КРИСТАЛЬ

СИНТЕЗ РАБОЧЕЙ ПРОГРАММЫ УПРАВЛЯЮЩЕЙ ЦИФРОВОЙ ЭВМ ИЗ ФУНКЦИОНАЛЬНО ИЗБЫТОЧНОГО НАБОРА МОДУЛЕЙ

Постановка задачи. При разработке программного обеспечения управляющих вычислительных систем с ограниченными ресурсами (мини- и микро-ЭВМ, используемых в системах управления подвижными объектами и некоторыми технологическими процессами) возникает проблема удовлетворения двум таким противоречивым критериям качества, как занимаемый рабочей программой объем памяти и время исполнения. Эта проблема может быть решена реализацией принципа функциональной избыточности, суть которого состоит в следующем: программное обеспечение построено по модульному принципу; каждый модуль является программным эквивалентом некоторого функционального признака (например, процедуры ввода-вывода, стандартные программы вычисления элементарных функций и т. п.); каждый функциональный признак реализуется созданием двух или более программных блоков, работающих по различным принципам; каждый из блоков, реализующих один и тот же функциональный признак, минимизирован по одному из критериев качества с различной степенью ущерба остальным (стандартная программа вычисления $\sin X$ может, например, иметь три реализации, из которых первая имеет минимальное время исполнения, вторая занимает минимальный объем памяти, а третья уступает первым двум и по времени исполнения, и по занимаемым объемам памяти, но превосходит их по точности вычислений).

Организованное таким образом программное обеспечение представляет собой функционально избыточную библиотеку модулей

$$B = \bigcup_i B_i, \quad (1)$$

где $B_i = \{b_1^i, b_2^i, \dots, b_{n_i}^i\}$ — подмножество программных модулей, реализующее функциональный признак a_i .

Задача генерации рабочей версии программного обеспечения управляющей мини- или микро-ЭВМ сводится к выбору из функционально избыточного множества B подмножества $B_k \subset B$, удовлетворяющего некоторому критерию, учитывающему в определенной зависимости критерия качества.

Математическую модель генерации можно представить в следующем виде.

Пусть $D = \|d_{ij}\|$ — матрица инцидентий функционально избыточного множества B ($d_{ij} = 1$, если i -й функциональный признак имеет j -ю реализацию в виде программного модуля b_j^i и $d_{ij} = 0$ — в противном случае); $T = \|t_{ij}\|$ — матрица временных характеристик элементов множества B (t_{ij} — время, необходимое для исполнения модуля b_j^i); $M = \|\mu_{ij}\|$ — матрица пространственных характеристик элементов множества B (μ_{ij} — объем памяти, занимаемый модулем b_j^i); $\tilde{C} = \|\tilde{c}_{ij}\|$ — матрица инцидентий множества B_k , где $i \in \{1, 2, \dots, k\}$; k — количество функциональных признаков конкретного объекта управления, $j \in \{1, 2, \dots, l\}$, $l = \max_{1 \leq i \leq k} l_i$.

Требуется найти такую матрицу инцидентий C , которая удовлетворяла бы следующим условиям:

$$\min_s Jt = \sum_{i=1}^k \sum_{j=1}^l t_{ij} c_{ij} \quad (2)$$

при

$$\bigvee_{j=1}^l c_{ij} \sim a_i, \quad (3)$$

$$\sum_{i=1}^k \sum_{j=1}^l \mu_{ij} c_{ij} \leq J_{\mu R}, \quad (4)$$

$$c_{ij} = \begin{cases} \tilde{c}_{ij} \& d_{ij}, & \text{если } (\forall r \in \{1, 2, \dots, l\}, r \neq j) (c_{ir} = 0), \\ 0, & \text{если } (\exists r \in \{1, 2, \dots, l\}, r \neq j). \end{cases} \quad (5)$$

Здесь $J_{\mu R}$ — запасы памяти.

В выражениях (2), (4) элементы матрицы C интерпретируются как десятичные числа, принимающие значения ноль или единица. В формулах (3), (5) значения c_{ij} интерпретируются как логические.

Модель генерации при минимизации занимаемой программным обеспечением памяти с удовлетворением заранее заданным ограничениям по времени исполнения аналогична предыдущей. Это дает возможность процедуры оптимизации по любому из критериев представить в виде стандартного, настраиваемого по соответствующим параметрам, блока.

Решение задачи. Решение задачи, поставленной в виде (2)–(5), можно выполнить перебором всех возможных вариантов с

вычисленным для каждого варианта времени исполнения и занимаемого объема памяти. При этом на каждом шаге реализуется процедура

$$J_{i\min} = \begin{cases} J_i & \text{при } J_i < J_{i\min} \text{ \& } J_p < J_{pR}; \\ J_{i\min} & \text{при } J_i \geq J_{i\min} \text{ \& } J_p \geq J_{pR}. \end{cases} \quad (6)$$

где $J_{i\min}$ — время исполнения, минимальное из всех просмотренных ранее вариантов; J_i — время исполнения набора, анализируемого на данном шаге; J_p — память, занимаемая набором, анализируемым на данном шаге.

Одновременно при выполнении условий (2), (4) фиксируется набор, для которого эти условия выполняются, т. е. вместо матрицы C достаточно найти вектор $N1$, каждая координата которого соответствует номеру реализации соответствующего функционального признака.

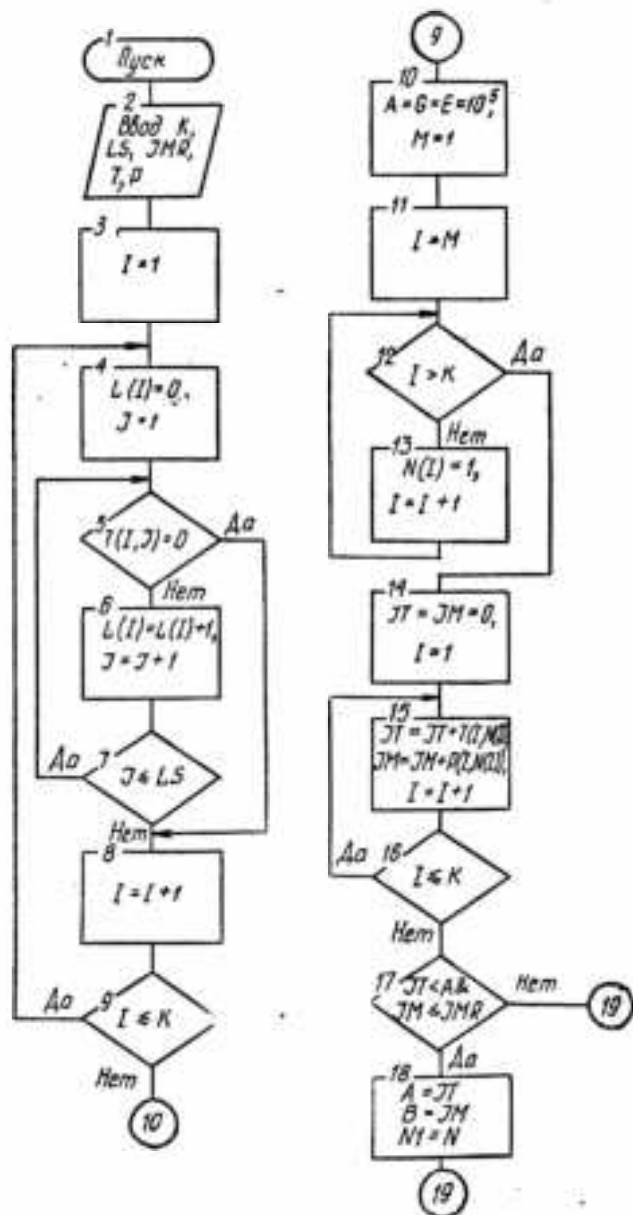
Очевидно, что количество просматриваемых вариантов при этом равно

$$\zeta = l^k, \quad (7)$$

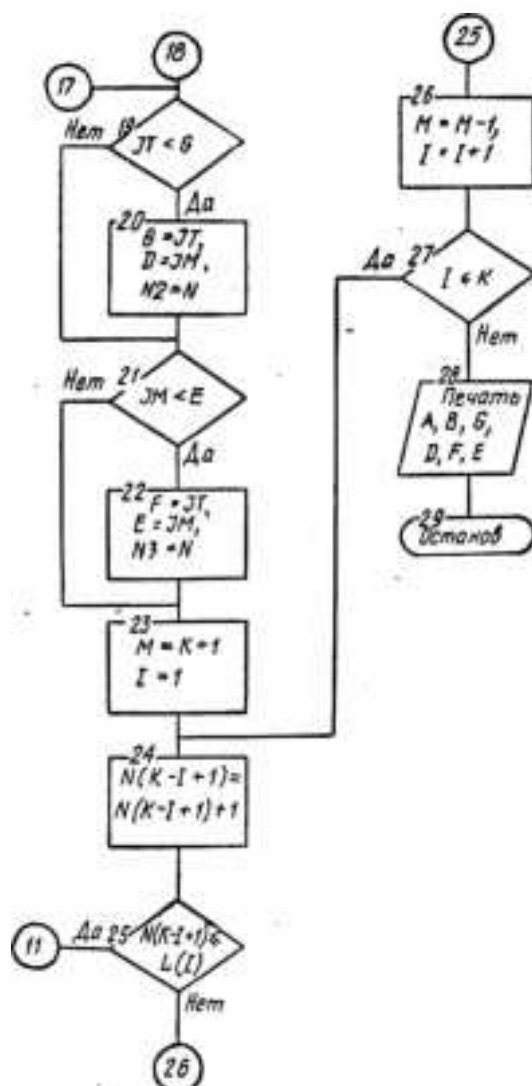
где k, l — количество строк и столбцов в матрицах соответственно.

Так как размеры матриц принципиально могут быть различными, хотя и не превышающими некоторых предельных значений, то перебор всех возможных вариантов сводится к сложному циклическому алгоритму с переменной кратностью вложений. Авторами предлагается алгоритм выбора оптимального набора модулей, образующих конкретную версию рабочей программы управляющей цифровой ЭВМ.

Полный алгоритм решения поставленной задачи изображен на рисунке. Здесь, кроме варианта, удовлетворяющего условиям (2), (4), отыскиваются также наборы, имеющие минимальное время исполнения без ограничений по занимаемой памяти и минимальный объем занимаемой памяти без ограничений на время исполнения. Принятые в алгоритме условные обозначения имеют следующий смысл: K — количество строк в матрицах T, P ; LS — количество столбцов в матрицах T, P ; JMR — ограничение на занимаемый выбранным вариантом объем памяти; T — матрица временных характеристик; P — матрица пространственных характеристик; N — вектор текущих значений индексов вложенных циклов; L — вектор предельных значений индексов вложенных циклов; $N1$ — вектор номеров реализаций, соответствующий оптимальному варианту; $N2$ — вектор номеров реализаций, соответствующий варианту, имеющему минимальное время исполнения без ограничений на занимаемый объем памяти; N — вектор номеров реализаций, соответствующий варианту, имеющему минимальный объем занимаемой памяти без ограничений на время исполнения; A, B — время исполнения и занимаемая память, соответствующие оптимальному варианту; G, D — время исполнения



и занимаемая память, соответствующие варианту с минимальным временем исполнения без ограничений на занимаемый объем памяти; F , E — время исполнения и занимаемая память, соответствующие варианту, имеющему минимальный объем памяти без ограничений на время исполнения; JT — время испол-



Алгоритм поиска оптимальной версии рабочей программы управляющей вычислительной машины

нения набора, анализируемого на данном шаге; JM — память, занимаемая набором, анализируемым на данным шаге.

В блок-схеме (см. рисунок) блоки 3+9—формирование вектора $L = \{L(1), L(2), \dots, L(K)\}$ предельных значений индексов вложенных циклов; блоки 11+13—организация K -кратного цикла; блоки 14+23— тело цикла, реализующее описанные в начале раздела процедуры; блоки 23+28— анализ конца K -крат-

ного цикла; блоки 2 и 5 — блоки ввода исходных данных и вывода результатов вычислений.

Заключение. Рассматриваемая задача возникла и была решена при создании программного обеспечения микропроцессорных комплексов управления технологическими процессами. Предложенный алгоритм генерации представляет собой, наряду со средствами автоматизированного выбора вектора функциональных признаков и формирования матриц временных и пространственных характеристик, фрагмент системы синтеза рабочей программы из функционально избыточного и функционально избирательного набора программных модулей.

Результаты контрольной отладки разработанного алгоритма и реализующей его программы на языке *PL/I* подтверждают справедливость принципов, сформулированных в постановке задачи.

Поступила в редакцию 12.12.83.

УДК 62—52:621

А. Л. ЛИТВИНОВ, канд. техн. наук

АНАЛИТИЧЕСКОЕ МОДЕЛИРОВАНИЕ АВТОМАТИЗИРОВАННЫХ СИСТЕМ ГРУППОВОГО УПРАВЛЕНИЯ СТАНКАМИ С ЧПУ ОТ ЭВМ. 2.

При подключении большого количества станков с числовым программным управлением (ЧПУ) к одному вычислительному комплексу необходимо использовать для хранения управляющих программ станков (УПС) внешнюю память, реализованную на магнитных дисках или на магнитных лентах. В оперативную память будут записываться только отдельные сегменты УПС, из которых будет производиться покадровая выдача информации в станки. После выдачи одного сегмента на его место будет записан новый и так до конца работы по данной УПС.

При такой организации обмена информации между ЭВМ и станками возможно несколько вариантов построения вычислительного комплекса (ВК).

1. Использование в качестве внешней памяти накопителя на магнитных дисках (НМД), который подключается к процессору комплекса через канал прямого доступа в память (КПДП). Диски обладают неплохими характеристиками по доступу к информации и большими возможностями по объему хранимой информации. Используя НМД, можно реализовать два варианта построения комплекса: однопроцессорный и двухпроцес-

сорный. При двухпроцессорном варианте построения ВК сам комплекс строится по иерархическому принципу. Процессор нижнего уровня (коммутирующий) осуществляет непосредственную раздачу кадров в станки. Процессор верхнего уровня (ведущий) осуществляет перекачку информации с внешней памяти в ОЗУ коммутирующего процессора, а также решает ряд задач по управлению цехом (задачи АСУП). При однопроцессорном варианте построения ВК функции раздачи кадров и перекачки информации возлагаются на один процессор. В этом случае возрастает объем программного обеспечения. Поиск информации на диске через КПДП может быть совмещен с раздачей информации станкам. Вопрос о количестве станков в системе, размерах сегментов информации должен быть решен с помощью моделирования.

2. Использование в качестве внешней памяти накопителя на магнитной ленте. По своим временным характеристикам (время доступа) магнитные ленты, будучи устройством памяти с чисто последовательным доступом, значительно уступают магнитным дискам. Однако магнитные ленты обладают более лучшими эксплуатационными характеристиками и более высокой надежностью, что играет существенную роль в условиях цеха, отличающихся сильными электромагнитными помехами, механическими воздействиями, запыленностью воздуха и другими неблагоприятными факторами.

Для построения модели системы группового управления (СГУ) с накопителем на магнитных дисках примем следующую структуру организации: область хранения управляющих программ станков разбита на l зон по числу станков; каждая зона состоит из двух одинаковых сегментов; по запросам от станков осуществляется выдача строк информации из одного сегмента. Как только этот сегмент будет исчерпан, происходит переключение на второй сегмент и выставляется требование на перепись нового сегмента из внешней памяти.

Пусть размер одного сегмента F байт, размер одного кадра B байт, интенсивность поступления запросов от одного станка на кадры информации λ . Тогда интенсивность поступления запросов на перепись сегментов от одной зоны ОЗУ $\lambda = \lambda(B + 3)/F$, выражение $(B + 3)$ учитывает межкадровый промежуток, который будет заполняться нулями.

Время передачи сегмента из НМД в ОЗУ складывается из следующих частей: а) формирование заявок на считывание (≈ 85 мкс); б) подвод считывающей головки НМД к нужному цилиндру (в среднем занимает 45 мкс); в) поиск записи на дорожке (в среднем занимает 12,5 мкс); г) считывание и передача сегмента из НМД с помощью КПДП в ОЗУ (равно $\frac{1}{156} \cdot \frac{F}{2} = \frac{F}{312}$; выражение $\frac{F}{2}$ учитывает передачу информации между НМД и ОЗУ

с помощью КПДП словами по два байта в каждом, а 156 (слов/мкс) — скорость передачи информации между НМД и ОЗУ).

Все приведенные характеристики выбраны исходя из использования в качестве внешней памяти НМД типа А 322 (ИЗОТ) для ВК М-6000.

Тогда среднее время обслуживания одной заявки на перепись сегмента

$$G = 45 + 12,5 + \frac{F}{312} \text{ (мкс)} = \left(57,5 + \frac{F}{312} \right) 10^{-3} \text{ (с)}.$$

Временем формирования заявки пренебрегаем ввиду его малости. Время в 57,5 мкс — это время поиска нужной информации на диске. Этот поиск осуществляется аппаратурой накопителя и не влияет на раздачу кадров станкам. На эту раздачу влияет лишь время перекачки сегмента из НМД в ОЗУ, в период которого КПДП блокирует ОЗУ. Средний интервал времени между поступлениями запросов на кадры со стороны станков равен $\tau = \frac{1}{\lambda}$. Оценим значение τ . По итогам статисти-

стической обработки УПС $\lambda = 0,1$ (запрос/с). Количество станков $n = 1 - 100$; таким образом, $n\tau = 100$ мкс. Размер сегмента может принимать значения от 60 до 300 (байт), т. е.

$\max \frac{F}{312} = 1$ мкс. Таким образом, $\frac{F}{213} \ll \tau$. Из этого можно сделать вывод, что процесс перекачки сегментов из НМД в ОЗУ практически не влияет на процесс раздачи кадров УПС отдельным станкам, а следовательно, вместо одной сложной модели СГУ можно использовать две более простые: одна модель нижнего уровня (коммутирующего процессора), которая будет описывать процесс раздачи информации из ОЗУ отдельным станкам, и вторая — модель верхнего уровня, которая будет описывать поиск и выдачу информации из накопителя на магнитных дисках с помощью КПДП в ОЗУ. При этом техническая реализация ВК (двухуровневый или одноуровневый) не играет существенной роли для моделирования, кроме ограничения на объем памяти под УПС.

Таким образом, в качестве модели нижнего уровня СГУ в двухуровневой системе можно использовать модели, разработанные раньше.

Разработаем модель СГУ верхнего уровня. В качестве источника заявок будет служить операционная система, которая при освобождении соответствующей половины зоны выставляет запрос в КПДП на перепись очередного сегмента. По одной зоне операционная система не будет выставлять нового запроса, пока не обслужится предыдущий. Обслуживающим устройством является КПДП вместе с устройством управления НМД. Процесс обслуживания запроса заключается в подводе головки на нужный цилиндр и считывании информации. Так как все головки НМД передвигаются одновременно, то КПДП вместе с НМД

является одноканальным устройством обслуживания, а весь верхний уровень также представляет собой однолинейную замкнутую систему обслуживания с ожиданием и конечным источником вызовов. Ее параметры следующие: интенсивность поступления заявок от одной зоны $\lambda = \lambda(B+3)/F$; интенсивность обслуживания $M = 1/G$; загрузка $\rho = \frac{\lambda(B+3)}{F} (57,5 + \frac{F}{312}) 10^{-3}$; количество источников вызовов L . Подставив эти параметры в выражения для вероятностей состояний и вероятность безотказной передачи информации, можно определить основные характеристики модели верхнего уровня СГУ, выраженные как функции от числа станков в системе, размера сегмента УПС, времени отработки одного кадра. Эти выражения для практического использования должны быть дополнены ограничением по объему оперативной памяти, которая отводится под управляющие программы станков. Для двухпроцессорной системы можно приближенно положить (учитывая опыт разработки одноканальной системы)

$$Fn < 20000. \quad (1)$$

Если СГУ строится на одном процессоре, то за счет программ, которые осуществляют перекачку информации с НМД в ОЗУ, область под УПС уменьшается; ориентировочно под программы связи с НМД необходимо отвести 10К слов памяти. Тогда ограничение по объему ОЗУ под УПС в случае одноканального комплекса имеет вид

$$Fn < 10000. \quad (2)$$

Графики зависимости вероятности безотказной работы R верхнего уровня СГУ по пересылке информации с НМД от количества станков в СГУ и объема сегмента в байтах приведены на рис. 1. Область, обозначенная двойной штриховкой, соответствует ограничению (1), т. е. $Fn < 20000$, а область, обозначенная однократной штриховкой, соответствует ограничению (2). Из графиков видно, что при принятых предположениях относительно частоты поступления запросов от станков и параметров вычислительного комплекса М-6000 вероятность сбоя в случае передачи информации в верхнем уровне системы очень мала, что позволяет организовать верхний уровень на НМД. Остается открытым только вопрос о свободной области памяти, который решится при конкретном программировании математического обеспечения.

Разработаем модель СГУ, в которой для хранения УПС используется накопитель на магнитной ленте (НМЛ). Для этого используем ту же структуру построения СГУ, что и в случае использования НМД, исследование которой было проведено выше. Разница между этими системами будет только в про-

граммном обеспечении верхнего уровня. Как исходную информацию при построении модели примем следующее.

1. Для реализации внешней памяти используется устройство внешней памяти на магнитных лентах АЗ11-3 (ЕС 5012-01).

2. В одной ленте в среднем A кадров ($A \approx 200$).

3. Средняя длина одного кадра B строк ($B \approx 21$), среднее расстояние между началами двух кадров $B + 3$ строк.

4. Длина одной ленты с УПС $C = A(B + 3)$ строк или байт ($C = 24 \cdot 200 = 4800$ (байт)).

5. Плотность записи равна D байт/мм (по ТУ $D = 8$ или 32).

6. Каждая лента занимает с учетом межзоновых промежутков и расстояний между двумя УПС ε строк: $\varepsilon = A(B + 3) + 2 \times 15,21D = 4800 + 30,42D$.

Предполагается, что каждая лента занимает, как минимум, две зоны; расстояние между зонами номинальное, равное 15,21 мм; коэффициент два учитывает два межзоновых промежутка, относящихся к одной ленте, первый — между двумя зонами, второй — между двумя лентами.

7. Скорость движения ленты v м/с (по ТУ $v = 2$ или 3 м/с).

8. На один станок в среднем в смену дается задание изготовить E типов деталей ($E \approx 3$).

9. В системе группового управления n станков.

10. Длина сегмента передаваемой информации из НМЛ в ОЗУ F байт.

11. Время между поступлениями запросов на считывание кадров от одного станка в нижнем уровне $\delta = 10$ с, а интенсивность поступления запросов $\lambda = 1/\delta = 0,1$ (запрос/с).

12. Для повышения эффективности работы верхнего уровня перед рабочей сменой готовится специальная магнитная лента, в которой записаны управляющие программы станков на данную смену. Для каждого станка УПС записываются подряд. Дублирование УПС исключено. Такая лента может подготавливаться на основе другой магнитной ленты, которая содержит всю библиотеку используемых УПС на предприятии. Эта магнитная лента (назовем ее Библиотека УПС) может включать в себя до

$$\frac{30 \cdot 10^6}{A(B + 3) + 30,42D} < \frac{30 \cdot 10^6}{4800 + 30,42 \cdot 32} \approx 5000$$

управляющих программ станков.

Так как общая структура и идеология построения системы с НМЛ такая же, как в случае использования НМД, то и математические модели этих систем будут одинаковы: системы массового обслуживания с конечным числом источников. Разница будет только в законах распределения потоков информации и количественных характеристиках этих законов.

Определим характеристики для исследуемой СГУ верхнего уровня. В целях упрощения исследований примем, что все по-

токи событий носят пуассоновский характер. Тогда входной поток запросов от одной зоны информации описывается следующей функцией распределения: $A(t) = 1 - e^{-\Lambda t}$, где Λ определяется выражением $\lambda(B+3)/F$. Это выражение для Λ то же, что и в случае модели системы с НМД. Функция распределения времени обработки запросов в КПДП вместе с НМЛ имеет вид $B(t) = 1 - e^{-\frac{t}{\omega}}$. Здесь ω — среднее время обслуживания одного запроса. Рассчитаем это время.

Для каждого станка отводится зона длиной α строки $\alpha = [A(B+3) + 2 \cdot 15,21D]E$ (строка или байт). Вся информация

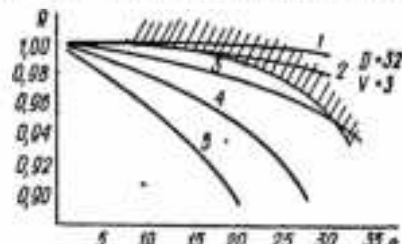
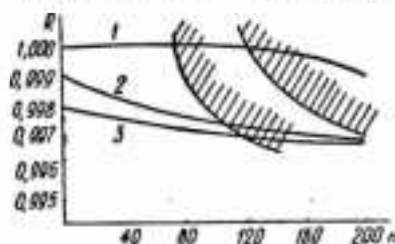


Рис. 1. Графики зависимости вероятности безотказной работы СГУ с НМД: 1 — $F = 130$; 2 — $F = 84$; 3 — $F = 64$

Рис. 2. Графики зависимости вероятности безотказной работы СГУ с НМЛ при максимальном режиме работы НМЛ:

1 — $F = 1000$; 2 — $F = 500$; 3 — $F = 300$; 4 — $F = 200$; 5 — $F = 100$

под УПС на магнитной ленте занимает β строки: $\beta = \alpha l = [A(B+3) + 30,42D]En$. Максимальное время поиска информации на ленте v (с)

$$v = \frac{\beta}{Dv10^3} = \frac{[A(B+3) + 30,42D]En10^{-3}}{Dv},$$

где $\beta/D \cdot 10^3$ — длина, занимаемая всеми УПС на ленте. Среднее время поиска

$$\xi = \frac{v}{2} = \frac{[A(B+3) + 30,42D]En}{2Dv} 10^{-3}.$$

Среднее время передачи одного сегмента

$$\tau = \frac{F}{Dv} 10^{-3} \text{ (с)}.$$

Среднее время обслуживания

$$\begin{aligned} \omega = \xi + \tau &= \left\{ \frac{[A(B+3) + 30,42D]En}{2Dv} + \frac{F}{Dv} \right\} 10^{-3} = \\ &= \frac{[A(B+3) + 30,42D]En + 2F}{2Dv} 10^{-3} \text{ (с)}. \end{aligned}$$

Интенсивность обслуживания

$$M = \frac{1}{w} = \frac{2Dv}{[A(B+3) + 30,42D]En + 2F} 10^3.$$

Загрузка

$$\rho = \frac{\lambda}{M} = \frac{\lambda(B+3)[A(B+3) + 30,42D]En + 2F}{2DvF} 10^{-3},$$

при этом предполагается, что интенсивность поступления запросов от одного станка λ такая же как в СГУ, исследование которой было проведено в первой части данной работы, т. е. $\lambda = 0,1$ (запрос/с). Используя конкретные значения для λ, B, E, ρ , можно переписать

$$\begin{aligned} \rho &= \frac{24 [314800 + 30,42D]n + 2F}{2DvF} 10^{-4} = \\ &= \frac{2,4 [17200 + 45,63D]n + F}{DvF} 10^{-3}. \end{aligned}$$

Основной расчет параметров будем проводить по вероятности безотказной работы, имеющей вид

$$R = \frac{1}{1+\rho} \cdot \frac{\sum_{i=0}^{n-1} \frac{\rho^i}{(1+\rho)^i (n-i-1)!}}{\sum_{i=0}^{n-1} \frac{\rho^i}{(n-i-1)!}}. \quad (3)$$

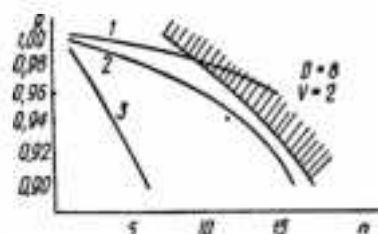


Рис. 3. Графики зависимости вероятности безотказной работы СГУ с НМЛ при минимальном режиме работы НМЛ:

1 — $F = 1000$; 2 — $F = 400$; 3 — $F = 100$

Из условия, что быстродействие НМЛ намного меньше быстродействия НМД, можно предположить, что не стоит ожидать возможности подключения к системе большого количества станков. Поэтому для расчета возьмем случай, когда двухуровневая система группового управления с НМЛ реализована на одном УВК, а в качестве ограничения возьмем условие, аналогичное условию (2): $Fn < 10\,000$. Графики зависимости вероятности без-

отказной работы для СГУ с НМЛ в случае максимальных параметров (т. е. скорость считывания равна 3 м/с, плотность записи информации — 32 бит/мм) приведены на рис. 2. Графики зависимости вероятности безотказной работы для СГУ с НМЛ в случае минимальных параметров работы НМЛ (т. е. скорость считывания — 2 м/с, а плотность записи информации — 8 бит/мм) приведены на рис. 3. Заштрихованная область соответствует условию (2). Из графиков видно, что минимальный режим работы НМЛ не подходит для организации СГУ. В случае максимального режима работы НМЛ к системе ориентировочно можно подключить 15—20 станков, что является

явно недостаточным при проектировании крупного цеха, заявленного в общую систему управления.

В заключение отметим, что при проектировании систем группового управления с большим количеством станков, подсоединенных к ЭВМ, необходимо использовать внешнюю память для хранения управляющих программ станков, а саму систему реализовывать по двухуровневому принципу. В качестве внешней памяти наиболее целесообразно использовать магнитные диски, удовлетворяющие требованиям как по объему хранимой информации, так и по быстродействию.

Поступила в редколлегию 12.12.83.

УДК 621—34

*А. Н. ГВОЗДИНСКИЙ, канд. техн. наук,
В. П. МОДНОВ, канд. техн. наук,
В. Н. КОБЗЕВ, канд. техн. наук*

РЕАЛИЗАЦИЯ ЧЕЛОВЕКО-МАШИННЫХ ПРОЦЕДУР ПОИСКА СТРУКТУРЫ ПРОЕКТИРУЕМОГО ИЗДЕЛИЯ

Процедура поиска структуры проектируемого изделия рассматривается в системе технической подготовки производства (ТПП), состоящей из подсистем конструкторской (КП), технологической (ТП) и организационной (ОП). Эта процедура представляется в виде процесса «горизонтальной» координации в иерархической структуре управления ТПП как способ ликвидации конфликта между подсистемами одного уровня на основе сотрудничества или взаимного обмена информацией.

Обмен информацией между подсистемами КП, ТП, ОП приводит к выполнению следующей задачи: создать изделие заданного качества, в заданные сроки с большей вероятностью, чем без такого обмена, поскольку с получением новой информации расширяется тезаурус. Следовательно, факторы, воздействующие на принятие решения и казавшиеся ранее неясными, случайными, становятся определенными. Такая трактовка взаимодействия подсистем одного уровня относится к результатам горизонтальной координации подсистем, но не затрагивает процесса взаимодействия, т. е. механизма горизонтальной координации.

Использование для этой цели адаптивного подхода [1] и эвристической процедуры [2] не дает возможности описать процесс горизонтальной координации при ТПП, направленной на получение оптимальной структуры изделия $X = X_{доп}$. При адаптивном подходе функции цели $J(x)$ и ограничения $g(x)$, как правило, не могут быть заданы в количественном виде.

Пространство структуры изделия имеет в качестве координат не только количественные, но и качественные признаки.

Так, на стадии разработки технического предложения структура изделия определяется следующими показателями: x_1 — предпосылки членения и компоновки конструкции; x_2 — предпосылки использования стандартных или унифицированных вариантов; x_3 — предпосылки использования в процессе производства типовых технологических процессов; x_4 — использования новых или малознакомых процессов; x_5 — предпосылки использования существующих средств технического оснащения; x_6 — возможность ограничения номенклатуры конструктивных материалов.

Такие координаты пространства X не позволяют выразить их в форме числа, а следовательно, не могут быть получены и функции J, g , определенные на X .

Эвристический подход в описании горизонтальной координации не использует возможности, которые возникают при решении задач математического программирования. Это касается целенаправленного взаимодействия подсистем. В случае эвристического программирования при осуществлении преобразования происходит реализация метода случайного поиска или перебора, и здесь большое значение имеет элемент творческого озарения. Реализация ненаправленных эвристических процедур требует слишком много времени и не гарантирует успеха.

Выбор желаемой структуры изделия осуществим с помощью человеко-машинной процедуры, использующей элементы эвристического и адаптивного подходов, которые реализуем следующим образом.

На множестве всех структур изделия (допустимых и недопустимых) зададим функции $f(x)$ такие, что $f^T = f_1, f_2, \dots, f_l(x)$ — частные показатели (совокупность которых F_0 характеризует структуру проектируемого изделия: трудоемкость изготовления, уровень технологичности конструкции по изготовлению, уровень технологичности конструкции по себестоимости и т. д. [3]).

Достоверно, что каждый из s показателей (критериев) выражается в виде функции f_i ($i = \overline{1, s}$) от n управляемых или неуправляемых переменных $x \in X$ признаков структуры конструкции. Примерами таких функций являются удельная трудоемкость изделия, технологическая себестоимость и другие показатели технологичности.

Обычно предполагают, что существует некоторая функция полезности $U(f)$, такая, что $U(f_1) > U(f_2)$ тогда и только тогда, когда ЛПР предпочитает значение f_1 значению f_2 [4].

Поэтому наша задача формулируется так: найти $\max_{x \in X'} U(f(x))$,

где X' — допустимое подмножество X .

Существование векторного показателя допустимо, однако в связи со сложностью и многомерностью задачи система $\{p_1, p_2, p_3\}$ не в состоянии задать явным образом функцию U .

Представим, что подсистемы ТП (П2) и ОП (П3) — источники информации f_2 и f_3 на какой-то стадии проектирования. Эта информация, а также информация координатора K (B_1, B_2, B_3) является исходной для подсистемы КП (П1) при определении некоторой структуры x^* , удовлетворяющей заданным ограничениям. Предположим, что координатор K в состоянии задать некоторый приемлемый уровень C_i каждого из технологических критериев f_i . Этот выбор производится таким образом, чтобы увеличение f_i сверх C_i не вызвало приращения полезности ($\partial U / \partial f_i = 0$ при $f_i > C_i$).

Полагаем также, что система $\{P_1, P_2, P_3\}$ в состоянии дать оценку весов отклонения f_i от C_i . Это дает нам право ввести некоторую меру относительной важности каждого из критериев w_i при его значении [4]: $w_i = \frac{\partial U / \partial f_i}{\partial U / \partial f_i}$, $i = \overline{1, s}$, где f_i — произвольно выбранный опорный критерий $i = \overline{1, s}$.

Допустим, что $\{P_1, P_2, P_3\}$ в состоянии подобрать такое приращение Δf_i , которое будет полностью компенсировано на единицу значения другого критерия, т. е. можно подобрать такое Δf_i , что для ЛПР (система $\{P_1, P_2, P_3\}$) будут безразличны комбинации значений критериев $(f_1, f_2, \dots, f_i)$ и $(f_1, f_2, \dots, f_i + \Delta f_i, \dots, f_i - 1, \dots, f_s)$, где f_i — значение опорного критерия. Тогда

$$w_i = \frac{1}{\Delta f_i} \approx \frac{\partial U / \partial f_i}{\partial U / \partial f_i}.$$

В общем случае размеры весов будут зависеть от значений $f_1, \dots, f_s$. В точке $\bar{f} = (\bar{f}_1, \bar{f}_2, \dots, \bar{f}_s)$ вектор весовых коэффициентов $w^T = (r_1, r_2, \dots, r_s)$. Здесь

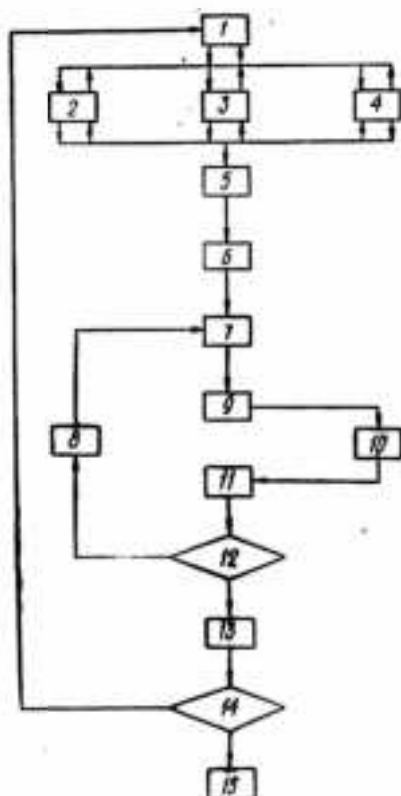
$$r_i \triangleq \left[\frac{\partial U(\bar{f} / \partial f_i)}{\partial U / \partial f_i} \right], \quad i = 2, \dots, s$$

означает некий коэффициент замещения между критерием f_i и опорным критерием f_1 в точке $\bar{f}$. Приближенные коэффициенты замещения могут быть описаны несколькими способами (подобно, например, описанным в [5]).

Исходя из изложенного человеко-машинная процедура горизонтальной координации при разработке конструкции изделия изображается в виде блок-схемы (рисунок).

Подсистемы ТПП совместно с K формируют требования, которым должно удовлетворять проектируемое изделие. Эти требования устанавливают области в пространстве критериев $F = (F_1 F_2 F_3)$, где $F_1 F_2 F_3$ — требования подсистем к проектируемому изделию.

Задача проектирования изделия заключается в формировании такой допустимой структуры x^* , чтобы вектор качества ее в этой точке $f(x^*) = (f_1(x^*), f_2(x^*), \dots, f_s(x^*)) \in F$.



Человеко-машинная процедура горизонтальной координации при разработке конструкции изделия:

1 — задание вектора показателей $F = (F_1, F_2, F_3, F_4)$; 2 — согласование вектора F_1 КП; 3 — согласование вектора F_2 ТП; 4 — согласование вектора F_3 ОП; 5 — $k = 0$; 6 — выбор $f^0 = (f_1^0, f_2^0, \dots, f_n^0) \in F$; 7 — оценка весов критериев

$$w_i^k = \left(\frac{\partial U / \partial f_i}{\sum_{j=1}^n \partial U / \partial f_j} \right)^k$$

f_1^k — опорный критерий; $k = k + 1$; $f^{k+1} = f^k + t^k d^k$; 8 — решение задачи поиска наилучшего направления $y^* = \max \sum_{i=1}^n w_i^k y_i$

9 — $y^k = f^k + d^k$; $f^{k+1} = f^k + t^k d^k$; 10 — определение экспертным методом оптимальной величины шага t^k : $H(f^k + t^k d^k) \rightarrow \max$, $0 \leq t^k \leq 1$; 11 — $H(f^k + t^k d^k) < H(f^k)$; 12 — определение структуры $\{X^k: \varphi(X^k) = -f^k \in F\}$; 13 — $t^k \neq 0$; 14 — утверждение k структуры X^k

Такое требование обеспечивается в течение нескольких итераций, начиная с нулевой, где подсистемой КП выбирается некоторый вариант конструкции $x^0: \varphi(x^0) = f^0 \in F$. Для каждой k -й итерации производится оценка весов критериев

$$w_i^k = \frac{\partial U / \partial f_i}{\sum_{j=1}^n \partial U / \partial f_j}$$

Здесь U — функция полезности $U = U(f_1, f_2, \dots, f_n)$.

В качестве опорного критерия выбран f_1 [4].

Поскольку, как уже отмечалось, функция U явным образом не выражена через вектор критериев f , она оценивается совместно экспертами — представителями КП, ТП, ОП.

Поиск наилучшего направления в пространстве критериев осуществляется решением на ЭВМ задачи математического программирования. Результатом решения этой задачи $\sum_{i=1}^n w_i^k y_i \rightarrow$

$\rightarrow \max$ является y^* — наилучшее направление в пространстве $f = (f_1, f_2, \dots, f_n)$, причем $y^* \in F$. Далее определяется разность $y^* - f^k = d^k$, $f^k + t d^k = y^*$, $0 \leq t \leq 1$, где значение t выбирается таким, чтобы (по мнению экспертов) максимизировать функцию полезности в направлении точки y^* через f^k . Тем самым максимизируется функция полезности $U(f^k + t^k d^k)$.

В результате сравнения $U(f^k + t^k d^k)$ и $U(f^k)$ (U получена к k -й итерации) в случае, если $U(f^k + t^k d^k) \leq U(f^k)$, КП генерирует структуру: $x = \varphi^{-1}(f^k)$. Если $U(f^k + t^k d^k) > U(f^k)$, итеративная процедура продолжается ($k = k + 1$ и $f^{k+1} = f^k +$

$+t^k d^k$). Если $x = \emptyset$, необходимо обращение к координатору K для изменения вектора ограничений.

Проиллюстрируем применение принципов горизонтальной координации на примере разработки гибочных прессов [6].

Задаана область значений нескольких критериев $F = (F_1, F_2, F_3, F_4)$, удовлетворяющих желательной структуре гибочного пресса (подсистемы КП, ТП, ОП совместно с F задают области значений критериев изделия — таблица). Эта область состоит из подобластей, задаваемых: F_1 — координатором K с критерием f_1 ; F_2 — подсистемой КП с критериями f_1, f_2, f_3, f_4 ; F_3 — подсистемой ТП с критериями f_5, f_6 ; F_4 — подсистемой ОП с критериями f_6, f_7 .

№	Критерий [f_i]	Область значений критериев F_1, F_2, F_3, F_4
1	Толщина материала, мм	2,5—3,5
2	Размер деталей (макс), мм	1500
3	Угол гибки, град	10—150
4	Конфигурация (по шкале сложности)	1—0,8
5	Тип производства (по шкале)	4—5
6	Габариты	по стандартной станине
7	Тип, мощность двигателя, кВт	1,5
8	Окупаемость, г	1
9	Коэффициент технологичности	0,8

Примечание. Критерии задаются соответственно: f_1, f_2, f_3, f_4 — подсистемой КП; f_5, f_6 — подсистемой ТП; f_6, f_7 — подсистемой ОП; f_8 — K .

Область F формируется в результате взаимодействия подсистем и координатора K .

Подсистемой КП с ТП и ОП выбирается некоторая точка $f^0 = (f_1^0, f_2^0, \dots, f_n^0) \in F$ — первое приближение в пространстве критериев f . Оцениваются веса критериев

$$w_i^0 = \frac{\partial U / \partial f_i}{\partial U / \partial f_8}$$

по опорному критерию, заданному K — f_8 (срок окупаемости).

Изменяя (ухудшая) этот критерий на единицу ($f_8^0 = 2$ года), находим вектор весов критериев (таблица).

Определяем y^* как решение задачи $\max \sum w_i^0 y_i$, т. е. определяем направление y^* , в сторону которого функция U изменяется наиболее быстро.

Найдено значение шага $t^0: U(f^0 + t^0 d^0) \rightarrow \max$, равное единице ($t^0 = 1,0$). Установлено, что условие $U(f^0 + t^0 d^0) \leq U(f^0)$ не выполняется. Произведена новая оценка весов критериев.

Вновь решается задача поиска наилучшего направления, определяется $t^1 = 0,6$ с помощью одномерной процедуры $U(f^1 + t^1 d^1) \rightarrow \max$. Условие $U(f^1 + t^1 d^1) \leq U(f^1)$ также выполняется.

Далее подсистема КП производит определение структуры $x^0: \varphi(x^0) = f^0 \in F$.

В нашем случае подходящей структуры пресса найти не удалось из-за того, что не удовлетворены первоначальные условия (критерии 1, 2, 4 — таблица). Происходит обращение к координатору K для изменения ограничений. Было увеличено ограничение 8, а также 7 (таблица), что повлекло немедленно к изменению неудовлетворенных прежде критериев 1, 2, 4.

Следующая итерация привела к тому, что для получения точки в пространстве критериев $\bar{f} \in F$, такой, что $U(f) \rightarrow \max$, и возможной структуры изделия $x: \varphi^{-1}(x) = \bar{f} \in F$ координатор вынужден принять решение о конструировании двух прессов, параметры которых покрывают значения первоначальных требований.

Описанная процедура позволяет рационально использовать ресурс, целевое назначение которого — разработка и изготовление специального технологического оборудования, предназначенного для развития производства, параметры которого удовлетворяют требованиям подсистем ТПП.

Список литературы: 1. Цыпкин Я. З. Адаптация и обучение в автоматических системах. — М.: Наука, 1968. — 399 с. 2. Ньюэлл А., Саймон Г. R-программа, моделирующая процесс человеческого мышления. — В кн.: Вычислительные машины и мышление. М, 1967. — 413 с. 3. Методика отработки конструкций на технологичность и оценки уровня технологичности изделий машиностроения и приборостроения. — М.: Изд-во стандартов, 1975. — 56 с. 4. Джоффрион А., Дайер Ж., Файнберг А. Решение задач оптимизации при многих критериях на основе человеко-машинных процедур. — В кн.: Вопросы анализа и процедуры принятия решений / Под ред. И. Ф. Шехнова. — М.: Мир, 1976. — 280 с. 5. Моднов В. П., Бек В. Н. Специальное технологическое оборудование для обработки листового материала. — Обзор по межотрасл. тематике, 1970, вып. 5/71—70, с. 84.

Поступило в редакцию 15.03.83.

УДК 668.011.56.019.3

А. Т. АШЕРОВ, канд. техн. наук

КОНЦЕПЦИЯ НАДЕЖНОСТИ АСУП (ЭЛЕМЕНТЫ РАСЧЕТА, СВОЙСТВА НАДЕЖНОСТИ, ОТКАЗЫ)

Исходные предпосылки. Решение ряда важных вопросов проектирования, производства и эксплуатации АСУП невозможно без использования рекомендаций теории надежности. Даже действующая методика определения научно-технического уровня автоматизированных систем управления предприятиями и производственными объединениями обязывает при защите проектов любых АСУ определять показатель устойчи-

ности функционирования системы при отказе отдельных элементов. Однако существующие традиционные методы анализа и оценки надежности изделий, а соответственно подходы и термины, ориентированные на простые изделия, не могут в полной мере удовлетворять потребности анализа организационных систем. ГОСТы по терминам систем «человек — машина», по основным положениям надежности АСУ ТП, по основным положениям и терминам АСУ, а также ГОСТы системы технической документации АСУ не содержат необходимые термины и определения надежности АСУП, номенклатуру рекомендуемых показателей, состав элементов расчета надежности и т. п. Вместе с тем опубликован в последнее пятилетие ряд работ, свидетельствующих о становлении информационно-надежного подхода к АСУП. Таким образом, назрела необходимость осуществить теоретическое обобщение разрозненных исследований в рамках определенной концепции надежности АСУП.

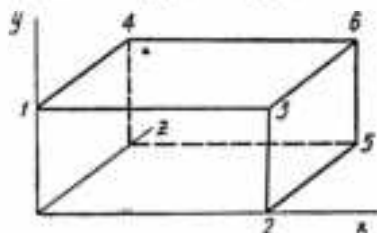
Понятие надежности АСУП. Надежность АСУП как многоаспективное понятие — это комплексное свойство, включающее безотказность, восстанавливаемость, сохраняемость и долговечность. Ввиду необходимости различать в работоспособном состоянии состояния правильного и неправильного функционирования безотказность рассматривается тоже как комплексное свойство, включающее структурную и функциональную безотказность. Понятие структурной безотказности связано с работоспособностью и тождественно безотказности в технике [1]. Функциональная безотказность связана с правильным функционированием в работоспособном состоянии и включает свойства безошибочности и своевременности. Свойство восстанавливаемости является обобщением свойства ремонтпригодности (в технике).

Надежность АСУП как многоуровневое понятие, учитывающее в структуре АСУП наличие функциональной и обеспечивающей частей, — это комплекс надежных свойств обеспечивающей части и персонала, проявляемых в надежных свойствах информации (безошибочности и своевременности).

Надежность АСУП как категория — это свойство АСУП обеспечивать в условиях случайных возмущений достижение заданного результата функционирования, заключающегося в решении задач оперативного управления, учета, анализа и планирования показателей к установленным срокам с определенным уровнем безошибочности выходной информации.

Элементы расчета надежности. Введем понятия структурных и функциональных элементов АСУП. Структурными элементами (С-элементами) являются специально обученный персонал (человек) и элементы обеспечивающей части. Персонал рассматривается как субъект труда в технологических процессах обработки информации (ТПОИ). Элементами обеспечивающей части являются техническое, математическое, програм-

мное, информационное, лингвистическое, правовое и организационное обеспечения. Организационное обеспечение принимается в объеме, определенном ГОСТ 24.209—80 [2], правовое — по [3], остальные обеспечения — по ГОСТ 19675—74 [4]. Персонал во многих случаях удобно рассматривать совместно с техническим, информационным, программным обеспечением и носителями информации в виде эрготехнического комплекса (ЭК). Такой комплекс является объектом переменной структуры, зависящей от тех функций (задач), относительно которых рассматривается ЭК. В зависимости от сочетания субъектов труда и орудий труда элементами ЭК являются технические,



Пространство выделения элементов расчета надежности АСУ

эргатические (человеко-технические) и человеческие элементы.

Функциональными элементами (Ф-элементами) являются: операции ТПОИ, задачи планирования, учета и анализа показателей АСУП, комплексы задач и функциональные подсистемы.

Элементами расчета надежности (РН) называются функционально обособленные части системы, для которых целесообразно

выполнить расчет показателей надежности. Выделение элементов РН проводится через С-элементы и Ф-элементы. При этом элементы обеспечивающей части принимают в объеме, необходимом для реализации Ф-элементов. Объем и наименование элементов РН определяется объемами Ф, С-элементов.

Элементы РН делятся на основные и вспомогательные. К основным относятся задачи и подсистемы с соответствующими обеспечивающими частями и персоналом. К вспомогательным — С-элементы и их сочетания, отнесенные к определенным Ф-элементам. Показатели надежности основных элементов служат для определения соответствия элементов АСУП задаваемым требованиям по надежности. Вспомогательные элементы служат для более полной характеристики АСУП и используются при оценке надежности отдельных элементов обеспечивающей части на этапах проектирования и испытаний.

Способ выделения элементов РН удобно представить в трехмерном пространстве, измерениями которого являются С-элементы (ось x), Ф-элементы (ось y) и свойства надежности (ось z) (рисунок). Точка 1 соответствует некоторому Ф-элементу; точка 2 — С-элементу; точка 3 — элементу РН; точка 4 — надежности основного элемента, обеспечиваемой всеми необходимыми С-элементами; точка 5 — надежности вспомогательного элемента, участвующего в реализации всех Ф-элементов; точка 6 — надежности элемента РН.

Свойства надежности АСУ. Наличие свойств надежности у элементов РН отражено в таблице. Дадим определения надежности элементам обеспечивающей части АСУП. Надежность технического обеспечения определяется традиционно как надежность в технике [1]. Надежность программного обеспечения (ПО) — комплексное свойство, включающее функциональную безотказность (безошибочность), долговечность, восстанавливаемость и сохраняемость. Указанными свойствами обладает в отдельности или в совокупности также любая часть ПО. Функциональная безотказность ПО — это свойство соответствовать функциональным спецификациям, т. е. способность стабильно вырабатывать желаемые результаты до тех пор, пока входные данные удовлетворяют спецификациям (условиям, требованиям). Отказ ПО (программная ошибка) имеет смысл конструкционного отказа. Долговечность ПО заключается в сохранении работоспособности до наступления предельного состояния. Предельное состояние — это такое состояние ПО, при котором его

Элементы РН	Учитываемые свойства надежности					
	Безотказность			Восста- навлива- емость	Долговеч- ность	Сохраня- емость
	Струк- турная	Функциональная				
		безошиб- очность	согласре- менность			
Подсистема	+	+	+	+	+	+
Функциональная задача управления		+	+	+	+	+
Функция (простая, сложная, составная)	+	+	+	+	+	+
Операция технологического процесса обработки информации		+	+	+		
Человек как субъект труда в технологических процессах обработки информации		+	+	+		
Эрготехнический комплекс или эргатический элемент	+	+	+	+	+	
Техническое обеспечение или его часть	+			+	+	+
Программное обеспечение или его часть		+		+	+	+
Информационное обеспечение или его часть		+		+	+	+
Математическое обеспечение или его часть		+		+	+	
Лингвистическое обеспечение или его часть		+		+	+	
Правовое обеспечение или его часть		+		+	+	

дальнейшая эксплуатация должна быть прекращена из-за неустраняемого снижения эффективности эксплуатации ниже допустимой. Предельное состояние наступает, как правило, при изменении внешних условий и связанного в ними математического обеспечения. Надежность ПО обеспечивается на стадии рабочего проектирования и поддерживается в процессе эксплуатации.

Надежность математического обеспечения (МО) — комплексное свойство, включающее функциональную безотказность (безошибочность), восстанавливаемость и долговечность. Функциональная безотказность МО — это свойство математических моделей и алгоритмов решения задач непрерывно описывать закономерности производственной, финансовой и хозяйственной деятельности с заданным уровнем адекватности в течение некоторого времени или некоторой наработки. Сохранение работоспособности МО заключается в способности математических моделей и алгоритмов реализовывать цели и задачи АСУП. Предельное состояние МО наступает по причине изменений в организации производства или организации труда. Надежность МО обеспечивается на стадии технического проектирования АСУП.

Надежность информационного обеспечения (ИО) — комплексное свойство, включающее функциональную безотказность (безошибочность), долговечность, восстанавливаемость и сохранность. Функциональная безотказность ИО — это свойство непрерывно способствовать реализации математических методов и алгоритмов решения задач и обработки информации в соответствии с функциональными спецификациями в течение некоторого времени или некоторой наработки. Сохраняемость ИО заключается в поддержании актуальности, обеспечении сохранности и защите данных от несанкционированного доступа. Актуальность данных в базе — это свойство информационной базы (массивов информации) непрерывно сохранять во времени способность адекватно отражать состояние объекта управления в заданных пределах. Поддержание актуальности данных достигается путем процедур обновления информации, т. е. путем добавления, исключения, изменения данных. Обеспечение сохранности данных в базе состоит в физической сохранности записей, в защите данных при машинных и программных сбоях, ошибках персонала и пользователей. Целями защиты данных от несанкционированного доступа является обеспечение секретности данных и защита их целостности с позиции полноты, согласованности и правильности. Сохранение работоспособности ИО заключается в способности систем классификации и кодирования технико-экономической информации, систем документации и массивов информации реализовывать МО АСУП. Надежность ИО обеспечивается на стадии технического и рабочего проектирования и поддерживается в процессе эксплуатации.

Надежность лингвистического обеспечения (ЛО) — комплексное свойство, включающее функциональную безотказность (безошибочность), восстанавливаемость и долговечность. Функциональная безотказность ЛО — это способность выбранных научно-технических терминов и других языковых средств, используемых в АСУП, а также правил формализации естественного языка непрерывно обеспечивать эффективную машинную обработку информации и общение человека с машиной. Надежность ЛО обеспечивается на стадии технического проектирования.

Надежность правового обеспечения (ПрО) — комплексное свойство, включающее функциональную безотказность (безошибочность), восстанавливаемость и долговечность. Функциональная безотказность ПрО — свойство непрерывно обеспечивать правовое регулирование функциональной и обеспечивающей частей АСУП, регулирование правового положения органов управления в условиях АСУП, совершенствование их организационной структуры, регулирование правового статуса и процесса создания АСУП в течение некоторого времени или некоторой наработки, сохранения во времени законодательную силу своих элементов в заданных условиях советского права. Сохранение работоспособности ПрО заключается в способности принятых нормативных актов и мероприятий организационно-правового и правоприменительного характера обеспечивать правовое регулирование функциональной и обеспечивающей частей системы, регулирование правового положения органов управления в условиях АСУ, совершенствование их организационной структуры, регулирование правового статуса и процесса создания АСУП. Надежность ПрО обеспечивается на стадиях технического и рабочего проектирования.

Состояние элементов АСУ. Различаются состояния С-элементов, Ф-элементов, элементов РН и АСУП. Для технических элементов и технического обеспечения виды состояний определены в ГОСТ 13377—75 [1]. Для человеческих и эргатических элементов, а также для ЭК появляются новые состояния. Наряду с исправным (неисправным) и работоспособным (неработоспособным) состоянием необходимо выделять рабочее, дежурное, правильное и неправильное состояние. Рабочее и дежурное являются разновидностями работоспособного состояния, а правильное и неправильное — разновидностями рабочего состояния. Более полно виды состояний эргатических элементов приведены в [5]. Другие С-элементы, Ф-элементы и элементы РН могут быть в следующих состояниях: работоспособном — неработоспособном, правильном — неправильном, предельном. Предельное состояние проявляется у тех элементов РН, для которых целесообразно учитывать свойство долговечности. Для комплексов задач (подсистем) АСУП и АСУП в целом целесообразно ввести понятие нормального и аномаль-

ного состояния. Нормальное состояние — вид состояния, при котором процесс функционирования обеспечивает реализацию целей АСУП.

Отказы элементов АСУ. Различаются отказы С-элементов, Ф-элементов и элементов РН. Среди Ф-элементов выделяются отказы задач, комплексов задач (подсистем), АСУП в целом. Отказ задачи — это событие, состоящее в нарушении требований по безошибочности или своевременности выходной информации. Отказ задачи может быть полным или частичным. Для комплексов задач (подсистем) и АСУП в целом могут быть определены только частичные отказы, состоящие в отказе любой задачи, входящей в комплекс задач (подсистему) или АСУП. Для технических элементов отказы определены в [1], для человеческих и эргатических элементов — в [5]. Для ЭК в связи с введением понятия «правильного состояния» поднимается необходимость отдельно рассматривать структурные (С) и функциональные (Ф) отказы, как это предложено делать в [6]. С — отказ заключается в потере работоспособности и относится к техническому обеспечению АСУП и его элементам, Ф-отказ — в нарушении правильности функционирования при условии сохранения работоспособности. С-отказ и Ф-отказ совместно формируют отказ ЭК как событие, заключающееся в невыполнении ЭК данной операции (задачи) к установленному сроку или выполнении с недопустимым уровнем безошибочности выходной информации. Специфическими видами отказа являются ошибки: ошибка деятельности человека в ЭК и в процессе проектирования обеспечивающей части и информационная ошибка — искажение сообщения, меняющее его смысл или численную характеристику.

Список литературы: 1. ГОСТ 13377—75. Надежность в технике, термины и определения. — Введ. 01.01.75. 2. ГОСТ 24.209—80. Система технической документации на АСУ. Требования к содержанию документов по организационному обеспечению. — Введ. 01.01.80. 3. Ольшанский А. Г. Правовое обеспечение автоматизированных систем управления. — М.: Юрид. лит., 1979. — 126 с. 4. ГОСТ 19675—74. Автоматизированные системы управления: Основные положения, термины и определения. — Введ. 01.01.75. 5. Губинский А. И., Гречко Ю. П., Еврафов В. Г. Методические рекомендации по терминологии в области надежности эргатических систем (систем «человек—техника»). — Л., 1978. — 24 с. (Препринт ЛЭТИ). 6. Губинский А. И., Еврафов В. Г. Эргономическое проектирование судовых систем управления. — Л.: Судостроение, 1977. — 224 с.

Поступила в редколлегию 10.05.83.

**СТОХАСТИЧЕСКИЙ МЕТОД ОЦЕНКИ ОПТИМАЛЬНЫХ
ВАРИАНТОВ РАЗВИТИЯ ОТРАСЛИ**

Одной из наиболее актуальных задач, решение которых направлено на совершенствование работы хозяйственного механизма, является проблема реконструкции сложившихся к настоящему времени отраслей промышленности. Комплексная задача формирования плана реконструкции должна учитывать ряд взаимно обусловленных критериев и целей, поставленных перед отраслью.

Рассмотрим один из возможных алгоритмов, который может быть использован в процессе составления плана реконструкции отрасли на верхнем уровне планирования. Задача реконструкции отрасли решается на основе вариантного подхода, то есть, при планировании используются подготовительные варианты развития и реконструкции для каждого предприятия отрасли. Вариантный подход позволяет использовать информацию о реальном состоянии отрасли в настоящее время, а также учитывает сложившуюся иерархию управления и планирования отрасли.

При постановке задачи по выбору оптимального плана реконструкции отрасли, производящей товары потребления, необходимо стремиться к тому, чтобы объем производства всей продукции отрасли соответствовал объему спроса (на прогнозируемый период). Далее для каждого предприятия отрасли вариант реконструкции выбирается таким образом, чтобы это приводило к улучшению технико-экономических показателей, характеризующих эффективность работы отрасли в целом. План реконструкции должен быть направлен на более рациональное использование материальных, трудовых, финансовых и других ресурсов.

Сформулированная задача оптимизации может рассматриваться только как одна из составных частей комплексной задачи формирования плана реконструкции отрасли. Определяющее значение имеет проблема выбора целевой функции, которая соответствовала бы программным установкам, поставленным перед отраслью. Принципиальным является также вопрос количественной оценки показателей, характеризующих экономическую эффективность работы отрасли.

Процесс разработки плана реконструкции неизбежно включает в себя ряд этапов, так как происходит многократное согласование вариантов плана развития для каждого предприятия с общими целевыми задачами, поставленными перед

отрасль в целом. На каждом этапе рассматривается вариант реконструкции, составленный на основе соответствующих вариантов развития предприятий. В этом случае алгоритм оптимизации представляет собой инструмент анализа, который позволяет сократить число рассматриваемых вариантов и сразу предложить для анализа на верхнем уровне планирования вариант, наилучший с точки зрения выборной целевой функции. Предполагается также, что в процессе последовательных согласований уточняется понятие целевой функции и выбираются значения весовых коэффициентов, входящих в выражение для целевой функции. Весовые коэффициенты в выражении целевой функции рассматриваются как управляющие параметры. Варьируя значениями параметров, выбираем вариант развития отрасли, удовлетворяющий программным установкам, поставленным перед данной отраслью.

Далее излагается математическая формулировка задачи выбора оптимального варианта реконструкции отрасли. Предполагается, что варианты развития для каждого предприятия на прогнозируемый период подготовлены заранее. Проблема формирования методики подготовки вариантов развития для всех предприятий отрасли требует углубленного технико-экономического анализа их производственной деятельности и выходит за рамки данной работы.

Итак, положим, что для i -го предприятия ($1 \leq i \leq I$) подготовлено N_i — вариантов развития. В целях упрощения индексации положим, что количество подготовительных вариантов для каждого предприятия одинаково и $N_i = N$. В процессе подготовки n -варианта развития ($1 \leq n \leq N$) для i -предприятия определяется прогнозируемый объем выпуска продукции C_i^n . Кроме того, эффективность работы предприятия характеризуется рядом технико-экономических показателей, a_{ik}^n — i -й технико-экономический показатель i -предприятия по n -варианту развития $1 \leq k \leq K$. Предполагается также заданным объем спроса C^0 на продукцию отрасли.

Алгоритм решения задачи оптимизации должен обеспечить выбор для предприятия n^* -варианта развития из числа подготовленных так, чтобы общий объем производимой продукции соответствовал объему спроса C^0 :

$$|C^0 - C| < \Delta C, \text{ где } C = \sum_{i=1}^I C_i^{n^*} \quad (1)$$

при максимуме значения S , характеризующего технико-экономическую эффективность работы отрасли в целом; ΔC — обозначает допустимое отклонение объема производства от объема прогнозируемого спроса C^0 .

Целевая функция S формируется в форме аддитивного критерия

$$S = \sum_{i=1}^I \gamma_i S_i^{n^*}. \quad (2)$$

В выражении (2) S_i^{**} соответствует значениям целевой функции для i -предприятия при использовании n^* варианта развития. Коэффициенты γ_i характеризуют степень приоритетности предприятий. Предприятия, продукция которых пользуется наибольшим спросом и которые имеют лучшие технико-экономические показатели, получают большие значения коэффициентов приоритетности. Предпочтение оказывается также и предприятиям, которые имеют большие возможности для своего развития. Это обуславливается местом расположения предприятия, наличием в данном экономическом районе резервов рабочей силы и соответствующими ресурсами строительной индустрии, необходимыми для осуществления капитального строительства.

В свою очередь целевая функция S_i^{**} , характеризующая работу i -предприятия по n^* варианту, определяется как

$$S_i^{**} = \sum_{k=1}^k \lambda_k a_{ik}^{**}, \quad (3)$$

λ_k в выражении (3) рассматривается как весовые множители, которые соответствуют вкладу различных технико-экономических показателей в оценку эффективности работы предприятий отрасли.

Подобным образом поставленная задача оптимизации целевой функции S при условии (1) относится к классу задач, которые определяются как варианты модели производственного типа. Как правило, для решения вариантных задач рекомендуются методы целочисленного программирования. Однако применение методов дискретной оптимизации для целочисленных задач большой размерности приводит к значительным трудностям в процессе их решения. Использование регулярно метода оптимизации неизбежно требует, чтобы поведение критерия качества удовлетворяло определенным предложениям. В то же время для вариантной модели обычно не удается упорядочить подготовительные варианты для каждого предприятия, так как при переходе от одного варианта к другому критерий качества изменяется нерегулярным образом.

Для задачи оптимизации в вариантной модели может быть применен метод статистических испытаний. В этом случае необходимо решить вопрос о потребном объеме статистического эксперимента для получения по результатам выборки приемлемой оценки значения оптимума целевой функции. Мы предлагаем для оценки оптимума вариантной модели по результатам статистического эксперимента использовать законы распределения крайних членов выборки. Применительно к задачам целочисленного программирования этот метод рассмотрен ранее¹.

¹ Киселев Н. И., Сидоров Н. М. Статистическая оценка глобального экстремума, — АВТ, 1974, № 4, с. 45—49.

В процессе формирования плана реконструкции отрасли использование метода статистических испытаний совместно с оценкой глобального экстремума на основе законов распределения крайних членов выборки является предпочтительным и, по нашему мнению, в наибольшей степени отвечает существу рассматриваемой задачи. Сформированный выше критерий качества, характеризующий работу отрасли, представляет собой определенную формализацию существа целей и программных установок, поставленных перед отраслью. Очевидно, что при любом выборе весовых коэффициентов и коэффициентов приоритетности будет иметь место неполная адекватность задачи оптимизации, соответствующей задаче планирования.

Алгоритм оптимизации может быть использован в процессе подготовки вариантов реконструкции отрасли для рассмотрения на верхнем уровне планирования, на основе использования подготовительных вариантов развития предприятий. Метод статистической оценки глобального экстремума удобен при работе в диалоговом режиме. Подобный метод допускает существенную вариацию постановки задачи оптимизации в процессе формирования плана и уточнения целевой функции, соответствует одному и тому же уровню вероятности, который в определенном смысле можно понимать как уровень надежности оценки. Весовые коэффициенты целевой функции согласно (2), (3) рассматриваются как управляющие параметры, с помощью которых производится предварительный выбор полученных на данном этапе планирования вариантов. Далее эти варианты развития отрасли анализируются на верхнем уровне планирования.

В процессе статистических испытаний из n -подготовленных вариантов развития i -предприятия производится с равной вероятностью выбор n^* -варианта. Далее проверяется условие (1), которое характеризует предполагаемый объем выпуска продукции по всей отрасли. Если это условие не выполняется, то производится выбор следующего варианта до тех пор, пока условие (1) не будет выполнено. При выполнении условия (1) для M варианта развития отрасли в соответствии с заданными значениями технико-экономических показателей a_{ij}^* согласно (2), (3) определяется значение целевой функции S_{ij} .

Таким образом, получим выборку M вариантов плана развития отрасли. Из этих вариантов в качестве решения выбирается вариант, которому соответствует наибольшее значение целевой функции S . Необходимо оценить объем выборки M так, чтобы полученное решение было достаточно близко к глобальному экстремуму. На основе результатов выборки можно оценить значение параметров законов распределения случайной величины S . Для получения достоверной оценки параметров закона распределения требуется располагать выборкой большого объема. Так как для задачи оптимизации необходи-

мо оценить только экстремальное значение, то в этом случае можно использовать законы распределения крайних членов выборки. При этом для оценки экстремума достаточна выборка меньшего объема. Для оценки экстремума вариантной модели предлагается использовать двойное экстремальное распределение (закон первого типа) в виде

$$F(z) = \exp[e^{-\alpha(z-\pi)}], \quad (4)$$

Оценки параметров распределения L на основе результатов статистических испытаний проводятся следующим образом.

Последовательность значений целевой функции S разбивается на m -групп по l -элементов в группе. В каждой из групп определяется максимальное значение S_m .

Оценка параметров α , n проводится по формулам

$$\frac{1}{\alpha} = \frac{T}{T_N}; \quad n = S_0 - J \frac{N}{T},$$

где

$$S_0 = \sum \frac{S_i}{N}; \quad T = \sqrt{\frac{\sum (S - S_0)^2}{N-1}} \quad (5)$$

T — определяется по таблицам, приведенным ранее (см. сноску).

Оценка оптимума задачи получается согласно соотношению

$$S_{0n} = n + \frac{J}{L}, \quad (6)$$

J — приведенная переменная для данного вида распределения, соответствующая заданному уровню вероятности P_0 .

Описанная выше методика оценки вариантов развития отрасли использована для разработки плана реконструкции предприятий, производящих музыкальные инструменты. Производство музыкальных инструментов является отраслью промышленности, имеющей около 50 предприятий, на которых производятся различные* по технологии изготовления музыкальные инструменты.

Нами был проведен расчет по описанной выше методике для клавишной группы музыкальных инструментов, насчитывающей 22 предприятия, выпускающих в год около 110 000 штук пианино.

В качестве исходных данных были подготовлены по 5 вариантов производства для каждого предприятия. Каждый из вариантов характеризовался следующими показателями: C_i^n — объем производства i -го предприятия по n -варианту производства; a_{ik}^n — технико-экономическими показателями i -го предприятия по n -му варианту производства, где $i = 1, 2, \dots, 22$; $n = 1, 2, \dots, 5$; $k = 1, 2, \dots, 11$.

Значение показателей C_i^n для всех 22-х предприятий по каждому из 5 вариантов производства приведены в табл. 1.

Таблица 1

Номер предприятия	Номер варианта					
	$n=1$	$n=2$	$n=3$	$n=4$	$n=4$	$n=5$
$i=1$	2 339	1 847	2 067	2 060	2 060	806
$i=2$	2 961	1 019	2 224	2 196	2 196	2 074
$i=3$	2 394	2 300	2 203	2 358	2 358	2 350
$i=4$	2 935	2 619	2 425	2 620	2 620	2 454
$i=5$	2 623	2 533	2 735	2 750	2 750	2 750
$i=6$	1 943	2 239	2 460	2 654	2 654	2 937
$i=7$	3 411	3 213	3 235	3 071	3 071	3 905
$i=8$	3 192	3 022	3 047	3 076	3 076	3 146
$i=9$	3 304	3 320	3 471	3 609	3 609	3 518
$i=10$	3 348	3 243	3 400	3 520	3 520	3 630
$i=11$	3 628	3 615	3 717	3 763	3 763	3 683
$i=12$	3 706	3 782	3 715	3 810	3 810	3 819
$i=13$	3 930	3 876	3 890	4 012	4 012	4 062
$i=14$	400	4 313	4 680	4 842	4 842	4 449
$i=15$	5 510	4 335	4 514	4 514	4 629	4 596
$i=16$	7 005	7 236	7 438	7 436	7 526	7 490
$i=17$	7 050	7 463	7 608	7 608	7 751	7 940
$i=18$	7 705	7 854	8 277	8 277	8 090	7 965
$i=19$	7 986	8 033	8 265	8 265	8 490	8 651
$i=20$	8 951	9 046	9 353	9 353	9 433	9 513
$i=21$	9 570	9 833	9 005	9 005	7 963	10 237
$i=22$	15 023	14 456	14 806	14 806	15 237	14 876

Значения показателей a_{ik}^* не приводятся в данной статье ввиду громоздкости, так как размерность матрицы a_{ik}^* равна $22 \times 11 \times 5$. Значение показателя спроса на всю продукцию клавишной группы музыкальных инструментов равно $C^0 = 109\,200$ штук пианино в год, а допустимое отклонение от этой величины равно $\Delta C = -200$ штук пианино в год.

Значение весовых множителей в формуле (3) λ_k — коэффициентов в приоритетности технико-экономических показателей соответствуют их номерам в табл. 2.

Значение весовых множителей γ_i в формуле (2), называемых коэффициентами предприятий, соответствуют их номерам в табл. 1.

Выбор варианта из пяти для каждого из 22-х предприятий осуществляется случайным образом.

Затем производится сравнение по формуле (1) и, если данный набор вариантов удовлетворяет условию (1), проводится расчет по формулам (3), (2).

Оценка необходимого количества выборок рассчитывается по формуле (4).

В результате расчетов на ЭВМ ЕС-1033 на языке PL получен оптимизированный набор вариантов по одному из пяти для 22-х предприятий, производящих пианино. Этот набор удовлетворяет условиям (1) и дает наибольшую величину целевой функции.

В табл. 2 представлены значения технико-экономических показателей (ТЭП) для предприятий с номером $i = 1$, для которого оказался выбранным вариант производства с номером $л = 2$.

Таблица 2

ТЭП, тыс. р.	Номер варианта				
	л = 1	л = 2	л = 3	л = 4	л = 5
Полная себестоимость товарной продукции	2741	2670	2683	2620	4172
Объем выпуска валовой продукции в оптовых ценах	3393	3193	3279	2838	4617
Производительность труда	6,170	6,017	6,129	5,558	7,100
Среднегодовая стоимость основных ПИИФ	829	945	1036	1032	1077
Средняя заработная плата	0,154	0,152	0,153	0,160	0,172
Фондоотдача	0,00235	0,00239	0,00281	0,00273	0,00429
Фондоёмкость	0,00027	0,00029	0,00031	0,00037	0,00023
Фондовооруженность	2,08	2,19	1,94	2,04	1,66
Прибыль от промышленной деятельности	225	110	194	131	376
Фонд заработной платы	953	957	981	979	1346
Объем реализуемой продукции	2966	2848	2918	2774	4410

Оптимизация для 22-х предприятий, производящих планино, дает возможность удовлетворить спрос на продукцию отрасли — выпустить $C = 109136$ шт. и $\Delta = 200$ шт.

УДК 681.142.2

И. Г. ВИНТИЗЕНКО, Г. А. ГЕРАСИМ, В. М. СЕМИШЕНКО

ПРОСТРАНСТВЕННО-ДЕКОМПОЗИЦИОННЫЙ ЯЗЫК ОПИСАНИЯ ПРИНЦИПИАЛЬНЫХ ЭЛЕКТРИЧЕСКИХ СХЕМ

Основной объем исходных сведений, необходимых для работы систем автоматизированного конструирования печатных плат, содержится в схеме электрической принципиальной и перечне ее элементов. Подготовка исходных данных и ввод их в ЭВМ — наиболее трудоемкий этап при конструировании печатных плат в автоматизированной системе. Именно на этом этапе в систему вводится наибольшее число ошибок, многие из которых выявляются только после изготовления опытного образца печатной платы.

Существует несколько способов описания топологии схемы и ее элементного состава. Но все эти способы имеют одну особенность, заключающуюся в том, что на первом этапе кодирования необходимо идентифицировать либо все эквипотенциальные узлы схемы, либо эквипотенциальные цепи, либо, определив их неявно, перечислить наименования и номера контактов радиоэлементов, связанных между собой электрически. При таком подходе усложнение кодируемой схемы приводит к непропорциональному увеличению сложности и объема кодирования, особенно при наличии в схеме большого числа групповых проводов («жгутов») и групповых элементов («блоков»). При кодировании приходится обозревать всю схему целиком, а нередко и отмечать цветным карандашом кодируемые соединения.

В статье предлагается новый подход — пространственной декомпозиции электрической схемы, кодирования содержимого фрагмента и контролируемого соединения всех фрагментов в единое описание. Метод пространственной декомпозиции реализован в виде языка KODOS описания схем принципиальных электрических.

В языке KODOS для реализации принципа пространственной декомпозиции на исходную принципиальную схему накладывается сетка из горизонтальных и вертикальных линий, при этом схема разбивается на ячейки.

Ячейка — это прямоугольник (в частности, квадрат), содержащий внутри себя часть схемы (элемент или его часть, несколько элементов, провода, групповые провода и их фрагменты и пр.). Каждая сторона ячейки имеет четыре точки (внешние пуанты), всего их 16. Линии, входящие (выходящие) в ячейку (из ячейки), должны быть однозначно отнесены к той или иной пуанте. Следовательно, каждая сторона ячейки может пропускать через себя до четырех линий, проводимых с некоторым шагом и отображающих электрические связи между элементами [1, 2].

Шаг расположения линий на чертеже определяется как минимальное расстояние между линиями. Как правило, разработчики и конструкторы пользуются миллиметровой бумагой и шагом, равным 5 мм. Ячейка может содержать произвольное число радиоэлементов, аналогично один радиоэлемент может располагаться в нескольких ячейках, может быть разбит на отдельные части, размещенные в разных ячейках, не обязательно смежных.

Задача кодирования принципиальной схемы сводится к описанию внутреннего содержания каждой ячейки и указанию пуант, через которые выводы элементов, провода, групповые провода (жгуты) покидают ячейку. Транслятор языка KODOS объединяет фрагменты электрических цепей из отдельных ячеек в цепи схемы, проводники жгутов — в жгуты схемы, части

элементов с одним именем из разных ячеек — в единые конструктивно-законченные элементы (корпуса), проверяя при этом в сопряженных парах пуант соответствие выходящих и входящих частей электрических цепей разных ячеек.

Приведем формальное металингвистическое описание декомпозиционного языка KODOS:

$\langle \text{пуанта} \rangle ::= \{ | \langle \text{внешняя пуанта} \rangle | \langle \text{внутренняя пуанта} \rangle | \}$
 $\langle \text{внешняя пуанта} \rangle ::= \{ | A | B | C | D | E | F | G | H | K | L | M | N | R | S | T | U | \}$
 $\langle \text{внутренняя пуанта} \rangle ::= X$
 $\langle \text{сопряженные пары пуант} \rangle ::= \{ | K - D | L - C | M - B | R - E | S - F | T - G | N - A | U - H | \}$
 $\langle \text{закодированная принципиальная схема} \rangle ::= \{ | \langle \text{строка} \rangle ; | \} \dots$
 $\langle \text{строка} \rangle ::= \{ | \langle \text{ячейка} \rangle \cdot | \} \dots$
 $\langle \text{ячейка} \rangle ::= \{ | \langle \text{провод} \rangle , | \langle \text{жгут} \rangle , | \langle \text{элемент} \rangle , | \} \dots$
 $\langle \text{провод} \rangle ::= \langle \text{пуанта} \rangle \{ | \langle \text{пуанта} \rangle | \} \dots$
 $\langle \text{жгут} \rangle ::= \langle \text{внешняя пуанта} \rangle \{ | \langle \text{внешняя пуанта} \rangle | \}$
 $\langle \text{ветвь} \rangle \{ | \langle \text{жгут} \rangle | \} \{ | \langle \text{жгут} \rangle \langle \text{ветвь} \rangle \}$
 $\langle \text{ветвь} \rangle ::= \langle \text{номер} \rangle \langle \text{пуанта} \rangle$
 $\langle \text{номер} \rangle ::= \langle \text{натуральное} \rangle$
 $\langle \text{элемент} \rangle ::= \{ | \langle \text{обозначение элемента} \rangle | \langle \text{описание контактов} \rangle \}$
 $\langle \text{обозначение элемента} \rangle ::= \langle \text{имя} \rangle \{ | \langle \text{номер} \rangle | \}$
 $\langle \text{имя} \rangle ::= \{ | \langle \text{буквенный символ} \rangle | \} \dots$
 $\langle \text{описание контактов} \rangle ::= \{ | \langle \text{явное описание контактов} \rangle | \}$
 $\langle \text{явное описание контактов} \rangle ::= \{ | \langle \text{номер} \rangle \langle \text{пуанта} \rangle | \} \dots$
 $\langle \text{неявное описание контактов} \rangle ::= \{ | \langle \text{пуанта} \rangle | + | \} \dots$

Рассмотрим провод, проходящий через некоторую ячейку (фрагмент схемы), один из них разветвляется в ячейке. Провода кодируются перечислением пуант, через которые они проходят: *HK, ANR*.

Порядок перечисления пуант безразличен: *APN, KH*. Еще одним проводом внутренняя пуанта *X* соединяется с внешней *T*, этот провод кодируется: *XT* (можно и *TX*).

Через ячейку проходит жгут, кодирующийся как простой провод парой внешних пуант: *EU*. Но в рассматриваемой ячейке он принимает и выделяет провода-ветви с номерами 8, 3 и 19. Тогда его описание дополняется: *EU3D8X19B* (можно и *EU8X19B3D, ...*).

Жгут, если он не принимает и не выделяет ветвей-проводов, кодируется как фиктивный проводник, проходящий через ячейку, парой внешних пуант. Ветвь-проводник, входящая или выходящая из жгута, кодируется номером провода, стоящим после общего кода группового провода, и пуантой, через которую входит (выходит) провод-ветвь в ячейку. Номер провода жгута задается разработчиком и не требует дополнительной работы со стороны кодировщика. Получая «адреса» жгутов из отдельных

ячеек, система «сшивает» его, считая, что выделенная ветвь-проводник остается в жгуте и может быть выделена еще сколько угодно раз.

В ячейке содержатся два элемента: резистор $R6$ и микросхема $У5$. Микросхема кодируется с явным указанием контактов: $(У5)5F12X8L3G$. Резистор кодируется с неявным указанием контактов: $(R6)MS$. Явное указание контактов постоянного резистора не имеет особого смысла, так как его контакты 1 и 2 разнозначны, но можно кодировать и так: $(R6)1M2S$ или $(R6)1S2M$, или $(R6)SM$.

Неявное указание контактов микросхемы $У5$ привело бы к такой единственно-правильной форме записи: $(5)++G++F++L+++X$, где знак $+$ означает пропущенный в перечислении (неиспользуемый) номер вывода элемента.

Перспективы декомпозиционного метода связаны с развитием средств автоматического считывания схем по частям непосредственно с чертежа. Предложенный способ кодирования позволяет кодировщику (оптическому считывающему устройству) работать одновременно только с фрагментом схемы, ограниченным ячейкой, поэтому объем кодирования определяется только количеством ячеек, на которые разбивается схема.

Достоинство этого способа проявляется, в частности, при внесении изменений в схему. При повторном кодировании изменению подлежит содержание только тех фрагментов (ячеек кодировочного бланка), в которые внесены исправления.

В языке KODOS при кодировании содержания ячейки частично повторяются сведения о связях в ячейках, смежных с кодируемой. Эта избыточность позволяет осуществлять жесткий контроль вводимой информации, что особенно актуально при применении оптических считывателей.

Все графические объекты, используемые при начертании принципиальной схемы (элемент, провод, жгут и пр.), могут быть однозначно описаны средствами языка KODOS, поэтому нет необходимости в предварительном перечерчивании принципиальной схемы. Простота языка позволяет доверить кодирование схем человеку без всякой квалификации.

Транслятор с языка KODOS является подсистемой системы автоматизированного конструирования печатных плат КАПЛ-3, разработанной в Томском институте автоматизированных систем управления и радиоэлектроники и успешно функционирующей на ряде предприятий страны. Система КАПЛ-3 и, естественно, транслятор и язык прошли проверку на примерно 700 студентах, каждый из которых кодировал индивидуальную схему. Не обнаружено синтаксических ошибок, которые прошли бы через систему контроля транслятора языка KODOS, несмотря на полное отсутствие опыта кодирования, общую инфантильность кодировщиков, «оригинальное» и просто невнима-

тельное прочтение инструкций, попытки «обмануть» систему и т. п.

Результаты эксплуатации системы КАРЛ-3 в части кодирования принципиальных схем и их перечней элементов:

1) время кодирования кодировщиков с минимальным опытом схемы средней сложности со 120 элементами, в том числе 40 микросхемами, при 200 цепях — 3 ч;

2) для исправления всех возникающих в процессе кодирования и перфорации (или ввода с дисплея) ошибок достаточно 1—3 подходов к ЭВМ;

3) время работы транслятора на одном подходе — 2—4 мин для ЭВМ ЕС-1022;

4) максимальное количество ячеек, на которые разбивается схема, ограничено только возможностями ЭВМ и при объеме памяти 256 кбайт составляет примерно 1500 ячеек, что можно считать практически неограниченными, так как самая сложная реальная схема требует не более 1000 ячеек;

5) количество кодируемых цепей, элементов, проводов, групповых проводов, групповых элементов, количество типов элементов, количество контактов на одной цепи и т. п. — практически не ограничено.

Список литературы: 1. Кашкан А. А. Система автоматизированного расчета функционирования устройств телевидения САРФУТ-12: Ч. 1. — *Вопр. телевиз. техники*, 1975, вып. 1, с. 211—219. 2. Кашкан А. А. Система автоматизированного расчета функционирования устройств телевидения САРФУТ-12: Ч. 2. — *Вопр. телевиз. техники*, 1977, вып. 2, с. 209—219.

Поступила в редколлегию 25.11.82.

УДК 681.3.06:621.3.049

С. Г. МАЗУР, Н. А. БОНДАРЕНКО, канд. техн. наук

МЕТОД РАСПРЕДЕЛЕНИЯ ПРОВОДНИКОВ В ДВУХСЛОЙНЫХ ПЕЧАТНЫХ ПЛАТАХ

Большое число межслойных переходов (МП) в двухслойных печатных платах, возникающее при использовании методов машинного проектирования печатных плат, отрицательно сказывается на надежности печатных плат, увеличивает их себестоимость и время изготовления. Особенно остро стоит задача уменьшения числа МП при использовании алгоритмов трассировки с ортогональным расслоением [1].

Постановка задачи. Пусть имеется расслоенный печатный монтаж. Вертикальные отрезки размещены на первой стороне платы, а горизонтальные — на другой. В местах, где цепи меняют

свое направление, размещаются МП. Необходимо путем переноса части отрезков с одной стороны платы на другую минимизировать число МП. При этом коммутационные характеристики монтажа должны оставаться неизменными.

Электрический монтаж в соответствии с [2] будем интерпретировать бихроматическим графом $G = (U, R)$. Каждой вершине графа U_i , $i = \overline{1, p}$ соответствует отрезок электрической цепи. Цвет вершины зависит от того, на какой стороне расположен отрезок. Вершины графа могут соединяться ребрами двух типов: R_A и R_B . Две вершины соединяются A -ребром, если соответствующие отрезки имеют общий МП, и B -ребром, если отрезки скрещиваются. Перекраска вершины графа эквивалентна перемещению отрезка с одной стороны платы на другую. Определим правила перекраски вершин графа G следующим образом.

Вершина U_k , которой инцидентны только A -ребра, может быть перекрашена в противоположный цвет. При этом никакая другая вершина не изменяет своего цвета. Вершина U_k , которой инцидентны B -ребра, может быть перекрашена в противоположный цвет. При этом одновременно перекрашиваются все вершины $U_{k1}, U_{k2}, \dots, U_{ki}$, из которых существует путь из B -ребер в вершину U_k .

Правила перекраски вершин графа G , сформулированные выше, гарантируют такое преобразование печатного монтажа, при котором его коммутационные характеристики не изменяются. Если после перекраски две вершины графа окрашены в один цвет и соединены A -ребром, то это означает, что соответствующие отрезки лежат в одном слое и соединены общим межслойным переходом. Такой МП не нужен для электрического соединения этих отрезков и может быть удален.

Решение поставленной задачи в терминах теории графов определяется следующим образом. Необходимо перекрасить вершины бихроматического графа G таким образом, чтобы минимальное число A -ребер соединяло разноокрашенные вершины, а относительно B -ребер граф оставался бихроматическим. Бихроматичность графа относительно B -ребер обеспечивает непересечение скрещивающихся отрезков.

Структура графа. Решение задачи минимизации числа МП состоит в оптимальной окраске графа $G = (U, R)$. Понятно, что существенную роль в оптимальной окраске графа играет его структура, обусловленная топологией печатных соединений, полученных на этапе трассировки.

Граф G является бихроматическим, а следовательно, все его простые циклы имеют четную длину [3]. Заметим, что G в общем случае непланарный и несвязный. Действительно, в соответствии с критерием Понтрягина — Куратовского граф планарен тогда и только тогда, когда он не содержит подграфов, гомеоморфных графам G_5 и $G_{3,3}$ [4]. Граф $G_{3,3}$ — бихро-

матический граф степени три. Монтаж, соответствующий $G_{3,3}$, является типичным фрагментом печатного монтажа платы.

Для анализа структуры графа G воспользуемся некоторыми определениями, введенными в [3]. Под блоком D_i графа понимается односвязный подграф, каждое ребро которого входит в простой цикл. Точкой сочленения U_c называется вершина графа, удаление которой увеличивает число компонент связности. Связный граф с большим числом точек сочленения похож на дерево. Эту черту графа можно оттенить четче, если сопоставить со связным графом соответствующее дерево, у которого множеством вершин служат $\{D_i\} \cup \{U_{c_i}\}$. Две вершины, соответствующие D_i и U_{c_j} , смежны, если $U_{c_j} \in D_i$. Такое дерево называется деревом блоков. Для графа G справедливо следующее утверждение.

Компонентами связности графа G могут быть: деревья, блоки и деревья блоков. Вполне очевидно, что задачу оптимальной окраски можно решать для каждой компоненты связности графа отдельно.

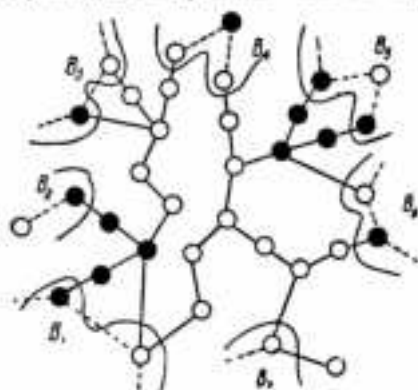
A и B — компоненты связности. Рассмотрим в графе G множество вершин, объединенных A -ребрами. Назовем A -компонентой частичный односвязный подграф графа G , содержащий только A -ребра. Аналогичным образом, для вершин графа, объединенных B -ребрами, определим понятие B -компоненты. Физический смысл A -компоненты — электрическая цепь, а B -компоненты — множество взаимно скрещивающихся отрезков, в котором каждый отрезок скрещивается хотя бы с одним отрезком из этого множества. Рассмотрим некоторые свойства A и B -компонент.

Свойство 1. Любая A -компонента A_i графа G является деревом. В графе G часть вершин A и B -компонент общая. Такие вершины назовем граничными. Весьма важно, что вершины A -компоненты, за исключением граничных, могут окрашиваться независимо. Будем понимать под оптимальной окраской A -компонент такую окраску их внутренних вершин, при которой минимальное число A -ребер объединяет разноокрашенные вершины. Частным случаем A -компонент являются A -цепи. Если в A -цепи концевые вершины имеют одинаковую окраску, то такую A -цепь можно окрасить так, что все A -ребра будут соединять одинаково окрашенные вершины. Для этого достаточно все внутренние вершины цепи окрасить в цвет любой концевой вершины. Если концевые вершины имеют разную окраску, то цепь можно окрасить так, что только одно A -ребро будет соединять разноокрашенные вершины.

Свойство 2. Для оптимальной окраски A -цепи необходимо и достаточно все ее внутренние вершины окрасить в цвет любой концевой вершины.

Используем свойство A -цепей, сформулированное выше, для оптимальной окраски A -компонент. Назовем узлами те внутренние вершины A -компоненты, локальная степень которых $r_A(U_i) \geq 3$.

Поскольку в A -компоненте узлы и граничные вершины соединяются A -цепями, то только окраска узлов и граничных вершин определяет число A -ребер, объединяющих разноокрашенные вершины после ее оптимальной окраски. Тогда с A -компонентой A_i можно выполнить следующее преобразование. Заменяем все A -цепи, соединяющие узлы и граничные вершины, A -ребрами. Полученную A -компоненту обозначим через $\bar{A}_i$. A -компоненты A_i и $\bar{A}_i$ с точки зрения оптимальной окраски полностью эквивалентны, поскольку будут содержать одинаковое число ребер, объединяющих разноокрашенные вершины. Ниже приводится алгоритм оптимальной окраски A -компонент. Он состоит из двух этапов. На первом этапе окрашивается $\bar{A}_i$, а на втором — все A -цепи A -компоненты.



Оптимальная окраска A -компоненты

1. Все граничные вершины $U_{m1}, U_{m2}, \dots, U_{mq}$ A -компоненты $\bar{A}_i$ помечаются символом «+». Не помеченные висятые вершины помечаются символом «-».

2. Для каждой вершины U_i , которой смежна только одна не помеченная вершина, вычисляется оценка $E(U_i) = d_{1i} - d_{2i}$, где d_{1i} — число смежных вершин, помеченных символом «+» и

имеющих ту же окраску, что и U_i ; d_{2i} — число смежных вершин, помеченных символом «+» и имеющих противоположную окраску, чем U_i . Если $E(U_i) > 0$, то вершина сохраняет цвет и помечается символом «+». Если $E(U_i) < 0$, то вершина перекрашивается в противоположный цвет и помечается символом «+». Когда $E(U_i) = 0$, вершина U_i сохраняет свой цвет и помечается символом «-». Шаг 2 повторяется до тех пор, пока все вершины A -компоненты $\bar{A}_i$ не будут помечены. Тогда поддеревья, помеченные символом «+», окрашены оптимально.

3. Каждое поддерево, помеченное символом «-», окрашивается снова. Пометки «-» у внутренних вершин $\bar{A}_i$ удаляются и возвращаемся к шагу 2. Процесс окраски $\bar{A}_i$ заканчивается тогда, когда все вершины будут помечены символом «+».

4. Внутренние вершины каждой A -цепи окрашиваются в цвет концевой вершины.

На рисунке приведен пример оптимальной окраски A -компоненты.

Окраска компонент связности. Рассмотрим оптимальную окраску каждой компоненты связности графа G : дерева, блока и дерева блоков.

Дерево T может быть окрашено так, что все его A -ребра будут объединять одинаково окрашенные вершины. Справедливость этого утверждения следует из того, что любая A -компонента, в которой зафиксирован цвет только одной граничной вершины, может быть окрашена так, что все ее A -ребра будут объединять одинаково окрашенные вершины. Тогда при последовательной окраске A и B -компонент дерева T в очередной A -компоненте, в силу ацикличности T , будет зафиксирован цвет только одной граничной вершины. Заметим, что существует ровно две оптимальные окраски дерева.

Рассмотрим оптимальную окраску блока.

Лемма. Вершины любой B -компоненты могут быть окрашены только двумя способами. Причем окраски вершин, соответствующие этим способам, противоположны.

Действительно, какую бы вершину U_* B -компоненты мы не перекрасили, одновременно необходимо перекрасить и все остальные вершины B -компоненты, поскольку, по определению, существует путь из B -ребер (B -путь) из вершины U_* во все остальные вершины B -компоненты. Но любая вершина может быть окрашена только в два цвета, тогда, очевидно, и B -компонента может быть окрашена лишь двумя способами.

Пусть задан блок D . Для D построим граф $G_D = (L, M)$, $|L| = n$, $|M| = m$, в котором вершины L_i , $i = \overline{1, n}$ соответствуют B -компонентам и узлам блока, а ребра M_{ij} , $i, j = \overline{1, n}$ отображают наличие A -цепей четной и (или) нечетной длины, соединяющих B -компоненты (узлы) B_i и B_j . В дальнейшем такой граф будем называть B -графом блока. Каждому ребру B -графа ставится в соответствие два веса t_{ij} и t'_{ij} , $i, j = \overline{1, n}$, где t_{ij} — число A -цепей четной длины, соединяющих i -ю и j -ю B -компоненты (узлы), а t'_{ij} — число A -цепей нечетной длины. Для весов t_{ij} и t'_{ij} , $i, j = \overline{1, n}$ справедливы следующие равенства:

$$t_{ij} = t_{ji}, t'_{ij} = t'_{ji}, i, j = \overline{1, n}.$$

Рассмотрим окраску графа G_D . Поскольку любая B -компонента (узел) может быть окрашена только двумя способами, то каждой вершине графа можно поставить в соответствие два условных цвета. Примем следующее условие. Если никакая вершина B -компонент или узлов не перекрашивалась, то все B -компоненты и узлы блока имеют одинаковую условную окраску, а следовательно, и все вершины B -графа имеют одинаковый цвет. Возьмем две любые B -компоненты (узлы) блока B_i и B_j и перекрасим оптимально все A -цепи, их соединяющие. Если B -компоненты B_i и B_j имеют одинаковую условную окраску, то в силу бихроматичности графа G , A -цепи четной длины имеют одинаково окрашенные концевые вершины, а нечетной — разноокрашенные. Тогда после оптимальной окраски A -цепей, число A -ребер, объединяющих разноокрашенные вершины, равно t'_{ij} — числу нечетных

A -цепей. Аналогично, если B_i и B_j имеют разную условную окраску, то число A -ребер, объединяющих разноокрашенные вершины после оптимальной окраски A -цепей, равно t_{ij} — числу четных A -цепей. Получаем, что в конечном итоге соотношение окрасок B_i и B_j влияет на число A -ребер, объединяющих разноокрашенные вершины. Допустим, что часть B -компонент (узлов) в блоке перекрашена. Тогда число A -ребер, объединяющих разноокрашенные вершины при любом сочетании окрасок B -компонент (узлов), определяется через B -граф блока следующим образом.

Каждому ребру M_{ij} в результате окраски вершин G_B будем приписывать один из двух возможных весов t_{ij} или $\dot{t}_{ij}$ по следующему правилу. Если вершины L_i и L_j окрашены в одинаковый цвет, тогда M_{ij} приписывается вес $\dot{t}_{ij}$, если в разный — то t_{ij} . Общее число A -ребер, объединяющих разноокрашенные вершины в блоке D при любом сочетании окрасок B -компонент (узлов), равно сумме приписанных весов соответствующего окрашенного B -графа. Введем булевы переменные x_i , $i = \overline{1, n}$, имеющие следующий смысл:

$$x_i = \begin{cases} 1, & \text{если никакая вершина } B\text{-компоненты } B_i \\ & \text{не перекрашивалась;} \\ 0 & \text{— в противном случае.} \end{cases}$$

Тогда значение суммы приписанных весов произвольно окрашенного B -графа можно записать так:

$$P = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n r_{ij} (x_i - x_j)^2 + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \dot{t}_{ij}, \quad (*)$$

где $r_{ij} = t_{ij} - \dot{t}_{ij}$. С учетом (*) задачу оптимальной окраски блока можно сформулировать следующим образом.

Окрасить вершины B -графа блока так, чтобы сумма приписанных весов окрашенного графа была минимальна. Найдя вектор $X = (x_1^0, x_2^0, \dots, x_n^0)$, при котором (*) достигает минимума, получим экстремальную окраску B -графа, а следовательно, и всего блока. Таким образом, задача оптимальной окраски блоков относится к классу задач дискретного программирования [5]. Для ее решения могут быть использованы как комбинаторные методы, так и методы целочисленного программирования.

В заключение рассмотрим оптимальную окраску дерева блоков. Очевидно, что блоки $D_{1n}, D_{2n}, \dots, D_{kn}$ в дереве блоков T_d могут иметь только одну общую вершину — точку сочленения. Отсюда следует, что блоки $D_{1n}, D_{2n}, \dots, D_{kn}$ могут окрашиваться независимо, без всяких ограничений в цвет отдельных вершин. После оптимальной окраски блоков часть вершин $U_1, U_2, \dots, U_p$ в дереве T_d получает фиксированный цвет. Оптимальная окраска дерева T_d состоит в оптимальной окраске ее A -компонент. Причем часть граничных вершин A -компонент дерева имеет фиксированную

окраску. Алгоритм оптимальной окраски таких A -компонент был рассмотрен выше. Таким образом, оптимальная окраска дерева блоков сводится к последовательной оптимальной окраске блоков и деревьев, у которых часть вершин имеет фиксированную окраску.

Предложенный в статье подход позволил сформулировать задачу минимизации числа МП как задачу оптимальной окраски бихроматического графа. Исследование структуры графа G дало возможность расчленить общую задачу оптимальной окраски графа на ряд подзадач окраски его компонент связности.

Показано, что структура деревьев допускает такую окраску вершин, при которой все A -ребра объединяют одинаково окрашенные вершины. Это означает, что электрический монтаж, соответствующий дереву, может быть реализован на двухслойной печатной плате без использования МП. Оптимальная окраска блоков является задачей дискретного программирования. Оптимальная окраска дерева блоков сводится к последовательной оптимальной окраске блоков и деревьев.

Описанный в статье подход реализован в отдельной подсистеме САПР ЭВА [2, 6]. Подсистема содержит 25 программных модулей, написанных на языке ПЛ/1. Время минимизации 2—6 мин при числе отрезков до 2000.

Список литературы: 1. Селютин В. А. Машинное конструирование электронных устройств.— М.: Сов. радио, 1977.— 383 с. 2. Комплекс программ для минимизации количества переходных точек печатного монтажа (ППП—ОПТ).— М.: изд. НИИМаш, 1982.— 28 с. 3. Харари Ф. Теория графов.— М.: Мир, 1973.— 300 с. 4. Мелихов А. Н., Берштейн Л. С., Куредчик В. М. Применение графов для проектирования дискретных устройств.— М.: Наука, 1974.— 303 с. 5. Серещенко И. В., Каспишская М. Ф. Модели и методы решения на ЭВМ комбинаторных задач оптимизации.— К.: Наук. думка, 1981.— 287 с. 6. Автоматизированная система проектирования двусторонних плат печатного монтажа (ППП—АСППП).— М.: изд. НИИМаш, 1982.— 60 с.

Поступила в редколлегию 14.06.83.

УДК 681.3.01:51 -

В. А. КАЛАШНИКОВ, В. В. ЯКШИН

ПОСТРОЕНИЕ ПЛОСКОЙ УКЛАДКИ ГРАФА ЭЛЕКТРИЧЕСКОЙ СХЕМЫ

При проектировании БИС к топологическому рисунку соединений предъявляются требования получения плоского изображения схемы либо ее частей, т. е. недопустимости пересечений проводников в пределах одного слоя. Таким образом, эта задача может быть сведена к определению планарности графа схемы либо ее частей [1].

Рассмотрим один из возможных путей решения данной задачи и его программную реализацию, ориентированную на использование в САПР микроэлектронных устройств.

Дан неориентированный простой связный граф G без петель с числом вершин N , представляемый матрицей $A = [a_{ij}]$, $N \times Y$, где Y — максимальная локальная степень вершины в G . Элемент $a(i, j) = n$ ($n \neq 0$), если в графе существует ребро (i, n) . Если G имеет остоновый подграф [2], который является деревом, то этот подграф называется покрывающим или остовным деревом. Дерево, получающееся в результате поиска в глубину на исходном графе, называется ориентированным покрывающим деревом, дуга орграфа $\vec{G}(V, W)$ является обратным ребром, если в ориентированном покрывающем дереве существует простой путь от вершины W к вершине V , т. е. ребра $\vec{G}$, не принадлежащие ориентированному покрывающему дереву, будут обратными ребрами [3].

Предлагаемый в работе алгоритм относится к матричным методам определения планарности графа. Его первая часть посвящена описанию построения $\vec{G}$, получаемого из G в результате поиска в глубину. Структура орграфа $\vec{G}$ проще, чем в исходном, и процесс определения путей и циклов в нем значительно облегчен. Во второй части алгоритма рассматриваются вопросы, связанные с нахождением циклов в $\vec{G}$. Третья часть описывает построение планарного графа путем перемещения обратных ребер, соответствующих ранее найденным циклам.

В качестве примера будем рассматривать фрагмент схемы, изображенный на рис. 1, матрица A которого представлена ниже.

	1	2	3	4	5	6	7
1	2	4	7	8	12	9	∅
2	15	6	3	4	8	7	∅
3	6	2	4	∅	∅	∅	∅
4	3	6	1	5	2	7	8
5	2	6	4	7	∅	∅	∅
6	5	2	3	4	∅	∅	∅
7	4	1	2	5	∅	∅	∅
8	4	1	2	10	9	11	12
9	10	8	1	11	12	∅	∅
10	8	9	11	∅	∅	∅	∅
11	9	10	8	12	∅	∅	∅
12	11	9	8	1	∅	∅	∅

В соответствии с приведенным ниже алгоритмом строится орграф $\vec{G}$.

Пусть V_i — номер вершины, рассматриваемой в настоящий момент времени.

1. Построение начинается с вершины V_i , $i = \overline{1, N}$.

2. Проходя по ребру (V_i, V_j) , $i \neq j$, инцидентному рассматриваемой вершине V_i , происходит переход к следующей вершине V_j .

3. Определяется, является ли V_j новой вершиной или была уже ранее пройдена, т. е. принадлежит ли ребро (V_i, V_j) ориентированному покрывающему дереву или нет. Если вершина ранее не посещалась, то переход к п. 4, иначе — к п. 5.

4. Ребро (V_i, V_j) является дугой в ориентированном покрывающем дереве. Рассматривается новая вершина V_j , т. е. $V_i = V_j$, переход к п. 2.

5. Ребро (V_i, V_j) , по которому был осуществлен переход в V_j , является обратным ребром, так как ведет назад в пройденные ранее вершины. Переход к п. 6.

6. Определяется, имеет ли рассматриваемая вершина еще одно инцидентное ребро, которое не было ранее пройдено, если да, то переход к п. 2, иначе — к п. 7.

7. Осуществляется проверка, полностью ли построен

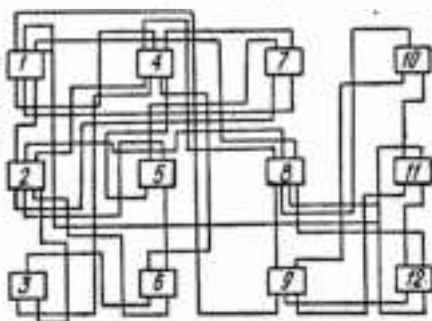


Рис. 1. Фрагмент схемы

орграф $\vec{G}$, т. е. все ли ребра исходного графа G были пройдены во время построения. Если нет, то переход к п. 8, иначе — к п. 9.

8. Происходит возвращение на ранее пройденную вершину V_{i-1} по дуге ориентированного покрывающего дерева, по которой осуществлялся переход в вершину V_i и рассматривается уже вершина V_{i-1} ($V_i = V_{i-1}$). Переход к п. 6.

9. Алгоритм прекращает свою работу. Построение G завершено.

Представленный выше алгоритм отличается от приведенного в [3] тем, что он прекращает свою работу, когда будет пройдено последнее неисследованное ребро графа G , в то время как алгоритм в [3] заканчивается при попытке вернуться назад из вершины, с которой началось построение $\vec{G}$.

Формируется матрица $B = [b_{ij}]$, $N \times Y$, описывающая орграф $\vec{G}$. В ней элемент $b(i, j) = n$ ($n \neq 0$), если дуга (i, n) принадлежит ориентированному покрывающему дереву и $b(i, j) = -n$, если дуга (i, n) является обратным ребром.

Построение начинается с вершины, имеющей номер 1. Элементу матрицы B присваивается значение первого отличного от нуля элемента, стоящего в первой строке матрицы A ($A(1, j) = n$, $n \neq \emptyset$, $B(1, j) = A(1, j)$), и осуществляется переход по ребру $(1, n)$ в вершину n , так как n ранее не посещалась. Происходит формирование последовательностей: $S(L)$, $L = \overline{1, N}$, в которую

входят номера пройденных вершин, и $C(L)$, $L = \overline{1, N}$, в которую заносятся номера столбцов, где стоят просмотренные вершины в A . $S(L)$ и $C(L)$ используются при необходимости возвращения назад на пройденную вершину и продолжения исследования ребер, инцидентных ей. Далее просматривается строка матрицы A с номером, равным значению новой, еще неисследованной вершины ($I = A(1, J) = n$), и выбирается новый, первый отличный от нуля элемент в ней ($A(n, J) = K$, $K \neq \emptyset$). Иными словами, происходит переход по ребру (n, K) . Определяется, была ли вершина K пройдена ранее. Для этого необходимо определить наличие в строке матрицы B с номером K элемента, значение которого равно номеру строки матрицы A , где стоит элемент K , взятому со знаком «+» или «-». Если такой элемент есть, то исследуемая вершина K не принадлежит множеству вершин, определяющих обратные ребра. Если же такого элемента нет, но в этой строке матрицы B есть любой положительный элемент, то вершина K была ранее пройдена и принадлежит множеству вершин, определяющих обратные ребра. В этом случае элементу матрицы B присваивается значение элемента K со знаком «-» ($B(n, J) = -K$). Затем осуществляется переход к рассмотрению следующего элемента в этой же строке матрицы A , т. е. $A(n, J+1)$, и процесс повторяется. В противном случае элемент K является новой вершиной в ориентированном покрывающем дереве ($B(n, J) = K$). Первая часть алгоритма заканчивает свою работу, когда будут исследованы все ребра графа G , т. е. построен оргграф $\vec{G}$, представляемый матрицей B . Так, для рассматриваемого примера получен оргграф $\vec{G}$ (рис. 2), матрица B которого приведена ниже.

	1	2	3	4	5	6	7
1	2	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
2	$\emptyset$	5	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
3	$\emptyset$	-2	4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
4	$\emptyset$	-6	-1	-5	-2	7	8
5	$\emptyset$	6	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
6	$\emptyset$	-2	3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
7	$\emptyset$	-1	-2	-5	$\emptyset$	$\emptyset$	$\emptyset$
8	$\emptyset$	-1	-2	10	$\emptyset$	$\emptyset$	$\emptyset$
9	$\emptyset$	-8	-1	11	$\emptyset$	$\emptyset$	$\emptyset$
10	$\emptyset$	9	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
11	$\emptyset$	-10	-8	12	$\emptyset$	$\emptyset$	$\emptyset$
12	$\emptyset$	-9	-8	-1	$\emptyset$	$\emptyset$	$\emptyset$

На рис. 2 обратные ребра обозначены пунктирными линиями, дуги, принадлежащие ориентированному покрывающему дереву, — сплошными.

Вторая часть алгоритма состоит в определении всех циклов

в $\bar{G}$ и построении матрицы D , описывающей их. В строку $D = [d_{Ml}, m_l]$, $M \times N$, где M — число обратных ребер, последовательно входят вершины, образующие цикл в $\bar{G}$. В первом столбце этой матрицы стоят конечные, а в последнем — начальные вершины обратных ребер. В связи с тем, что все циклы в $\bar{G}$ состоят из простой тропы дуг в ориентированном покрывающем дереве и одного обратного ребра, число циклов в орграфе равно числу обратных ребер в нем (M), т. е. числу отрицательных элементов в матрице B . В соответствии с [3] $M = E - N + 1$, где E — число ребер в исходном графе G . Для того чтобы G был планарен, должно выполняться неравенство $E < 3N - 6$ [4]. $\bar{G}E$ складывается из суммы дуг в покрывающем дереве и обратных ребер, следовательно, в планарном орграфе $M < 3N - 6 - N + 1$, т. е. $M < 2N - 5$.

Поиск цикла начинается из конечной вершины V соответствующего ему обратного ребра. Проходя по дуге (V, W) ориентированного покрывающего дерева, посещается вершина W . Если эта вершина не является началом какого-либо обратного ребра, то осуществляется переход по дуге, инцидентной ей в ориентированном покрывающем дереве, и рассматривается следующая вершина и т. д. Если же W есть начало какого-либо обратного ребра, то определяется, является ли его конечная вершина той, с которой начался поиск. Если да, то цикл найден, если нет, то проходится следующая дуга, инцидентная W . Когда все дуги, инцидентные W , пройдены, а цикл не найден, то возникает необходимость вернуться назад по дуге (U, W) , по которой пришли в W , и продолжать двигаться по другой дуге, инцидентной вершине U , до тех пор, пока не встретится обратное ребро, соответствующее искомому циклу.



Рис. 2. Орграф

В строке матрицы B выбирается отрицательный элемент $(B(I, J) = -K)$. Затем осуществляется переход на строку с номером, равным значению отрицательного элемента, взятому со знаком «+» ($I := K$), т. е. начинается поиск цикла с началом в вершине K , которому соответствует обратное ребро (I, K) . В этой строке выбирается положительный элемент $(B(K, J) = n, n > 0)$ и формируется список $Y(LJ)$, куда входят номера столбцов, в которых стоят вершины в матрице B , образующие отдельный цикл. $Y(LJ)$ используется, когда возникает необходимость вернуться из некоторой вершины W по дуге (U, W) в некоторую ранее пройденную вершину U и обнуляется перед нахождением нового цикла. Далее элементу матрицы D присваивается значение номера строки, в которой стоит положительный элемент в B ($B(K, J) = n$, $D(Ml,$

$MJ) = K)$, т. е. в строку D заносится вершина, входящая в искомый цикл.

Осуществляется переход на строку в B с номером, равным значению элемента n , и в ней определяются элементы, отличные от нуля ($B(n, J) = m, m \neq \emptyset$). Это означает, что совершен переход по дуге (n, m) в вершину m . Затем необходимо выяснить, является ли m конечной вершиной какого-либо обратного ребра. Если $m > \emptyset$, то она не является конечной вершиной обратного ребра, посещается и процесс повторяется. Если же $m < \emptyset$, то дуга (n, m) есть обратное ребро. Определяется, будет ли m исходной вершиной, с которой начался поиск данного цикла, т. е. является ли обратное ребро (n, m) замыкающим в искомом цикле. Если значение отрицательного элемента $(-m)$ равно значению первоэлемента, с которого начался поиск данного цикла (а он стоит в той же строке матрицы B , что и этот первоэлемент), то цикл найден, т. е. сформирована строка матрицы D . Затем определяется новый цикл и формируется следующая строка матрицы D как было описано выше. Если же m не является исходной вершиной и обратное ребро не будет замыкающей дугой искомого цикла, то рассматривается следующий за этим отрицательным элементом ($B(n, J) = -m$) элемент в строке матрицы B , т. е. $B(n, J+1) = v$, и процесс повторяется. Вторая часть алгоритма прекращает свою работу после того, как будут найдены все циклы в $\bar{G}$ и полностью сформирована матрица D . Для рассматриваемого в примере орграфа $\bar{G}$ матрица D , описывающая циклы в нем представлена ниже.

	1	2	3	4	5	6	7	8	9	10	11	12
1	2	5	6	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	3
2	6	3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	4
3	1	2	5	6	3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	4
4	5	6	3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	4
5	2	5	6	3	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	4
6	2	5	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	6
7	1	2	5	6	3	4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	7
8	2	5	6	3	4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	7
$D =$ 9	5	6	3	4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	7
10	1	2	5	6	3	4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	8
11	2	5	6	3	4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	8
12	8	10	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	9
13	1	2	5	6	3	4	8	10	$\emptyset$	$\emptyset$	$\emptyset$	9
14	10	9	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	11
15	8	10	9	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	11
16	9	11	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	12
17	8	10	9	11	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	12
18	1	2	5	6	3	4	8	10	9	11	$\emptyset$	12

Третья часть алгоритма заключается в построении планарного графа из орграфа $\bar{G}$. Так как $\bar{G}$ состоит из ориентированного покрывающего дерева и обратных ребер и каждый цикл в нем состоит из тропы ребер в ориентированном покрывающем дереве и одного обратного ребра, то построение планарного графа сводится к размещению обратных ребер в его орграфе $\bar{G}$ без пересечений. Предполагается, что обратные ребра могут располагаться по отношению к покрывающему дереву в двух плоскостях. Одна из них помечается знаком «+», другая — знаком «-». Допускается, что обратное ребро, соответствующее циклу, расположенному в первой строке матрицы D , размещается в «положительной» плоскости. Попытаемся разместить в этой плоскости обратное ребро, соответствующее циклу во второй строке D . Алгоритм размещения двух циклов в одной плоскости без пересечения их обратных ребер приводится ниже.

1. Если в новом, располагаемом в данный момент времени цикле есть вершина, которая является конечной для обратного ребра, соответствующего ранее уложенному циклу, то под первым рассматриваемым циклом будем понимать новый цикл, а под вторым — ранее уложенный, переход к п. 3, иначе — к п. 2.

2. Если конечная вершина обратного ребра, соответствующего новому циклу, принадлежит ранее расположенному, то под первым рассматриваемым циклом будем понимать ранее уложенный, а под вторым — новый цикл, переход к п. 3, иначе — к п. 10.

3. Если все вершины второго цикла принадлежат первому, то переход к п. 10, иначе — к п. 4.

4. Если второй цикл имеет с первым только одну общую вершину, то переход к п. 10, иначе — к п. 5.

5. Если последняя совпадающая вершина в циклах V_i является разветвлением в ориентированном покрывающем дереве, т. е. имеет локальную степень исхода >1 , то переход к п. 6, иначе — к п. 9.

6. Если второй цикл содержит дугу ориентированного покрывающего дерева, которая является ответвлением в нем с началом в вершине V_i , то переход к п. 7, иначе — к п. 9.

7. Если V_i является началом обратного ребра первого цикла, то переход к п. 8, иначе — к п. 10.

8. Если этот цикл одновременно входит в цикл, содержащий дугу ориентированного покрывающего дерева, которая является ответвлением и охватывает другой цикл, содержащий дугу ориентированного покрывающего дерева, которая является тем же ответвлением, то переход к п. 9, иначе — к п. 10.

9. Ранее уложенный цикл переносится в другую плоскость. Знаки элементов в строке матрицы D , соответствующей этому

перемещенному циклу, меняются на противоположные, переход к п. 11.

10. Циклы размещаются в одной плоскости без пересечений их обратных ребер, переход к п. 11.

11. Алгоритм прекращает свою работу.

Если в результате работы описанного выше алгоритма новый цикл располагается в «положительной» плоскости с ранее



Рис. 3. Планарный граф

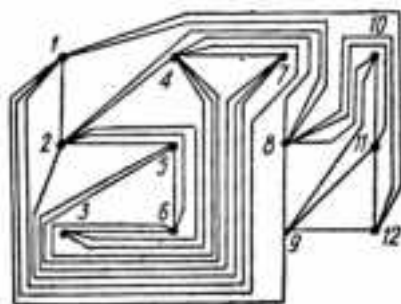


Рис. 4. Планарный граф схемы

размещенным в ней, то рассматривается следующий новый цикл. В противном случае ранее уложенный цикл перемещается в «отрицательную» плоскость и аналогично проверяется, может ли он быть расположен с ранее уложенными в ней циклами. Если же такое размещение невозможно, то ранее уложенный в «отрицательной» плоскости цикл переносится в «положительную» и проверяется возможность его укладки вместе с ранее расположенными в этой плоскости циклами без пересечения их обратных ребер. Если на этом шаге перенесенный в «положительную» плоскость цикл вызывает перемещение в «отрицательную» плоскость первоначального цикла, с которого началось построение планарного графа на заданном этапе, то считаем, что граф непланарен. Этот процесс будет продолжаться до тех пор, пока не окажется, что граф непланарен, либо пока не будут расположены все обратные ребра в G без пересечений, т. е. получен планарный граф, описываемый матрицей D' .

Для рассматриваемого примера построен планарный граф (рис. 3), матрица D' которого приведена ниже.

	1	2	3	4	5	6	7	8	9	10	11	12
1	2	5	6	∅	∅	∅	∅	∅	∅	∅	∅	3
2	-6	-3	∅	∅	∅	∅	∅	∅	∅	∅	∅	-4
3	-1	-2	-5	-6	-3	∅	∅	∅	∅	∅	∅	-4
4	-5	-6	-3	∅	∅	∅	∅	∅	∅	∅	∅	-4
5	2	5	6	3	∅	∅	∅	∅	∅	∅	∅	4
6	2	5	∅	∅	∅	∅	∅	∅	∅	∅	∅	6
7	-1	-2	-5	-6	-3	-4	∅	∅	∅	∅	∅	-7
8	-2	-5	-6	-3	-4	∅	∅	∅	∅	∅	∅	-7
$D' =$ 9	-5	-6	-3	-4	∅	∅	∅	∅	∅	∅	∅	-7
10	1	2	5	6	3	4	∅	∅	∅	∅	∅	8
11	2	5	6	3	4	∅	∅	∅	∅	∅	∅	8
12	-8	-10	∅	∅	∅	∅	∅	∅	∅	∅	∅	-9
13	-1	-2	-5	-6	-3	-4	-8	-10	∅	∅	∅	-9
14	10	9	∅	∅	∅	∅	∅	∅	∅	∅	∅	11
15	8	10	9	∅	∅	∅	∅	∅	∅	∅	∅	11
16	-9	-11	∅	∅	∅	∅	∅	∅	∅	∅	∅	-12
17	8	10	9	11	∅	∅	∅	∅	∅	∅	∅	12
18	1	2	5	6	3	4	8	10	9	11	∅	12

Необходимо отметить, что в общем случае, при заданном фиксированном размещении элементов после получения плоской укладки графа схемы возможно увеличение общей суммарной длины связей. В этой связи целесообразным является совмещение процессов определения планарности графа схемы и размещения его вершин по критерию минимальной суммарной длины ребер.

Таким образом, предложенный алгоритм позволяет при исследовании матричного представления неориентированного графа G получить матрицу, определяющую оргграф, структура которого проще, чем в исходном графе, найти все циклы в нем и путем перемещения соответствующих им обратных ребер получить планарный граф. Так, для рассматриваемого примера получены планарный граф схемы (рис. 4) и ее плоская укладка (рис. 5). Разработана программная реализация этого алгоритма, что дает возможность использовать его при решении задачи автоматизации конструирования микроэлектронных устройств на этапе получения плоского изображения графа соединений элементов электрической схемы. Программа выполнена на алгоритмическом языке ФОРТРАН IV и проходила отладку на ЭВМ ЕС-1022, содержит 160 операторов, время решения для графа, имеющего 40 вершин и 69 ребер, составляет 196 с.

После выполнения программы на печать выводятся матрицы D' и B , причем если граф планарен, то печатаются обе матрицы, если же нет, то только одна — B . При выбранном виде печати результата допускается максимальное число вершин в графе, равное 42, максимальная локальная степень вершины в графе 41, число ребер — 861. Максимальное количество столбцов в выводимой матрице D' , а следовательно, и число вершин в графе определяется количеством знаков в строке устройства печати и форматом, по которому выводятся эле-

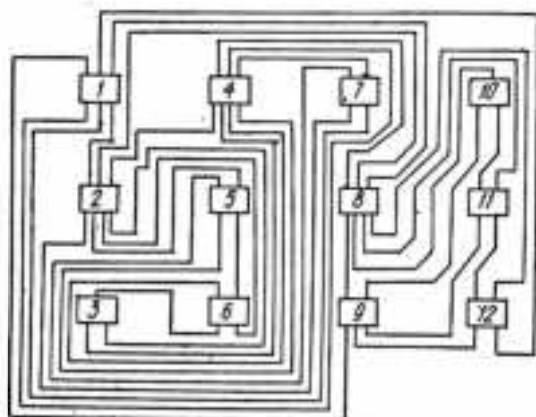


Рис. 5. Плоская укладка схемы

менты матриц. При изменении вида печати результата, например вывода матриц по частям, возможно увеличение числа вершин и ребер в исследуемом графе. В этом случае это число будет ограничено памятью ЭВМ, необходимой для хранения матриц A , B , D и списков S , C , Y . Весь алгоритм построения плоской укладки графа имеет общую сложность $O(N^2)$.

Список литературы: 1. Мелихов А. Н., Берштейн Л. С., Курейник В. М. Применение графов для проектирования дискретных устройств. — М.: Наука, 1974. — 303 с. 2. Кристофидес Н. Теория графов: Алгоритмический подход. — М.: Мир, 1978. — 432 с. 3. Рейнгольд Э., Нисерсальт Ю., Део Н. Комбинаторные алгоритмы. Теория и практика. — М.: Мир, 1980. — 476 с. 4. Харари Ф. Теория графов. — М.: Мир, 1973. — 300 с.

Поступила в редакцию 02.11.83.

ОБ ОДНОМ МЕТОДЕ ИДЕНТИФИКАЦИИ ЛИНЕЙНЫХ ОБЪЕКТОВ, РАБОТАЮЩИХ В ЗАМКНУТЫХ СИСТЕМАХ УПРАВЛЕНИЯ

Введение. Идентификация объектов управления (ОУ) по результатам их нормальной эксплуатации в замкнутых системах управления (СУ) является одной из основных задач, возникающих при построении оптимальных СУ действующими технологическими объектами [1]. В [2, 3] показано, что при неизвестной структуре ОУ такая задача неразрешима.

В работе [4] рассматриваются линейные по параметрам ОУ известной структуры, описываемые уравнениями авторегрессии или авторегрессии со скользящим средним. Приведены необходимые и достаточные условия оцениваемости параметров, сформулированные в виде требований к самим параметрам, структуре ОУ и входным сигналам. С использованием этих условий анализируется оцениваемость параметров ОУ, которые описываются уравнениями авторегрессии, по результатам их функционирования в замкнутом контуре. Показано, что обратная связь (ОС) в общем случае нарушает идентифицируемость и приведены некоторые способы ее восстановления.

В работе [5] рассматривается применение метода максимального правдоподобия для оценивания параметров замкнутых СУ, в которых шумы ОУ и ОС могут быть сведены к «белым шумам невязок» [3] их уравнений. Показано, что оценки максимального правдоподобия состоятельны.

В настоящей статье рассматривается задача оценивания параметров ОУ по результатам их нормального функционирования в замкнутом контуре управления при наличии как собственных шумов объекта, так и ненаблюдаемых ошибок измерений управляющего воздействия и выходной координаты. Основное отличие указанной задачи от исследованных в [5] заключается в том, что шумы ОУ и ошибки измерения не сводятся к белым шумам невязок его уравнений. Это затрудняет использование метода максимального правдоподобия для решения рассматриваемой задачи оценивания даже при известных вероятностных характеристиках ошибок измерения и требует разработки практически более приемлемых процедур.

Постановка задачи. Рассматривается устойчивая СУ, приведенная на рисунке. Структура СУ и параметры ОС известны. Уравнения, описывающие работу системы, имеют вид

$$w_n = \sum_{i=1}^N a_i w_{n-i} + \sum_{j=1}^N b_j x_{n-j} + \eta_n, \quad (1)$$

$$y_n = w_n + z_n, \quad (2)$$

$$u_n = \sum_{i=1}^K c_i u_{n-i} - \sum_{j=0}^K d_j y_{n-j}, \quad (3)$$

$$x_n = u_n + f_n + \lambda_n, \quad (4)$$

где x_n , w_n — ненаблюдаемые последовательности на входе и выходе ОУ; y_n , u_n — наблюдаемые последовательности на входе и выходе ОС; f_n , z_n — случайные ошибки реализации управляющего воздействия и измерения выходной координаты, представляющие собой стационарные белые шумы.

Дисперсии θ_i^2 и σ_i^2 процессов f_n и ε_n в предположении о нормальности их одномерных законов распределения однозначно определяются паспортными характеристиками точности измерительных приборов или методов контроля и потому считаются известными; λ — постоянная на интервале идентификации неизвестная ошибка реализации управляющего воздействия; η_n — собственный шум объекта, представляющий собой стационарный белый шум с неизвестной конечной дисперсией.

При этом

$$\begin{aligned} M[f_n] &= M[\varepsilon_n] = M[\eta_n] = 0; \quad M[f_n u_{n-1}] = \\ &= M[\varepsilon_n \cdot u_{n-1}] = M[\eta_n \cdot u_{n-1}] = \\ &= M[f_n \cdot y_{n-1}] = M[\varepsilon_n \cdot y_{n-1}] = \\ &= M[\eta_n \cdot y_{n-1}] = 0; \quad \forall i \in \overline{0, \infty}, \\ &\quad \forall j \in \overline{1, \infty}; \\ M[\varepsilon_i f_j] &= M[\eta_i f_j] = \\ &= M[\varepsilon_i \cdot \eta_j] = 0; \quad \forall (i, j) \in \overline{0, \infty}. \end{aligned} \quad (5)$$

Устойчивая система управления

Здесь M — символ математического ожидания.

Необходимо оценить неизвестные параметры ОУ: a_i , b_i и λ по результатам его нормальной эксплуатации на интервале $\overline{1, T}$, где $T \gg 2N$.

Представим данные о функционировании ОУ, используемые для оценивания его параметров, в матричной форме:

$$Y = HG + PG + E + \Omega = HG + E_1, \quad (6)$$

где $E_1 = PG + E + \Omega$ — вектор ненаблюдаемых величин; $Y' = (y_{1+1}, \dots, y_T)$; $E' = (\varepsilon_{1+1}, \dots, \varepsilon_m)$; $\Omega = \eta_{1+1}, \dots, \eta_m$;

$$\begin{aligned} H &= \begin{bmatrix} y_1 & \dots & y_{1+1-N} & u_1 & \dots & u_{1+1-N} & 1 \\ \vdots & & & \vdots & & \vdots & \vdots \\ y_{T-1} & \dots & y_{T-N} & u_{T-1} & \dots & u_{T-N} & 1 \end{bmatrix}; \\ G' &= (a_1, \dots, a_N, b_1, \dots, b_N, \gamma); \quad \gamma = \lambda \sum_{i=1}^N b_i; \\ P &= \begin{bmatrix} -\varepsilon_1 & \dots & -\varepsilon_{1+1-N} & f_1 & \dots & f_{1+1-N} & 0 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ -\varepsilon_{T-1} & \dots & -\varepsilon_{T-N} & f_{T-1} & \dots & f_{T-N} & 0 \end{bmatrix}; \\ S &= \begin{cases} N & \text{при } N \leq K; \\ 2N - K & \text{при } N > K. \end{cases} \end{aligned}$$

Если $N < K$, то состоятельная оценка идентифицируемых параметров $\hat{G}$ может быть получена при помощи метода вспомогательных переменных $\hat{G} = (\Phi'H)^{-1} \Phi'Y$, где Φ — матрица наблюдений, коррелированных с H , но не коррелированных с E_1 . Для того чтобы $M[\Phi'E_1] = 0$, наблюдения Φ должны быть поэлементно

сдвинуты относительно H в сторону меньшего времени не менее, чем на N тактов. Однако с ростом интервала между наблюдениями снижается их взаимная корреляция, и при ограниченном T норма отбрасываемого вектора $\Phi'E$, чаще всего соизмерима с нормами $\Phi'H$ и $\Phi'Y$, что приводит к значительной дисперсии $\hat{G}$.

При $N > K$ столбцы матрицы H линейно зависимы, поэтому $(\Phi'H)^{-1}$ не существует и метод вспомогательных переменных не применим.

Процедура идентификации. Для решения рассматриваемой задачи предлагается метод, основанный на умножении (6) слева на матрицу экспериментальных данных V' , коррелированную как с H , так и с E :

$$V'Y = V'HG + V'PG + V'(E + Q), \quad (8)$$

где

$$V = \begin{bmatrix} y_s & \dots & y_1 & u_s & \dots & u_{s-r} & 1 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ y_{T-1} & \dots & y_{T-N} & u_{T-1} & \dots & u_{T-r+1} & 1 \end{bmatrix},$$

$$r = \begin{cases} K-1 & \text{при } K < N; \\ N-1 & \text{при } K > N. \end{cases}$$

Если $N \leq K$, матрица V совпадает с H . Отличие V от H при $N > K$ объясняется тем, что для обеспечения линейной независимости уравнений (8) столбцы V должны быть также линейно независимы.

Разделим (8) на $T-s$ и заменим ненаблюдаемые слагаемые правой части пределами

$$p \lim_{T \rightarrow \infty} \frac{1}{T-s} V'(E + Q) = 0, \quad (9)$$

$$p \lim_{T \rightarrow \infty} \frac{1}{T-s} V'PG = Q \neq 0, \quad (10)$$

к которым они сходятся по вероятности.

В (10)

$$Q = \begin{bmatrix} -\sum_{i=1}^N a_i K_{is}(i-1) + \sum_{j=1}^N b_j K_{js}(j-1) \\ \vdots \\ -\sum_{i=1}^N a_i K_{is}(i-s) + \sum_{j=1}^N b_j K_{js}(j-s) \\ \vdots \\ -\sum_{i=1}^N a_i K_{is}(i-1) + \sum_{j=1}^N b_j K_{js}(j-1) \\ \vdots \\ -\sum_{i=1}^N a_i K_{is}(i-r) + \sum_{j=1}^N b_j K_{js}(j-r) \\ \vdots \\ -\sum_{i=1}^N a_i M[\varepsilon_N] + \sum_{j=1}^N b_j M[f_N] \end{bmatrix}. \quad (11)$$

$$K_{xy}(i) = \rho \lim_{T \rightarrow \infty} \frac{1}{T-s} \sum_{l=1}^{T-s} x_l y_{l+i} = \alpha_i^2 g_{xy}(i); \quad (12)$$

$$K_{ly}(i) = \rho \lim_{T \rightarrow \infty} \frac{1}{T-s} \sum_{l=1}^{T-s} l y_{l+i} = \alpha_i^2 g_{ly}(i); \quad (13)$$

$g_{xy}(i)$, $g_{ly}(i)$ — весовые функции замкнутой системы по соответствующим каналам; $K_{xy}(i)$ и $K_{ly}(i)$ определяются аналогично.

Выразим значения $g_{xy}(i)$, $g_{ly}(i)$, $g_{yx}(i)$ и $g_{lx}(i)$ через параметры СУ. Если передаточная функция замкнутой СУ по каналу $x-y$ имеет вид

$$W_{xy}(z) = \frac{\beta_0 z^{N+K} + \beta_1 z^{N+K-1} + \dots + \beta_{N+K}}{z^{N+K} - \alpha_1 z^{N+K-1} - \dots - \alpha_{N+K}},$$

где α_i , β_i линейны относительно неизвестных параметров ОУ, то

$$g_{xy}(0) = \beta_0; \quad g_{xy}(1) = \alpha_1 g_{xy}(0) + \beta_1;$$

$$g_{xy}(i) = \sum_{l=1}^i \alpha_l g_{xy}(i-l) + \beta_i; \quad (i \leq N-1). \quad (14)$$

Значения весовой функции по каналу $f-y$ определяются аналогично:

$$g_{ly}(i) = \sum_{l=1}^i \varphi_l g_{xy}(i-l) + \varphi_i; \quad (i \leq N-1), \quad (15)$$

где φ_i — коэффициенты при соответствующих степенях z в числителе передаточной функции $W_{ly}(z)$, линейные относительно параметров ОУ.

Учитывая уравнение (3) ОС, выразим функции $g_{yx}(i)$, $g_{lx}(i)$ через $g_{xy}(i)$ и $g_{ly}(i)$:

$$g_{yx}(i) = \sum_{l=1}^K c_l g_{xy}(i-l) - \sum_{l=0}^K d_l g_{xy}(i-l), \quad (16)$$

$$g_{lx}(i) = \sum_{l=1}^K c_l g_{ly}(i-l) - \sum_{l=0}^K d_l g_{ly}(i-l). \quad (17)$$

Нелинейные уравнения предлагаемого метода получаются подстановкой в (8) выражений (9) — (17):

$$(T-s)^{-1} V' Y = (T-s)^{-1} V' H \bar{G} + Q(\sigma_i^2, \theta_j^2, \bar{G}), \quad (18)$$

где $Q(\sigma_i^2, \theta_j^2, \bar{G})$ — вектор, составляющие которого являются полиномами относительно оцениваемых параметров ОУ.

Для упрощения процедуры решения задачи перейдем от (18) к эквивалентной системе при помощи преобразований, которые, например, при $N \leq K$ имеют вид:

$$L_{N+i}^2 = L_{N+i} - \sum_{l=1}^{N-i} c_l L_{N+i+l} + \sum_{l=0}^{N-i} d_l L_l, \quad i \in \overline{1, N-1}, \quad (19)$$

$$L_i^* = L_i - \sum_{j=1}^{N-1} a_{ij} L_{i+j}, \quad i \in \overline{1, N-2}, \quad (20)$$

де L_i — i -е уравнение системы (18); L_i^* — i -е уравнение эквивалентной системы. Преобразования (19) в соответствии с (16), (17) позволяют исключить нелинейные члены (составляющие вектора Q) из уравнений L_{N+i} ($i \in \overline{1, r}$) системы (18). Преобразования (20) снижает порядок L_i ($i \in \overline{1, N-2}$) до второго. Таким образом, эквивалентная система будет содержать $N+2$ линейных и $N-1$ уравнений второго порядка относительно оцениваемых параметров OY и λ .

В дальнейшем под (18) будем подразумевать именно эквивалентную систему. В данной статье рассматривается аналитический подход к ее решению, основанный на последовательном исключении переменных при помощи результата [6]. Подход гарантирует определение всех решений, если их число конечно. Однако из-за большого числа функционально различных операций он может быть реализован на цифровой ЭВМ с приемлемым объемом работы по программированию при $N \leq 4$.

Необходимость исследования ограниченности числа решений системы (18) возникает при $N \geq 3$. Можно показать, что система сводится к двум взаимно простым нелинейным уравнениям порядка $(N-1)^2$, может быть, за исключением изолированных точек на плоскости с координатами σ_i^2 , θ_i^2 . Отсюда в соответствии с теоремой Безу [7] следует, что система (18), «как правило», имеет конечное число решений, не превышающее 2^N .

При $T \rightarrow \infty$ элементы матриц $(T-s)^{-1}V'Y$ и $(T-s)^{-1}V'H$ сходятся по вероятности к соответствующим корреляционным моментам, а одно из решений системы (18) сходится [7] в комплексном пространстве к вектору истинных значений оцениваемых параметров. Отсюда следует, что состоятельные оценки параметров содержатся среди действительных корней или вещественных частей комплексных корней системы (18).

Для выделения оценки вектора идентифицируемых параметров решения системы (18), обеспечивающие устойчивость замкнутой системы, подвергаются проверке. При этом для каждого решения определяются корреляционные моменты, соответствующие элементам матриц $(T-s)^{-1}V'Y$ и $(T-s)^{-1}V'H$, и средние квадратические отклонения элементов при ограниченном числе T экспериментальных наблюдений. Вычисления выполняются по формулам [8]:

$$K_{uv}(r) = \sigma_u^2 \sum_{l=0}^{\infty} g_{ul}(l) g_{uv}(r+l) + \theta_v^2 \sum_{l=0}^{\infty} g_{lv}(l) g_{lv}(r+l), \quad (r \geq 0);$$

$$\sigma_{uv}(r) = \sigma_u \sigma_v \sqrt{2/(T-s-r-1)}, \quad (21)$$

где $K_{uv}(r)$ имеет смысл (12), (13); $\sigma_{uv}(r)$ — среднеквадратическое отклонение соответствующего элемента матриц $(T-s)^{-1}V'Y$ и

$(T-s)^{-1}V'H$; σ_{u_n} , σ_{y_n} — средние квадратические отклонения процессов u_n и y_n при проверяемом решении системы (18).

В качестве оценки вектора параметров принимается решение, при котором все элементы матриц $(T-s)^{-1}V'Y$ и $(T-s)^{-1}V'H$ удовлетворяют правилу трех сигм, и сумма квадратов их отклонений от соответствующих корреляционных моментов минимальна.

Параметр	Метод	T			
		100	200	500	1000
a_1	МВП	13,6	2,5	2,1	0,3
	ПМ	0,15	0,14	0,11	0,08
a_2	МВП	5,4	1,4	1,2	0,04
	ПМ	0,12	0,1	0,07	0,009
b_1	МВП	0,91	0,15	0,14	0,002
	ПМ	0,01	0,003	0,001	0,0004
b_2	МВП	1,33	0,29	0,33	0,008
	ПМ	0,024	0,016	0,011	0,0018
λ	МВП	1,14	0,72	0,69	0,24
	ПМ	0,43	0,36	0,27	0,14

Если ни одно из решений системы (18) не обеспечивает устойчивости замкнутой СУ, следует признать, что для оценивания идентифицируемых параметров объем экспериментальных данных недостаточен.

В качестве примера рассматривалась система (1) — (4) при $K = N = 2$; $a_1 = 0,7$; $a_2 = -0,1$; $b_1 = 0,3$; $b_2 = 0,1$; $c_1 = 0,6$; $c_2 = 0,4$; $d_0 = 1,6$; $d_1 = 1,2$; $d^2 = 0,8$; $\lambda = 0$; $\sigma_u^2 = 1$; $\sigma_y^2 = 4$; $u_n = 0$. Работа системы моделировалась на цифровой ЭВМ. Параметры a_i , b_i , λ оценивались предлагаемым методом и методом вспомогательных переменных.

Качество оценивания по каждому из параметров характеризовалось функционалом

$$F = \frac{1}{L} \sum_{j=1}^L (\hat{p}_j - p)^2,$$

где $L = 10\,000/T$ — число повторений эксперимента, длительность которого равна T ; $\hat{p}_j$ — оценка параметра p по данным j -й реализации. В таблице приведены значения F для метода вспомогательных переменных (МВП) и предлагаемого метода (ПМ), полученные при разных T . Как видно из таблицы, предлагаемый метод имеет значительно более высокую эффективность оценивания.

В некоторых случаях система (18) имела два действительных решения в области устойчивости замкнутой СУ. Например, для одной из реализаций экспериментальных данных при $T = 200$ они имели вид: 1) $\hat{a}_1 = 0,777$; $\hat{a}_2 = -0,158$; $\hat{b}_1 = 0,314$; $\hat{b}_2 = 0,082$; 2) $\hat{a}_1 = -0,008$; $\hat{a}_2 = 0,368$; $\hat{b}_1 = 0,481$; $\hat{b}_2 = 0,338$.

Дисперсии выходных координат ОУ, параметры которых совпадают с первым и вторым решениями системы (18), соответственно равны: $K'_{yy}(0) = 2,52$; $K''_{yy}(0) = 3,63$. Статистические оценки этих дисперсий в соответствии с (21) могут находиться в диапазонах

$$1,764 < \bar{K}_{yy}(0) < 3,276; \quad 2,54 < \bar{K}'_{yy}(0) < 4,72.$$

Экспериментально полученная оценка $\bar{K}_{yy}(0) = 2,11$. Уже эти данные позволяют исключить второе решение системы (18).

Список литературы: 1. Ротач В. Я. Расчет динамики промышленных автоматических систем регулирования. — М.: Энергия, 1973. — 440 с. 2. Ротач В. Я. По поводу работ, связанных с идентификацией объектов в условиях их нормального функционирования. — Автоматика и телемеханика, 1969, № 6, с. 201—202. 3. Эйхгофф П. Основы идентификации систем управления. — М.: Мир, 1975. — 663 с. 4. Кашин Р. Л., Рао А. Р. Построение динамических стохастических моделей по экспериментальным данным. — М.: Наука, 1983. — 384 с. 5. Ворчик Б. Г. Асимптотические оценки максимального правдоподобия параметров замкнутой стохастической системы. — Автоматика и телемеханика, 1976, № 12, с. 32—48. 6. Киром А. Г. Курс высшей алгебры. — М.: Наука, 1975. — 432 с. 7. Шафаревич И. Р. Основы алгебраической геометрии. — М.: Наука, 1972. — 568 с. 8. Литвиц Н. А., Пугачев В. Н. Вероятностный анализ систем автоматического управления. — М.: Сов. радио, 1963. — 809 с.

Поступила в редакцию 10.05.83.

УДК 621.372

И. Ф. ОГОРОДНЕЙЧУК, д-р техн. наук, Ю. Н. ЗОЛОТОВЕРХИЙ, М. Я. ЖИТОМИРСКИЙ

О ПРИЛОЖЕНИИ ТЕОРИИ ФЛОКЕ — ЛЯПУНОВА К ИССЛЕДОВАНИЮ ПАРАМЕТРИЧЕСКИХ ЦЕПЕЙ

Цепи с переменными во времени параметрами находят все более широкое применение в радиотехнике и электронике. Примером таких цепей могут служить параметрические генераторы, усилители, преобразователи частоты, параметрические модуляторы и фильтры. Принцип действия таких устройств основан на использовании параметрического (линейного и нелинейного) резонанса. Анализ процессов в цепях с переменными во времени параметрами без использования специальных методов представляет значительные математические трудности. Поэтому в настоящее время анализ таких цепей является актуальной задачей.

К важным вопросам анализа процессов в параметрических цепях относятся вопросы определения периодических режимов работы и нахождения областей устойчивости для данного класса устройств.

Рассмотрим некоторые аспекты анализа процессов в нестационарных линейных цепях, параметры которых изменяются периодически. Такие цепи можно описать неоднородной системой линейных дифференциальных уравнений с периодическими коэффициентами вида

$$\frac{dx}{dt} = A(t)x + f(t) \quad (1)$$

с начальными условиями $x(0) = x_0$. Здесь $A(t)$ — $(n \times n)$ — периодическая с периодом T матрица; $x(t) = (x_1(t), x_2(t), \dots, x_n(t))^T$ — вектор состояний системы; $f(t)$ — периодическая с периодом T вектор-функция возбуждения системы.

Пусть $x(t)$ — некоторое решение системы (1). Если допустимо представление $x(t) = u(t) + v(t)$, где $v(t) \rightarrow 0$ при $t \rightarrow \infty$, а $u(t)$ — периодическая с периодом T функция, то функцию $u(t)$ назовем существенным решением (1), а значение времени τ , такое что $|v(t)| < \varepsilon$ при $t > \tau$, — временем установления при данной точности ε .

При анализе периодических режимов работы параметрической цепи возникают следующие вопросы. При каких условиях система (1) является устойчивой? Какой вид имеет ее существенное решение? Чему равно время установления для данной системы (1)?

В настоящей работе эти вопросы решаются на основании преобразования Флоке — Ляпунова [2], приводящего систему (1) к системе с постоянной матрицей в линейной части.

Будем искать решение системы (1) в виде

$$x(t) = B(t)y(t), \quad (2)$$

где $B(t)$ — T периодическая матрица. Из (1) получим при подходящем выборе $B(t)$ систему следующего вида для $y(t)$:

$$\frac{dy(t)}{dt} = Ky(t) + g(t). \quad (3)$$

Здесь K — постоянная матрица, которая связана следующим образом с фундаментальной матрицей решений $X(t)$ в момент времени $t = T$:

$$K = \frac{1}{T} \ln [X(T)]. \quad (4)$$

Матрица $X(T)$ вычисляется непосредственным интегрированием (1) на временном интервале $[0, T]$. Пусть $e_j = (0, \dots, 0, 1, 0, \dots, 0)$ — единичный орт (с единицы на j -м месте); $\bar{x}_j$ — решение (1)

при начальных условиях $x(0) = e_j$. Тогда $\bar{x}_j$ совпадает с вектором, представляющим j -й столбец $X(T)$. Заметим, что матрица K имеет, вообще говоря, комплексные коэффициенты, возникающие при логарифмировании. Таким образом, для того чтобы получить в явной форме систему (3), остается определить $g(t)$. Поскольку

$g(t) = B^{-1}(t) f(t)$, то достаточно определить матрицу преобразования $B(t)$. Для этого необходимо решить уравнение

$$B'(t) = A(t) B(t) - B(t) K \quad (5)$$

при начальном условии $B(0) = E$ (единичная матрица) во временном интервале $[0, T]$. Для интегрирования (5) можно исследовать конечно-разностную формулу

$$B(t_0 + h) = B(t_0) + h [A(t_0) B(t_0) - B(t_0) K]. \quad (6)$$

Условия устойчивости (1) совпадают с условиями устойчивости (3) и могут быть выражены в терминах жордановых инвариантов матрицы K [1]. Жорданова форма матрицы K должна быть диагональна, а собственные числа λ_j должны обладать свойством $\operatorname{Re} \lambda_j < 0$, ($j = 1, \dots, n$). Будем предполагать теперь, что указанные условия устойчивости выполнены.

Теорема. Если система (1) устойчива, то она имеет вещественное решение, т. е. ее решение $x(t)$ можно разложить на составляющие $x(t) = u(t) + v(t)$, где $v(t) \rightarrow 0$ ($t \rightarrow \infty$), а $u(t)$ периодическая функция.

Доказательство. Пусть $\gamma(t) = y(t+T) - y(t)$. Тогда $\gamma(t)$ подчинено системе уравнений

$$\frac{d\gamma(t)}{dt} = K\gamma(t), \quad (7)$$

решая которую находим

$$\gamma(t) = \exp(Kt) C. \quad (8)$$

Неизвестный вектор C можно определить из начальных условий

$$C = \gamma(0) = y(T) - y(0) = y(T) - x_0 = (\exp(KT) - I) x_0 + \\ + \exp(KT) \int_0^T \exp(-K\mu) g(\mu) d\mu. \quad (9)$$

(Заметим, что $y(0) = y_0 = x_0$, поскольку $B(0) = E$).

Итак,

$$\gamma(t) = \exp(Kt) \left[(\exp(KT) - I) x_0 + \exp(KT) \int_0^T \exp(-K\mu) g(\mu) d\mu \right]. \quad (10)$$

Но тогда $\gamma(t) \rightarrow 0$ при $t \rightarrow \infty$, что следует из условий устойчивости (1). Это и означает, что решение $y(t)$ раскладывается на указанные в условии теоремы составляющие. Очевидно, тем же свойством обладает и решение (1).

Перейдем теперь к следующему вопросу — оценке времени установления.

Оценим время установления в периодическом режиме на выходе схемы. Обозначим

$$B_1 = [\exp(KT) - I] x_0$$

$$B_2 = \exp(KT) \int_0^T \exp(-K\mu) g(\mu) d\mu.$$

Для того чтобы

$$\|\gamma_1(t) + \gamma_2(t)\| < \epsilon, \quad (11)$$

достаточно выполнения неравенства

$$\|\gamma_1(t)\| + \|\gamma_2(t)\| < \epsilon$$

или

$$\|\exp(Kt)\| \cdot \|B_1\| + \|\exp(Kt)\| \cdot \|B_2\| < \epsilon,$$

или

$$\|\exp(Kt)\| (\|B_1\| + \|B_2\|) < \epsilon.$$

Так как

$$\|\exp(Kt)\| < \frac{1}{\|\exp(-Kt)\|},$$

то

$$\|\exp(Kt)\| > \frac{\epsilon}{\|B_1\| + \|B_2\|}.$$

Пусть

$$\|\exp(-Kt)\| = t |\lambda_{\max}|,$$

где $|\lambda_{\max}|$ — максимальное собственное число матрицы K . Тогда можно оценить время τ , начиная с которого процесс на выходе параметрической схемы будет периодическим:

$$\tau > \frac{1}{|\lambda_{\max}|} \ln \left(\frac{\epsilon}{\|B_1\| + \|B_2\|} \right) = \frac{1}{\|K\|} \ln \left(\frac{\epsilon}{\|B_1\| + \|B_2\|} \right). \quad (12)$$

Для того чтобы углубить исследование установившегося режима, заметим следующее. Пусть $y(t)$ — решение (3); $y(t) = y_1(t) + y_2(t)$, $y_1(t)$ — периодическая вектор-функция; $y_2(t) \rightarrow 0$ ($t \rightarrow \infty$). Обозначим

$$z(t) = \exp(Kt) \int_0^t \exp(-K\mu) g(\mu) d\mu. \quad (13)$$

Тогда допустимо разложение $z(t) = z_1(t) + z_2(t)$, где $z_1(t)$ — периодическая вектор-функция, а $z_2(t) \rightarrow 0$ ($t \rightarrow \infty$) и при этом $z_1(t) = y_1(t)$. Таким образом, существенное решение (т.е. установившийся выходной режим, соответствующий входному сигналу $g(t)$), достаточно выделять из вектор-функции $z(t)$. Поскольку жорданова форма K диагональна, то не нарушая общности, можно считать диагональной саму матрицу K . Тогда система (3) сводится к n одномерным системам и существенный отклик достаточ-

но определять по координатно, выделяя его для j -й координаты $g(t)$ из

$$z_j(t) = \exp(\lambda_j t) \int_0^t \exp(-\lambda_j \mu) g_j(\mu) d\mu.$$

Представляя j -ю координату $g_j(t)$ рядом Фурье

$$g_j(t) = \sum_{m=0}^{\infty} \left(a_{jm}^* \sin \frac{2\pi m}{T} t + b_{jm}^* \cos \frac{2\pi m}{T} t \right), \quad (14)$$

получим выражение для $z_j(t)$ в виде

$$z_j(t) = \sum_{m=0}^{\infty} \exp(\lambda_j t) \left(a_{jm}^* \int_0^t \exp(-\lambda_j \mu) \sin \frac{2\pi m}{T} \mu d\mu + \right. \\ \left. + b_{jm}^* \int_0^t \exp(-\lambda_j \mu) \cos \frac{2\pi m}{T} \mu d\mu \right). \quad (15)$$

Вычислим интегралы, входящие в (15):

$$\int_0^t \exp(-\lambda_j \mu) \sin \frac{2\pi m}{T} \mu d\mu = \\ = \frac{\left[\frac{1}{-\lambda_j} \exp(-\lambda_j t) \sin \frac{2\pi m}{T} t - \frac{2\pi m}{T \lambda_j^2} \exp(-\lambda_j t) \cos \frac{2\pi m}{T} t + \frac{2\pi m}{T \lambda_j^2} \right]}{1 + 4\pi^2 m^2 / \lambda_j^2 T^2}; \\ \int_0^t \exp(-\lambda_j \mu) \cos \frac{2\pi m}{T} \mu d\mu = \\ = \frac{\left[\frac{1}{-\lambda_j} \exp(-\lambda_j t) \cos \frac{2\pi m}{T} t + \frac{2\pi m}{\lambda_j^2 T} \exp(-\lambda_j t) \sin \frac{2\pi m}{T} t + \frac{1}{\lambda_j} \right]}{1 + 4\pi^2 m^2 / \lambda_j^2 T^2}. \quad (16)$$

Учитывая (16), существенное решение $z_{1j}(t)$ запишется как

$$z_{1j}(t) = \sum_{m=0}^{\infty} \frac{\lambda_j^2 T^2}{T^2 \lambda_j^2 + 4\pi^2 m^2} \left[a_{jm}^* \left(\frac{1}{-\lambda_j} \sin \frac{2\pi m}{T} t - \frac{2\pi m}{T \lambda_j^2} \cos \frac{2\pi m}{T} t \right) + \right. \\ \left. + b_{jm}^* \left(-\frac{1}{\lambda_j} \cos \frac{2\pi m}{T} t + \frac{2\pi m}{T \lambda_j^2} \sin \frac{2\pi m}{T} t \right) \right]. \quad (17)$$

Пример. Рассмотрим цепь (рисунок, поз. а, содержащую емкостной параметрический элемент $C(t)$ и источник гармонического тока $i(t) = I \cos \frac{2\pi}{T_0} t$. Пусть емкость изменяется по закону (рисунок, поз. б)

$$C(t) = C_0 \left(1 + m_c \sin \frac{2\pi}{T_0} t \right). \quad (18)$$

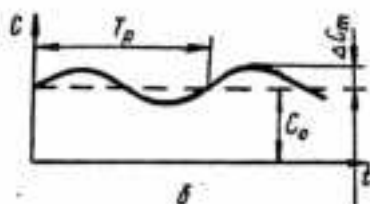
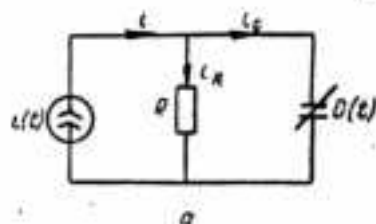
где $m_c = \Delta C_m / C_0$ — коэффициент вариации емкости; $T_p = \frac{T_0}{2}$. Тогда дифференциальное уравнение относительно заряда q на конденсаторе, описывающее эту цепь, имеет вид

$$\frac{dq}{dt} = a(t)q + i(t) \quad (19)$$

при начальных условиях $q(0) = 0$.

Для принятого закона изменения емкости (18) функция

$$a(t) = \frac{-1}{\tau_0 \left(1 + m_c \sin \frac{2\pi}{T_p} t\right)}$$



Параметрическая цепь с емкостью

при $m_c \ll 1$

$$a(t) \approx \frac{-1}{RC_0} \left(1 - m_c \sin \frac{2\pi}{T_p} t\right) = -\frac{1}{\tau_0} \left(1 - m_c \sin \frac{2\pi}{T_p} t\right).$$

Здесь $\tau_0 = RC_0$.

Воспользуемся преобразованием Флоке — Ляпунова. Будем искать решение уравнения (19) в виде

$$q(t) = b(t)y(t). \quad (20)$$

Подстановка (20) в (19) приводит к уравнению с постоянным коэффициентом в линейной части

$$\frac{dy(t)}{dt} = Ky(t) + g(t); \quad y(0) = 0, \quad (21)$$

если

$$b(t) = \exp \left[\int_0^t (a(\tau) - K) d\tau \right] = \exp \left[\frac{m_c T_p}{2\pi \tau_0} \left(1 - \cos \frac{2\pi}{T_p} t\right) \right].$$

При этом

$$K = \frac{1}{T_0} \int_0^{T_0} a(t) dt = \frac{1}{\tau_0 T_0} \int_0^{T_0} \left(-1 + m_c \sin \frac{2\pi}{T_p} t\right) dt = -\frac{1}{\tau_0},$$

а

$$g(t) = b^{-1}(t)i(t) = \exp \left[-\frac{m_c T_p}{2\pi \tau_0} \left(1 - \cos \frac{2\pi}{T_p} t\right) \right] i \cos \frac{2\pi}{T_0} t.$$

Если $m_c \ll 1$, то

$$\exp \left[-\frac{m_c T_p}{2\pi\tau_0} \left(1 - \cos \frac{2\pi}{T_p} t \right) \right] \approx 1 - \frac{m_c T_p}{2\pi\tau_0} \left(1 - \cos \frac{2\pi}{T_p} t \right)$$

и выражение для $g(t)$ сводится к виду

$$g(t) = \left[1 - \frac{m_c T_p}{2\pi\tau_0} \left(1 - \cos \frac{2\pi}{T_p} t \right) \right] / \cos \frac{2\pi}{T_0} t.$$

Представим функцию $g(t)$ в виде ряда Фурье

$$g(t) = \sum_{m=0}^{\infty} \left(a_{jm} \sin \frac{2\pi m}{T_0} t + b_{jm} \cos \frac{2\pi m}{T_0} t \right),$$

где коэффициенты a_{jm} и b_{jm} вычисляются по известным формулам

$$b_0 = \frac{1}{T_0} \int_0^{T_0} g(t) dt = 0, \quad a_{jm} = \frac{2}{T_0} \int_0^{T_0} g(t) \sin \frac{2\pi m}{T_0} t dt = 0,$$

$$b_{jm} = \frac{2}{T_0} \int_0^{T_0} g(t) \cos \frac{2\pi m}{T_0} t dt = \begin{cases} 1 \left(1 - \frac{m_c T_0}{8\pi\tau_0} \right) & \text{при } m = 1, \\ 1 \frac{m_c T_0}{8\pi\tau_0} & \text{при } m = 3, \\ 0 & \text{при } m \neq 1, 3. \end{cases}$$

Тем самым согласно (17) существенное решение уравнения (19), т. е. периодический сигнал на выходе параметрической схемы, имеет следующий вид:

$$q(t) = \frac{IT_0^2}{T_0^2 + 4\pi\tau_0^2} \left[\left(1 - \frac{m_c T_0}{8\pi\tau_0} \right) \left(\tau_0 \cos \frac{2\pi}{T_0} t + \frac{2\pi\tau_0^2}{T_0} \sin \frac{2\pi}{T_0} t \right) + \right. \\ \left. + \frac{m_c T_0}{8\pi\tau_0} \left(\tau_0 \cos \frac{6\pi}{T_0} t + \frac{2\pi\tau_0^2}{T_0} \sin \frac{6\pi}{T_0} t \right) \right] b(t),$$

где

$$b(t) = 1 - \frac{m_c T_p}{2\pi\tau_0} \left(1 - \cos \frac{2\pi}{T_p} t \right).$$

Список литературы: 1. Демидович Б. П. Лекции по математической теории устойчивости. М.: Наука, 1967.—472 с. 2. Якубович В. А., Старжинский В. М. Линейные дифференциальные уравнения с периодическими коэффициентами. — М.: Наука, 1972.—718 с.

Поступила в редакцию 20.06.82.

СОДЕРЖАНИЕ

Путятин Е. П., Лискин В. М., Тоница В. С. Автоматизация построения методом R-функций аналитических описаний многогранных геометрических объектов трехмерных сцен	3
Тарасенко А. П., Машталир В. П., Хитров Б. В. Последовательная нормализация проектных преобразований	11
Шостак В. Ф., Котляревский С. В. Задача синтеза последовательности допустимых погрешностей итерационных алгоритмов координатного одного класса	14
Куник Е. Г., Храмцов В. Н. Об алгоритмах выделения перето-оптимальных решений в задачах векторной оптимизации	20
Салыга В. И., Сидоренко И. Г., Роговченко В. В. Исследование и разработка модифицированных адаптивных алгоритмов идентификации	24
Романий П. Г. К вопросу решения ближайших моделей сетевых систем	27
Тевяшев А. Д., Федоров Н. В., Козыренко С. И. О агрегировании и решении уравнений модели установившегося потокораспределения в инженерных сетях	32
Данилкин М. Я., Лубенский Ю. М. Оптимальное распределение ресурсов в групповой системе водоснабжения	37
Недин Н. В., Сердюк Б. Н. Информационно-алгоритмическое обеспечение ситуационного моделирования эксплуатационных режимов работы электросетей	41
Таубе В. М., Вайсфельд В. А. Условия существования допустимого дискретного управления насосной станцией, работающей на резервуары	47
Пакишев А. В., Данильченко А. М. Об одной задаче управления транспортным процессом	55
Филиппенко И. Г., Карасюк В. В., Меркулов В. С. Элементы автоматизации эксперимента при имитационном исследовании однородной дискретной структуры	58
Гильшин О. Т., Крутиков С. А. Формирование последовательности запросов к большой базе данных	64
Кудрявцев В. А., Яковлева Н. А. Об одном подходе к рациональному размещению информации в системе управления базой данных	71
Борисенко А. А., Губарева С. И., Куно Г. В. Алгоритмы построения кодов с постоянным весом на основе бивомимальных чисел	77
Добрянский В. М., Домина С. Л., Кристаль Н. П. Синтез рабочей программы управляющей цифровой ЭВМ из функционально избыточного набора модулей	81
Литвинов А. Л. Аналитическое моделирование автоматизированных систем группового управления станками с ЧПУ от ЭВМ (Часть вторая)	86
Гасизимский А. Н., Моднов В. П., Кобзев В. Н. Реализация человеко-машинных процедур поиска структуры проектируемого изделия	93
Ашерев А. Т. Концепция надежности АСУП. I (элементы расчета, свойства надежности, отказы)	98
Бородина Е. П. Стохастический метод оценки оптимальных вариантов развития отрасли	105
Винтизенко Н. Г., Герасим Г. А., Семисошенко В. Н. Пространственно-декомпозиционный язык описания принципиальных электрических схем	111
Мазур С. Г., Бондаренко Н. А. Метод распределения проводников в двухслойных печатных платах	115
Калашиников В. А., Якишин В. В. Построение плоской укладки графа электрической схемы	121
Еременко И. Ф., Кривоносов В. А. Об одном методе идентификации линейных объектов, работающих в замкнутых системах управления	131
Огороднейчук И. Ф., Золотоверхий Ю. Н., Житомирский М. Я. О приложении теории Флойда — Ляпунова к исследованию параметрических цепей	137

1 р. 40 к.

АВТОМАТИЗИ- РОВАННЫЕ СИСТЕ- МЫ УПРАВЛЕНИЯ

И ПРИБОРЫ АВТОМАТИКИ

**РЕСПУБЛИКАНСКИЙ
МЕЖДУВЕДОМСТВЕННЫЙ
НАУЧНО-ТЕХНИЧЕСКИЙ
СБОРНИК**

Основан в 1965г.

АСУ и приборы автоматики, 1983, вып. 74, 1—144

