

## О СИНТЕЗЕ ОДНОГО КЛАССА МНОГОРЕЖИМНЫХ СИСТЕМ

А.Ш. Абакаров,\*Ю.А. Сушков †‡

*Санкт-Петербургский государственный университет, г. Санкт-Петербург*

В работе рассматриваются вопросы синтеза систем, у которых настройка на определенный режим работы осуществляется путем изменения связей между составляющими ее элементами, в качестве которых используются двухполюсные  $\lambda$ -обратимые элементы. Такие системы называются многорежимными (МРС), или структурно управляемыми системами. Представлен алгоритм минимизации элементов управления в схемах с максимальным числом режимов и неполным их использованием. Минимизация элементов управления базируется на комбинированном использовании модифицированного алгоритма нахождения минимального гамильтонова цикла и метода Магу.

## Основные понятия и определения

В этой работе под многорежимной системой (МРС), как и в [3,4], будет пониматься система с переменной структурой, у которой настройка на определенный режим работы (т.е. на определенную выходную функцию) осуществляется путем дискретного изменения связей между функциональными элементами, составляющими систему.

Исследуемый здесь класс МРС состоит из двухполюсных функциональных элементов и бинарных элементов управления. В нашем случае для описания таких систем удобнее использовать графовые модели.

**Определение 1.** Многорежимная система представляет собой совокупность объектов:

$$\Sigma = \langle \Gamma, v, w \rangle, \quad (1)$$

где  $\Gamma = \langle Z, D + U \rangle$  – взвешенный граф, у которого:

$Z$  – множество вершин, соответствующих полюсам системы;

$D$  – множество дуг, соответствующих двухполюсным функциональным элементам системы;

$U$  – множество ребер, определяющих элементы управления системы;

$v \in Z$  – входная вершина графа, соответствующая входному полюсу системы и

$w \in Z$  – выходная вершина системы.

Каждой вершине графа  $i \in Z$  поставлено в соответствие значение сигнала  $\omega_i$  на  $i$ -ом полюсе.

Дуге  $(i, j) \in D$  соответствует передаточная функция элемента системы, имеющая в общем случае вид:

$$\omega_j = f(\omega_i, \lambda_{ij}), \quad (2)$$

где  $\lambda_{ij} \in \Lambda$  – скалярный параметр, определяющий конкретный вид зависимости (2),  $\omega_i, \omega_j \in \Omega \subset \mathbb{R}$ .

**Определение 2.** Функциональный элемент называется  $\lambda$ -обратимым [3], если для любых двух значений  $\omega_j = b$ ,  $\omega_i = a$  из  $\Omega$ , удовлетворяющих уравнению (2), при некотором фиксированном  $\lambda' \in \Lambda$ , найдется такое значение  $\lambda'' \in \Lambda$ , что  $a = f(b, \lambda'')$ .

\*alik@vega.math.spbu.ru.

†sushkov@vega.math.spbu.ru.

‡Работа выполнялась при поддержке гранта МО N E02-2.0-28.

Так как структура предлагаемых в этой работе алгоритмов синтеза МРС практически не зависит от вида функции (2), то далее считается, что все функциональные элементы однотипны и описываются линейным  $\lambda$ -обратимым элементом:

$$\omega_j = \lambda_{ij}\omega_i, \quad \lambda_{ij} \in \Lambda. \quad (3)$$

Без ограничения общности можно считать, что  $\Lambda \subset (0, \infty)$ , т.е.  $\lambda_{ij}$  и  $\omega_i$ , ( $i, j \in Z$ ) принимают значения строго больше нуля. Можно показать, что в этом случае

$$\Lambda = \Lambda_{01} \cup \Lambda_{1\infty}, \quad (4)$$

где  $\Lambda_{01} \subseteq (0, 1)$  и  $\Lambda_{1\infty} \subseteq (1, \infty)$ .

В виду линейности и однородности функциональных уравнений (3) будем считать, что значение входного сигнала  $\omega_v = 1$ .

Для ребер  $(i, j)$  из множества  $U$ , характеризующих элементы управления, будем считать, что они описываются уравнениями вида

$$\omega_i = \omega_j. \quad (5)$$

**Определение 3.** Граф  $\Gamma = \langle Z, D + U \rangle$  называется структурным графом системы,  $\Gamma_D = \langle Z, D \rangle$  – функциональным графом, а граф  $\Gamma_U = \langle Z, U \rangle$  – графом управления.

Выделим из  $U$  подмножество ребер  $R$  такое, чтобы в графе  $\Gamma_{DR} = \langle Z, D + R \rangle$  существовала бы единственная цепь

$$i_1, i_2, \dots, i_n, \quad i_n, \quad (6)$$

где  $i_1 = v$ ,  $i_n = w$ ,  $i_j \in Z$  и  $n \leq d$ ,  $d = |D|$ . Тогда эта цепь определяет режим работы системы, заданный элементами управления из  $R \subseteq U$ .

Согласно (3) и (5) значение сигнала  $\omega_w$  на выходном полюсе в режиме  $R$  можно подсчитать по формуле

$$y_R = \omega_w = \lambda_{i_1 i_2} \lambda_{i_2 i_3} \dots \lambda_{i_{n-1} i_n} \omega_v, \quad R \subseteq U, \quad (7)$$

где  $\lambda_{i_j i_{j+1}} = 1$ , если ребро  $(i_j, i_{j+1}) \in U$ , т.е. является элементом управления. Тогда функция

$$f_R(\lambda) = \lambda_{i_1 i_2} \lambda_{i_2 i_3} \lambda_{i_{n-1} i_n} \quad (8)$$

называется передаточной функцией системы в режиме  $R$ .

Пусть  $g = \langle g_1, g_2, \dots, g_l \rangle$  – вектор чисел, называемый выходной гаммой, и пусть для любого  $i \in 1 : l$  существует такое  $R_i \subseteq U$ , что значение передаточной функции при фиксированном векторе  $\lambda$  с заданной точностью равно  $g_i$ . Тогда можно говорить, что режим  $R_i$  реализует значение  $g_i$  выходной гаммы, а система реализует саму выходную гамму.

Существование для каждого режима единой цепи от входной вершины к выходной равносильно тому, что функциональный граф должен быть лесом [3], т.е. необходимо выполнение условия независимости (9): для всякого  $A \subseteq D$ ,  $A \neq \emptyset$ , должно выполняться неравенство:

$$|\Gamma A| \geq |A| + 1, \quad (9)$$

где  $\Gamma A$  – множество вершин, инцидентных ребрам из  $A$ .

**Определение 4.** Компоненту связности функционального графа будем называть каскадом. Систему, состоящую из  $n$  каскадов назовем  $n$ -каскадной.

Двухкаскадную систему, у которой одна компонента связности функционального графа состоит из одной (голой) вершины для определенности будем называть однокаскадной. Число каскадов определяет степень свободы МРС  $\sigma$ .

Очевидно, для того, чтобы в МРС можно было получить более одного режима, число каскадов должно быть не менее двух.

Число степеней свободы определяет минимальное число связей, которое необходимо наложить на систему уравнений (7), чтобы однозначно получить значения выходной функции при фиксированных  $\lambda_{ij}$  и различных  $R \subseteq U$ .

В нашем случае одна связь задается уравнением  $\omega_v = 1$ , а  $\sigma - 1$  связей определяются  $|R| = \sigma - 1$  элементами управлений вида  $\omega_i = \omega$ ,  $(i, j) \in R$ .

Далее, для обозначения неизвестных  $\lambda_{ij}$ , будем также использовать один индекс  $i$ . Для этого перенумеруем дуги функционального графа числами от 1 до  $d$  и будем считать  $\lambda_i$  параметром  $i$ -го функционального элемента.

Пусть матрица  $h[1 : r, 1 : d]$  состоит из элементов, принимающих значения из множества  $\{-1, 0, +1\}$ ,  $r \geq d$ . Тогда множество передаточных функций, соответствующих матрице  $h$ , в общем случае можно записать в виде:

$$\begin{aligned} y_1 &= \lambda_1^{h_{11}} \lambda_2^{h_{12}} \dots \lambda_d^{h_{1d}}, \\ y_2 &= \lambda_1^{h_{21}} \lambda_2^{h_{22}} \dots \lambda_d^{h_{2d}}, \\ &\vdots \\ y_r &= \lambda_1^{h_{r1}} \lambda_2^{h_{r2}} \dots \lambda_d^{h_{rd}}. \end{aligned} \quad (10)$$

Независимо от вида функционального графа передаточные функции всех режимов МРС можно представить в виде (10).

Так как величины  $h_{ij}$  принимают только три значения, то общее число различных режимов (4), которое можно получить в МРС с  $d$  функциональными элементами, подчиняется неравенству:

$$r \leq 3^d. \quad (11)$$

Если  $d > r$ , то понятно, что положив  $y_i(\lambda) = g_i$ ,  $i \in 1 : r$ , всегда можно так подобрать параметры вектора  $\lambda = \langle \lambda_1, \lambda_2, \dots, \lambda_d \rangle$ , что эти равенства будут выполняться точно. Однако задача воспроизведения заданной гаммы  $g$  резко усложняется, если учесть, что  $d < r$ , и поэтому выходная гамма в общем случае может быть воспроизведена приближенно.

Условимся, что выходные сигналы должны лежать внутри заданного интервала  $[gl_i, gr_i]$ , а параметры  $\lambda_i$  – принадлежать области  $\Lambda$ , причем как можно ближе к некоторому желательному значению  $\lambda_0 \in (0, 1)$ , или  $1/\lambda_0 \in (1, \infty)$ .

Все эти ограничения резко усложняют задачу синтеза МРС. В этой работе рассматриваются частные постановки задач синтеза и подходы к их решению. Однако предварительно сформулируем общую, неформальную постановку задачи синтеза МРС [3].

## Общая постановка задачи синтеза

Во всех рассматриваемых ниже частных задачах синтеза предполагаются заданными:

- 1)  $l$  – число выходных сигналов, которые необходимо реализовать в синтезируемой МРС,
- 2)  $g = \langle g_1, g_2, \dots, g_l \rangle$ ,  $g_i > 0$ ,  $i \in 1 : l$ , – вектор выходных сигналов, или выходная гамма сигналов,
- 3)  $\{[gl_i, gr_i] | i \in 1 : l\}$  – интервалы, внутри которых должны находиться получаемые значения выходных сигналов на соответствующих режимах,
- 4)  $\Lambda = \Lambda_{01} \cup \Lambda_{1\infty}$  – область возможных значений параметров функциональных элементов,
- 5)  $\lambda_0 \in \Lambda$  – желательное значение параметра функционального элемента.

При синтезе МРС в общем случае требуется найти:

- 1)  $d = |D|$  – минимально необходимое число функциональных элементов,
- 2)  $u = |U|$  – минимально необходимое число элементов управления,
- 3)  $\Gamma = \langle Z, D + U \rangle$  – структурный граф системы,
- 4)  $\varphi : (1 : l) \rightarrow (1 : r)$  – инъективное отображение, определяющее, на каком из  $r$  режимов значение выходной функции  $f_{\varphi i}(\lambda)$  находится как можно ближе к заданному значению выходной гаммы  $g_i$ ,  $i \in 1 : l$ ,
- 5)  $\lambda_{ij} \in \Lambda$  – значения конструктивных параметров функциональных элементов, лежащих как можно ближе к  $\lambda_0$ .

В качестве критериев оптимизации в зависимости от рассматриваемой задачи синтеза использовались разные функции. В частности, функция

$$F_g(\lambda) = \max_{i \in 1:l} \left| \frac{y_{\varphi i} - g_i}{g_i} \right| \Rightarrow \min_{\lambda} \quad (12)$$

характеризует степень относительного отклонения реализации заданного числа  $g_i$  выходной гаммы  $g$  от соответствующей передаточной функции  $y_{\varphi i}(\lambda)$ .

Введем функцию, определенную на  $\Lambda$ :

$$\Theta(\lambda_j) = \begin{cases} \lambda_j, & \text{если } 0 < \lambda_j < 1, \\ 1/\lambda_j, & \text{если } 1 < \lambda_j < \infty, \end{cases} \quad (13)$$

Тогда критерий

$$F_{\lambda} = \max_j |\lambda_0 - \Theta(\lambda_j)|, j \in 1 : d. \quad (14)$$

будет характеризовать степень отклонения параметров функциональных элементов  $\lambda$  от желаемого значения  $\lambda_0 \in \Lambda_{01}$  ( $1/\lambda_0 \in \Lambda_{1\infty}$ ).

В этой работе также использовался более сложный критерий оптимизации, учитывающий тот факт, что параметры функциональных элементов  $\lambda_j$  должны принимать значения только из  $\Lambda$ ,  $j \in 1 : d$ . Это учитывается в критерии с помощью специальной штрафной функции.

Пусть функциональные элементы и их параметры перенумерованы числами от 1 до  $d$ . И пусть

$$\Lambda_{01} = [\lambda_l, \lambda_r], \quad \Lambda_{1\infty} = [1/\lambda_r, 1/\lambda_l]. \quad (15)$$

Введем такую штрафную функцию

$$\Phi_{\lambda_j} = \left| \lambda_l - \Theta(\lambda_j) \right| + \left| \lambda_r - \Theta(\lambda_j) \right| + (\lambda_l - \lambda_r), \quad (16)$$

которая равна нулю, если  $\lambda_j \in \Lambda$ , и не равна нулю в противном случае.

Тогда функция

$$\Phi_{\lambda} = A \sum_{j=1}^d \Phi_{\lambda_j} \quad (17)$$

равна нулю только тогда, когда  $\lambda_j \in \Lambda = \Lambda_{01} \cup \Lambda_{1\infty}$  для всякого  $j \in 1 : d$  (здесь  $A$  – достаточно большое число).

Теперь, если в задаче синтеза требуется найти такую МРС, для которой функция (12) принимает минимальное значение при условии  $\lambda_j \in \Lambda$ ,  $j \in 1 : d$ , то достаточно найти минимум функции

$$F(\lambda) = F_g(\lambda) + \Phi_{\lambda}.$$

## Общие подходы к решению задачи синтеза МРС

Как видно из вышесказанного, задача синтеза МРС сводится к минимизации функции цели (критерия), заданного как на непрерывном множестве, так и на дискретном. Если в качестве первого во всех рассматриваемых задачах синтеза выступает  $\Lambda^d$ , то в качестве дискретного, в зависимости от постановки задачи, могут выбираться, например, функциональные графы систем, отображения  $\varphi$ , графы управления и т.д.

Пусть  $S$  обозначает дискретное множество, на котором задана функция цели. Для определенности будем называть элемент  $s \in S$  структурой. Тогда задача синтеза может быть описана в виде

$$F(\lambda, s) \Rightarrow \min_{\lambda \in \Lambda^d, s \in S}, \quad (18)$$

где возможные функциональные ограничения учитываются с помощью штрафных функций.

Для рассматриваемых в этой работе задач возможны два принципиально отличающихся подхода [3] (с условным названием):

- 1) прямой,
- 2) обратный.

Суть прямого состоит в том, что тем или иным способом перебираются все структуры из  $S$  и для каждой из них решается задача минимизации функции (18) по  $\lambda \in \Lambda^d$ .

Прямой подход алгоритмически очень прост, универсален, однако во многих случаях весьма трудоемок. Снижение трудоемкости возможно лишь за счет использования конкретных особенностей задачи синтеза и элементов множества  $S$ .

Обратный подход существенно использует то, что передаточные функции (8) имеют общий и достаточно простой вид, полностью определяемый матрицей  $h[1 : r, 1 : d]$ . Поэтому, если найти отображение  $\varphi : (1 : l) \rightarrow (1 : r)$ , при котором функция (12) минимальна на  $\Lambda^d$ , то тем самым была бы найдена матрица  $h[1 : r, 1 : d]$ . И тогда задача определения, например, функционального графа состоит в том, чтобы найти, какой лес соответствует полученной матрице  $h$ . Эта задача в каком-то смысле обратная первой и не всегда решается просто.

Для обоих описанных подходов иногда может быть использован один способ, который не требует перебора структур из  $S$ .

Предположим, что минимум функции цели (18) на  $S$  при фиксированном значении  $\lambda$  может быть найден достаточно просто. Введем обозначение:

$$\Psi(\lambda) = \min_{s \in S} F(\lambda, s) \quad (19)$$

и предположим, что имеется метод, который позволяет найти минимум функции (19) на множестве  $\Lambda^d$  при фиксированном  $\lambda$ , т.е. решить задачу:

$$\Psi(\lambda) \rightarrow \min_{\lambda \in \Lambda^d} . \quad (20)$$

Очевидно, что решения задач (18) и (19),(20) совпадают.

В качестве итерационного процесса нахождения минимума функции  $\Phi(\lambda)$  будем использовать случайный поиск [1], а для нахождения оптимальной структуры  $s \in S$  при фиксированном значении  $\lambda$  – различные жадные алгоритмы, метод динамического программирования и др.

Далее будут рассмотрены задачи синтеза для МРС определенного класса и приведены соответствующие алгоритмы.

## Однокаскадные схемы

Условно под однокаскадной здесь понимается такая двухкаскадная система, у которой функциональный граф  $\langle Z, D \rangle$  состоит из компонент связности:  $\Gamma_1 = \langle Z - w, D \rangle$  и отдельной вершины  $w$  (компоненты связности  $\Gamma_2 = \langle \{w\}, \emptyset \rangle$ ).

Далее предполагаем, что все вершины графа  $\Gamma_1$  перенумерованы числами от 1 до  $z - 1$ ,  $z = |Z|$ , причем входная вершина имеет номер 1, а выходная номер  $z$ .

Так как на каждом режиме существует единственная цепь от входной вершины к выходной (3), то отсюда следует, что граф  $\Gamma_1$  должен быть деревом (3) и поэтому  $d = |D| = z - 2$ .

Сначала рассмотрим синтез однокаскадных схем с полным использованием функциональных элементов и элементов управления. Это означает, что для каждого из этих элементов существует хотя бы один режим, участвующий в реализации заданной гаммы, в котором они используются. Вышесказанное означает, что

$$l = d = z - 2 = r_{\max}, \quad (21)$$

где  $r_{\max}$  – максимальное число режимов, которое можно получить в такой системе.

Так как для любого дерева  $\Gamma_1$  параметр каждого ребра  $\lambda_{ij}$  можно выразить согласно (3) с помощью формулы

$$\lambda_{ij} = \lambda_{i1} \lambda_{1j} = \frac{\lambda_{1j}}{\lambda_{1i}}, \quad i, j \in 2 : (z - 1), \quad (22)$$

то параметры любого дерева с  $z - 1$  вершинами могут быть найдены, если известны параметры  $\lambda_{1i}$ ,  $i \in 2 : (z - 1)$ , звездообразного дерева. Далее будем считать такую схему канонической, а переменные  $\lambda_{1i}$ ,  $i \in 2 : (z - 1)$ , возьмем в качестве базиса. Общий вид канонического структурного графа представлен на рис 1.

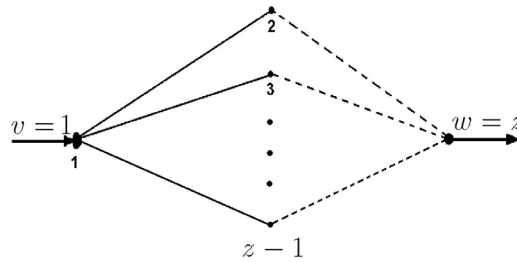


Рис.1. Канонический структурный граф.

Теперь рассмотрим несколько постановок задач синтеза однокаскадной схемы.

**Задача 1.** Пусть задано:

– выходная гамма  $g = \langle g_1, g_2, \dots, g_l \rangle$ ,

– область существования параметров функциональных элементов  $\Lambda = [\lambda_l, \lambda_r] \cup [1/\lambda_r, 1/\lambda_l]$ .

Необходимо определить такие значения параметров  $\lambda_{ij} \in \Lambda$  и такое дерево  $\Gamma_1$ , при которых выходная гамма воспроизводилась бы совершенно точно, т.е. функция (12) равнялась бы нулю (если, конечно, такое решение существует).

Во-первых, отметим, что число вершин в  $\Gamma_1$  согласно (21) равно

$$z = l + 2.$$

Во-вторых, в канонической структуре на множестве вершин  $2 : (z - 1)$  имеет место симметрическая группа. В связи с этим можно считать, что отображение  $\varphi$  имеет вид:

$$\varphi i = i + 1, \quad i \in 1 : l.$$

Теперь приведем алгоритм синтеза МРС, удовлетворяющий поставленным требованиям.

Шаг 1. Приравняем параметры  $\lambda_{1,i+1}$  соответствующим значениям  $g_i$  выходной гаммы  $g$ .

Шаг 2. Используя формулу (22), находим значения параметров  $\lambda_{ij}$  для всех  $i, j \in 1 : (z - 1)$ .

Шаг 3. Каждому ребру  $(i, j)$  приписываем вес, равный штрафной функции  $\Phi_{\lambda_j}$  (16).

Шаг 4. Используя жадный алгоритм, находим минимальное дерево.

Если в результате выполнения алгоритма будет найдено дерево, для которого штрафную функцию  $\Phi_\lambda$  (17) равна нулю, то полученное решение – искомое. В противном случае поставленная задача решения не имеет.

Отметим, что если допустить неполное использование функциональных элементов, то поставленная задача всегда имеет решение. Более точно это утверждение можно выразить следующей теоремой.

**Теорема 1.** Для любой выходной гаммы  $g = \langle g_1, g_2, \dots, g_l \rangle$  и любого ограничения  $\lambda \in \Lambda$  существует такое  $z \geq l + 2$ , при котором задача 1 имеет точно решение.

Для доказательства теоремы достаточно показать, что задачу 1 можно решить путем добавления вершин в канонический структурный граф (рис 1).

Выберем ребро  $(1,2)$  и покажем, что путем добавления вершин  $1', 2', \dots, k'$  всегда можно добиться того, что параметры  $\lambda_{11'}, \lambda_{1'2'}, \dots, \lambda_{k'-1,k'}, \lambda_{k'2}$  будут принимать свои значения из области  $\Lambda$ , а произведение этих параметров будет в точности равно  $g_1$ , т.е.

$$g_1 = \lambda_{11'} \lambda_{1'2'} \dots \lambda_{k'2} = \lambda_{12}.$$

Далее удобно использовать не сами величины  $g_i$ ,  $\lambda$  и др., а их логарифмы, введя такие обозначения:  $\lg g_i = \bar{g}_i$ ,  $\lg \lambda = \bar{\lambda}$ ,  $[\lg \lambda l, \lg \lambda r] = \bar{\Lambda}_{01}$  и т.д. Тогда предыдущее равенство будет иметь вид:

$$\bar{g}_1 = \bar{\lambda}_{11'} + \bar{\lambda}_{1'2'} + \dots + \bar{\lambda}_{k'2} = \bar{\lambda}_{12}.$$

Положим  $\bar{\lambda}_{11'} = \bar{\lambda}r$ ,  $\bar{\lambda}_{1'2'} = -\bar{\lambda}l$ ,  $\bar{\lambda}_{2'3'} = \bar{\lambda}r$ ,  $\bar{\lambda}_{3'4'} = -\bar{\lambda}l \dots$ . В этом случае значения параметров, стоящих на нечетных местах, принадлежат  $\bar{\Lambda}_{01}$ , а на четных —  $\bar{\Lambda}_{1\infty}$ . Отсюда видно, что  $\lambda_{11'} + \lambda_{1'2'} = \bar{\lambda}r - \bar{\lambda}l$ ,  $\lambda_{2'3'} + \lambda_{3'4'} = \bar{\lambda}r - \bar{\lambda}l$  и т.д. Выберем такое нечетное  $k'$ , что

$$\frac{k' - 1}{2}(\bar{\lambda}r - \bar{\lambda}l) \leq \bar{g}_1 \leq \frac{k' - 1}{2}(\bar{\lambda}r - \bar{\lambda}l) + (\bar{\lambda}r - \bar{\lambda}l).$$

В этом случае, выбирая  $\bar{\lambda}_{k'-1,k'}$  из  $\bar{\Lambda}_{01}$ , можно всегда добиться того, чтобы правая часть в точности равнялась бы  $\bar{g}_1$ .

Понятно, что аналогичную процедуру представления ребра в виде цепи можно сделать над всеми остальными ребрами канонического структурного графа.

В связи с этим в случае неполного использования функциональных элементов описанный выше алгоритм должен быть изменен при выполнении четвертого шага: в случае, если при заданном числе вершин  $z$  задача решения не имеет, необходимо добавить одну вершину, положить  $w = z + 1$  и перейти к шагу 1.

Пример 1. Задано:

– выходная гамма из 10 элементов:

$$g = \langle 0.30, 0.52, 0.65, 0.85, 1.05, 2.10, 3.00, 4.00, 4.60, 5.00 \rangle,$$

– область существования параметров функциональных элементов

$$\Lambda = [0.4166, 0.833] \cup [1.2, 2.40].$$

Полученный функциональный граф изображен на рис. (2).

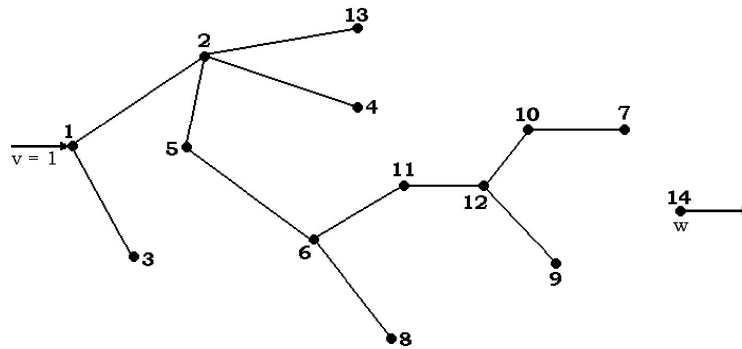


Рис. 2. Функциональный граф для примера 1.

Значения параметров ФЭ при этом равны.

$$\begin{aligned} \lambda_{21} &= 1.923, & \lambda_{52} &= 2.019, \\ \lambda_{65} &= 1.999, & \lambda_{86} &= 1.905, \\ \lambda_{116} &= 1.949, & \lambda_{31} &= 1.538, \\ \lambda_{132} &= 1.733, & \lambda_{1211} &= 1.336, \\ \lambda_{1012} &= 1.632, & \lambda_{912} &= 1.501, \\ \lambda_{710} &= 1.666, & \lambda_{42} &= 1.635. \end{aligned}$$

Время, затраченное компьютером с процессором Pentium-133 и ОЗУ 64 Мб на решение этой задачи составляет 1.3 сек.

Теперь рассмотрим несколько иную постановку задачи синтеза однокаскадной схемы.

Задача 2.

Она отличается от первой задачи тем, что в ней не требуется точного воспроизведения выходной гаммы, требуется лишь, чтобы соответствующий  $i$ -ый выходной сигнал лежал в заданной области  $[gl_i, gr_i]$ .

Задано:

- выходная гамма  $g = \langle g_1, g_2, \dots, g_l \rangle$ ,
- область существования параметров функционального элемента  $\Lambda = [\lambda_l, \lambda_r] \cup [1/\lambda_r, 1/\lambda_l]$ ,
- границы существования выходных сигналов, реализующих число  $g_i$ ,  $gl_i \leq g_i \leq gr_i$ .

Необходимо определить:

- такой функциональный граф  $\langle Z, D \rangle$ ,
- такие значения параметров элементов этого графа  $\lambda_{ij}$ ,
- такие выходные сигналы  $y_i(\lambda) = \lambda_{1,i+1}$ ,

при которых:

- $gl_i \leq y_i(\lambda) \leq gr_i$ ,  $i \in 1:l$ ,
- $\lambda_{ij} \in \Lambda$ ,  $(i, j) \in D$ ,
- критерий качества реализуемости выходной гаммы  $F_g$  принимал бы минимальное значение.

При решении этой задачи каждому ребру будем приписывать вес, определяющий штраф за то, что параметр этого ребра  $\lambda_{ij} \notin \Lambda$ . В качестве штрафной функции используем (17). Таким образом минимизируемый критерий будет иметь следующий вид:

$$F(\lambda) = F_g(\lambda) + \Phi_\lambda \Rightarrow \min_{\lambda \in \Lambda, t \in T}, \quad (23)$$

где  $T$  – множество деревьев в полном графе на  $z - 1$  вершинах.

Для минимизации (23) используем подход, задаваемый формулами (19) и (20) и основанный на теореме 1. С этой целью методом случайного поиска будем искать минимум функции (23), а на каждом шаге случайного поиска при фиксированном  $\lambda$  находить минимальное дерево жадным алгоритмом.

Помимо этого, учтем два существенных момента, позволяющих значительно уменьшить трудоемкость расчетов.

Во-первых, без ограничения общности можно считать, что  $g_1 < g_2 < \dots < g_l$ . Поэтому при случайном выборе параметров функциональных элементов лучше на каждом шаге поиска упорядочивать их тоже в порядке возрастания:  $\lambda_{12} < \lambda_{13} < \dots < \lambda_{1,z-1}$ . Это позволяет сократить область поиска в  $d$ -мерном пространстве в  $2^{d-1}$  раз.

Во-вторых, так как в процессе поиска параметр  $\lambda_{1,i+1}$  стремится к  $g_i$  и его значения не должны выходить за пределы интервала  $[gl_i, gr_i]$ ,  $i \in 1:l$ , то гораздо эффективнее моделировать значения  $\lambda_{1,i+1}$  прямо непосредственно в этом интервале.

Теперь кратко алгоритм решения задачи 2 можно описать следующим образом. На каждом этапе случайного поиска выполняются следующие действия.

Шаг 1. Моделируются значения параметров  $\lambda_{12} < \lambda_{13} < \dots < \lambda_{1,z-1}$  в области  $[gl_i, gr_i]$ ,  $i \in 1:l$ .

Шаг 2. Каждому ребру  $(i, j)$  приписывается вес, равный значению штрафной функции.

Шаг 3. Жадным алгоритмом находится минимальное дерево.

Шаг 4. Если процесс случайного поиска не закончился, осуществляется переход к его новому этапу (шаг 1). Иначе конец.

И в этой задаче решение может не существовать, если число полюсов  $z = l + 2$ . При этом штрафная функция не равна нулю. Как и в предыдущей задаче, в описанный выше алгоритм можно добавить еще один шаг:

Шаг 5. Если случайный поиск завершил работу и решение не найдено, то добавляется новая вершина в  $Z_1$ , полагается  $w = z + 1$  и процесс поиска возобновляется с первого шага.

Отметим, что здесь параметры  $\lambda_{1k}$ ,  $k > z - 1$ , где  $k$  – вновь добавленная вершина, необходимо моделировать уже в области  $\Lambda$ .

Пример 2. Задано:

– выходная гамма из 10 элементов:

$g = \langle 0.30, 0.52, 0.65, 0.85, 1.05, 2.10, 3.00, 4.00, 4.60, 5.00 \rangle$ ;

– границы существования выходных сигналов  $[gl_i, gr_i]$ :  $[0.25, 0.35]$ ,  $[0.40, 0.58]$ ,  $[0.60, 0.70]$ ,  $[0.80, 0.90]$ ,  $[1.00, 1.10]$ ,  $[1.80, 2.50]$ ,  $[2.80, 3.20]$ ,  $[3.90, 4.10]$ ,  $[4.50, 4.70]$ ,  $[4.90, 5.10]$ ;



– область существования параметров функциональных элементов  
 $\Lambda = [0.4166, 0.833] \cup [1.2, 2.40]$ .  
 Значения параметров ФЭ при этом равны.

$$\begin{aligned} \lambda_{21} &= 1.924, & \lambda_{52} &= 2.017, \\ \lambda_{65} &= 2.003, & \lambda_{86} &= 1.879, \\ \lambda_{112} &= 1.733, & \lambda_{31} &= 1.541, \\ \lambda_{42} &= 1.635, & \lambda_{96} &= 2.184, \\ \lambda_{79} &= 1.527, & \lambda_{107} &= 1.648, \end{aligned}$$

Значения выходных сигналов.

$$\lambda_{111} = 0.299, \lambda_{12} = 0.519, \lambda_{13} = 0.649, \lambda_{14} = 0.849, \lambda_{15} = 1.049, \\ \lambda_{16} = 2.100, \lambda_{17} = 3.005, \lambda_{18} = 3.948, \lambda_{19} = 4.588, \lambda_{110} = 4.953.$$

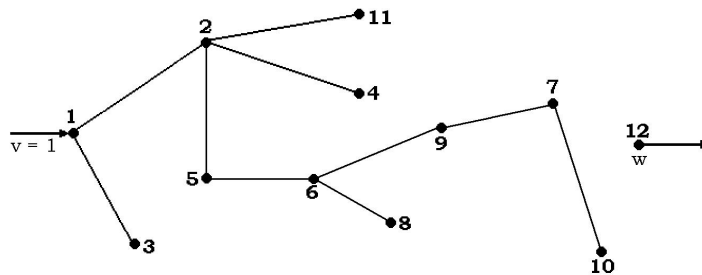


Рис. 3. Функциональный граф для примера 2.

### Синтез (двухкаскадных) многокаскадных схем

Условимся, что двухкаскадная схема содержит три компоненты связности  $\langle Z_1, D_1 \rangle$  и  $\langle Z_2, D_2 \rangle$ ,  $\langle Z_3, D_3 \rangle$ , где  $Z_1 \cup Z_2 \cup Z_3 = Z$  и  $D_1 \cup D_2 = D$ , при этом  $|D_1| = d_1 > 0$ ,  $|D_2| = d_2 > 0$ ,  $Z_3 = 1$ ,  $|D_3| = 0$ .

Так как в двухкаскадной схеме максимальное число режимов равно

$$r_m = \begin{cases} z^2/4, & \text{если } z - \text{четное,} \\ (z+1)(z-1)/4, & \text{в противном случае,} \end{cases}$$

то отображение  $\varphi$  в данном случае выглядит так

$$\varphi : (1 : l) \rightarrow (1 : r_m).$$

Передаточная функция двухкаскадной схемы, использующая функциональные элементы  $(1, i)$  и  $(j, z)$  имеет следующий вид:

$$y_{ij} = \lambda_{1i} \lambda_{jz}.$$

Режим, соответствующий этой передаточной функции, обозначим парой  $(i, j)$ . Тогда все возможные в схеме режимы определим как множество:

$$R = \{(i, j) | i \in Z_1, j \in Z_2\}.$$

Рассмотрим следующую постановку задачи синтеза двухкаскадной схемы.

Задача 3.

Задано:

– выходная гамма  $g = \langle g_1, g_2, \dots, g_l \rangle$ ,

– область существования параметров функционального элемента  $\Lambda = [\lambda l, \lambda r] \cup [1/\lambda r, 1/\lambda l]$ ,

– границы существования выходных сигналов, реализующих число  $g_i$ ,  $gl_i \leq g_i \leq gr_i$ .

Необходимо определить:

– такие компоненты связности (деревья) функционального графа  $\langle Z_1, D_1 \rangle$  и  $\langle Z_2, D_2 \rangle$ ,

– такие значения параметров элементов этого графа  $\lambda_{ij}$ ,

– такие выходные сигналы  $y_i(\lambda)$

при которых:

–  $gl_i \leq y_i(\lambda) \leq gr_i$ ,  $i \in 1:l$ ,

–  $\lambda_{ij} \in \Lambda$ ,  $(i, j) \in D$ ,

– критерий качества реализуемости выходной гаммы  $F_g$  принимал бы минимальное значение.

При решении задачи 3 также будем каждому ребру приписывать вес, определяющий штраф за то, что параметр этого ребра  $\lambda_{ij} \notin \Lambda$ . В качестве штрафной функции используем функцию (17). Таким образом получим следующий критерий:

$$F = F_g + [\Phi_\lambda^1 + \Phi_\lambda^2] \Rightarrow \min_{\lambda \in \Lambda, t_1 \in T_1, t_2 \in T_2},$$

где  $T_1, T_2$  – множества деревьев в полных графах на  $(z_1 - 1)$  и  $(z_2 - 2)$  вершинах соответственно.

Опишем алгоритм решения задачи 3. На каждом этапе случайного поиска выполняются следующие действия.

Шаг 1. Моделируются значения параметров функционального графа  $\lambda_{ij} \in \Lambda$ .

Шаг 2. Каждому ребру  $(i, j)$  приписывается вес, равный значению штрафной функции.

Шаг 3. Жадным алгоритмом для каждого каскада находится минимальное дерево.

Шаг 4. Формируется множество всех возможных выходных функций  $R$ .

Шаг 5. Алгоритмом сдвига [3,4] находится отображение  $\varphi$ .

Шаг 4. Если процесс случайного поиска не закончился, осуществляется переход к его новому этапу (шаг 1). Иначе конец.

Задача 3 Задано:

– выходная гамма из 10 элементов:

$g = \langle 0.30, 0.52, 0.65, 0.85, 1.05, 2.10, 3.00, 4.00, 4.60, 5.00 \rangle$ ;

– область существования параметров функциональных элементов

$\Lambda = [0.4166, 0.8333] \cup [1.2, 2.40]$ .

Полученный функциональный граф изображен на рис. 4.

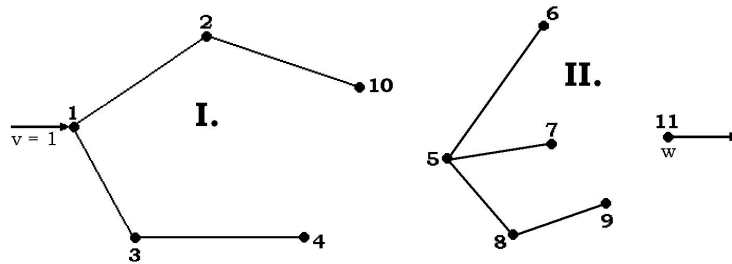


Рис. 4. Функциональный граф для примера 3.

Значения параметров ФЭ при этом равны.

$$\lambda_{12} = 2.047, \quad \lambda_{56} = 2.110,$$

$$\lambda_{13} = 2.219, \quad \lambda_{57} = 1.771,$$

$$\lambda_{34} = 2.269, \quad \lambda_{58} = 1.770,$$

$$\lambda_{210} = 2.223, \quad \lambda_{89} = 1.785,$$

Значения полученных выходных сигналов: 0.295, 0.527, 0.649, 0.865, 1.045, 2.110, 2.991, 3.988, 4.588, 4.953.

Время, затраченное компьютером с процессором Pentium-133 и ОЗУ 64 Мб на решение этой задачи, составляет 3.2 сек.

Приведенный алгоритм можно применить также для синтеза схем с числом каскадов больше двух.

## Минимизация элементов управления

Здесь будет рассмотрен класс многокаскадных схем, реализующих максимальное число режимов. Известно [3], что такие схемы имеют единственную структуру (рис 5), а значит отпадает необходимость в нахождении функционального графа.

Далее рассмотрим задачу минимизации элементов управления в таких многокаскадных схемах.

Пусть матрица  $h[1 : r_m, 1 : d]$  содержит как положительные, так и отрицательные единицы. Понятно, что в этом случае максимальное число режимов  $r_m = 3^d$ . Обозначим такую систему  $\Sigma_{\max}$ .

Структурный граф  $\Gamma = \langle Z, E \rangle$  изображен на рис. 5.

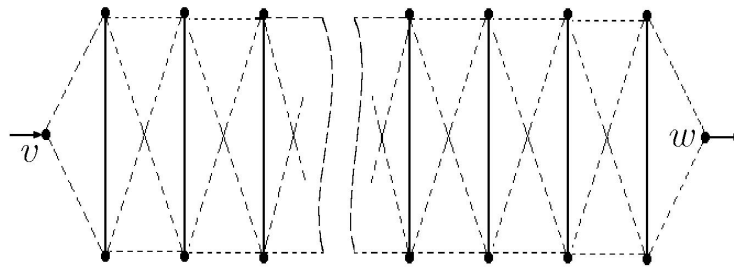


Рис. 5. Структурный граф системы  $\Sigma_{\max}$ .

Сплошные отрезки на рис. 5 соответствуют функциональным элементам (ФЭ), а пунктирные линии – элементам управления (ЭУ) графа  $\Gamma$ ; вход и выход как и прежде обозначены через  $v$  и  $w$  соответственно.

Максимальное число элементов управления в этом графе равно  $4d$ .

Пусть каждому режиму  $i$ , а, значит, и строке  $h[i, 1 : d]$  на графе  $\Gamma$  соответствует путь (цепь) от входной вершины к выходной и пусть этот путь формируется по следующим правилам:

– если параметр  $\lambda_k$  имеет степень равную  $+1$  (т.е.  $h[i, k] = +1$ ), то цепь режима  $i$  проходит по  $k$ -му ребру графа снизу вверх;

– если  $\lambda_k$  имеет степень равную  $-1$  (т.е.  $h[i, k] = -1$ ), то цепь режима  $i$  проходит по  $k$ -му ФЭ сверху вниз;

– в противном случае (т.е.  $h[i, k] = 0$ )  $\lambda_k$  цепь не содержит  $k$ -е ребро.

Предположим, что для системы  $\Sigma_{\max}$ , приведенной на рисунке 5, решена следующая задача синтеза.

Задано:

– число функциональных элементов  $d$ ,

– выходная гамма  $g = \langle g_1, g_2, \dots, g_l \rangle$ ,  $l \leq 3^d$ ,

– область существования параметров ФЭ –  $\Lambda = [1/\lambda^+, 1/\lambda^-] \cup [\lambda^-, \lambda^+]$ .

Необходимо определить:

– такие значения параметров  $\lambda_k \in \Lambda$ ,

– такое инъективное отображение  $\varphi : (1 : l) \rightarrow (1 : r_m)$ , определяющее, на каком из  $r_m$  режимов реализуется заданное значение выходной гаммы  $g_i$ , при которых критерий  $F_g$  принимает минимальное значение.

Алгоритм решения этой задачи аналогичен описанным выше. Поэтому условимся, что она решена и получена матрица  $h[P, 1 : d]$ , где  $P = \varphi(1 : l) \subseteq 1 : r_m$ , содержащая набор степеней для соответствующих выходных функций (10).

В матрице  $h[P, 1 : d]$  число режимов  $l < 3^d$ , поэтому и необходимое для их реализации число элементов управления может быть меньше, чем  $4d$ .

В связи с этим поставим такую задачу: для матрицы  $h[P, 1 : d]$ ,  $|P| = l$ ,  $l \leq 3^d$ , найти такое минимальное число ЭУ, при котором реализуются режимы, определяемые этой матрицей.

Для решения этой задачи предлагается подход, представляющий собой модернизацию эвристического метода, предложенного в [4].

Рассмотрим два соседних каскада  $t$  и  $t + 1$ ,  $t \in 1 : (d - 1)$ , и перенумеруем возможные между ними ЭУ так, как указано на рис. 6.

Пусть  $x_i$ ,  $i \in 0 : 3$ , логическая переменная, такая, что  $x_i = 1$ , если  $i$ -ый элемент управления присутствует между каскадами, и  $x_i = 0$ , в противном случае (рис. 6).

И пусть  $S_i^{t,t+1}$  логическая функция, описывающая возможные элементы управления на  $i$ -ом режиме между каскадами  $t$  и  $t + 1$ . Например, если  $h(i, t) = 0$  и  $h(i, t + 1) = 1$ , то возможные ЭУ имеют номера 1 и 2. На рис. 6 это означает, что соответствующая цепь должна пройти мимо  $t$ -го каскада и снизу вверх —  $(t + 1)$ -ый каскад. Соответствующее логическое уравнение имеет вид:  $x_1 \vee x_2 = 1$ .

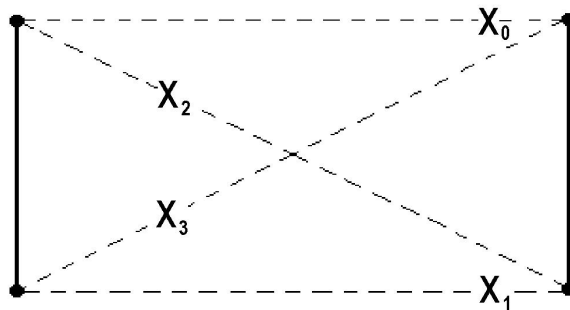


Рис. 6. Размещение ЭУ между двумя соседними каскадами.

В таблице 1 приведены все возможные логические уравнения, описывающие наличие элементов управления, которые могут существовать между соседними каскадами.

Понятно, что минимальное число сомножителей в конъюнкции

$$S_P^{t,t+1} = \bigwedge_{i \in P} S_i^{t,t+1} = 1, \quad (24)$$

определяет минимальное число элементов управления, необходимых для реализации  $l$  выходных функций и находящихся между функциональными элементами  $t$  и  $t + 1$ .

Тогда как минимальная конъюнкция в функции

$$S_{1:d} = \bigwedge_{i \in P, t \in 1:(d-1)} S_i^{t,t+1} = 1 \quad (25)$$

определяет минимально необходимое число элементов управления между всеми ЭУ, необходимое для реализации  $l$  выходных функций при заданном порядке расположения столбцов матрицы.

Таблица. 1. Возможные логические уравнения

N	$h(i, t)$	$h(i, t + 1)$	$S_i^{t,t+1}$
1	0	0	$x_0 \vee x_1 \vee x_2 \vee x_3 = 1$
2	0	1	$x_1 \vee x_2 = 1$
3	1	0	$x_0 \vee x_2 = 1$
4	0	-1	$x_0 \vee x_3 = 1$
5	-1	0	$x_1 \vee x_3 = 1$
6	1	-1	$x_0 = 1$
7	-1	1	$x_1 = 1$
6	1	1	$x_2 = 1$
9	-1	-1	$x_3 = 1$

Отсюда следует очевидное решение задачи минимизации числа ЭУ для заданного набора  $l$  выходных функций. Оно заключается в полном переборе всех возможных перестановок столбцов матрицы  $h$ , построении для  $h$  логической функции (25), получении ее дизъюнктивно-нормальной формы и отыскании минимальной конъюнкции.

Однако полный перебор требует больших временных затрат, поэтому здесь предлагается модернизация эвристического подхода, предложенного в [4].

Построим полный граф  $\Gamma_p = \langle V, E \rangle$ , где вершинам соответствуют функциональные элементы системы, входной и выходной полюса.

Назначим каждому ребру  $(k, s) \in E$  вес, равный числу ЭУ, необходимых для реализации всех выходных функций для функциональных элементов  $k$  и  $s$ . Это число можно определить из (24) нахождением конъюнктивного члена с минимальным количеством элементов. Однако этот процесс можно оптимизировать с точки зрения машинного времени и в этом заключается первая модификация. Поясним на примере суть предлагаемой модификации.

Пусть

$$\begin{pmatrix} h(k) & h(s) \\ 0 & 1 \\ 0 & -1 \\ 0 & 0 \\ 1 & 1 \\ 1 & -1 \\ 1 & 0 \end{pmatrix} \quad (26)$$

значения степеней при  $\lambda_k$  и  $\lambda_s$  в выходных функциях между каскадами  $k$  и  $s$ . Логическое уравнение, соответствующее (26) выглядит следующим образом:

$$(x_1 \vee x_2) \wedge (x_0 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (x_0 \vee x_2) \wedge x_0 \wedge x_2 = 1. \quad (27)$$

Раскрывая скобки и применяя операцию поглощения ( $ab \vee b = a$ ), можно получить:  $x_0 \wedge x_2 = 1$ .

Произведенные выше операции можно значительно упростить с точки зрения использования машинного времени. Для этого над (26), перед тем как выписывать логическое уравнение типа (27), необходимо выполнить следующие преобразования:

- исключить пары  $(i = 0, j = 0)$ :
- любую пару  $(i, j)$ , такую, что  $i = 0, j = b, b \in \{1, -1\}$ , исключить, если существует пара  $(1, b)$  или  $(-1, b)$ :
- любую пару  $(i, j)$ , такую, что  $i = b, j = 0, b \in \{1, -1\}$ , исключить, если существует пара  $(b, 1)$  или  $(b, -1)$ .

После применения перечисленных выше преобразований над (26) получим:

$$\begin{pmatrix} i & j \\ 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Таким образом, описанные преобразования позволяют значительно упростить процесс взвешивания полного графа  $\Gamma$  для случая большого числа выходных функций.

После взвешивания полного графа необходимо найти в нем все минимальные гамильтоновы цепи, которые дадут нижнюю границу числа ЭУ и, тем самым, обоснованные перестановки ФЭ или столбцов матрицы  $h$ . Далее для каждой такой перестановки, определив конъюнктивный член с минимальным количеством элементов, найдем необходимое число элементов управления и тем самым определим минимальное количество элементов управления, необходимое для реализации всех заданных выходных функций. Заметим, что в работе [4] не делалось предположений о том, что минимальных гамильтоновых цепей может быть несколько (а находилась лишь одна) и для полученных перестановок не находился минимальный конъюнктивный член, а проводился анализ матрицы  $h$ , что не всегда дает минимальное число элементов управления. В этом заключаются вторая и третья модификации способа, предложенного в [4].

Далее несколько слов о способе, использованном в этой работе для нахождения всех гамильтоновых цепей минимальной длины.

Так как на сегодня не существует известного способа получения всех гамильтоновых цепей минимальной длины, кроме полного перебора, здесь для этих целей использовалась комбинация алгоритма Кристофидеса [5] и метода ветвей и границ, т.е. поиск гамильтоновых цепей осуществлялся путем построения полного дерева решений с отсечением веток на основе минимальной гамильтоновой цепи, полученной алгоритмом Кристофидеса.

### Пример 3.

Исходная матрица:

$$\begin{pmatrix} -1 & 0 & 1 & -1 & 0 \\ -1 & 1 & 0 & -1 & -1 \\ 0 & -1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ -1 & 1 & 0 & -1 & -1 \end{pmatrix}$$

Для исходной матрицы было найдено 15 гамильтоновых цепей минимальной длины. Матрица, соответствующая одной из таких цепей имеет вид:

$$\begin{pmatrix} 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & -1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & -1 & -1 \end{pmatrix}$$

Схема, соответствующая этой матрице, представлена на рис. 7.

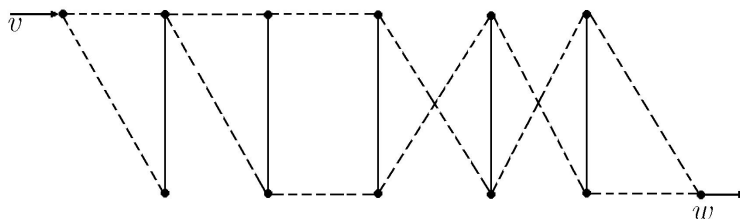


Рис. 7. Одна из возможных схем с минимальным числом ЭУ.

В заключении отметим, что все описанные выше алгоритмы синтеза доведены до программного уровня и позволяют решать задачи синтеза в упрощенном диалоговом режиме.

### Литература

1. Сушков Ю.А. Метод, алгоритм и программа случайного поиска // – Л. ВНИИТрансМаш, 1969. – 43 с.
2. Сушков Ю.А. Связность в гиперграфах и матроидах. Исследование операций и статистическое моделирование, – С-Пб.: 1994. Вып.6. – С.111-138.

3. *Сушков Ю.А.* Структурно управляемые системы. – Автореферат диссертации на соискание ученой степени доктора физико-математических наук, С-Пб.: 2001, 20с.
4. *Сушков Ю.А., Фатташ И.* Синтез многорежимных систем из двухполюсников // Дискретные системы и их программное обеспечение. – Л., ЛГУ. 1991. Вып.26. – С.30-43.
5. *Кристофидес Н.* Теория графов. Алгоритмический подход. – М., Мир, 1978. – 432 с.

## On synthesis of one type multimode systems

**A. Sh. Abakarov, Yu. A. Sushkhov**

In paper the questions of synthesis systems by tuning on specific regime with change of connections between its component on the basis of bipolar  $\lambda$ -reversible elements are considered. These systems denominate the multimode or structural operated systems. The algorithm of minimization of operated elements in schemes with maximal number routines and partial using is represented. The minimization of operated elements is based on the modified algorithm of searching minimal cycle hamiltonian and Mary method.