

ВРЕМЕННОЙ АНАЛИЗ КРУПНЫХ СЕТЕВЫХ ГРАФИКОВ БЛОЧНОЙ СТРУКТУРЫ

Ю. Н. ЕФИМОВ

(Представлена научным семинаром УВЛ ТПИ)

Известные программы [1—4] анализа сетевых графиков используют только оперативную память (ОЗУ) электронных машин и обрабатывают сравнительно небольшие сети. Анализ же крупных сетевых графиков на ЭЦВМ с ограниченным объемом оперативной памяти представляет собой довольно трудную задачу, требующую привлечения внешней памяти. Для анализа крупных сетей сравнительно просто разработать программы, реализующие те же алгоритмы и использующие внешний накопитель в качестве оперативной памяти. Однако частое обращение к НМЛ снижает надежность вычислений и настолько увеличивает машинное время, что делает временной анализ большинства графиков нереальным. Возникает проблема разработки методов и приемов наиболее эффективного использования ОЗУ с тем, чтобы при анализе крупных сетей сократить до минимума число обращений к НМЛ. Очевидно, что эти методы должны быть основаны на разбиении исходной сети на части и анализе этих частей в отдельности.

Всякая моделируемая разработка состоит из отдельных, более или менее обособленных частей, что находит отражение и на модели в виде графически обособленных комплексов работ и событий. Каждый такой комплекс (назовем его блоком) представляет собой подграфик $G_k = (I_k, U_k)$ исходного сетевого графика $G = (I, U)$, где $I_k U_k$ — соответственно множество событий и работ этого подграфика, причем $I_k \subset I, U_k \subset U$. Ставится задача построения по известным подграфикам G_k укрупненного (менее детализированного) сетевого графика G^1 , сохраняющего основные параметры и взаимосвязи исходного графика G . Очевидно, что укрупненный график G^1 будет содержать значительно меньшее число работ, чем исходный G , и может быть обработан обычными методами. Таким образом, необходимо разработать систему эквивалентных преобразований, при помощи которой блок можно свести к одной или нескольким работам.

Введем некоторые определения и обозначения.

События блока G_k будем обозначать i^k, j^k , где k — номер блока. Событие $i_0^k \in I_k$ назовем исходным событием блока G_k , если

$$\Gamma^{-1}i_0^k \cap I_k \neq \Gamma^{-1}i_0^k. \quad (1)$$

В общем случае блок может иметь некоторое подмножество Ei_0^k исходных событий. Очевидно, что исходные события Ei_0^k графика G являют-

ся исходными событиями и блока, в который они входят (для них $\Gamma^{-1}Ei_0 \cap I = \emptyset$), т. е. имеем

$$Ei_0 \subset Ei_0^k \text{ при } Ei_0 \subset I_k. \quad (2)$$

Событие $i_n^k \in I_k$ назовем завершающим событием блока G_k , если

$$\Gamma i_n^k \cap I_k \neq \Gamma i_n^k. \quad (3)$$

По аналогии с исходными событиями вводим понятие подмножества Ei_n^k завершающих событий блока G_k и получаем выражение

$$Ei_n \subset Ei_n^k \text{ при } Ei_n \subset I_k \text{ (так как } \Gamma Ei_n \cap I = \emptyset \text{)}. \quad (4)$$

Очевидно, что

$$Ei_n^k \equiv Ei_0^{k+1}. \quad (5)$$

Исходные и завершающие события блока G_k составляют подмножество граничных событий \bar{I}_k этого блока

$$\bar{I}_k = Ei_0^k \cup Ei_n^k. \quad (6)$$

Укрупненным сетевым графиком G^1 данного графа G будем называть сетевой график, событиями которого являются все события множества I_s :

$$I_s = \bigcup_{k=1}^m \bar{I}_k, \quad (7)$$

где m — число блоков в G .

Всякая работа (i^s, j^s) длиной $t(i^s, j^s)$ будет являться работой укрупненного графика G^1 при соблюдении следующих условий:

а) $i^s \in Ei_0^k, j^s \in Ei_n^k$;

б) существует один или несколько путей, ведущих из i^s в j^s в графе G ;

в) величина $t(i^s, j^s)$ равна длине максимального из этих путей.

Нетрудно показать, что для построенного описанным способом укрупненного графика G^1 имеют место равенства

$$t_p'(j) = t_p(j), \quad (8)$$

$$t_n'(j) = t_n(j), \quad (9)$$

где $t_p'(j), t_n'(j), t_p(i), t_n(j)$ — ранние и поздние времена свершения события j соответственно в G^1 и G .

Таким образом, основную задачу обработки сетевой модели блочной структуры можно свести к следующему:

а) найти в блоке критические пути, ведущие из Ei_0^k в Ei_n^k ;

б) вычислить длины этих путей;

в) сформировать информацию о графе G^1 в виде

$$(m) i^s j^s t(i^s, j^s), \quad (10)$$

где

m — номер ячейки ОЗУ машины,

i^s, j^s — соответственно начальное и конечное события работы искомого графа G' ,

$t(i^s, j^s)$ — продолжительность этой работы.

Для решения поставленной задачи применим следующую нумерацию исходного графика G (рис. 1). Каждое событие $i \in I_s$ разделим на три сектора. Пронумеруем события подмножества I_s по порядку (от

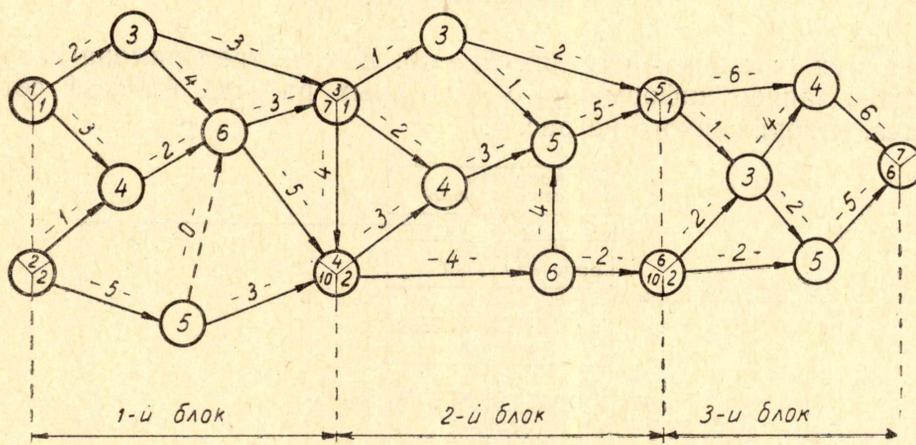


Рис. 1. Сетевая модель системы с нумерацией событий удобной для машинного анализа. (Выделены события подмножества I_s)

единицы подряд) в восьмиричной системе счисления, и номера запишем в верхнем секторе. Это будут номера (внешние) искомого графика G' . Далее пронумеруем события $i^k \in I_k$ внутри каждого блока, причем начальные номера присвоим исходным событиям блока. Полученные внутренние номера событий блока записываются в правом секторе для $i^k \in Ei_0^k$ и в левом для $i^k \in Ei_n^k$. Таким образом, каждое событие исходного графика G оказывается пронумерованным, причем граничные события имеют по три номера, а исходные и завершающие — по два номера (рис. 1). В процессе анализа в зависимости от того, в какой системе рассматривается событие, берется один из трех номеров. Следует отметить, что предлагаемая нумерация событий позволяет анализировать отдельные блоки независимо друг от друга, рассматривая их в любом порядке.

Рассмотрим методику и программу анализа сетевых графиков с описанной нумерацией событий на ЭЦВМ «Минск-1».

Исходная информация о графе G записывается на магнитной ленте в виде отдельных массивов. Каждый массив содержит информацию об одном блоке, которая включает в себя описание блока и перечень всех работ. В описании блока содержатся списки исходных и завершающих событий блока (их внешние номера), а также номер блока и контрольная сумма массива. Топология блока и временные характеристики работ задаются в виде перечня, каждая строка которого характеризует одну работу и имеет вид

$$P_2 i^k j^k t(i^k, j^k) P_1, \quad (11)$$

где

i^k, j^k — внутренние номера событий, определяющих работу;

$t(i^k, j^k)$ — длина работы;

P_1 — признак завершающей работы.

$P_1 > 0$ для работ, входящих в завершающее событие блока, причем величина P_1 берется равной порядковому номеру этого события в описании рассматриваемого блока.

$P_1 = 0$ для всех остальных работ блока.

$P_2 = 1$ только для завершающих работ, конечные события которых удовлетворяют условию:

$$\Gamma E i_n^k \cap I_k \neq \Phi. \quad (12)$$

Во всех остальных случаях $P_2 = 0$. В качестве примера в табл. 1 приведена запись информации о некоторых работах блока 1 сетевого графика, изображенного на рис. 1.

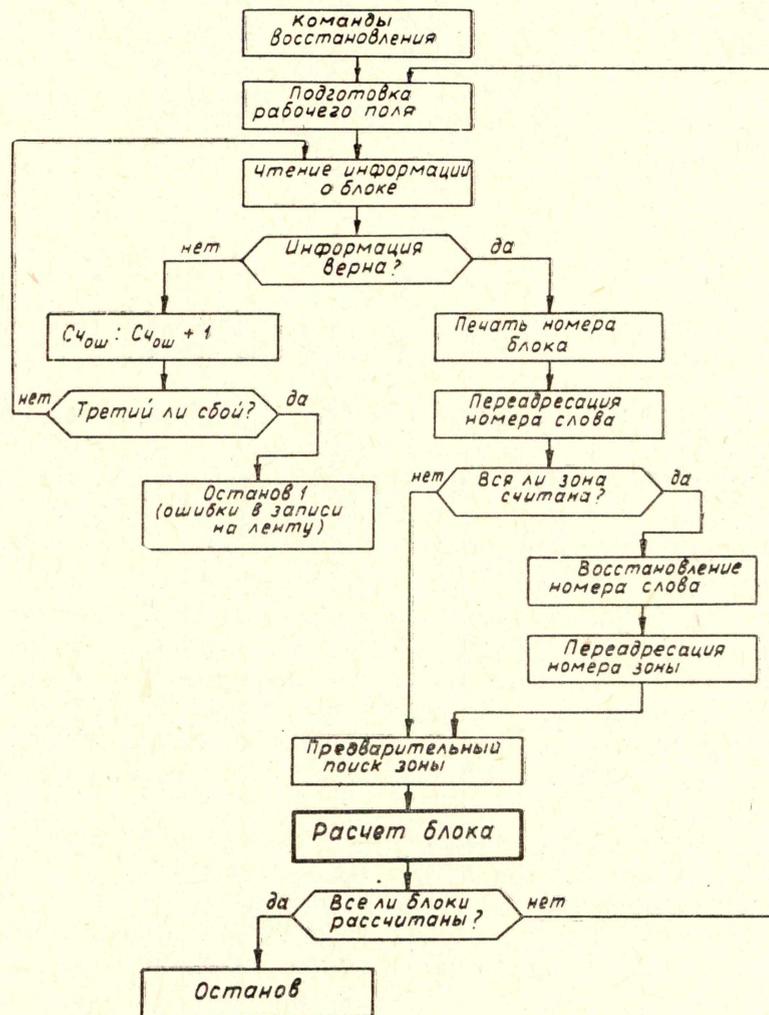


Рис. 2. Блок-схема программы анализа крупных сетевых графиков блочной структуры

На рис. 2 представлена блок-схема программы анализа крупных сетевых графиков блочной структуры. Информация о каждом блоке считывается с магнитной ленты в ОЗУ машины. После проверки правильности считанной информации печатается номер анализируемого блока, организуется предварительный поиск зоны магнитной ленты с информацией о следующем блоке и затем производится расчет блока.

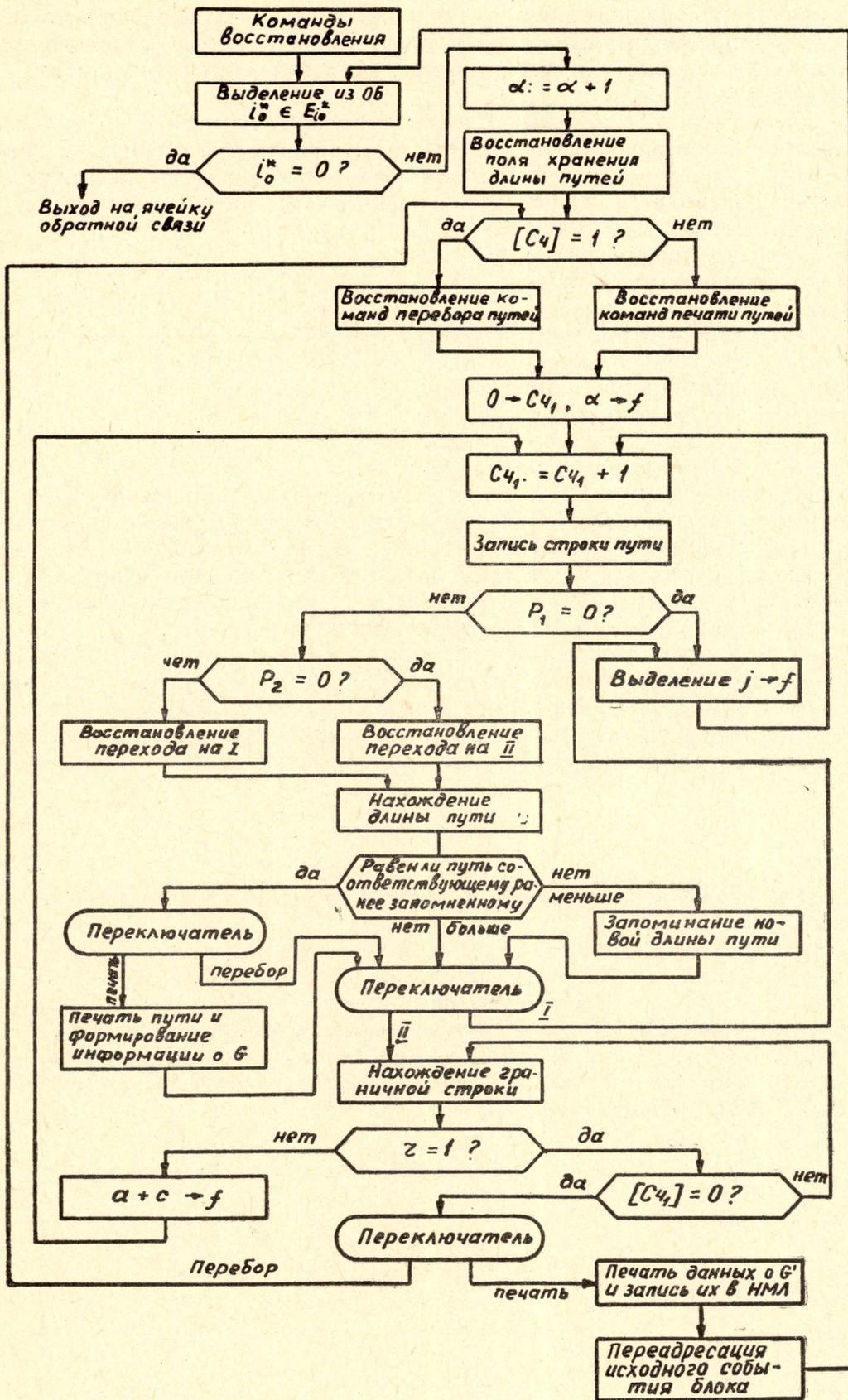


Рис. 3. Блок-схема программы временного анализа блоков

Такая процедура повторяется для каждого $G_k \subset G$. Необходимо отметить, что предварительный поиск зоны намного сокращает общее время анализа, так как практически за время поиска зоны производится полный расчет блока.

Для расчета отдельных блоков используется несколько видоизмененный алгоритм полного перебора [3], приспособленный для анализа сетевых графиков многоцелевых разработок. Обработка информации о блоке производится в два этапа (блок-схема на рис. 3), причем на каждом из них используются одни и те же участки программы, по-разному соединенные с помощью переключателей.

На первом этапе производится построение всех возможных путей в блоке и запоминание n длин максимальных путей, соответственно для каждого завершающего события блока (значение n равно количеству завершающих событий в блоке). На втором этапе производится повторное построение матриц путей и сравнение их длин с ранее запомненными. При совпадении получаемых длин с соответствующими ранее запомненными печатаются матрицы путей и формируется информация о графе G' в форме (10). Информация о графе G' выводится на перфоратор или по мере накопления переписывается на ленту.

В результате анализа всех блоков получаем исходную информацию об укрупненном графике G' , и детальную расшифровку всех его работ в виде множества матриц, выведенных на печать. Очевидно, что после анализа укрупненного графика G' , используя эту расшифровку, легко построить критический путь в исходном графике G .

Таблица 1

P_2	i^k	j^k	$t(i^k, j^k)$	P_1
0	1	3	2	0
0	4	6	2	0
1	6	7	3	1
0	6	10	5	2
0	5	6	0	0

ЛИТЕРАТУРА

1. А. М. Горлов. Расчет сетевых графиков по методу критического пути. Гипротис. М., 1965.
2. Г. М. Адельсон-Вельский, Ф. М. Филлер. Программа вычисления сетевых графиков. Ж. вычислит. мат. и мат. физики, том. 5, № 1. 1965.
3. Ю. Н. Ефимов. Анализ сетевых графиков на ЭЦВМ «Минск-1». Аннотации программ, решенных на цифровых вычислительных машинах. Вып. 4. М., 1967.
4. А. Т. Петрова, Н. Н. Карнаухова. Об одном алгоритме нахождения критического пути сетевого графика. Сб. «Вычислительные системы». Вып. 11. Изд. ИМ СО АН СССР, Новосибирск. 1964.