

АЛГОРИТМ НАХОЖДЕНИЯ ПОРЯДКОВОЙ ФУНКЦИИ СЕТЕВОГО ГРАФИКА

Ю. Н. ЕФИМОВ

(Представлена научным семинаром вычислительной лаборатории ТПИ)

При машинном анализе сложного сетевого графика, являющегося динамической моделью комплекса разработок, часто возникает задача предварительного упорядочения событий. Сущность упорядочения сводится к разбиению множества событий сетевого графика на отдельные подмножества — классы [1]. Такое разбиение полезно, когда необходимо ввести в график новые события и работы, вычертить сетевой график или часть его. Предварительное упорядочение событий по классам позволяет также ускорить процесс временного анализа сетевого графика. Так, для нахождения времени наиболее раннего свершения события $[tp(i)]$ [1, 2, 3] достаточно всего лишь одного просмотра исходной информации.

Для разбиения событий сетевого графика на классы находят для каждого события число, называемое порядком. Затем исходную информацию упорядочивают по мере возрастания порядка начальных (для расчета $tp(i)$) или конечных (для $tn(i)$) событий работ. Дадим более строгие определения.

Пусть задан сетевой график без контуров [4] $G = (I, \Gamma^{-1})$, где I — непустое множество событий графика $I \neq \emptyset$;

Γ^{-1} отображение множества I в I .

Рассмотрим некоторые подмножества I .

$$I(0) = \{i | \Gamma i^{-1} = \emptyset\}$$

$$I(1) = \{i | \Gamma i^{-1} \subset I(0)\}$$

$$\dots \dots \dots$$

$$I(k) = \{i | \Gamma i^{-1} \subset I(k-1)\}.$$

В общем случае $I(n) = \bigcup_{k=1}^n I(k)$.

Очевидно, если $\alpha < \beta$, то $I(\alpha) \subset I(\beta)$. Порядком события i сетевого графика $G = (I, \Gamma^{-1})$ назовем наименьшее порядковое число α , для которого $i \in I(\alpha)$; $i \notin I(\beta)$ при всех $\beta < \alpha$. Полагаем тогда, что $\alpha = \varphi(i)$, где $\varphi(i)$ — порядковая функция сетевого графика. Используя отображение Γ , обратное Γ^{-1} , можно получить подмножества

$$I^{-1}(0) = \{i | \Gamma i = \emptyset\}$$

$$\dots \dots \dots$$

$$I^{-1}(k) = \{i | \Gamma i \subset I^{-1}(k-1)\}.$$

В общем случае $I^{-1}(n) = \bigcup_{k=0}^n I^{-1}(k)$.

Аналогично получим обратный порядок события α^{-1} и обратную порядковую функцию сетевого графика $\varphi^{-1}(i)$. Очевидно, что между функциями $\varphi(i)$ и $\varphi^{-1}(i)$ существует однозначная зависимость:

$$\varphi^{-1}(i) = \varphi_{\max_{i \in I}}(i) - \varphi(i), \text{ причем}$$

$$\varphi_{\max_{i \in I}}^{-1}(i) = \varphi_{\max_{i \in I}}(i), \text{ где максимум}$$

берется по всем событиям $i \in I$. Приняв длительность каждой работы сетевого графика равной единице, получим $\varphi(i) = tp(i)$, где $tp(i)$ — наиболее ранний срок свершения события i .

Так что для нахождения порядковой функции можно воспользоваться любым алгоритмом вычисления $tp(i)$ [1, 3, 5, 6]. Нахождение $\varphi(i)$ с целью ускорения расчетов временных параметров (используя упорядочение по классам) не имеет смысла, если для вычисления порядковой функции использовать те же алгоритмы, что и для временного анализа. Представляется целесообразным разработать специальные алгоритмы, которые бы полнее учитывали как специфику поставленной задачи, так и выигрышные особенности ЭЦВМ, на которой они реализуются.

Опишем один из возможных вариантов решения задачи нахождения порядковой функции сетевого графика на двухадресной вычислительной машине „Минск-1“. Исходная информация (UU) о сетевом графике задается в виде списка работ и их длин, расположенных в любой последовательности. Производим упорядочение работ UU по ходу их начал [6] и получаем информацию вида

$$i) \ r \ a \ j t_{ij} \varphi(i), \text{ где}$$

i — номер ячейки памяти машины, равный номеру начального события работы;

j — номер конечного события одной из работ, исходящей из события i ;

r — число работ, исходящих из события i ;

a — условный адрес другой работы, исходящей из события i .

Действительный адрес $(a + c)$, где $c = \text{const}$. Обычно c принимают равной наибольшему номеру события сетевого графика.

$\varphi(i)$ — порядковая функция. Пока $\varphi(i) = 0$ для всех $i \in I$.

Из определения порядковой функции сетевого графика следует, что если все события, имеющие $\varphi(i) = \rho$, образуют известное подмножество $I_1(\rho) \subset I$, то все события подмножества $I_1(\rho + 1) = \{i/i \in I, I_1(\rho)\}$ принимают значения $\varphi(i) = \rho + 1$. При этом, если какое-нибудь событие подмножества $I_1(\rho + 1)$ имеет уже вычисленную функцию порядка, то ее следует заменить новой. Отсюда работа алгоритма сводится к следующему. Исходному событию i_0 присвоим значение $\varphi(i_0) = 0$. На первом шаге находим подмножество событий Γi_0 и присваиваем им значение $\varphi(i) = 1$. Для этого из содержимого ячейки i_0 выделяются величины $j \in \Gamma i_0$ и a . По адресу j заносится значение $\varphi(j) = 1$, а из содержимого ячейки $(a + c)$ вновь выделяются величины j и a . Процесс продолжается до выделения величины $a = 0$, что свидетельствует о присвоении всем событиям подмножества Γi_0 значения $\varphi(i) = 1$. На втором шаге для каждого события i , имеющего $\varphi(i) = 1$, находим подмножество Γi только что описанным способом. Каждому событию подмножества Γi присваиваем $\varphi(i) = 2$. На k -м шаге для событий, имеющих $\varphi(i) = k$, находим подмножества Γi и присваиваем значение $\varphi(i) = k + 1$.

Процесс продолжается до тех пор, пока на каком-то шаге все найденные подмножества $\Gamma_i = \emptyset$. На этом вычисление порядковой функции сетевого графика заканчивается. Очевидно, что количество описанных шагов равно числу $[\varphi_{\max}(i) + 1]$. Почти полное отсутствие арифметических операций, а также необходимость просмотра на каждом шаге только списка событий, а не общего списка работ, делают алгоритм более быстродействующим по сравнению с известными алгоритмами для расчета $tp(i)$ и $tn(i)$.

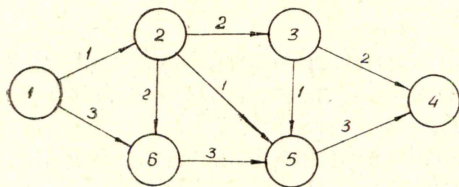


Рис. 1. Пример сетевого графика

Рассмотрим описанный алгоритм на примере. Пусть UU о сетевом графике (рис. 1) задана на табл. 1. Программой упорядочения [6] при $C = 6$ (т.к. 6—наибольший номер события) информация пре-

образуется к виду, представленному левой частью табл. 2. Временные оценки и значения r во внимание не принимаются, так как они не участвуют в работе алгоритма. В правой части табл. 2

Таблица 1

i	j	t_{ij}
1	2	1
2	3	2
3	4	2
2	5	1
5	4	3
3	5	1
6	5	3
2	6	2
1	6	3

i	r	a	j	$\varphi(i)$ на каждом шаге						$\varphi(i)$
				0	1	2	3	4	5	
1	2	4	2	0						0
2	3	1	3	0	1					1
3	2	2	4	0		2				2
4				0			3	4		4
5	1	0	4	0		2	3			3
6	1	0	5	0	1	2				2
7	2	3	5							
10	1	0	5							
11	1	0	6							
12	1	0	6							

даны значения $\varphi(i)$, вычисляемые на каждом шаге. Пустые клетки означают, что значение $\varphi(i)$ осталось без изменения. На первом шаге из $i_0 = 1$ выделяем $j = 2$ и $a = 4$. В ячейку 2 заносим $\varphi(2) = 1$, а из ячейки $a + c = 4 + 6 = 12_8$ выделяем $j = 6$, $a = 0$. В ячейку 6 заносим $\varphi(6) = 1$ и на этом заканчиваем первый шаг, так как $a = 0$. На втором шаге из ячейки 2 [так как $\varphi(2) = 1$] выделяем $j = 3$, $a = 1$. Вносим $\varphi(3) = 2$. Из ячейки $a + c = 6 + 1 = 7_8$ выделяем $j = 5$, $a = 3$. Вносим $\varphi(5) = 2$. Из ячейки $6 + 3 = 11_8$ выделяем $j = 6$, $a = 0$. Вносим $\varphi(6) = 2$. Так как событие 6 к моменту его рассмотрения изменило уже значение $\varphi(6)$ с 1 на 2, то для него Γ_6 не находим (за счет этого процесс расчета несколько ускоряется). Действуя таким образом, заполняем правую часть табл. 2. На 5-м шаге никаких изменений в значения $\varphi(i)$ не вносится, так как событие 4 не имеет исходящих работ ($\Gamma_4 = \emptyset$). В последнем столбце табл. 2 выписаны получившиеся значения $\varphi(i)$.

ЛИТЕРАТУРА

1. К. А. Багриновский, И. Б. Рабинович. Постановка задачи анализа сетевого графика. Сб. «Вычислительные системы», вып. II, Новосибирск, из-во ИМ СО АН СССР, 1964.
2. Г. С. Поспелов, А. И. Тейман. Автоматизация процессов управления работками больших систем или сложных комплексов. Изд-во АН СССР, серия Техн. кибернетики, № 4, 1963.
3. L. R. Ford. Network flow-theory, Rand Corp. paper, P-923, 1956.
4. К. Берж. Теория графов и ее применения. М., Издательство ИЛ., 1962.
5. Г. М. Адельсон-Вельский, Ф. М. Филлер. Программа вычисления сетевых графиков. Ж. вычислительной математики и матем. физики, т. 5, № 1, 1965.
6. Ю. Н. Ефимов, В. И. Кизёв, В. М. Разин. Об одном алгоритме автоматизации расчетов для процессов управления разработками сложных систем. Изв. ТПИ, т. 154, изд. ТГУ, 1967.