А К А Д Е М И Я Н А У К С С С Р
СИБИРСКОЕ ОТДЕЛЕНИЕ
А В Т О М Е Т Р И Я

№ 1 1972

УДК 519.241.5

В. И. ЗНАК
(Новосибирск)

АЛГОРИТМ АПРИОРНОЙ ОЦЕНКИ ВРЕМЕНИ МАШИННОЙ РЕАЛИЗАЦИИ ПРОГРАММЫ

I. При создании математического обеспечения цифровых вычислительных машин (ЦВМ), работающих в комплексах обработки экспериментальных данных, в ряде случаев требуется знание времени выполнения алгоритмов, подлежащих реализации на ЦВМ. Прежде чем сформулировать задачу данной работы, отметим, что основой работы ЦВМ является программа, т. е. некоторая наперед заданная система предписаний (обозначим U), определяющая порядок элементарных актов обработки информации, поступающей в ЦВМ. Тогда задача состоит в том, чтобы оценить время реализации $T(U)$ программы U, представляющей тот или иной алгоритм. В предлагаемой работе построен достаточно общий алгоритм априорной оценки времени реализации программы (см. II), даны решающие схемы алгоритма (см. III) и приведены некоторые результаты исследования названных схем (см. IV). Основные идеи раздела II статьи изложены в [1].

II. Всякую программу (или алгоритм, реализуемый такой программой) можно интерпретировать в виде ориентированного графа [2]. Операторы программы (или соответствующего алгоритма) задают вершины такого графа (и могут иметь, в свою очередь, достаточно сложную структуру). Дуги же графа определяются передачами управления между операторами программы.

Рассмотрим такой граф $G=G(V)$, из каждой вершины которого $R_j \in V (j=1,\dots,m)$ исходит не более двух дуг. При этом на графе G имеется одна исходная вершина (или начало графа) $\Lambda \in V$, в которую не входит ни одна дуга, и одна конечная вершина (или просто конец графа) $Q \in V$, из которой не исходит ни одной дуги.

Выделим далее на графе $G=G(V)$ множество $L \subset V$ вершин таких, что из каждой $R_j \in L$ исходит пара дуг — r_j^1 и r_j^0. Дуги, исходящие из вершин $R_j \not\subset L$, обозначим r_j^*. Когда вид конкретной дуги, исходящей из вершины $R_j \in V$, не будет представлять интереса, обозначим ее $r_j^{\Delta_j}$. Здесь Δ_j есть или *, или σ_j, или $\bar{\sigma}_j$, где σ_j (как и $\bar{\sigma}_j$) равно или 0, или 1 ($\bar{\sigma}_j$ инверсно σ_j, т. е. если $\sigma_j=0$, то $\bar{\sigma}=1$; если $\sigma_j=1$, то $\sigma_j=0$).

Путем на графе G назовем такую последовательность дуг

$$\alpha = \left(\dots r_{j_0}^{\Delta_{j_0}} r_{j_1}^{\Delta_{j_1}} \dots r_{j_n}^{\Delta_{j_n}} \dots\right), \tag{1}$$

что каждые две соседних дуги $r_{j_{l-1}}^{\Delta j_{l-1}}$ и $r_{j_l}^{\Delta j_l}$ $(l=1, \ldots, n)$ имеют общую вершину R_{jl} и дуги проходят в направлении их ориентации.

Путь, для которого никакая вершина не повторяется, назовем простым путем. Простым полным путем назовем такой путь ψ, для которого $R_{j_0}=\Lambda$, $R_{j_n}=Q$ и никакие промежуточные вершины не повторяются. Путь φ, для которого $R_{j_0}=R_{j_n}$, назовем простым циклом (и заключим в фигурные скобки $\{\varphi\}$), если R_{j_0} не является промежуточной вершиной в $\{\varphi\}$ и никакие другие вершины не повторяются. Отметим для дальнейшего, что любая циклическая перестановка элементов простого цикла дает верное изображение этого цикла.

Алгоритмы выделения простых путей на графе существуют (рассмотрены, например, в [3, 4]). Далее будем полагать, что с помощью одного из известных методов для графа $G=G(V)$ выделено множество Ψ простых полных путей ψ_i $(i=1,\ldots,n)$ и множество Φ простых циклов $\{\varphi_i\}$ $(j=1, \ldots, m)$.

Говорят, что пути α и β «зацепляют» друг друга, если они могут быть представлены в виде

$$\alpha' r_j^{\sigma_j} \alpha'', \quad \beta' r_j^{\bar{\sigma}_j} \beta'', \tag{2}$$

где α', α'', β', β'' — подпути путей α и β соответственно. Если один из таких путей есть простой полный путь $\psi_i \in \Psi$, а другой — простой цикл $\varphi_l\} \in \Phi$, то скажем, что путь ψ_i «порождает» цикл $\{\varphi_l\}$ и запишем новый путь

$$\gamma = \psi_i' \left\{ r_j^{\sigma_j} \varphi_l' \right\} r_j^{\bar{\sigma}_j} \psi_i'', \tag{3}$$

который, естественно, не является простым. Сейчас на множестве Φ можно искать циклы, порождаемые уже рассмотренным циклом $\{\varphi_l\}$. Простые полные пути назовем путями нулевого ранга. Циклы, зацепляющие один из путей ранга $\nu=0$, назовем путями ранга $\nu=1$ и т. д.

Рассмотрим некоторый путь нулевого ранга $\psi_i \in \Psi$ (обозначим его ψ^{i_0}), элементы которого $r_j^{\sigma_j}$ (и только такие элементы) представим в виде $\left(r_j^{\sigma_j}\right)_l^{i_0}$ $(l=l(i_0)=1,\ldots, m(i_0)$; 0 — ранг рассмотренного пути). Далее такие элементы будем обозначать символом $r_l^{i_0}$, полагая, однако, что каждому $r_l^{i_0}$ соответствует дуга r_j^0 (или r_j^1) на графе G. Элемент $r_m^{i_0}$ рассмотренного пути ψ_i назовем старшим. Первыми из циклов $\{\varphi_l\} \in \Phi$, порождаемых таким $\psi_i \in \Psi$, выделим самые внешние циклы (т. е. такие, что каждый из них содержит дугу $r_j^{\bar{\sigma}_j}$, инверсную дуге $r_{j_0}^{\sigma_j}$, соответствующую старшему элементу $r_m^{i_0} \in \psi^{i_0}$), что дает множество $M_m^{i_0} \subset \Phi$. Далее производим циклическую перестановку элементов выделенных циклов таким образом, чтобы первым стоял инверсный элемент $r_j^{\bar{\sigma}_j}$. Циклы $\{\varphi_l\} \in M_m^{i_0}$ представим в виде $\left[\begin{smallmatrix} i_0 \\ m \end{smallmatrix}\right] \varphi^{i_1}$ и их элементы $r_j^{\sigma_j}$ — в виде $\left[\begin{smallmatrix} i_0 \\ m \end{smallmatrix}\right] r_s^{i_1}$ ($i_1=1,\ldots, n_1$ — порядковый номер цикла $\{\varphi_l\}$ на множестве $M_m^{i_0}$} $s=s(i_1)=1, \ldots, m_1(i_1)$ — порядковый номер элемента $r_j^{\sigma_j}$ в цикле $\left[\begin{smallmatrix} i_0 \\ m \end{smallmatrix}\right] \varphi^{i_1}$). Подобную последовательность действий, позволяющую организовать индексы пути ранга $\nu+1$, назовем процедурой порождения путей.

Запишем преобразование

$$\Phi / M_m^{i_0} = \Phi_{m-1}^{i_0} = \left[\begin{smallmatrix} i_0 \\ m \end{smallmatrix}\right] \Phi_{m_1}^{i_1}. \tag{4}$$

Затем в качестве старших рассматриваются: элемент $r^{i_0}_{m-1} \in \psi^{i_0}$ и элементы $\begin{bmatrix} i_0 \\ m \end{bmatrix} r^{i_1}_{m_1}$ циклов $\begin{bmatrix} i_0 \\ m \end{bmatrix} \varphi^{i_1}$, т. е. на множестве $\Phi^{i_0}_{m-1}$ выделяются циклы, содержащие элемент $r^{\sigma_j}_j$, инверсный элементу $r^{i_0}_{m-1}$ и т. д.

Таким образом, приходим к рекурсивной процедуре, которую назовем процедурой ранжировки. Пусть на l-м этапе процедуры ранжировки имеется множество $\begin{bmatrix} i_0, \ldots, i_{\nu-1} \\ b, \ldots, l \end{bmatrix} \Phi^{i_\nu}_s$ циклов и элемент $\begin{bmatrix} i_0, \ldots, i_{\nu-1} \\ b, \ldots, l \end{bmatrix} \times \times r^{i_\nu}_s$ рассматривается в качестве старшего $(i_\nu = 1, \ldots, n_\nu;\ s = = s(i_\nu) = 1, \ldots, m(i_\nu))$.

1. Произведем процедуру порождения путей, в результате которой получаем множество $\begin{bmatrix} i_0, \ldots, i_{\nu-1} \\ b, \ldots, l \end{bmatrix} M^{i_\nu}_s$ путей $\begin{bmatrix} i_0, \ldots, i_{\nu-1}, i_\nu \\ b, \ldots, l, s \end{bmatrix} \varphi^{i_{\nu+1}}$, состоящих из элементов $\begin{bmatrix} i_0, \ldots, i_{\nu-1}, i_\nu \\ b, \ldots, l, s \end{bmatrix} r^{i_{\nu+1}}_x$, где $i_{\nu+1} = 1, \ldots, n_{\nu+1}$; $x = x(i_{\nu+1}) = 1, \ldots, m(i_{\nu+1})$.

2. Из множества $\begin{bmatrix} i_0, \ldots, i_{\nu-1} \\ b, \ldots, l \end{bmatrix} \Phi^{i_\nu}_s$ исключим множество путей $\begin{bmatrix} i_0, \ldots, i_{\nu-1} \\ b, \ldots, l \end{bmatrix} M^{i_\nu}_s$, в результате чего получаем

$$\begin{bmatrix} i_0, \ldots, i_{\nu-1} \\ b, \ldots, l \end{bmatrix} \Phi^{i_\nu}_s \Big/ \begin{bmatrix} i_0, \ldots, i_{\nu-1} \\ b, \ldots, l \end{bmatrix} M^{i_\nu}_s =$$

$$= \begin{bmatrix} i_0, \ldots, i_{\nu-1} \\ b, \ldots, l \end{bmatrix} \Phi^{i_\nu}_{s-1} = \begin{bmatrix} i_0, \ldots, i_{\nu-1}, i_\nu \\ b, \ldots, l, s \end{bmatrix} \Phi^{i_{\nu+1}}_m. \qquad (5)$$

3. Выделим в качестве старших элемент $\begin{bmatrix} i_0, \ldots, i_{\nu-1} \\ b, \ldots, l \end{bmatrix} r^{i_\nu}_{s-1}$ пути $\begin{bmatrix} i_0, \ldots, i_{\nu-1} \\ b, \ldots, l \end{bmatrix} \varphi^{i_\nu}$ и элементы $\begin{bmatrix} i_0, \ldots, i_{\nu-1}, i_\nu \\ b, \ldots, l, s \end{bmatrix} r^{i_{\nu+1}}_m$ [здесь $m = = m(i_{\nu+1})$] путей $\begin{bmatrix} i_0, \ldots, i_{\nu-1}, i_\nu \\ b, \ldots, l, s \end{bmatrix} \varphi^{i_{\nu+1}}$.

Процедура заканчивается, когда на некотором шаге N элемент, рассматриваемый в качестве старшего, порождает пустое множество путей ранга $\nu+1$ и отсутствует элемент $\begin{bmatrix} i_0, \ldots, i_{\mu-1} \\ a, \ldots, l \end{bmatrix} r^{\mu}_d$, для которого не произведена процедура порождения последовательностей ранга $\mu+1$.

В результате рассмотренной процедуры получаем множество Γ ранжированных путей $\begin{bmatrix} i_0, \ldots, i_{\nu-1} \\ b, \ldots, l \end{bmatrix} \varphi^{i_\nu}$. Здесь ν — ранг соответствующего цикла $\{\varphi\}$ $(\nu = 1, \ldots, k)$; i_ν — порядковый номер цикла $\{\varphi\}$ на множестве $\begin{bmatrix} i_0, \ldots, i_{\nu-2} \\ b, \ldots, d \end{bmatrix} M^{i_{\nu-1}}_l$ $(i_\nu = 1, \ldots, n_\nu)$; l — порядковый номер элемента $r^{\sigma_j}_j$ в простом цикле $\begin{bmatrix} i_0, \ldots, i_{\nu-2} \\ b, \ldots, d \end{bmatrix} \varphi^{i_{\nu-1}}$ (при $\nu > 1$) или в простом полном пути ψ^{i_0} (при $\nu = 1$).

В дальнейшем для упрощения формы записи вместо набора индексов $\begin{bmatrix} i_0, \ldots, i_{\nu-1} \\ b, \ldots, l \end{bmatrix}$ будем пользоваться символом $\overline{il}$.

Положим, что процесс выполнения программы сопровождается движением по рассмотренному графу $G = G(V)$ некоторой характеристической точки. При этом каждой вершине $R_j \in V$ соответствует время t_j (время задержки характеристической точки в данной вершине), а каждой дуге $r_j^{\sigma_j}$ — условная вероятность $p_j^{\sigma_j}$ ее выбора (при условии, что достигнута такая вершина R_j, из которой исходит данная дуга):

$$p_j^{\sigma_j} = P\left(r_j^{\sigma_j} \middle| R_j\right), \tag{6}$$

где $R_j \in L$. Тогда каждому пути $\overline{il}\,\varphi^{i_\nu} \in \Gamma$ соответствует трудоемкость однократной реализации

$$\overline{il}\,\Theta^{i_\nu} = \sum_{R_j \in \overline{il}\varphi^{i_\nu}} t_j \tag{7}$$

и условная вероятность однократной реализации (при условии, что достигнута одна из вершин такого пути)

$$\overline{il}\,P^{i_\nu} = \prod_{r_j^{\sigma_j} \in \overline{il}\varphi^{i_\nu}} p_j^{\sigma_j}. \tag{8}$$

Для простых полных путей соответствующие величины записываются аналогично.

Кроме того, каждому пути $\overline{il}\,\varphi^{i_\nu}$ множества Γ поставим в соответствие коэффициент $\overline{il}\,n^{i_\nu}$ $(\overline{il}\,n^{i_\nu} = 0, 1, 2, \ldots)$ — возможное количество реализаций соответствующего цикла при условии однократной реализации пути $\overline{ib}\,\varphi^{i_{\nu-1}} \in \Gamma$ (или простого полного пути $\psi_i \in \Psi$ при $\nu = 1$), порождающего данный.

Тогда для математического ожидания времени реализации программы можно записать соотношение

$$M[T(U)] = \lim \sum_{i_0=1}^{n_0} \sum_{\overline{ib}n_j^{i_{\nu-1}}\,\overline{ij}n^{1_\nu} + \ldots + \overline{ij}n^{n_\nu} = \overline{ib}n_j^{i_{\nu-1}}} \Bigg\{\Theta^{i_0} + \sum_{\nu=1}^{k} \sum_{j(i_{\nu-1})=1}^{m(i_{\nu-1})} \times$$

$$\times \sum_{i_\nu=1}^{n_\nu} \overline{ij}\Theta^{i_\nu} \prod_{\mu=1}^{\nu} \overline{il}n^{i_\mu}\Bigg\} f(P^{i_0}) \tag{9}$$

при $\overline{ib}n_j^{i_{\nu-1}} = \left(\overline{ij}n^{1_\nu} + \ldots + \overline{ij}n^{n_\nu}\right) \to \infty$ $(\nu = 1, \ldots, k;\ i_\nu = 1, \ldots, n_\nu;$ $j = j(i_{\nu-1}) = 1, \ldots, m(i_{\nu-1}))$,

где
$$f(P^{i_0}) = P^{i_0} \prod_{l(i_0)=1}^{m(i_0)} \left\{ \prod_{i_1=1}^{n_1} \left[f\left(\overline{il}\,P^{i_1}\right)\right]^{\overline{il}n^{i_1}} \frac{n_l^{i_0}!}{\overline{il}n^{1_1}! \ldots \overline{il}n^{n_1}!} \right\};$$

. . . .

$$f\left(\overline{ij}\,P^{i_{k-1}}\right) = \overline{ij}\,P^{i_{k-1}} \prod_{s(i_{k-1})=1}^{m(i_{k-1})} \left\{ \prod_{i_k=1}^{n_k} \left[f\left(\overline{is}\,P^{i_k}\right)\right]^{\overline{is}n^{i_k}} \frac{\overline{ij}n_s^{i_{k-1}}!}{\overline{is}n^{1_k}! \ldots \overline{is}n^{n_k}!} \right\};$$

произведение $\prod_{\mu=1}^{\nu} \overline{il}\,n^{i_\mu}$ включает только коэффициенты путей, последовательно порождающих соответствующий путь $\overline{ij}\,\varphi^{i_\nu}$.

Предел соотношения (9) существует и равен

$$M[T(U)] = \sum_{i_0=1}^{n_0} \left\{ \Theta^{i_0} P^{i_0} + \sum_{\nu=1}^{k} \sum_{j(i_{i-1})=1}^{m(i_{\nu-1})} \sum_{i_\nu=1}^{n_\nu} M\left[\overline{ij}\, T^{i_\nu}(U)\right] \right\}, \qquad (10)$$

где

$$M\left[\overline{ij}\, T^{i_\nu}(U)\right] = \overline{ij}\,\Theta^{i_\nu} \frac{F_1}{\left(1-\sum_{i_1=1}^{n_1} \overline{i1}\,\widetilde{P}^{i_1}\right)\ldots\left(1-\sum_{i_1=1}^{n_1} \overline{il}\,\widetilde{P}^{i_1}\right)^2\ldots\left(1-\sum_{i_1=1}^{n_1} \overline{im}\,\widetilde{P}^{i_1}\right)};$$

$$F_1 = P^{i_0} \frac{F_2}{\left(1-\sum_{i_2=1}^{n_2} \overline{i1}\,\widetilde{P}^{i_2}\right)\ldots\left(1-\sum_{i_2=1}^{n_2} \overline{ic}\,\widetilde{P}^{i_2}\right)^2\ldots\left(1-\sum_{i_2=1}^{n_2} \overline{im}\,\widetilde{P}^{i_2}\right)}; \qquad (11)$$

. . . .

$$F_{\nu-1} = \overline{ib} P^{i_{\nu-2}} \frac{F_\nu}{\left(1-\sum_{i_\nu=1}^{n_\nu} \overline{i1}\,\widetilde{P}^{i_\nu}\right)\ldots\left(1-\sum_{i_\nu=1}^{n_\nu} \overline{ij}\,\widetilde{P}^{i_\nu}\right)^2\ldots\left(1-\sum_{i_\nu=1}^{n_\nu} \overline{im}\,\widetilde{P}^{i_\nu}\right)};$$

$$F_\nu = \overline{id}\, P^{i_{\nu-1}}\, \overline{ij}\, \widetilde{P}^{i_\nu};$$

$$\overline{is}\,\widetilde{P}^{i_\mu} = \frac{\overline{is} P^{i_\mu}}{\prod_{g(i_\mu)=1}^{m(i_\mu)} \left(1-\sum_{i_{\mu+1}=1}^{n_{\mu+1}} \overline{ig}\widetilde{P}^{i_{\mu+1}}\right)}$$

$$\left(\mu = 0, 1, \ldots, \nu, \ldots, k-1;\ \overline{is}\,\widetilde{P}^{i_k} = \overline{is}\,P^{i_k}\right). \qquad (12)$$

При этом

$$\sum_{i_0=1}^{n_0} \widetilde{P}^{i_0} = \sum_{i_0=1}^{n_0} \frac{P^{i_0}}{\prod_{l(i_0)=1}^{m(i_0)} \left(1-\sum_{i_1=1}^{n_1} \overline{il}\widetilde{P}^{i_1}\right)} = 1. \qquad (13)$$

Пример. Пусть исследуемая программа представлена графом (рис. 1). Проставим на нем только наименования дуг. Для такого графа

$$\psi_1 = r_\Lambda^* r_4^* r_1^1 r_2^1 r_3^* \text{ — простой полный путь;}$$

$$\{\varphi_1\} = r_1^0 r_4^*; \quad \{\varphi_2\} = r_1^1 r_2^0 r_5^* \text{ — простые циклы.}$$

В результате процедуры ранжировки путей получим:

$$\psi^{1_0} = r_\Lambda^* r_4^* r_1^{1_0} r_2^{1_0} r_3^* = \psi_1 \text{ — путь 0-го ранга;}$$

$$\begin{bmatrix} 1_0 \\ 2 \end{bmatrix} \varphi^{1_1} = \begin{bmatrix} 1_0 \\ 2 \end{bmatrix} r_1^{1_1} r_5^* \begin{bmatrix} 1_0 \\ 2 \end{bmatrix} r_2^{1_1} = \{\varphi_2\}, \qquad \begin{bmatrix} 1_0 \\ 1 \end{bmatrix} \varphi^{1_1} = \begin{bmatrix} 1_0 \\ 1 \end{bmatrix} r_1^{1_1} r_4^* = \{\varphi_1\} \text{ —}$$

пути 1-го ранга;

$$\begin{bmatrix} 1_0, & 1_1 \\ 2, & 2 \end{bmatrix} \varphi^{1_2} = \begin{bmatrix} 1_0, & 1_1 \\ 2, & 2 \end{bmatrix} r_1^{1_2} r_4^* = \{\varphi_1\} \text{ — путь 2-го ранга.}$$

В соответствии с (7) и (8) имеем (полагая $t_Q = t_\Lambda = 0$):

$$\Theta^{1_0} = t_4 + t_1 + t_2 + t_3;$$

$$P^{1_0} = P_1^1 P_2^1; \begin{bmatrix} 1_0 \\ 2 \end{bmatrix} \Theta^{1_1} = t_2 + t_5 + t_1;$$

$$\begin{bmatrix} 1_0 \\ 2 \end{bmatrix} P^{1_1} = P_2^0 P_1^1;$$

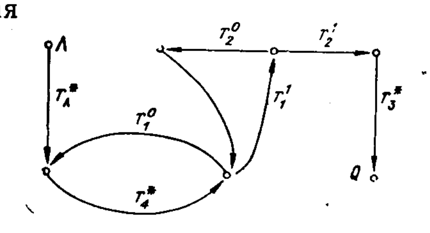

Рис. 1.

$$\begin{bmatrix}1_0\\1\end{bmatrix}\Theta^{1_1}=\begin{bmatrix}1_0, & 1_1\\2, & 2\end{bmatrix}\Theta^{1_2}=t_1+t_4;$$

$$\begin{bmatrix}1_0\\1\end{bmatrix}P^{1_1}=\begin{bmatrix}1_0, & 1_1\\2, & 2\end{bmatrix}P^{1_2}=p_1^0.$$

Тогда математическое ожидание времени реализации программы, представленной графом на рис. 1, есть

$$M(T)=\frac{P^{1_0}}{\alpha\beta}\left(\Theta^{1_0}+\begin{bmatrix}1_0\\1\end{bmatrix}\Theta^{1_1}\begin{bmatrix}1_0\\1\end{bmatrix}P^{1_1}+\begin{bmatrix}1_0\\2\end{bmatrix}\Theta^{1_1}\begin{bmatrix}1_0\\2\end{bmatrix}P^{1_1}\frac{\beta}{2}+\right.$$
$$\left.+\begin{bmatrix}1_0, & 1_1\\2, & 2\end{bmatrix}\Theta^{1_2}\begin{bmatrix}1_0, & 1_1\\2, & 2\end{bmatrix}P^{1_2}\begin{bmatrix}1_0\\2\end{bmatrix}P^{1_1}\frac{\beta}{\alpha^2}\right), \tag{14}$$

где

$$\alpha=1-\begin{bmatrix}1_0\\1\end{bmatrix}P^{1_1};\quad \beta=1-\frac{\begin{bmatrix}1_0\\2\end{bmatrix}P^{1_1}}{1-\begin{bmatrix}1_0, & 1_1\\2, & 2\end{bmatrix}P^{1_2}}.$$

В соответствии с (13), учитывая, что $P_j^0=1-P_j^1$:

$$\widetilde{P}^{1_0}=\frac{P^{1_0}}{\alpha\beta}=\frac{p_1^1\,p_2^1}{\left(1-p_1^0\right)\left(1-\dfrac{p_2^0\,p_1^1}{1-p_1^0}\right)}=1 \text{ (условие нормировки)}. \tag{15}$$

III. Для реализации процедуры ранжировки путей может быть использована решающая схема, представленная на рис. 2. Исходными для нее являются: множество Ψ (мощности x) простых полных путей; множество Φ (мощности y) простых циклов; время однократной реализации каждого из путей ψ_i и $\{\varphi_j\}$ и условная вероятность однократной реализации рассмотренных путей.

Здесь операторы A_i $(i=1,\ldots,14)$ обеспечивают выполнение следующих действий:

A_1 подготавливает проверку логического условия P_1; A_2, A_{12} реализуют п. 2 процедуры ранжировки, в результате чего организуется множество циклов $\overline{is}\,\Phi_m^{i_\nu+1}$ мощности $\overline{is}\,y_m^{i_\nu+1}$ (или множество $\overline{il}\Phi_{s-1}^{i_\nu}$ мощности $\overline{il}\,y_{s-1}^{i_\nu}$); A_3, A_8, A_{11} подготавливают проверку логических условий P_2, P_5, P_7 соответственно; A_4 подготавливает к анализу очередной путь; A_5: 1) подготавливает к анализу очередной цикл $\{\varphi\}\in\overline{il}\,M_j^{i_\nu-1}$; 2) формирует индексы $\overline{ij}$, i^ν такого цикла; A_6 реализует п.1 процедуры ранжировки и затем п. 2 оператора A_5; A_7 производит учет путей нулевого ранга; A_9 проверяет выполнение ограничения на ранг путей исследуемой схемы (при невыполнении — аварийный останов); A_{10}, A_{13} производят операцию вычитания $(j(i_\nu)-1)$; A_{14} подготавливает к работе решающую схему собственно оценки величины $T(U)$.

Операторы P_j $(j=1,\ldots, 8)$ производят проверку следующих условий: P_1 — $(i_{\nu=1}=n_{\nu-1})$ для путей множества $\overline{il}\,M_j^{i_\nu-1}$; P_2, P_8 — «мощность множества $\overline{il}\,\Phi_s^{i_\nu}$ равна нулю»; P_3, P_6 — $(\nu-1=0)$ для путей множества $\overline{il}\,M_j^{i_\nu-1}$; P_4 — $\left(\overline{il}\,M_j^{i_\nu}=\varnothing\right)$, т. е. множество $\overline{il}\,M_j^{i_\nu}$ — пустое; P_5, P_7 — $s(i_\nu)=1$ для пути $\overline{is}\,\varphi^{i_\nu+1}$.

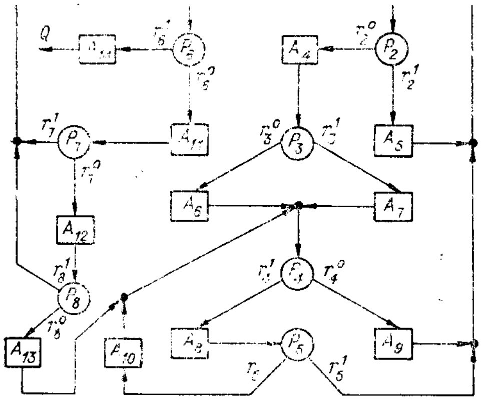

Рис. 2.

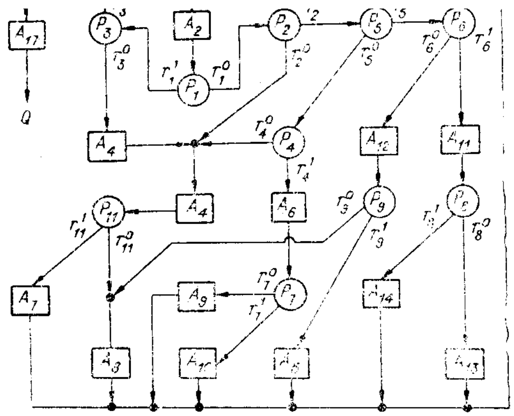

Рис. 3.

Для получения собственно оценки времени реализации исследуемой программы U в соответствии с соотношением (10) может быть предложена следующая решающая схема (рис. 3).

Здесь операторы A_i ($i=1,\ldots,17$) выполняют следующие действия: A_1 выделяет на множестве Γ ранжированных путей множество Ξ^{i_0} путей, последовательно порождаемых путем нулевого ранга ψ^{i_0}; A_2 для исследуемого пути $\overline{il}\,\varphi^{i_\nu}$ множества Ξ^{i_0} находит порождающий путь $\overline{ij}\,\varphi^{i_\nu-1}$; A_3: 1) вычисляет $\overline{il}\,\widetilde{P}^{i_\nu}$ в соответствии с формулой (12); 2) вычисляет $M\left[\overline{il}\,T^{i_\nu}(U)\right]$ в соответствии с (11); 3) складывает $M\left[\overline{il}\,T^{i}_{\nu}(U)\right]$ с ранее полученными $M\left[\overline{ij}\,T^{i_\mu}(U)\right]$; A_4 передает управление оператору A'_4; A'_4 подготавливает проверку логического оператора P_{11}; A_6, A_{11}; A_{12} подготавливают проверку логических операторов P_7, P_8, P_9 соответственно; A_7: 1) вычисляет соотношение $\prod\limits_{i(i_\nu)=1}^{m(i_\nu)}\left(1-\sum\limits_{i_{\nu+1}=1}^{n_{\nu+1}}\overline{ij}\,\widetilde{P}^{i_{\nu+1}}\right)$, 2) подготавливает для анализа следующий путь $\overline{il}\,\varphi^{i_\mu}$; A_8: 1) вычисляет $\sum\limits_{i_{\nu+1}=1}^{u_{\nu+1}}\overline{is}\,\widetilde{P}^{i_{\nu+1}}$, 2) реализует п. 2 оператора A_7; A_9: 1) вычисляет $\overline{il}\,\widetilde{P}^{i_\nu}$ в соответствии с формулой (12); 2) вычисляет произведение $\prod\limits_{\mu=1}^{\nu}\overline{il}\,P^{i_\mu}$; 3) реализует п. 2 оператора A_7; A_{10}, A_5 аналогичны операторам A_9, A_7 соответственно; A_{13}: 1) вычисляет $\sum\limits_{i_{\nu+1}=1}^{n_{\nu+1}}\overline{is}\,\widetilde{P}^{i_{\nu+1}}$; 2) вычисляет $\left(1-\sum\limits_{i_{\nu+1}=1}^{n_{\nu+1}}\overline{ij}\,\widetilde{P}^{i_{\nu+1}}\right)$; 3) реализует пункты оператора A_8; A_{14}: 1) реализует п. п. 1 и 2 оператора A_{13}; 2) вычисляет $\overline{il}\,\widetilde{P}^{i_\nu}$ в соответствии с (12); 3) реализует пункты оператора A_7; A_{16} подготавливает для анализа следующий путь $\overline{ij}\ \varphi^{i}{}_{\nu}$; A_{17} обеспечивает вывод результатов.

Операторы P_j ($j=1,\ldots,11$) проверяют следующие логические условия: P_1 — $\nu=1$ для $\overline{il}\,\varphi^{i_\nu}$; P_2 — «ранг μ исследуемого пути $\overline{ij}\,\varphi^{i_\mu}$ равен рангу ν пути, для которого должно быть вычислено $M\left[\overline{il}\,T^{i_\nu}(U)\right]$»; P_3 аналогичен оператору P_1; P_4 — «исследуемый путь $\overline{il}\,\varphi^{i_\nu}$ есть путь, для которого должно быть вычислено $M\left[\overline{il}\,T^{i_\nu}(U)\right]$»; P_5 — «имеется необходимость проверить значение логического условия оператора P_4»; P_6 — «на предыдущем шаге ранг исследуемого пути уменьшен»; P_7, P_8, P_9, P_{11} — $i_\nu = 1$ для исследуемого пути $\overline{il}\,\varphi^{i}_{\nu}$; P_{10} — «проанализированы все пути множества Г».

IV. Рассмотренные решающие схемы были реализованы на ЦВМ М-20 в машинном языке. Программа, представляющая процедуру ранжировки, включала $640_{10}(1200_8)$ команд (включая исходные константы и команды переадресации). Программа, обеспечивающая вычисление искомой оценки (10) в соответствии с индексацией ранжированных путей множества Г, содержала $256_{10}(400_8)$ команд. На исследуемый граф были наложены ограничения: 1) исходное количество всех простых путей не должно превышать 24; 2) количество простых путей, содержащих символ одной и той же дуги $r^{\sigma j}_{j}$, не должно превышать 7.

Отметим, что анализу подвергается схема программы, т. е. объект более обозримый, чем сама программа. В случае достаточно сложной программы анализ соответствующего графа может производиться по частям (части графа и затем обобщенный граф анализируются по общей схеме).

Частота реализации операторов рассмотренных решающих схем (а следовательно, и время работы соответствующих программ) определяется структурной сложностью графа, представляющего исследуемую программу. Результаты исследования программы, представляющей первую из рассмотренных схем, и программы, представляющей вторую схему, даны в табл. 1 и 2 соответственно (при этом в процессе анализа программ использован изложенный метод). В табл. 1 и 2 в качестве первого рассмотрен случай полной неопределенности (т. е. когда отсутствует информация о структурной сложности графа, представляющего программу, подлежащую анализу). Соответствующие P_j^0 определяются соотношением $P_j^0 = 1 - P_j^1$. Для контроля результатов во всех случаях ис-

Таблица 1

p_1^1	p_2^1	p_3^1	p_4^1	p_5^1	p_6^1	p_7^1	p_8^1	$M(T)$, мс
0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	35,5
0,4	0,3	0,4	0,6	0,3	0,4	0,3	0,3	80,68
0,1	0,1	0,2	0,3	0,05	0,1	0,05	0,1	1684,12

Таблица 2

p_1^1	p_2^1	p_3^1	p_4^1	p_5^1	p_6^1	p_7^1	p_8^1	p_9^1	p_{10}^1	p_{11}^1	$M(T)$, мс
0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	15,52
0,25	0,35	0,4	0,25	0,5	0,6	0,6	0,6	0,4	0,2	0,6	94,17
0,2	0,2	0,02	0,06	0,5	0,3	0,15	0,15	0,2	0,02	0,02	215156,39

пользовалось условие нормировки (13). В табл. 1а и 2а приведено время (в миллисекундах) реализации операторов решающих схем.

Таблица 1а

A_i	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}
t	0,144	4,07	0,124	2,72	5,36	8,9	6,843	1,135	0,217	3,83	1,16	5,17	3,83	1,17

P_i	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
t	0,048	0,048	0,048	0,1	0,048	0,048	0,048	0,048

Таблица 2а

A_i	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}
t	0,611	1,183	0,75	0,024	0,048	0,072	0,707	0,573	1,345	1,766	0,048	0,048	0,887	1,308	1,054	0,024	0,438

P_i	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}
t	0,12	0,12	0,12	0,048	0,024	0,024	0,66	0,66	0,66	0,072	0,66

ЗАКЛЮЧЕНИЕ

Основным содержанием изложенной работы является разработка метода априорной оценки трудоемкости реализации программ на цифровых вычислительных машинах. Существо предлагаемого метода составляет: процедура ранжировки путей и выражение математического ожидания времени реализации программы [см. (10)] (основанное на результатах процедуры ранжировки). Собственно объектом исследования являлась граф-схема программы. Как отмечается в работе [5], понятие программы является весьма сложным во всей его полноте. Поэтому схемы программ и алгоритмов являются вполне естественными объектами анализа.

В целом алгоритм априорной оценки времени программ представляется в следующем виде: 1) задание схемы программы (в данном случае граф-схемы или блок-схемы); 2) задание времени реализации операторов схемы программы; 3) задание вероятностей переходов между операторами схемы; 4) перечисление простых путей исследуемой схемы программы; 5) реализация процедуры ранжировки путей; 6) вычисление математического ожидания времени реализации программы.

В том случае, когда исходным объектом исследования выступает схема алгоритма, определенные результаты могут быть получены до того, как данный алгоритм представлен в виде машинной программы.

Основополагающей работой, посвященной вопросам изготовления граф-схем алгоритмов, является, очевидно, работа Л. А. Калужнина [6]. Обзор работ, рассматривающих вопросы построения блок-схем программ, дан в [7].

Задание времени реализации операторов схемы, как правило, не представляет особых затруднений.

Природа вероятностных связей алгоритмов и программ в данной работе не исследовалась. Представляется, что этот вопрос требует специального рассмотрения. Отметим только, что вероятностные связи между элементами алгоритмов и программ определяются содержанием решаемой задачи (и в определенных случаях могут быть заданы аналитически). Так, в частности, вероятности p_j^1 передач управления в рассмотренных решающих схемах (табл. 1, 2) определяются количеством простых путей в исследуемой схеме и их длиной (если в качестве длины l пути α принять количество символов $r_j^{\sigma j} \in \alpha$).

Для перечисления путей в ориентированном графе могут быть использованы методы, рассмотренные в [3, 4] (или в [8, 9], если алгоритм или соответствующая программа заданы блок-схемой).

Ввиду отсутствия достаточно законченных решений задачи априорной оценки трудоемкости алгоритмов и программ, представляется затруднительной сравнительная оценка различных методов по всем возможным параметрам. Поэтому в данном случае отметим только, что изложенный метод позволяет (по сравнению с наиболее известным методом, составляющим основу марковского подхода [2]):

1) обойтись без трудоемких операций с матрицами, так как особенности структуры программы учитываются процедурой ранжировки;

2) учитывать время реализации распознавателей (логических операторов) алгоритмов и программ;

3) производить оценку времени реализации алгоритмов и программ достаточно просто при наличии конкретного соотношения (10) (что весьма важно в том случае, когда возникают трудности с заданием точных значений вероятностей $p_j^{\sigma j}$ передач управления и встает задача анализа таких алгоритмов для различных наборов вероятностей);

4) использовать условие нормировки (13) в качестве одного из средств контроля результатов, которое весьма просто получается в процессе работы алгоритма.

И наконец, отметим, что всякий достаточно общий алгоритм допускает различные варианты своей конкретной (в том числе программной) реализации, т. е. ограничения 1 и 2 (см. IV) следует рассматривать как ограничения, свойственные только данной реализации алгоритма, а не алгоритму в целом.

ЛИТЕРАТУРА

1. В. И. Знак. Об одном методе формализации процесса анализа структуры алгоритма при оценке времени его машинной реализации.— Вычислительные системы. Материалы II Всесоюзной конференции. Секция III. Новосибирск, 1969.
2. Р. М. Карп. О приложении теории графов к программированию.— Кибернетический сборник, 4. М., Изд-во иностр. лит., 1964.
3. А. А. Зыков. Теория конечных графов. Новосибирск, «Наука», 1969.
4. Ф. Е. Хон, С. Сешу, Д. Д. Ауфенкамп. Теория сетей. — В сб. «Математика», 1959, 3, № 3.
5. А. П. Ершов, А. А. Ляпунов. О формализации понятия программы.— Кибернетика, 1967, № 5.
6. Л. А. Калужнин. Об алгоритмизации математических задач.— Проблемы кибернетики, вып. 2. М., 1959.
7. M. D. Abrams. A comparative sampling of the systems for producing computer — drawn flowcharts.— Proc. 23rd ACM (Assoc. Comput. Mach.). Nat. Conf., 1968. Princeton, N. Y.— London, 1968.
8. В. Ф. Дъяченко, В. Г. Лазарев, Г. Г. Саввин. Управление на сетях связи. М., «Наука», 1967.
9. В. И. Знак. К методам анализа блок-схем и программ.— В сб. «Цифровая вычислительная техника и программирование», вып. 6. М., «Советское радио», 1971.

Поступила в редакцию 25 декабря 1970 г.,
окончательный вариант —
7 июня 1971 г.