ЭКОНОМИКА
И МАТЕМАТИЧЕСКИЕ МЕТОДЫ

1967
том III, вып. 6

ВЫБОР ОПТИМАЛЬНЫХ ТРАСС ДВИЖЕНИЯ ПАССАЖИРОВ ПРИ ЗАДАННОЙ ТРАНСПОРТНОЙ СЕТИ ГОРОДА

С. Л. АМБАРЯН

(*Ереван*)

Расчетное обоснование маршрутных схем городского пассажирского транспорта требует прогноза ожидаемых потоков пассажиров по заданной (проектируемой) уличной магистральной сети города. В качестве исходной информации для подобного рода расчетов могут служить данные о тяготении населения между отдельными территориями (районами) города, изображаемыми в виде квадратных матриц $\|s_{ij}\|$, где число s_{ij} показывает количество лиц, перемещающихся из зоны i в зону j. Однако по данным о тяготении населения можно определить пассажиропотоки на отдельных участках магистральной сети города, если известна сама трасса передвижения.

Таким образом, центральным вопросом прогноза интенсивности пассажиропотоков по проектируемым маршрутам городского транспорта является выбор оптимальных трасс внутригородских передвижений населения.

В настоящей статье описывается решение на ЭВМ следующей задачи.

Пусть имеется план некоторого города с заданной сетью маршрутов городского пассажирского транспорта (автобус, трамвай, троллейбус). Требуется определить путь L, по которому из любой заданной зоны (области) городской территории A можно попасть в другую область B и притом так, чтобы затраты времени на передвижение из A и B по пути L были меньше, чем по любому другому пути. Путь (трассу), удовлетворяющий этому условию, назовем оптимальным.

1. ОСНОВНЫЕ ПОНЯТИЯ ОБ ИСХОДНЫХ ДАННЫХ

Для математического представления условий и исходных данных указанной задачи городская территория на основе конкретного анализа (выполняемого обычно специалистами-транспортниками) «разумным» образом разбивается на непересекающиеся области D_i, $i = 1, 2, \ldots, N$, и в каждой области D_i выбирается точка x_i, называемая узловой. При отыскании оптимального пути, ведущего из любой точки области D_i в любую точку области D_j, мы будем считать, что передвижение совершается из узловой точки (из узла) $x_i \in D_i$ в узловую точку $x_j \in D_j$. Иными словами, наша задача сводится к следующей: для заданной системы узловых точек $x_1, x_2, \ldots, x_N$ (индекс при узловой точке назовем ее номером) и заданной сети городского транспорта найти оптимальный путь из любой точки x_i в любую другую точку x_j $(i, j = 1, 2, \ldots, N; i \neq j)$.

Для дальнейшего нам нужно ввести понятие трассы. Мы будем различать трассы трех типов. Трассы первого типа образуются из линий, являющихся обычным маршрутом городского транспорта. Несколько точнее — это непрерывная линия, соединяющая фиксированные узловые точки, по которой движется один и тот же вид городского транспорта и притом с одним и тем же маршрутным номером. Таким образом (см. рис. 1, *а*), если, например, путь трамвая № 2 начинается в точке x_1 и кончается в точке x_7, после чего тот же трамвай возвращается в точку x_1 (не обязательно по тому же самому пути), то этот путь (в оба конца) образует трассу. Трассы этого типа мы назовем м а р ш р у т н ы м и или просто маршрутами. Максимальную общую (с учетом совпадения направления движения транспорта) часть двух (или трех, или четырех...) любых маршрутов, соединяющую две узловые точки, мы назовем искусственной трассой. Это второй тип трассы. Заметим, что при решении транспортных задач имеет смысл рассматривать не все возможные искусственные трассы, а только часть из них, обычно выделяемых специалистами.

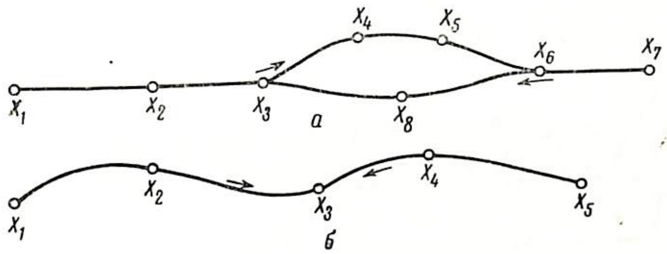

Рис. 1

Наконец, трасса третьего типа строится следующим образом. При заданном плане города с выбранными на нем узловыми точками некоторые пары узлов x_n и x_k (множество таких пар нам задано) могут быть соединены линиями, по которым перемещение из узла x_n в узел x_k осуществляется пешком. Кроме того, перемещение пешком разрешается из любого узла x_i в любой узел x_j, если эти узлы находятся на одном и том же маршруте (т. е. маршрут проходит через эти точки), не содержащем между ними другой узловой точки. При этом, если через два узла x_i и x_j проходят два различных маршрута $\Gamma^{(1)}$ и $\Gamma^{(2)}$, не содержащих между узлами x_i и x_j других узлов, то мы считаем, что линии, образующие маршруты $\Gamma^{(1)}$ и $\Gamma^{(2)}$ на участке (x_i, x_j), совпадают (см. ниже описание таблицы). Таким образом, все указанные выше линии пешего передвижения, а также все маршрутные трассы мы объединяем в одну трассу, которую в отличие от остальных назовем п е ш е х о д н о й. Обозначим через $\Gamma_1, \Gamma_2, \ldots, \Gamma_M$ множество всех трасс, причем через Γ_1 обозначена пешеходная трасса (индекс при трассе назовем ее номером).

Любые две точки x_i и x_j, входящие в упорядоченную пару (x_i, x_j), называются смежными (смежной парой), если существует хоть одна трасса (в том числе и пешеходная), соединяющая эти две точки в указанном порядке (в смысле направления движения по трассе) и не содержащая между узлами x_i и x_j никакой другой узловой точки. Каждой смежной паре (x_i, x_j) сопоставлено некоторое положительное число $\rho(x_i, x_j)$, равное длине трассы, соединяющей точки x_i и x_j. При этом предполагается, что все трассы, которые образуют смежную пару (x_i, x_j), на отрезке (x_i, x_j) совпадают. Число $\rho(x_i, x_j)$ (вообще говоря, не равное числу $\rho(x_i, x_j)$) называется расстоянием узловой точки x_i от узловой точки x_j.

При помощи сети городского транспорта и множества узловых точек каждую трассу можно задать как последовательность узловых точек, через которые от предыдущей узловой точки к последующей проходит данная трасса. При этом для маршрутных трасс первый и последний члены последовательности должны быть одинаковыми (для других трасс это не обязательно). Например, на рис. 1, *а* и *б* показаны некоторые трассы, последовательности узловых точек которых будут соответственно:

а) $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_6, x_8, x_3, x_2, x_1\}$,

б) $\{x_1, x_2, x_3, x_4, x_5, x_4, x_3, x_2, x_1\}$.

Отметим, что последовательность, характеризующую данную трассу, можно задать, начиная с любой узловой точки, через которую проходит данная трасса, лишь циклически меняя очередность следования по узло-

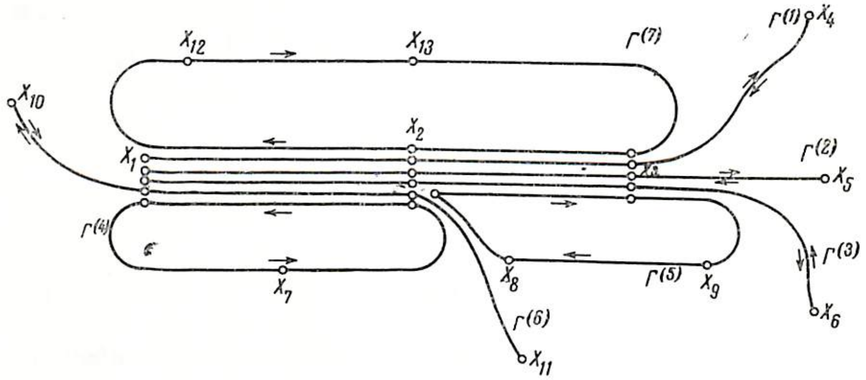

Рис. 2

вым точкам. Например, если некоторая последовательность $\{x_1, x_2, x_3, x_4, x_3, x_2, x_1\}$ характеризует трассу Г, то трассу Г можно задать также при помощи последовательностей: $\{x_2, x_3, x_4, x_3, x_2, x_1, x_2\}$, $\{x_3, x_4, x_3, x_2, x_1, x_2, x_3\}$, $\{x_4, x_3, x_2, x_1, x_2, x_3, x_4\}$.

Для иллюстрации искусственных трасс рассмотрим рис. 2, на котором изображена сеть маршрутов городского транспорта и узловые точки, через которые проходят эти маршруты.

Выпишем маршрутные трассы при помощи последовательностей узловых точек:

$$\Gamma^{(1)} = \{x_1, x_2, x_3, x_4, x_3, x_2, x_1\},$$
$$\Gamma^{(2)} = \{x_1, x_2, x_3, x_5, x_3, x_2, x_2\},$$
$$\Gamma^{(3)} = \{x_1, x_2, x_3, x_6, x_3, x_2, x_1\},$$
$$\Gamma^{(4)} = \{x_1, x_7, x_2, x_1\},$$
$$\Gamma^{(5)} = \{x_2, x_3, x_9, x_8, x_2\},$$
$$\Gamma^{(6)} = \{x_{10}, x_1, x_2, x_{11}, x_2, x_1, x_{10}\},$$
$$\Gamma^{(7)} = \{x_1, x_{12}, x_{13}, x_3, x_2, x_1\}.$$

Маршрутные трассы $\Gamma^{(1)}, \Gamma^{(2)}, \Gamma^{(3)}, \ldots, \Gamma^{(7)}$ порождают множество искусственных трасс. Выпишем некоторые из них:

$$\Gamma^{(8)} = \Gamma^{(1)} \cap \Gamma^{(2)} \cap \Gamma^{(3)} = \{x_1, x_2, x_3, x_2, x_1\},$$
$$\Gamma^{(9)} = \Gamma^{(8)} \cap \Gamma^{(6)} = \{x_1, x_2, x_1\},$$
$$\Gamma^{(10)} = \Gamma^{(9)} \cap \Gamma^{(4)} = \{x_2, x_1\},$$
$$\Gamma^{(11)} = \Gamma^{(8)} \cap \Gamma^{(5)} = \{x_2, x_3\},$$
$$\Gamma^{(12)} = \Gamma^{(8)} \cap \Gamma^{(7)} = \{x_3, x_2, x_1\},$$
$$\Gamma^{(13)} = \Gamma^{(1)} \cap \Gamma^{(2)} = \{x_1, x_2, x_3, x_2, x_1\},$$
$$\Gamma^{(14)} = \Gamma^{(1)} \cap \Gamma^{(7)} = \{x_3, x_2, x_1\}.$$

Отметим, что выделение искусственных трасс (см. ниже) значительно проще выполнять от руки, хотя, конечно, можно задать довольно простой алгоритм, по которому машина сама построит искусственные трассы, исходя из информации о маршрутных трассах.

Пусть $\mu(\Gamma)$ — длина трассы Γ и $t_{\Gamma}^{(1)}$ — время, за которое транспорт, движущийся по трассе Γ, проходит весь путь. Тогда величина $\mu(\Gamma)\,/\,t_{\Gamma}^{(1)}$ будет средней (эксплуатационной) скоростью (ν) транспорта, движущегося по трассе Γ. Предположим, что по трассе Γ через каждые $t_{\Gamma}^{(2)}$ единиц времени через данную узловую точку x_i проходит транспортное средство. Тогда пассажир, который подходит к этой узловой точке, будет ожидать транспортное средство в среднем время (τ), равное $t_{\Gamma}^{(2)}/2$. Действительно, время ожидания — это случайная величина, распределение которой можно считать таковым:

Время ожидания, t, ед. вр.	0	1	2	...	$t_{\Gamma}^{(2)}$
Вероятность, p	p_0	p_1	p_2	...	$p_{t_{\Gamma}^{(2)}}$

Здесь

$$\sum_{k=0}^{t_{\Gamma}^{(2)}} p_k = 1, \quad p_0 = p_1 = p_2 = \ldots = p_{t_{\Gamma}^{(2)}} = 1/(t_{\Gamma}^{(2)} + 1).$$

Тогда среднее время ожидания будет

$$\tau = M \cdot 0(t) = \sum_{k=0}^{t_{\Gamma}^{(2)}} p_k \cdot t_k = \frac{1}{t_{\Gamma}^{(2)} + 1}(0 + 1 + 2 + \ldots + t_{\Gamma}^{(2)}) =$$

$$= \frac{1}{t_{\Gamma}^{(2)} + 1} \cdot \frac{(0 + t_{\Gamma}^{(2)})(t_{\Gamma}^{(2)} + 1)}{2} = \frac{t_{\Gamma}^{(2)}}{2},$$

$$\tau = t_{\Gamma}^{(2)}/2.$$

Для маршрутных трасс среднее время ожидания и средняя скорость движения транспортного средства бывают заданы в условиях задачи. Однако для искусственных трасс среднее время ожидания и средняя скорость должны быть вычислены. Можно задать простой алгоритм, при помощи которого эти вычисления будут выполнены программно, но ввиду простоты вычислений и малого объема это также делается от руки. Вычисления ведутся по формулам, которые мы сейчас выведем.

Предположим, что некоторая искусственная трасса $\Gamma = \Gamma^{(1)} \cap \Gamma^{(2)} \cap \ldots \cap \Gamma^{(l)}$ является общей частью маршрутных трасс $\Gamma^{(1)}, \Gamma^{(2)}, \ldots, \Gamma^{(l)}$ со следующими величинами ожидания и скорости: $\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(l)}$; $\nu^{(1)}, \nu^{(2)}, \ldots, \nu^{(l)}$. Тогда величины $1\,/\,2\tau^{(\rho)}$ и $\sum_{\rho=1}^{l} 1/2\tau^{(\rho)}$ будут равны числу транспортных средств (автобус, трамвай, троллейбус), проходящих в единицу времени через любую узловую точку соответственно трасс $\Gamma^{(\rho)}$ и Γ. Таким образом, будем иметь:

$$\tau = 1/2 \sum_{\rho=1}^{l} \frac{1}{2\tau^{(\rho)}} = 1/\sum_{\rho=1}^{l} \frac{1}{\tau^{(\rho)}},$$

$$\nu = \sum_{\rho=1}^{l} \frac{1/\tau^{(\rho)}}{1/\tau} \nu^{(\rho)} = \tau \sum_{\rho=1}^{l} \frac{1}{\tau^{(\rho)}} \nu^{(\rho)}.$$

С точки зрения среднего времени ожидания и средней скорости движения трассы $\Gamma^{(8)}$ и $\Gamma^{(13)}$ различны. Вместе с тем (как отмечалось выше), непосредственно очевидно, что рассмотрение трассы $\Gamma^{(13)}$ излишне. Точно так же излишне рассмотрение трассы $\Gamma^{(14)}$. Пусть, наконец, нам задана скорость пешехода, которую соответственно с нумерацией трасс мы обозначим через ν_1. Время ожидания на трассе Γ_1, равное нулю, мы обозначим через τ_1.

Таким образом, исходной информацией для решения задачи является перечень всех трасс $\Gamma_1, \Gamma_2, \Gamma_3, \ldots, \Gamma_M$ с параметрами $\tau_1, \tau_2, \ldots, \tau_M$; $\nu_1, \nu_2, \ldots, \nu_M$, причем каждая трасса задана последовательностью узловых точек. Заметим, что, согласно определению, пешеходная трасса задана, во-первых, всеми маршрутами и, во-вторых, перечнем тех пар узловых точек, которые не лежат ни на каком маршруте, но считаются соединенными пешеходным путем. Указанную исходную информацию мы изобразим в виде таблицы, называемой информационной. В этой таблице во втором столбце выписаны все различные пары смежных точек (если точки x_i, x_j и x_j, x_i образуют смежные пары, то обе они выписаны). В третьем столбце указаны расстояния между этими точками. Остальные столбцы таблицы соответствуют всем трассам $\Gamma_1, \Gamma_2, \Gamma_3, \ldots, \Gamma_M$. В строке пары (x_i, x_j) в клетке, соответствующей трассе Γ_k, ставится 1 или 0 в зависимости от того, являются или нет точки x_i и x_j смежными на трассе Γ_k (для простоты нули в таблице опущены). Очевидно, что во всех клетках столбца Γ_1 стоят единицы. Порядок смежных пар во втором столбце безразличен. В первом столбце дан их порядковый номер в данной таблице.

2. АЛГОРИТМ ФОРДА И ПОСТРОЕНИЕ ГРАФА

Как известно, для решения транспортной задачи с успехом может быть применен алгоритм Форда. Однако непосредственно использовать этот алгоритм для решения нашей задачи нельзя, так как мы связаны уже существующими маршрутами городского транспорта (чего нет в обычной транспортной задаче) и поэтому при определении оптимальной трассы должны учитывать не только время на движение по трассе (причем на каждой трассе скорость движения своя), но и время, затрачиваемое на пересадки. Основная идея решения нашей задачи заключается в замене транспортной сети искусственным графом, в котором каждая узловая точка «расщепляется» в столько вершин графа, сколько маршрутов прохо-

№ пп	Список смежных пар	Расстояние смежных узловых точек	Трассы							
			Γ_1	Γ_2	Γ_3	…	Γ_l	…	Γ_{M-1}	Γ_M
1	2	3	4	5	6				$M+2$	$M+3$
1			1							
2			1							
3			1							
.										
.										
.										
p	x_i, x_j	$\rho(x_i, x_j)$	$z_1^{(p)}$	$z_2^{(p)}$	$z_3^{(p)}$	…	$z_l^{(p)}$	…	$z_{M-1}^{(p)}$	$z_M^{(p)}$
.										
.										
.										
P			1					…		

дит через данную узловую точку. Эти «расщепленные» вершины соединяются искусственными путями, «приведенная» длина которых определяется в зависимости от скорости движения по трассе и времени ожида-

Рис. 3

ния при перемене одного транспортного средства на другое. Ниже приводится строгое описание указанного графа.

Рассмотрим две узловые точки x_i, x_j, образующие смежную пару по трассам $\Gamma_{l_1}, \Gamma_{l_2}, \dots, \Gamma_{l_s}$, где $l_1 = 1$ (рис. 3). Узловым точкам x_i и x_j мы сопоставим $2s$ — вершины графа, которые обозначим через

$$(l_1, i),\ (l_2, i),\ \dots,\ (l_s, i);\ (l_1, j),\ (l_2, j),\ \dots,\ (l_s, j).$$

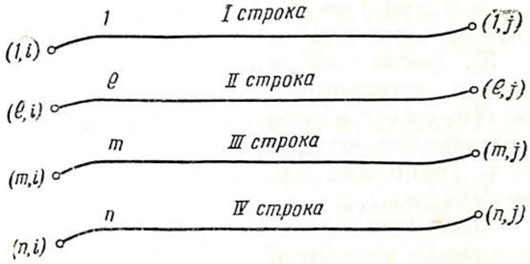

Рис. 4

Соответственные вершины, т. е. вершины (l_k, i) и (l_k, j), $k = 1, 2, \dots, s$ (взятые в том же порядке, что и узлы x_i, x_j), мы соединим дугами (рис. 4, где количество дуг (строк) равно s), «приведенная» длина которых равна

$$\mu\{(l_k, i),\ (l_k, j)\} = \rho(x_i, x_j) / V_{l_k}$$

(«приведенная» длина l_k-й дуги определяется в зависимости от скорости движения транспортного средства по трассе Γ_{l_k}). Такое «расщепление» узлов выполняется для всех смежных пар.

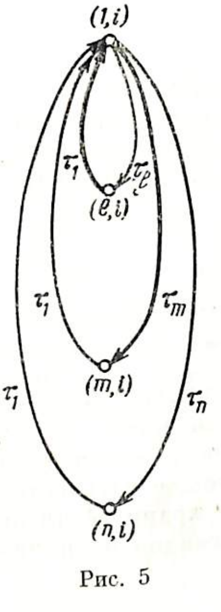

Рис. 5

Пусть в результате этого узел x_i «расщепился» на множество вершин $(m_1, i), (m_2, i), \dots, (m_p, i)$; $m_1 = 1$ (одна и та же вершина x_i входит в различные смежные пары). Для данного i вершина (m_1, i) соединяется со всеми вершинами (m_k, i), $k = 2, 3, \dots, p$; при помощи дуг, «приведенные» длины которых равны τ_{m_k}, $k = 2, 3, \dots, p$, и, наоборот, все вершины (m_k, i), $k = 2, 3, \dots, p$, соединяются с вершиной (m_1, i) дугами, «приведенные» длины которых равны τ_{m_1} ($\tau_{m_1} = \tau_1$ по существу равно нулю, но для возможности реализации алгоритма Форда мы берем в качестве τ_1 достаточно малое, но отличное от нуля число).

Описанное множество вершин и дуг образует граф. Заметим, что из этого описания видна специфическая роль пешеходного маршрута Γ_1 для реализации пересадок. Действительно, пересадка в узловой точке x_i с трассы Γ_ν на трассу Γ_λ осуществляется с помощью перехода с вершины (ν, i) на вершину $(1, i)$ и последующим переходом на вершину (λ, i). Вместе с тем, за счет рассмотрения «приведенных» длин нам удается избежать трудностей в применении алгоритма Форда, возникающих из различных скоростей различных транспортных средств (рис. 5).

Для простоты машинной реализации мы перенумеруем все вершины графа последовательными числами, однако так, чтобы, во-первых, вершинам $(1, i)$ соответствовал номер i, и, во-вторых, по номеру вершины можно было восстановить номер узловой точки и трассы, из которых образована данная вершина. Мы не будем описывать примененного нами метода нумерации.

Для связности изложения мы приведем описание алгоритма Форда [1, стр. 79]. Обозначим вершины графа точками $\alpha_1, \alpha_2, \ldots, \alpha_{N_0}$. Отыскание кратчайшего пути из точки α_{i_1} в точку α_{i_2} $(i_1, i_2 = 1, 2, \ldots, N_0)$ осуществляется следующим образом.

1°. Сопоставим каждой вершине (точке) α_i параметр λ_i; первоначально берем $\lambda_{i_1} = 0$ и $\lambda_i = +\infty$ при $i \neq i_1$.

2°. Ищем такую дугу (α_i, α_j), для которой $\lambda_j - \lambda_i > l(\alpha_i, \alpha_j)$; затем заменяем параметр λ_j параметром $\lambda_j' = \lambda_i + l(\alpha_i, \alpha_j) < \lambda_j$; заметим, что $\lambda_j' > 0$ при $j \neq i_1$. Продолжаем проверку до тех пор, пока еще остается хотя бы одна дуга, для которой можно уменьшить λ_i.

3°. После того как параметры установятся (никакой параметр больше нельзя уменьшить), найдется такая вершина α_{p_1}, что $\lambda_{i_2} - \lambda_{p_1} = l(\alpha_{p_1}, \alpha_{i_2})$; в самом деле, параметр λ_{i_2} при нашем процессе монотонно уменьшался, а α_{p_1} — последняя вершина, послужившая для его уменьшения. Точно так же найдется вершина α_{p_2}, для которой $\lambda_{p_1} - \lambda_{p_2} = l(\alpha_{p_2}, \alpha_{p_1})$, и т. д.

Последовательность $\lambda_{i_2}, \lambda_{p_1}, \lambda_{p_2}, \ldots$ — строго убывающая (поэтому мы положили $\tau_1 > 0$) и, следовательно, в некоторый момент будет $\alpha_{p_{k+1}} = \alpha_{i_1}$. Образовавшееся при этом значение параметра λ_{i_2} $(i_2 = 1, 2, \ldots, N_0)$ есть длина кратчайшего пути из α_{i_1} в α_{i_2}, причем этот путь проходит через вершины $\alpha_{i_1}, \alpha_{p_k}, \alpha_{p_{k-1}}, \ldots, \alpha_{p_2}, \alpha_{p_1}, \alpha_{i_2}$, т. е. $\mu = [\alpha_{i_1}, \alpha_{p_k}, \alpha_{p_{k-1}}, \ldots, \alpha_{p_2}, \alpha_{p_1}, \alpha_{i_2}]$ есть кратчайший путь между вершинами α_{i_1} и α_{i_2} в указанном направлении.

3. МАШИННАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА

Кодирование исходной информации. Реализация алгоритма на ЭВМ в значительной степени зависит от метода кодирования исходной информации, который мы сейчас опишем. Как следует из предыдущего, мы должны закодировать информационную таблицу, таблицу скоростей и времен ожиданий и матрицу приведенных корреспонденций.

Рассмотрим p-ю строку информационной таблицы

p	$x_i,\ x$	$\rho(x_i,\ x_j)$	$z_1^{(p)}$	$z_2^{(p)}$	$z_3^{(p)}$	$\ldots$	$z_l^{(p)}$	$\ldots$	$z_{M-1}^{(p)}$	$z_M^{(p)}$

в которой $z_l^{(p)}$ равняется 1 или 0 в зависимости от того, является или нет пара (x_i, x_j) смежной на трассе Γ_l. Эта строка кодируется нами в трех ячейках $\rho + p$, $\alpha + p$ и $\beta + p$ (мы предполагаем, что рассматриваемая машина имеет трехадресную систему команд и запоминающее устройство машины содержит не менее 4096 ячеек). В ячейке $\rho + p$ хранится число $\rho(x_i, x_j)$. Распределение информации в ячейке $\alpha + p$ очевидно из нижеследующего рисунка:

		I A	II A	III A
$\alpha + p$:	—	$\rho + p$	i	j

Наконец, предположим, что число трасс M совпадает (или во всяком случае не больше) с числом разрядов в ячейке. Сопоставим l-й разряд ячейки l-й трассе. Тогда набор $z^{(p)} = z_1^{(p)} z_2^{(p)}, \ldots, z_M^{(p)}$ помещается в ячейку $\beta + p$.

Числа τ_l и ν_l мы храним в ячейках $T + l$ и $V + l$ $(l = 1, 2, \ldots, M)$. Элементы матрицы $S = \|s_{ij}\|$ размещены в запоминающем устройстве в порядке их следования по строкам, т. е.

$$s_{ij} \to S + (i-1)N + j; \qquad i, j = 1, 2, \ldots, N.$$

Схема решения задачи. Обозначим через D_p оператор, который по ячейкам $\alpha + p$ и $\beta + p$ формирует ячейки:

	коп				
$\gamma_p + q_l$	$t_p + q_l$	l	i	l	j

(1)

Здесь в разрядах кода операции записан адрес ячейки, в которой хранится число $\rho(x_i, x_j) / \nu_{q_l}$ (также вычисляемое оператором D_p); далее, все остальные разряды ячейки $\gamma_p + q_l$ разбиты на четыре части, и в них вписаны последовательно числа l, i, l, j; числа q_l определяются из формулы $q_l = \sum_{s=1}^{l} z_s^{(p)}$ для тех и только тех значений l, для которых $z_l^{(p)} \neq 0$; наконец,

$$t_{p+1} = t_p + \sum_{s=1}^{M} z_s^{(p)} + 1, \quad \gamma_{p+1} = \gamma_p + \sum_{s=1}^{M} z_s^{(p)} + 1,$$

$p = 1, 2, \ldots, P$; t_1 и γ_1 — некоторые ячейки.

Применяя оператор $D = \prod_{p=1}^{P} [D_p]$, мы в ячейках $\gamma_1 + q$ $(q = 1, 2, \ldots, q^*$; где $q^* = \sum_{p=1}^{P} \sum_{s=1}^{M} z_s^{(p)})$ получим информацию о смежных вершинах графа (расщепление узловых точек x_i и x_j по трассе Γ_l) (l, i) и (l, j), $i \neq j$, $i, j = 1, 2, \ldots, N$; $l = 1, 2, \ldots, M$; а в ячейках $t_1 + q$ $(q = 1, 2, \ldots, q^*)$ — приведенные длины дуг, соединяющих эти вершины.

Нам остается построить ячейки, в которых будет записана информация о смежных вершинах $(1, i)$ и (l, i), соответствующих расщеплению узловой точки x_i по всем трассам, проходящим через нее. Эта информация (вырабатываемая оператором F_1) имеет вид

$T + l$	1	i	l	i

(2)

или

$T + 1$	l	i	1	i

(2′)

Отметим, что приведенная длина соответствующих дуг равна времени ожидания l-го транспорта и поэтому не требует специального вычисления.

Эту информацию мы помещаем в последовательные ячейки, начиная с номера $\gamma_1 + q^* + 1$.

Чтобы построить ячейки типа (2), рассмотрим (при фиксированном p) ячейки $\gamma_p + 1$, $\gamma_p + 2$, $\ldots$, $\gamma_p + \sum\limits_{s=1}^{M} z_s^{(p)}$. Выделим из ячейки $\gamma_p + 1$ те разряды, в которых записана информация о вершине (l, i) (см. рис. 1). Эта информация объединяется с информацией, выделенной из тех же разрядов ячеек $\gamma_p + 2$, $\gamma_p + 3$, $\ldots$, $\gamma_p + \sum\limits_{s=1}^{M} z_s^{(p)}$. Формирование ячеек (2′) очевидно. Однако одна и та же узловая точка x_i может входить в различные смежные пары по одной и той же трассе. Поэтому оператор F_1 при переходе от одного p к следующему проверяет все ранее построенные им информационные ячейки (типа (2)) и при каждом данном p формирует новые информационные ячейки только для тех узловых точек и тех трасс, которые не встречались ни в одной из предыдущих строк информационной таблицы. Делается это исключительно с целью экономии объема запоминающего устройства машины и главным образом ради времени работы алгоритма Форда. Как можно видеть из описания алгоритма Форда, порядок нумерации дуг безразличен, и поэтому мы не дали более подробного описания работы оператора F_1.

Итак, мы получили последовательность ячеек $\gamma_1 + q$; $q = 1, 2, \ldots, Q$ (выработанных операторами D и F_1), содержащих информацию вида:

	l_1	i_1	l_2	i_2	
Адрес приведенной длины дуги	Номер трассы	Номер узловой точки	Номер трассы	Номер узловой точки	(3)

Далее оператор F_2 перенумеровывает *последовательными* номерами все различные вершины графа и выводит на печать таблицу старых обозначений (l, i) и новых номеров (s). При этом вершине $(1, i)$ сопоставляется номер i, а в остальном метод нумерации совершенно произволен. Эта нумерация вводится для того, чтобы сопоставить s-й вершине графа ячейку с номером $L + s$, в которой будет храниться параметр λ_s. Отметим также, что поиск оптимальной трассы имеет смысл лишь для вершин $(1, i)$, $i = 1, 2, \ldots, N$, т. е. для первых N вершин графа (в новой нумерации).

Наконец, каждую ячейку типа (3) мы заменяем четырьмя командами вида

$k + 1.$	$+$	l_{ij}	λ_i	χ
$2.$	$>$	λ_j	χ	$k + 5$
$3.$	ПЧ	χ	—	λ_j
$4.$	ПУ	—	Начало	—
$5.$				

где λ_i и λ_j — номера соответственно ячеек, хранящих параметры λ_i и λ_j (соответствующие вершинам графа (l_1, i_1) и (l_2, i_2)). Команда ПУ — Начало — передает управление к ячейке, аналогичной ячейке $k + 1$, но полученной из ячейки $\gamma_1 + 1$. Легко видеть, что указанные четыре команды осуществляют один шаг алгоритма Форда.

Осуществив указанную замену для всех ячеек типа (3) и записав полученные четыре команды последовательно одну после другой, мы получим программу, реализующую весь алгоритм Форда.

Заметим, наконец, что с точки зрения экономии машинного времени выгоднее указанную замену выполнить один раз и записать в запоминающее устройство машины. Если, однако, эта программа не помещается в запоминающем устройстве, то эту замену можно выполнять каждый раз заново. Отыскание самого оптимального пути (номеров вершин, через которые он проходит) выполняется на основании правил, указанных в п. 3° алгоритма Форда (раздел 2). На отыскании суммарных пассажиро-потоков мы здесь не останавливаемся.

Решение задачи для города Ростова-на-Дону. Описанный метод был запрограммирован на трехадресную машину «Арагац», ячейка памяти которой содержит 42 разряда, число команд в программе — меньше 200.

В поставленной задаче имелись следующие данные: 43 узловые точки (на самом деле число узловых точек было больше, однако некоторые из них были отброшены, что будет описано ниже) и 38 трасс (из них — 37 маршрутных, 1 пешеходная).

Отбрасывание некоторых узловых точек делалось исходя из следующего: если некоторая узловая точка x_i с общей городской сетью соединена при помощи одной и только одной узловой точки x_j, то узловую точку x_i можно отбросить, так как кратчайший путь, ведущий из узловой точки x_i в другие узловые точки и наоборот, обязательно проходит через узловую точку x_j; при этом корреспонденция области $Д_i$ прибавляется к корреспонденции области $Д_j$. Это делается до тех пор, пока ни одной другой узловой точки нельзя отбросить (это отбрасывание было сделано для экономии машинного времени). Конечно, задачу можно решить и для всех исходных узловых точек.

Искусственные трассы при решении задачи для Ростова-на-Дону не рассматривались.

В информационной таблице было 66 строк. Построив граф городского транспорта города (на это ушло около 6 мин.), мы получили граф с 254 вершинами и около 1000 дуг (дуг разной длины — около 250, остальные дуги были пересадочные, имеющие длины, соответствующие временам ожидания транспорта, движущегося по данным трассам). Было рассчитано $43 \cdot 42 = 1806$ кратчайших путей. Время отыскания всех путей из одной вершины ко всем остальным колеблется от 7 до 30 мин. (см. ниже «Замечание по поводу применения алгоритма Форда»). Всего на решение задачи потрачено около 20 час. (с выводом данных).

Задача была решена в ВЦ АН АрмССР в ЕрГУ в лаборатории экономико-математических исследований под руководством кандидата экономических наук А. Х. Карапетяна.

Замечание по поводу применения алгоритма Форда. С 5 по 8 мая 1965 г. в г. Ереване проходил Всесоюзный научный семинар, посвященный комплексным задачам городского пассажирского транспорта с использованием математических методов и ЭВМ. Содержание данной статьи было доложено на семинаре. Сотрудницей ЦЭМИ АН СССР Т. М. Великановой было сделано замечание о применении алгоритма Форда в описанном толковании. Она предложила следующее: после получения графа городского транспорта в машине хранить его в виде, удобном для применения модифицированного метода алгоритма Форда (метод «Метлы»), сущность которого заключается в разумном выборе порядка проверки дуг (движение по вершинам, а не по дугам). С применением указанного метода «Метла» машинное время резко сокращается. В этой связи автор выражает свою благодарность Т. М. Великановой.

ЛИТЕРАТУРА

1. К. Берж. Теория графов и ее применение. М., Изд-во иностр. лит., 1962.

Поступила в редакцию
13 VIII 1966