

ОПРЕДЕЛЕНИЕ ОПТИМАЛЬНОЙ СТРУКТУРЫ ТЕХНИЧЕСКИХ СРЕДСТВ И ТЕХНОЛОГИИ ОБРАБОТКИ ИНФОРМАЦИИ

Э. С. ХАЗАНОВИЧ, И. К. ЖАБИН

(Тула)

ПОСТАНОВКА ЗАДАЧИ

При проектировании автоматизированных систем управления (АСУ) возникает задача выбора варианта технологии обработки данных и используемых при этом технических средств. Решение ее должно дать оптимальный вариант технологии, обеспечивающий при заданных ограничениях по времени и объемам информации минимум затрат на ее обработку.

Для каждой из функций АСУ совокупность различных вариантов операционной технологии представляется в виде графа, узлы которого — используемые в данном случае технические средства и носители информации, а дуги — взаимосвязи и показатели последовательности каждой операции. Стоимость использования данных технических средств задается в виде оценки длины соответствующей дуги. Например, при проектировании АСУ материально-техническим снабжением в экономическом районе различные варианты организации расчета сводных порайонных потребностей могут быть представлены в виде графа (рис. 1).

Для любой из функций АСУ, рассматриваемых отдельно, оптимальный вариант технологии находится как путь минимальной длины в вышеуказанном графе.

Взаимосвязь функций АСУ усложняет проблему выбора варианта и выдвигает требование поиска глобального оптимума для всей совокупности функций системы.

Математически такая задача может быть представлена как поиск совокупностей минимальной длины N путей в N графах, взаимосвязанных между собой так, что выбор пути в i -м графе влияет на выбор пути в $(i+1)$, $(i+2)$, ..., N -х графах.

Ниже рассматривается модель этой задачи и предлагается алгоритм ее решения, обеспечивающий получение строгого оптимума. В основе алгоритма лежит процедура построения графа на множестве N графов с последующим отысканием в этом графе пути минимальной длины.

МАТЕМАТИЧЕСКАЯ ФОРМУЛИРОВКА ЗАДАЧИ

Пусть имеется N частных ориентированных сетей, составленных для решения N задач проблемы. Требуется отыскать также N полных путей (в каждой сети по одному), общая стоимость которых C была бы минимальной. Множество номеров задач упорядочено, и задачи, номера которых отличаются на 1, назовем соседними задачами.

Введем обозначения. Каждая частная сеть состоит из множества вершин A и дуг B . Взаимосвязанная последовательность дуг от начальной до

конечной вершин i -й сети $(b_0, \dots, b_k, \dots, b_p)$ образует путь графа l_j , который будет определять j -й вариант решения i -й задачи, где $j = 1, \dots, m_i$. Обозначим j -й вариант решения i -й задачи через e_{ij} .

Множество путей i -й сети будет представлять множество $E_i = e_{i1}, \dots, e_{im}$ вариантов решения i -й задачи, где $i = 1, \dots, N$. Для каждой задачи m_i имеет свое значение.

Каждой дуге b_{hk} из вершин h в вершину k в i -й задаче будет соответствовать s_{hk} — стоимость выполнения какой-то одной операции.

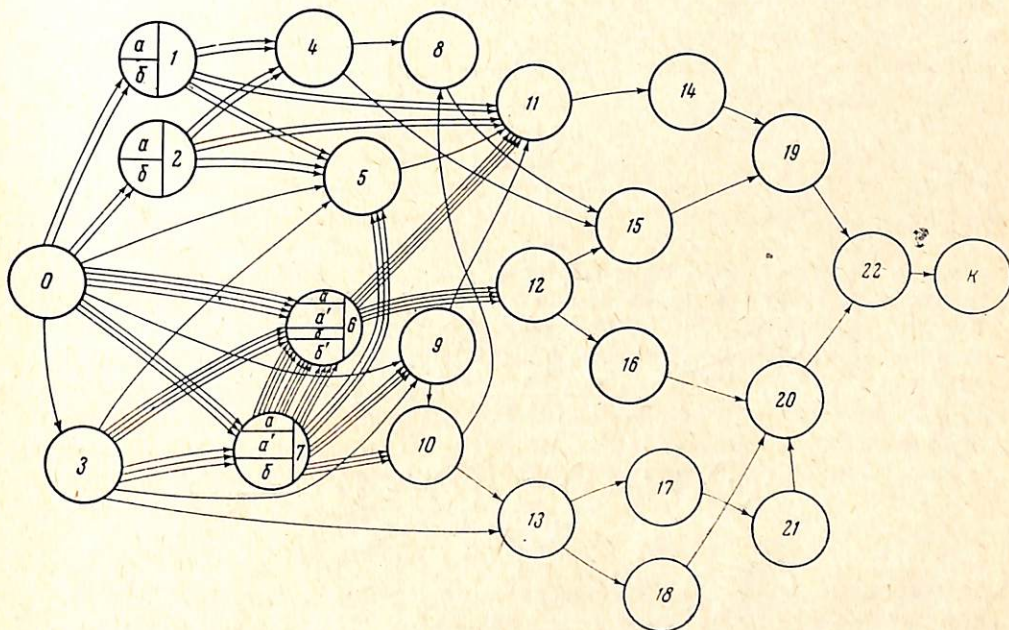


Рис. 1. Сеть вариантов организационно-технических решений (расчет сводных порайонных заявок): 1 — бланк с графическими отметками: a — обычный, b — с заранее отмеченными постоянными признаками; 2 — бланк со стилизованным текстом: a — обычный, b — с заранее отмеченными постоянными признаками; 3 — ведомость; 4 — отсортированный массив бланков; 5 — перфолента; 6 — перфокарта: a — обычная, a' — обычная с заранее отперфорированными постоянными признаками, b — дуаль, b' — дуаль-карта с заранее отперфорированными постоянными признаками; 7 — карта: a — обычная, a' — с краевой перфорацией обычной, b — с заранее отперфорированными постоянными признаками; 8 — перфолента с упорядоченным расположением информации; 9 — ленточная карта; 10 — массив отсортированных карт; 11 — информация, введенная в ЭВМ в неупорядоченной форме; 12 — массив перфокарт, отсортированный по СПМ; 13 — массив информации отсортированной; 14 — массив информации, отсортированной на ЭВМ; 15 — информация, введенная в ЭВМ в упорядоченной форме; 16 — сводная ведомость порайонной потребности-табулаграмма; 17 — сводная ведомость — счетно-клавишные машины; 18 — сводная ведомость — счетно-клавишные машины с автоматическим получением перфоленты; 19 — сводная ведомость порайонной потребности; 20 — информация введена в ЭВМ в упорядоченной форме после расчета потребности; 21 — перфолента с упорядоченным расположением информации; 22 — информация, записанная на магнитную ленту

Сумма $\sum s_{hk}$ для дуг, составляющих полный j -й путь i -й сети l_{ij} , определит стоимость варианта решения задачи S_{ij} , т. е.

$$S_{ij} = \sum_{s_{hk} \in l_{ij}} s_{hk}.$$

Учитывая принятые обозначения, нам необходимо найти множество

вариантов $R = e_{1j_i}, e_{2j_i}, \dots, e_{Nj_i}$, при которых функция $C = \sum_{i=1}^N S_{ij}$ достигала бы минимума.

При решении i -й задачи j -м вариантом в общем случае могут использоваться результаты решений (как промежуточные, так и окончательные) других задач какими-то вариантами; в свою очередь результаты (как промежуточные, так и окончательные) решения i -й задачи j -м вариантом могут быть использованы при решении других задач.

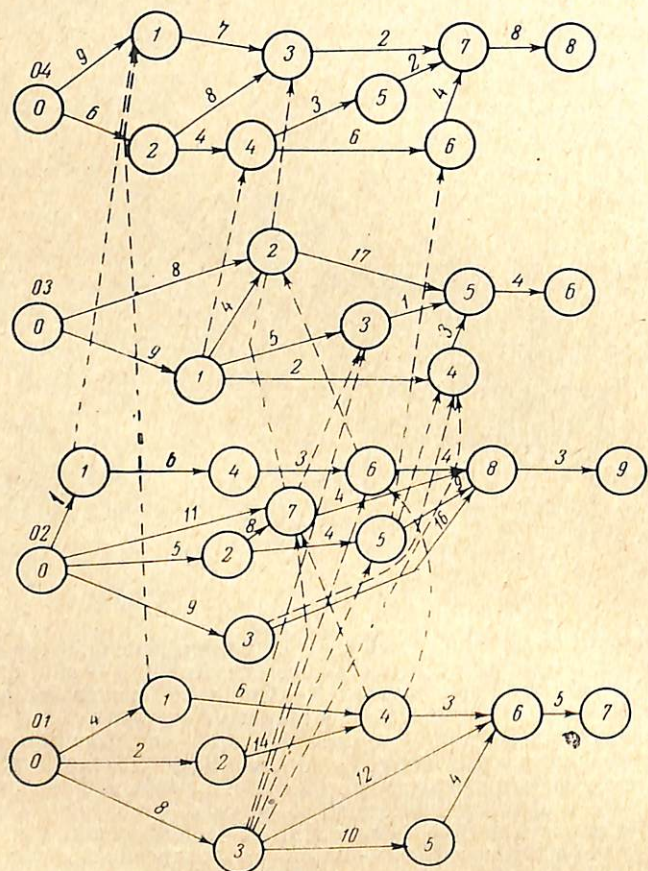


Рис. 2. Пример связи задач

Приведем определения. Дугу, отображающую использование промежуточного или конечного результата решения одной задачи при решении другой задачи, назовем *связью* и обозначим через W_{ijg} , где $i = 1, 2, \dots, N$, а $g = 1, 2, \dots, N$, т. е. i -я задача может иметь связи как с $1, 2, \dots, (i+1)$ -й задачами, так и с $(i+1), \dots, N$ -й задачами.

Вариант e_{gf} решения g -й задачи назовем *связанным*, если хотя бы к одной вершине соответствующего ему пути l_{gf} подходит хотя бы одна связь от какого-либо из путей любой из N задач общей проблемы и отходит хотя бы одна связь к вариантам решения других задач.

Вариант e_{gf} множества E_g назовем *свободным*, если ни к одной из вершин соответствующего ему пути l_{gf} не подходит и ни из одной из вершин этого пути не отходит ни одной связи.

Вариант, к которому подходят связи, но из которого не отходит ни одной связи, назовем *частично связанным*.

Вариант, к которому не подходит ни одной связи, но от которого отходит связи, назовем *частично свободным*.

Соответственно задачу, один из вариантов решения которой связан, назовем *связанной*. Будем считать, что если e_{gf} -й вариант полностью или частично связан с одним или несколькими вариантами i -й задачи, то g -я задача может быть решена этим вариантом в том и только в том случае, если i -я задача решается каким-либо из связанных с e_{gf} вариантов.

На решения всех задач проблемы накладываем следующие ограничения: любая из N задач должна быть решена одним и только одним из m_i -х вариантов. Поэтому, если какой-либо из вариантов g -й задачи имеет связи с несколькими вариантами i -й задачи, то такие связи относятся к классу связей «или-или». Это означает, что для решения g -й задачи f -м вариантом выбирается один и только один из связанных с ним вариантов решения i -й задачи. При этом, если разные вершины l_{gf} -го пути в g -й задаче связаны с вершинами разных путей в i -й задаче, то вариант e_{gf} по условию недопустим и должен быть отброшен.

Если же какой-либо из вариантов g -й задачи имеет связи с несколькими вариантами в различных задачах (в каждой из них хотя бы с одним), то такие связи относятся к классу связей «и-и». Это означает, что для решения g -й задачи f -м вариантом должны быть реализованы все связанные с ним варианты других задач.

В общем случае при решении рассматриваемых задач имеются как варианты, так и задачи свободные, связанные типом связи «или-или» и связи «и-и», т. е. будем иметь N частных сетей, определенным образом связанных друг с другом (рис. 2).

АЛГОРИТМ РЕШЕНИЯ ЗАДАЧ

Первый вариант. Нахождение оптимума для описанного комплекса задач проблемы (см. рис. 2) с помощью известных алгоритмов расчета сетей не представляется возможным. Воспользуемся поэтому следующей процедурой поиска решения.

I этап вычислений. Определяем все m_i вариантов решения каждой из N задач, их связи с вариантами решения других задач и стоимость. Используем для этого следующий алгоритм.

I -й шаг. Начальной вершине a_0 в каждой сети ставим в соответствие число $\lambda_0^q = 0$.

Общий q -й шаг. Просматриваем все вершины сети от a_0 до a_p и заменяем числа $\lambda_{k\xi}^{q-1}$, полученные на предыдущем $(q-1)$ -м шаге, на числа $\lambda_{k\xi}^q > \lambda_{k\xi}^{q-1}$ по формуле

$$\lambda_{k\xi}^q = \lambda_{k\xi}^{q-1} + s_{hk}, \quad \xi = 1, \dots, v; \quad h = 1, \dots, \alpha. \quad (1)$$

v — определяет количество чисел λ , соответствующих какой-то вершине. В выражении (1) для каждого h ставим свое значение s_{hk} . Все полученные числа $\lambda_{k\xi}^q$ будут соответствовать рассматриваемой вершине. Каждому числу ставится в соответствие последовательность вершин, которую мы прошли при определении числа $\lambda_{k\xi}^q$ от начальной вершины.

При рассмотрении вершины на $(q+1)$ -м шаге, при определении чисел λ_{k+1}^{q+1} , соответствующих вершине $k+1$, в выражение (1) ставим значения $\lambda_{k\xi}^q$, если из вершины k идет дуга в вершину $k+1$. Следовательно, вершина $k+1$ получает γ_k значений чисел λ_{k+1}^{q+1} от данной дуги. Если же

в эту вершину входят, в общем случае, дуги от α вершин, непосредственно предшествующих вершине $(k+1)$, то количество чисел λ для вершины $(k+1)$ будет

$$\gamma_{k+1} = \sum_{\beta=1}^{\alpha} \gamma_{\beta}.$$

Величина γ_{k+1} , следовательно, будет определять количество путей, проходящих через данную вершину.

Общий шаг повторяем до тех пор, пока не просмотрим все вершины сети и по всем входящим в каждую вершину дугам.

В результате мы получим для каждой из N задач множество $E_i = e_{i1}, \dots, e_{im}$ со значениями S_{ij} и с рядом определяющих вариант вершин $a \in A$.

Из общей сетевой схемы решения задач выписываем связи вида W_{ijgf} . Эти связи определяются также вершинами $a_h \in e_{ij}$ и $a_k \in e_{gf}$. Если вершина a_h входит в ряд других вариантов задачи g , тогда она будет определять связанность этих вариантов с вариантом e_{ij} .

На основе этого построим матрицу смежности вариантов решения связанных задач. Эта матрица D будет иметь порядок mim_g .

Общий элемент ее d_{jgf} определит связанность варианта e_{ij} с вариантом e_{gf} : $d = 0$, когда варианты не связаны; $d = 1$, когда варианты e_{ij} и e_{gf} связаны между собой. В соответствии с приведенным выше алгоритмом определим для частных сетей (рис. 2) все варианты решения задач e_{ij} , значения S_{ij} и построим матрицы смежности их друг с другом.

Для задачи 01. Вариант, соответствующий пути l , проходящему через вершины 0, 1, 4, 6, 7, обозначим $e_{01,1}$:

$$\begin{aligned} S_{01,1} &= \sum s_{hk} = 4 + 6 + 3 + 5 = 18; \\ l(0, 2, 4, 6, 7) &- e_{01,2}; & S_{01,2} &= 24; \\ l(0, 3, 6, 7) &- e_{01,3}; & S_{01,3} &= 25; \\ l(0, 3, 5, 6, 7) &- e_{01,4}; & S_{01,4} &= 27. \end{aligned}$$

Для задачи 02.

$$\begin{aligned} l(0, 1, 4, 6, 8, 9) &- e_{02,1}; & S_{02,1} &= 17; \\ l(0, 7, 8, 9) &- e_{02,2}; & S_{02,2} &= 18; \\ l(0, 2, 7, 8, 9) &- e_{02,3}; & S_{02,3} &= 20; \\ l(0, 2, 5, 8, 9) &- e_{02,4}; & S_{02,4} &= 21; \\ l(0, 3, 8, 9) &- e_{02,5}; & S_{02,5} &= 28. \end{aligned}$$

Для задачи 03.

$$\begin{aligned} l(0, 2, 5, 6) &- e_{03,1}; & S_{03,1} &= 28; \\ l(0, 1, 2, 5, 6) &- e_{03,2}; & S_{03,2} &= 34; \\ l(0, 1, 3, 5, 6) &- e_{03,3}; & S_{03,3} &= 19; \\ l(0, 1, 4, 5, 6) &- e_{03,4}; & S_{03,4} &= 18. \end{aligned}$$

Для задачи 04.

$$\begin{aligned} l(0, 1, 3, 7, 8) &- e_{04,1}; & S_{04,1} &= 26; \\ l(0, 2, 3, 7, 8) &- e_{04,2}; & S_{04,2} &= 24; \\ l(0, 2, 4, 5, 7, 8) &- e_{04,3}; & S_{04,3} &= 23; \\ l(0, 2, 4, 6, 7, 8) &- e_{04,4}; & S_{04,4} &= 28. \end{aligned}$$

Определяем матрицы смежности для задачи 01 и 02.

Между частными сетями этих задач есть связи W , выходящие из вершины 3 сети 01 и подходящие к вершине 5, 6 и 7 сети 02, и связи W , выходящие из вершины 4 сети 01 и подходящие к вершинам 6 и 7 сети 02.

Все варианты задачи 02, содержащие вершины 5, 6 и 7 сети 02, связаны со всеми вариантами задачи 01; содержащие вершины 6 и 7 сети 02, связаны со всеми вариантами задачи 01, содержащими вершину 4.

Просматривая состав путей в задачах 01 и 02, находим матрицу смежности $D_{01, 02}$

$$D_{01,02} = \begin{vmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{vmatrix}$$

Аналогично определяются матрицы смежности для всех остальных наборов задач. Для задач 01 и 04 имеем $D_{01, 04}(W_{01,04_1}; W_{01,04_2}; W_{01,04_3}; W_{01,04_4})$;

$$D_{01,04} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix}$$

Для задачи 01 и 03 — $D_{01, 03}(W_{01,03_1}; W_{01,03_2}; W_{01,03_3}; W_{01,03_4})$;

$$D_{01,03} = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Для задач 02 и 03 — $D_{02, 03}(W_{02,03_1}; W_{02,03_2}; W_{02,03_3}; W_{02,03_4}; W_{02,03_5}; W_{02,03_6})$;

$$D_{02,03} = \begin{vmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Для задач 02 и 04 — $D_{02, 04}(W_{02,04_1}; W_{02,04_2}; W_{02,04_3}; W_{02,04_4}; W_{02,04_5}; W_{02,04_6})$;

$$D_{02,04} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix}$$

Для задач 03 и 04 — $D_{03, 04}(W_{03,04_1}; W_{03,04_2}; W_{03,04_3}; W_{03,04_4}; W_{03,04_5}; W_{03,04_6})$;

$$D_{03,04} = \begin{vmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{vmatrix}$$

В матрице $D_{01, 02}$ в последнем столбце все элементы $d = 0$. Это значит, что вариант $e_{02, 5}$ является частично свободным по отношению к задаче 01.

Поскольку свободные варианты также должны участвовать в решении проблемы, вариант $e_{02,5}$ может быть связан со всеми вариантами задачи 01, т. е. все $d = 0$ последнего столбца заменяются на $d = 1$. Это возможно, однако, только для матриц смежности соседних задач.

В случае частично связанных вариантов для соседних задач i и $i + 1$ строка в матрице смежности со всеми $d = 0$ может быть заменена на $d = 1$ только в таком столбце этой матрицы, все элементы которого $d = 0$,

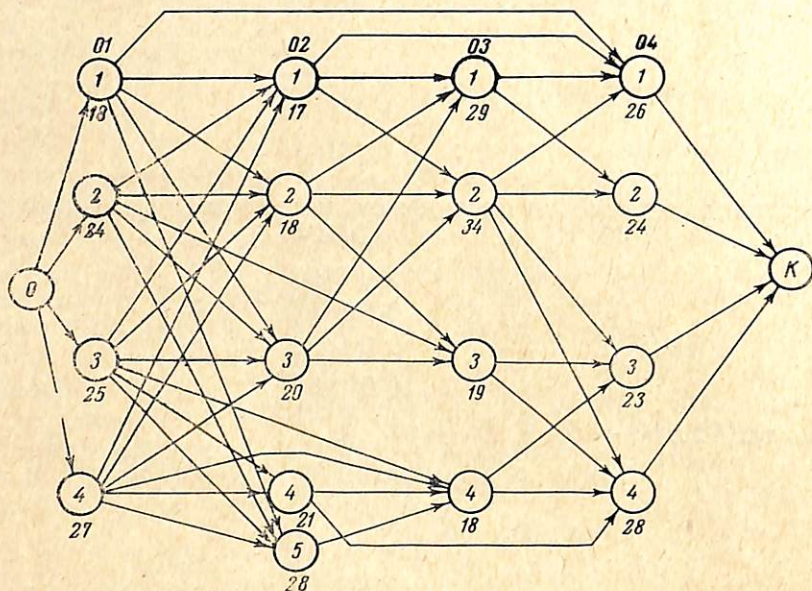


Рис. 3. Сеть вариантов решения проблемы

т. е. частично связанные варианты, могут включаться на последующих этапах решения как связанные, но только с вариантами, свободными полностью или частично.

Каждому варианту e_{ij} можно поставить в соответствие вершину графа v_{ij} . Расположим все вершины i -й задачи в один столбец, $(i + 1)$ -й — в другой и т. д. В результате будем иметь множество вершин графа, расположенных определенным образом, соответствующих всем вариантам e_{ij} решения всех N задач проблемы.

Все вершины графа соединены между собой дугами u , когда элемент соответствующей матрицы D $d = 1$. Преобразуем граф в сеть путем добавления начальной и конечной вершин, которые соединим соответственно дугами с вершинами первой задачи и последней. Каждой дуге $u_{ij(i+1)}$ поставим в соответствие число $S_{i+1,j}$, определенное нами ранее и представляющее собой стоимость решения $(i + 1)$ -й задачи вариантом $e_{i+1,j}$.

В результате получим сеть специальной структуры (рис. 3). Все входящие в нее дуги u будут иметь оценку $S_{i+1,j}$ (оценки $S_{i+1,j}$ указаны под каждой из вершин графа).

Для упрощения все дуги $W_{i,j,gf}$ приняты нами с нулевой оценкой. В случае, если влияние связи e_{ij} -го варианта с e_{gf} -м выражается в необходимости произвести затраты $s'_{ij,gj}$ и $g = i + 1$, т. е. задачи смежные, величина $s_{ij,gj}$ добавляется к оценке дуги $u_{ij(i+1)}$.

Дуги вида $u_{ij(i+\theta)_j}$, где $\theta > 1$, приняты с оценкой $S_{i+\theta, j} = 0$. Дуги, входящие в конечную вершину, также будут иметь оценку $S = 0$.

Очевидно, что решение задач проблемы будет заключаться в отыскании полного пути L общей сети минимальной длины.

Но необходимо, чтобы этот путь к какой-то одной вершине при условии наличия связей «и — и» включал в себя все те вершины сети, из которых выходят такие связи, т. е. дуги вида $u_{ij(i+\theta)_j}$, где $\theta > 1$.

II этап вычислений. *1-й шаг.* Отыскиваем начальную вершину сети (вершина, в которую не входит ни одной дуги).

Общий шаг. 1. Просматриваем все дуги, выходящие из вершин i -го ряда, и отыскиваем те из них, которые входят в вершину $(i + \theta)$ -го ряда, т. е. дуги вида $u_{ij(i+\theta)_j}$, где $\theta > 1$; $i = 1, \dots, N$. (Ряды вершин просматриваются последовательно, начиная с 1-го ряда.) Эти дуги соответствуют связям при условии «и — и». Фиксируем вершины v_{ij} и вершину $v_{i+\theta, j}$.

2. Просматриваем все дуги, входящие в вершину $v_{i+\theta, j}$, и отыскиваем среди них дуги вида $u_{(i+t)_j(i+\theta)_j}$, где $1 \leq t \leq (\theta - 2)$. Фиксируем вершины $v_{(i+t)_j}$. Исключаем из сети связи класса «и — и», входящие в вершину $v_{i+\theta, j}$.

3. Используя известные алгоритмы построения путей в сети, подходящих к заданным вершинам, определяем пути l_n , подходящие к вершине $v_{i+\theta, j}$ и включающие в себя все вершины, из которых выходят к вершине $v_{i+\theta, j}$ связи при условии «и-и», т. е. получаем частные пути общей сети, начальными вершинами которых будут вершины i -го ряда и конечной вершиной $v_{i+\theta, j}$. Причем, один путь l включает в себя вершины, которые относительно как вершины $v_{i+\theta, j}$, так и друг друга, по классу связей, выходящих из них, находятся в связи «и-и».

Эти пути проходят через одну из вершин каждого ряда от $i + 1$ до $i + (\theta - 1)$.

Тогда $l_n = (v_{ij}; v_{i+1, j}, \dots, v_{i+\theta, j})$.

Если какая-то вершина $(i + t)$ -го ряда сети, из которой выходит связь «и-и» к вершине $v_{i+\theta, j}$, не принадлежит ни одному из путей l_n , и если ни один из l_n путей не проходит через другие вершины этого ряда, из которых выходит связь «и-и», то вершина $v_{i+\theta, j}$ и все дуги, входящие в нее, из сети исключаются (как недопустимый вариант решения $(i + \theta)$ -й задачи).

Общий шаг вычислений продолжаем до тех пор, пока не просмотрим все вершины $(N - 2)$ -го ряда вершин сети.

4. Строим все пути от начального до конечного события сети, которые включают в себя частные пути l_n , т. е. получим r полных путей сети L .

Каждый из r полных путей сети может включать в себя как частные пути от вершины i -го ряда сети до $(i + \theta)$ -го ряда, так и i_1 -го ряда до $(i_1 + \theta_1)$ -го ряда сети, где $i_1 > i$.

Для построения путей L также можно воспользоваться известными алгоритмами построения путей в сети, но при условии, что любой из этих путей содержит в себе какой-то один или более частных путей l_n . Пути L можно выделить из общей сети и представить в виде ее части G_1 , но допустимыми эти пути в такой сети будут тогда, когда в ней не образуется дополнительного пути. Поэтому сеть G_1 можно построить, не допуская объединения любых двух или более путей L через какую-то одну общую для них вершину v . Ограничимся тогда тем, что назовем совокупность путей L частью сети G_1 .

Вторую часть сети — G_2 получим путем исключения из общей сети всех вершин v , к которым подходят связи «и-и», и всех дуг, входящих в них.

Это исключение вершин и дуг позволит получить сеть G_2 допустимых

вариантов решения задач, в которой не содержатся варианты со связями класса «и-и».

Полные пути сетей G_1 и G_2 будут соответствовать вариантам решения всех задач проблемы.

III этап вычислений. В сети G_1 определяем продолжительность каждого из r путей L

$$L = \sum_i S_{(i-1)_j i_j}.$$

Из всех путей L определяем путь минимальной длины L_1 .

Далее находим путь минимальной длины в сети G_2 , используя известные методы поиска такого пути.

1-й шаг. Приписываем начальной вершине v_0 число $\lambda_0 = 0$.

Общий шаг. Просматриваем все вершины i -го ряда сети и приписываем им числа

$$\lambda_{ij} = \min (\lambda_{i+1, j} + S_{(i-1)_j i_j}). \quad (2)$$

Каждой вершине i -го ряда приписываем также индекс вершины $(i-1)$ -го ряда, при котором выражение (2) достигает минимума.

Общий шаг вычислений продолжаем до конечной вершины сети. В результате определим путь L_2 сети минимальной длины, который будет обязательно проходить через одну вершину каждого ряда.

Сравнивая пути $L_{\min}$ для сети G_1 и сети G_2 , выбираем наименьший из них

$$L = \min \left\{ \begin{matrix} L_1 \\ L_2 \end{matrix} \right\}.$$

Найденный путь дает нам искомое множество $R = e_{1j}, \dots, e_{Nj}$, причем это решение строго оптимально.

Второй вариант. При использовании первого варианта алгоритма решения задач для построения допустимых вариантов решения по сети G_1 практически происходит перебор всех вариантов. Поэтому целесообразно его использовать при решении сети G , имеющей незначительное количество вершин v , в которые входят связи «и-и».

В случае, если количество таких вершин в сети G велико, то возможно использование такого алгоритма решения задач, который не требует расчленения общей сети G .

1. Используя III этап вычислений 1-го варианта алгоритма, находим в сети G путь $L(v_{1j}, v_{2j}, \dots, v_{Nj})$ минимальной продолжительности.

2. Просматриваем все вершины пути L и отыскиваем среди них такие, в которые входят связи «и-и», т. е. дуги $u_{i(i+\theta)}$, где $\theta > 1$. Каждой вершине $v_{i+\theta, f}$ приписываем множество Q таких вершин v , из которых к ней выходят связи «и-и». Связи «и-и» по отношению к $v_{i+\theta, f}$ могут находиться в связи «или-или», т. е. вершины, из которых они выходят, принадлежат одному ряду вершины сети. В этом случае пути L должна принадлежать только одна из таких вершин. Поэтому в дальнейшем, говоря, что все множество вершин Q , соответствующее вершине $v_{(i+\theta)}$, должно принадлежать пути L , мы будем иметь в виду, что из всех вершин, из которых выходят дуги, находящиеся в отношении «или-или», пути L должна принадлежать только одна из них.

Если путь L не содержит таких вершин, то решение найдено и является допустимым. Если есть хотя бы одна такая вершина, то переходим к следующему шагу.

3. Рассматриваем вершины пути L и проверяем, содержится ли в нем множество Q вершин, относящихся к вершине $v_{i+\theta, f}$. Такую проверку про-

изводим для всех вершин, в которые входят связи «и-и» и через которые проходит путь L .

Если множество вершин принадлежит пути L , то решение найдено и является допустимым.

Если хотя бы одна $v \in Q$ не принадлежит L , то решение недопустимо. В этом случае отыскиваем новый путь L с наименьшей продолжительностью и снова проверяем найденное решение на допустимость.

Вычисление продолжаем до тех пор, пока не будет найдено допустимое решение.

РЕШЕНИЕ ЗАДАЧ НА ОСНОВЕ ПРИВЕДЕННОГО АЛГОРИТМА

Покажем процесс получения решения для сети на рис. 3.

Используем первый вариант алгоритма. Становимся в начальную вершину v_0 сети и из нее переходим в вершину первого ряда сети.

Просматриваем дуги, выходящие из вершины $v_{01,1}$. Имеем дугу $u_{01,04,1}$.

В вершину $v_{04,1}$ входят дуги $u_{02,04,1}$, $u_{03,04,1}$ и $u_{03,04,1}$. Из них дуги $u_{01,04,1}$, $u_{02,04,1}$ характеризуют связи «и-и».

Пути сети, допускающие решение

$$l_1(v_{01,1}; v_{02,1}; v_{03,1}; v_{04,1}); \quad l_2(v_{01,1}; v_{02,1}; v_{03,2}; v_{04,1}).$$

Просматриваем дуги, выходящие из вершины $v_{01,2}$.

К классу дуг при условии «и-и» относится дуга $u_{01,03,2}$. Других дуг при условии «и-и», входящих в вершину $v_{03,3}$, нет. Тогда будем иметь: $l_3(v_{01,2}; v_{02,2}; v_{03,3}); l_4(v_{01,2}; v_{02,3}; v_{03,3})$.

Просматриваем дуги, выходящие из вершины $v_{01,3}$. Дуга $u_{01,03,3}$ соответствует связи «и-и». Из остальных дуг, входящих в вершину $v_{03,4}$ при связи «и-и», будет дуга $u_{01,03,4}$. Дуги $u_{01,03,4}$, $u_{01,03,4}$ объединены вершиной $v_{03,4}$ и по отношению к ней рассматриваем их как связи «или-или».

Тогда будем иметь

$$l_5(v_{01,3}; v_{02,4}; v_{03,4}); \quad l_7(v_{01,4}; v_{02,4}; v_{03,4}); \\ l_6(v_{01,3}; v_{02,5}; v_{03,4}); \quad l_8(v_{01,4}; v_{02,5}; v_{03,4}).$$

Переходим в вершины ряда 02 сети.

При просмотре их имеем еще не выделенную дугу $u_{02,04}$ при условии «и-и». Тогда будем иметь $l_9(v_{02,4}; v_{03,4}; v_{04,4})$. Определим полные пути в общей сети, включающие в себя все 9 частных путей, допускающих решение.

$$\text{Для } l_1 \text{ будем иметь } L_1(v_0; v_{01,1}; v_{02,1}; v_{03,1}; v_{04,1}; v_k), \\ \text{для } l_2 - L_2(v_0; v_{01,1}; v_{02,1}; v_{03,2}; v_{04,1}; v_k), \\ \text{для } l_3 - L_3(v_0; v_{01,2}; v_{02,2}; v_{03,3}; v_{04,3}; v_k), \\ \text{для } l_4 - L_4(v_0; v_{01,2}; v_{02,3}; v_{03,3}; v_{04,3}; v_k), \\ \text{для } l_5 - L_5(v_0; v_{01,3}; v_{02,4}; v_{03,4}; v_{04,3}; v_k), \\ L_6(v_0; v_{01,3}; v_{02,4}; v_{03,4}; v_{04,4}; v_k).$$

Вершина $v_{04,4} \in L_6$, но $v_{04,4} \in l_9$, в то же время $l_9 \in L_6$. Отсюда L_6 соответствует допустимому варианту решения задач.

$$\text{Для } l_6 \text{ будем иметь } L_7(v_0; v_{01,3}; v_{02,5}; v_{03,4}; v_{04,3}; v_k), \\ \text{для } l_7 - L_8(v_0; v_{01,4}; v_{02,4}; v_{03,4}; v_{04,3}; v_k), \\ L_9(v_0; v_{01,4}; v_{02,4}; v_{03,4}; v_{04,4}; v_k).$$

Путь $l_9 \in L_9$.

$$\text{Для } l_8 \text{ будем иметь } L_{10}(v_0; v_{01,4}; v_{02,5}; v_{03,4}; v_{04,3}; v_k).$$

Для частного пути l_9 построены полные пути L_6 и L_9 .

Других путей L , которые соответствовали бы допустимому варианту решения, нет.

Совокупность всех путей L назовем частью общей сети G_1 .

Вторую часть сети G_2 получим путем исключения из общей сети G вершин, к которым подходят пути при условии «и-и» и всех дуг, подходящих к ним.

Тогда сеть G_2 будет иметь вид, показанный на рис. 4.

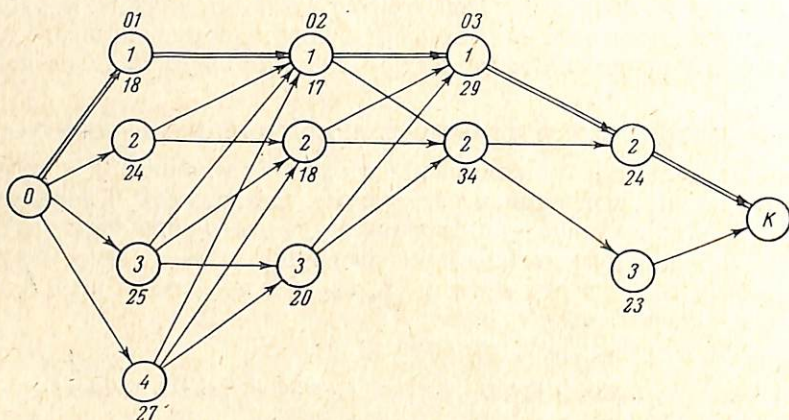


Рис. 4. Часть G_2 общей сети

В результате членения общей сети G на части G_1 и G_2 исключены все недопустимые варианты решения задач проблемы. Используя известный алгоритм определения длины минимального пути сети и состава вершин, через которые он проходит, для сети G_1 будем иметь

$$L_{\min}(v_0; v_{01, 2}; v_{02, 2}; v_{03, 3}; v_{04, 3}; v_k) = 84.$$

Для сети G_2

$$L_{\min}(v_0; v_{01, 1}; v_{02, 1}; v_{03, 1}; v_{04, 2}; v_k) = 88.$$

Сравнивая $L_{\min}$ для сети G_1 и G_2 , определяем, что искомый путь принадлежит сети G_1 ; тогда $R = (e_{01, 2}; e_{02, 2}; e_{03, 3}; e_{04, 3})$; $C = 84$.

Обратимся к решению задач вторым вариантом алгоритма для сети, изображенной на рис. 3. Используем при этом алгоритм нахождения в сети пути минимальной продолжительности.

1. Для начальной вершины $\lambda_0 = 0$.
2. Для вершины первого ряда сети $\lambda_{01, 1} = 18$; $\lambda_{01, 2} = 24$; $\lambda_{01, 3} = 25$; $\lambda_{01, 4} = 24$.
3. Для вершины второго ряда сети $\lambda_{02, j} = \min_j \lambda_{01, j} + S_{01j, 02j}$.

$$\begin{aligned} \lambda_{02, 1} &= 18 + 17 = 35, \text{ индекс } (01, 1); \\ \lambda_{02, 2} &= 18 + 18 = 36, \text{ » } (01, 1); \\ \lambda_{02, 3} &= 18 + 20 = 38, \text{ » } (01, 1); \\ \lambda_{02, 4} &= 25 + 21 = 46, \text{ » } (01, 3); \\ \lambda_{02, 5} &= 18 + 28 = 46, \text{ » } (01, 1). \end{aligned}$$

4. Для вершины третьего ряда

$$\begin{aligned} \lambda_{03, 1} &= 35 + 26 = 61, \text{ индекс } (02, 1); \\ \lambda_{03, 2} &= 35 + 34 = 69, \text{ » } (02, 1); \\ \lambda_{03, 3} &= 36 + 19 = 55, \text{ » } (02, 2); \\ \lambda_{03, 4} &= 46 + 18 = 64, \text{ » } (02, 4). \end{aligned}$$

5. Для вершины четвертого ряда

$$\begin{aligned}\lambda_{04,1} &= 64 + 26 = 90, \text{ индекс } (03,1); \\ \lambda_{04,2} &= 64 + 24 = 88, \quad \text{»} \quad (03,1); \\ \lambda_{04,3} &= 55 + 23 = 78, \quad \text{»} \quad (03,3); \\ \lambda_{04,4} &= 55 + 28 = 83, \quad \text{»} \quad (03,3).\end{aligned}$$

Количество вершин последнего ряда сети определяет количество полных путей сети, каждый из которых является минимальным для соответствующей вершины.

Минимальный путь сети $L_{\min}$ будет $L_{\min} = \min_j (\lambda_{04,j}) = 78$, индекс (04, 3).

Просматривая вершины сети от $v_{04,3}$ до v_0 , определяем состав

$$L_{\min}(v_{04,3}; v_{03,3}; v_{02,2}; v_{01,1}) = 78.$$

Среди вершин пути $L_{\min}$ имеем вершину $v_{03,3}$, в которую входит связь $W_{01,03}$ класса «и-и». Решение будет допустимо, только если соответствующий путь $L_{\min}$ включает вершину $v_{01,2}$. Найденный путь $L_{\min}$ не содержит вершины $v_{01,2}$. Следовательно, решение недопустимо. Для вершины $v_{04,3}$ находим новое число λ' , которое будет равно $\lambda_{04,3} = 84$. Следующий минимальный путь сети будет $L_{\min} = \min_j (\lambda_{04,j}) = 83$.

Состав пути $L_{\min}(v_{04,4}; v_{03,3}; v_{02,2}; v_{01,1}) = 83$. Вершина $v_{03,3}$ имеет связь класса «и-и» $W_{01,03}$ и вершина $v_{04,4}$ — связь $W_{02,04}$.

Путь $L_{\min}$ не содержит вершин $v_{01,2}$ и $v_{02,4}$. Находим новое число λ' для вершины $v_{04,4}$. $\lambda'_{04,4} = 89$. Новый минимальный путь сети будет $L'_{\min} = \min_j |\lambda_{04,j}| = 84$. Состав пути $L'_{\min}(v_{04,3}; v_{03,3}; v_{02,2}; v_{01,2}) = 84$.

Связь «и-и» $W_{01,03}$ вершин $v_{01,2}$ и $v_{03,3}$ принадлежит пути $L'_{\min}$. Решение допустимо и оптимально.

Тогда $R = (e_{01,1}; e_{02,2}; e_{03,3}; e_{04,3})$; $C = 84$.

При решении данного примера целесообразно использование второго варианта алгоритма, так как в этом случае мы строим частные пути сети, а повторное нахождение чисел λ делаем только для одной из вершин последнего ряда сети. Пути $L'_{\min}$ или $L''_{\min}$ находятся определением вершины конечного ряда, для которой число λ является наименьшим из всех.

В каждом конкретном случае необходимо определить целесообразность применения того или иного варианта алгоритма.

Поступила в редакцию
13 IX 1967