Задачи сетевого планирования

Методы анализа задач управления проектами, объединенные под общим названием *сетевое планирование и управление*, (СПУ), являются важнейшим инструментом в арсенале современного менеджера. Впервые эти методы стали широко применяться в середине 60-х годов прошлого века в США в системе ПЕРТ (PERT, Program Evaluation Research Task), разработанной специально для управления научно-исследовательскими и конструкторскими проектами при создании ракеты «Поларис». Позже они получили широкое применение при разработке ракет «Аполлон» и в целом программы высадки людей на Луну. И в СССР внедрению этих методов в практику планирования и управления уделялось большое значение. В настоящее время во многих странах государственные органы рассматривают только те проекты, анализ которых проведен, в том числе, и методами СПУ.

Математической основой этих методов является теория графов. Некоторым математическим задачам, которые возникают на графах в рамках систем СПУ, и будет посвящен данный раздел.

Рассмотрим один пример составления сетевого графика при строительстве загородного дома. Чтобы построить дом, необходимо выполнить некоторый набор работ, примерный список которых приведен ниже:

a) разработка проекта;

b) получение разрешения на строительство;

c) подписание контракта;

d) подвоз материалов к стройке;

e) подвод электричества и воды к стройке;

f) строительство фундамента;

g) возведение несущих стен;

h) плотницкие работы;

i) крыша;

j) грунтовка;

k) возведение стен, разделяющих комнаты;

l)прокладка коммуникаций для водопровода, центрального отопления, газа;

m) внутренняя электропроводка;

n) установка рам и окон;

o) обработка стен и перегородок;

p) установка дымовых труб;

q) плиточная облицовка, санитарное и кухонное оборудование;

r) штукатурка;

s) установка электрических выключателей и розеток;

t) установка дверей;

u) покраска стен и оклейка обоев;

v) подвоз земли для сада;

w) подвод коммуникаций для различных видов связи;

x) уборка строительной площадки;

y) планировка (выравнивание) участка;

z) предварительный прием.

Некоторые работы (операции) из этого списка нельзя выполнять, не завершив другие. Например, возвести крышу нельзя, не построив стены. Другие же операции можно выполнять параллельно, например, m), n), o). Вообще, правильное расписание технологической последовательности выполнения тех или иных работ, является делом ответственным и требует специальных знаний. Ниже приведен приблизительный сетевой график выполнения операций из нашего примера (рис. 4.26).

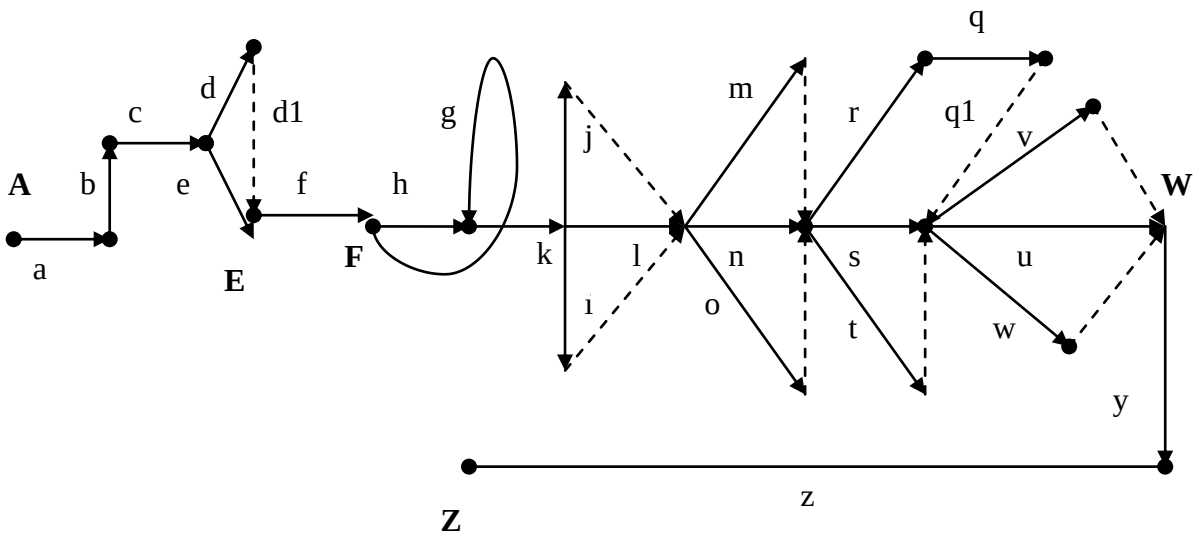

Рис. 4.26.

На этом рисунке дуги обозначают выполняемые работы, а вершины – это события. Например, вершины A, Z обозначают два события: начало всего комплекса работ и его завершение соответственно. Вершина F - это событие, состоящее в том, что завершена операция f, выполнение которой нельзя было начать, пока не выполнены все работы, входящие в вершину E. Эти работы изображены дугами e и $d1$. При этом $d1$ фиктивная операция (все фиктивные операции изображены пунктирными стрелками). Она понадобилась для того, чтобы графически зафиксировать тот факт, что операция f не может быть начата до того момента, пока не завершены операции e и d. Конечно, можно было бы обойтись без этой фиктивной работы, проведя стрелку d прямо в вершину E так, как это сделано с операцией g, но иногда введение таких работ более удобно. Вершина W обозначает событие, заключающееся в том, что завершены работы u,v, и w. А последние нельзя начать, если не выполнены операции q,s и t. Как правило, длительность всех фиктивных работ нулевая и для них не требуются никакие

ресурсы. Предположим, что длительность всех работ, изображенных горизонтальными дугами, равна 2, а всех остальных работ 1. Тогда время выполнения всего комплекса операций равно длине самого длинного пути из вершины A в Z. Один из таких путей, записанный в виде последовательности дуг, есть (a,b,c,e,f,h,k,l,n,r,q,q1,u,y,z) длины T=2+1+2+1+2+2+2+2+2+1+2+0+2+1+2=24. То есть загородный дом можно «сдать под ключ» за 24 условных дня.

В дальнейшем сетевой график выполнения комплекса операций (проекта) будем представлять в виде ориентированного графа без контуров. При этом дуги графа – это элементарные операции (работы), а вершины – события. Эти события будут иметь следующее толкование: завершены все операции, изображенные дугами, входящими в данную вершину. Логика выполнения всего комплекса работ такова. Никакая операция (дуга), выходящая из вершины, не может начаться, если не выполнены все операции (дуги), входящие в эту вершину. В таком графе присутствуют две специальных вершины. Из первой дуги только выходят, а во вторую дуги только входят. Первая вершина обозначает начало выполнения проекта, а вторая его завершение.

Пусть все вершины графа занумерованы числами от 1 до n, причем начальная вершина имеет номер 1, а конечная n. Пусть, как и ранее, дуга (i, j) ведет из вершины с номером i в вершину j. Если $t_{ij} \geq 0$ время выполнения работы (i, j), то очевидно, что время выполнения всего проекта равна длине максимального пути, ведущего из вершины 1 в n. При этом под длиной пути подразумевается сумма длин t_{ij} всех входящих в него дуг. Действительно, конечное событие наступит только тогда, когда будут выполнены все операции, принадлежащие любому пути, ведущему из начальной вершины в конечную, следовательно, и все операции, принадлежащие самому длинному из них. Последние пути называются *критическими.* Смысл этого названия заключается в том, что, если хотя бы

одна из работ (i, j) этого пути будет выполнена не за плановое время t_{ij}, а за большее время $t_{ij} + \Delta$, то и время выполнения всего проекта увеличится на Δ. Наименьшее время выполнения всего комплекса операций называется *критическим временем* и, как было сказано, оно равно длине максимального пути.

Определение 4.2. *Ранним сроком наступления события* i *называется число* $t_p(i)$, *равное длине максимального пути, ведущего из начальной вершины 1 в вершину* i.

Из этого определения следует, что раньше момента $t_p(i)$ событие i наступить не может. В частности, ранний срок $t_p(n)$ есть критическое время выполнения проекта.

Обозначим $U_i^{(-)}$ множество вершин, из которых выходят дуги, оканчивающиеся в вершине i, а $U_i^{(+)}$- множество вершин, в которые ведут дуги из вершины i.

Теорема 4.13. *Справедливо равенство*

$$t_p(j) = \max_{(ij) \in E_j^{(+)}} (t_p(i) + t_{ij}). \qquad (3.1)$$

▼ Событие j не может наступить раньше, чем произойдет любое из событий $i \in U_i^{(-)}$ плюс время, необходимое на выполнение операции (i, j). То есть $t_p(j) \geq t_p(i) + t_{ij}$ для всех $(i, j) \in E_j^{(+)}$. Причем хотя бы для одной работы $(i, j) \in E_j^{(+)}$ выполняется равенство. ▲

Ниже будет приведен модифицированный метод Беллмана – Калаба построения максимальных путей. Его описание выглядит особенно просто для правильно занумерованных графов.

Определение 4.3. *Орграф без контуров называется правильно занумерованным, если его вершины занумерованы так, что все дуги ведут из вершин с меньшими номерами в вершины с большими.*

Правильную нумерацию графа можно осуществить с помощью разбиения графа без контуров на *слои.*

Определение 4.4. *Будем говорить, что множество вершин орграфа без контуров разбито на слои* $V_1,V_2,...,V_r$, *если каждая вершина* u *принадлежит ровно одному из слоев* V_k, *и все дуги, ведущие из* u, *входят в вершины из слоев с номерами большими* k.

Если такое разбиение возможно, то вершины из слоя V_1 можно занумеровать в произвольном порядке от 1 и далее, затем перейти к слою V_2, продолжив нумерацию, и так вплоть до последнего слоя V_r.

Способ построения слоев состоит в следующем. В графе без контуров найдем висячие вершины, то есть такие вершины, из которых не выходит ни одна дуга. В конечном графе такие вершины всегда существуют иначе в нем можно построить контур. Все такие вершины отнесем к последнему слою V_r. Затем удалим их вместе с входящими в них дугами и повторим эту процедуру для оставшегося подграфа. Вновь найденные висячие вершины отнесем к слою V_{r-1}. Очевидно, что для любой вершины u из V_{r-1} дуги ведут лишь в вершины слоя V_r. Продолжая эту процедуру, мы разобьем множество вершин графа на слои с указанными выше свойствами. На рис. 4.27, 4.28 приведены два изоморфных графа, вершины одного из которых разбиты на слои.

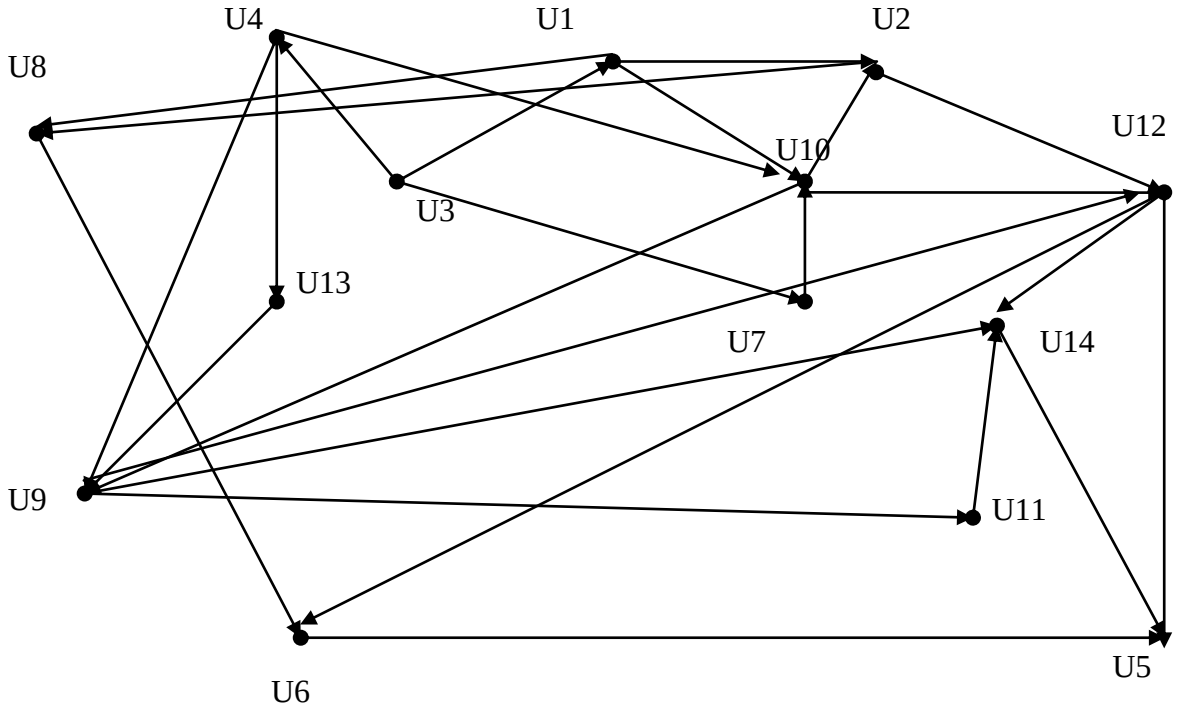

Рис. 4.27.

Изучая граф на рис. 4.29, мы видим, что единственной висячей вершиной в нем является u_5. Отнесем ее к последнему слою и удалим из графа вместе с инцидентными ей дугами. В новом графе висячими окажутся вершины u_6 и u_{14}. Отнесем их к предпоследнему слою и удалим вместе с входящими в них дугами. В оставшемся графе висячими станут вершины u_8, u_{11}, u_{12}. Это будет новый слой, третий от конца. Продолжая эту процедуру, мы разобьем граф на семь слоев (см. рис.4.30). Слои обозначены пунктирными овалами. Занумеруем вершины графа в соответствии с таблицей 4.1.

Таблица 4.1.

U3	U4	U7	U1	U13	U10	U9	U2	U11	U12	U8	U14	U6	U5
1	2	3	4	5	6	7	8	9	10	11	12	13	14

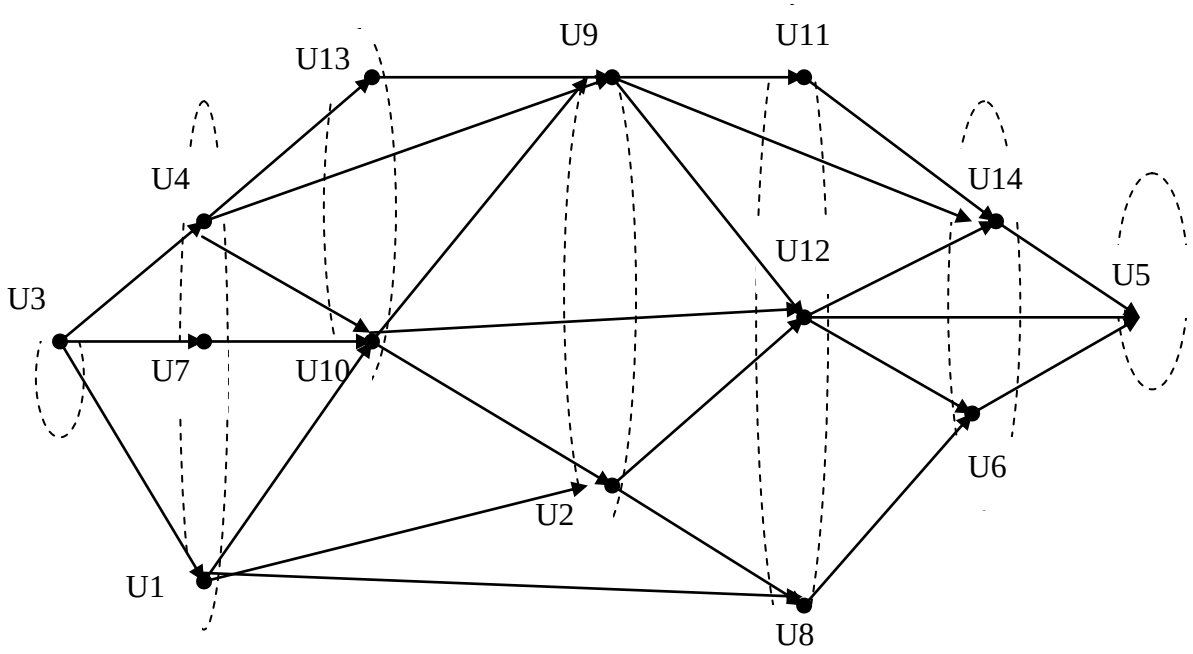

Рис. 4.28.

Теперь наш граф правильно занумерован (см. рис.4.31), в котором жирно обозначены номера вершин).

Графический метод разбиения графа без контуров на слои можно изложить на языке матриц смежности вершин. В таблице 4.2. приведена матрица смежности вершин графа, изображенного на рис. 4.27. В ней, по умолчанию, пустые клетки заполнены нулями.

Найдем нулевые строки. В нашем примере это пятая строка. Это значит, что вершина U5 является висячей. Отнесем ее к последнему слою. В матрице 4.2. вычеркнем пятые строку и столбец. Геометрически это равносильно тому, что в исходном графе удалена вершина U5 вместе с инцидентными ей дугами.

В новой матрице (см. таб. 4.3.) произведем аналогичную операцию. В ней нулевыми являются строки U6 и U14. Отнесем соответствующие вершины к предпоследнему слою, и вычеркнем строки и столбцы с номерами U6 и U14. Такое вычеркивание означает удаление из графа соответствующих вершин и инцидентных им дуг.

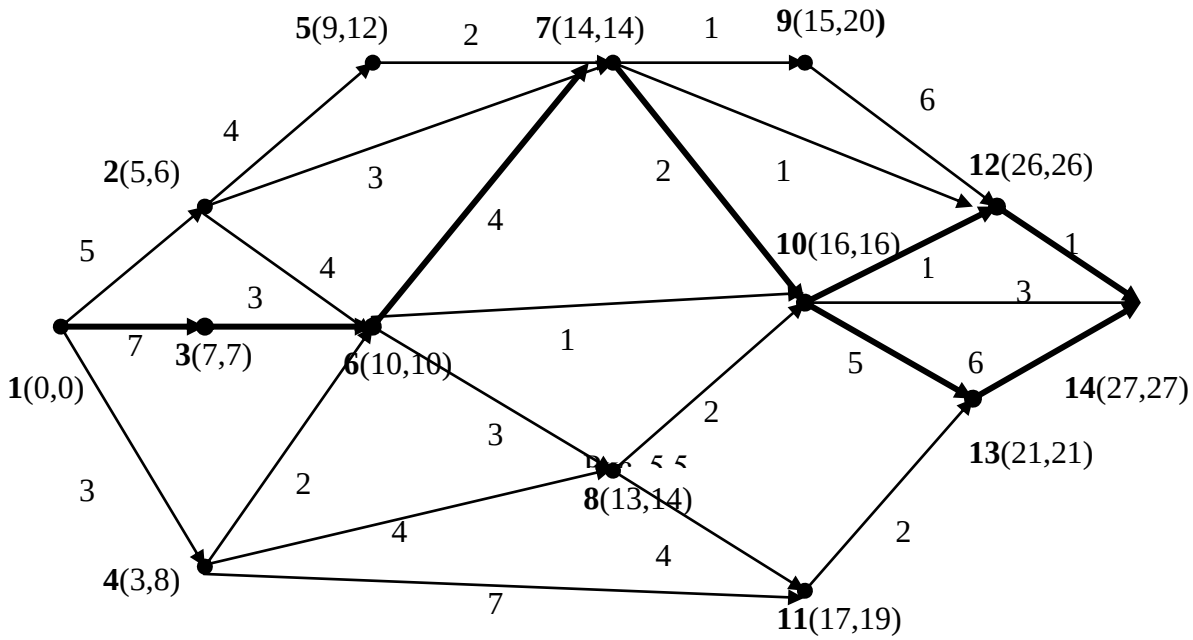

Рис.4.29.

Теперь нулевыми окажутся строки U8, U11, U12 (см. таблицу 4.4.). Соответствующие им вершины принадлежат третьему от конца слою. Предлагаем читателю самостоятельно довести процедуру разбиения графа на слои до конца и убедиться в справедливости представления графа в виде рис. 4.28.

Таблица 4.2. (Матрица смежности вершин графа на рис. 4.27.)

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14
U1		1						1		1				
U2								1				1		
U3	1			1			1							
U4									1	1			1	

U5													
U6					1								
U7										1			
U8						1							
U9											1	1		1
U10		1							1			1		
U11														1
U12					1	1								1
U13									1					
U14					1									

Таблица 4.3.

	U1	U2	U3	U4	U6	U7	U8	U9	U10	U11	U12	U13	U14
U1		1					1		1				
U2							1				1		
U3	1			1		1							
U4								1	1			1	
U6													
U7									1				
U8					1								
U9										1	1		1
U10		1						1					
U11													1
U12					1								1
U13								1					
U14													

Таблица 4.4.

	U1	U2	U3	U4	U7	U8	U9	U10	U11	U12	U13
U1		1				1		1			
U2						1				1	
U3	1			1	1						
U4							1	1			1
U7								1			
U8											
U9									1	1	
U10		1					1				
U11											
U12											
U13							1				

Пусть вершины графа правильно занумерованы. В основе приводимых ниже двух алгоритмов лежит метод динамического программирования, примененный к данной конкретной задаче.

Модифицированный алгоритм Беллмана-Калаба построения максимальных путей в орграфе.

1. Вычисление ранних сроков наступления событий t_p.

1 шаг.

$t_p(1) := 0;$

Общий шаг $j = \overline{2,n}$.

$t_p(j) := \max\limits_{(ij)\in E_j^{(+)}} (t_p(i) + t_{ij})$. Отметим, что величины $t_p(i)$, входящие в правую часть оператора присвоения, определены на предыдущих шагах, так

как из правильной нумерации сети следует, что $i < j$. Если указанный максимум достигается на входящих в j-ю вершину дугах $(i_1, j) \in E_j^{(+)}$, $(i_2, j) \in E_j^{(+)}$,…, то все эти дуги вносятся в список E_j^*.

2. Построение критических путей.

Из списка E_n^* выбираем любую дугу (i_1, n). Затем из списка $E_{i_1}^*$ выбираем дугу (i_2, i_1), из $E_{i_2}^*$ выбираем дугу (i_3, i_2), из $E_{i_3}^*$ выбираем дугу (i_4, i_3) и т.д. Этот процесс заканчивается после k шагов выбором дуги (i_k, i_{k-1}), причем $i_k = 1$. Построенный путь $(1 = i_k, i_{k-1}, ..., i_1, n)$ и будет критическим.

Обоснование 1 части работы алгоритма следует из теоремы 4.13 и правильной нумерации графа. Далее из этой же теоремы следует, что для любой вершины j графа длина максимального пути в нее из первой вершины равна $t_p(j)$, а дуга (i^*, j), на которой достигается максимум в (3.1), принадлежит этому пути. Поэтому дуга (i_1, n) принадлежит максимальному (критическому) пути из первой вершины в n. Аналогичные рассуждения справедливы и для дуг (i_2, i_1), (i_3, i_2),…, (i_k, i_{k-1}).

Проиллюстрируем работу алгоритма на примере сети, приведенной на рис. 4.29. На дугах (i, j) графа 4.29. проставлены числа t_{ij}, равные продолжительности выполнения соответствующих операций.

1. Вычисление ранних сроков наступления событий t_p.

1 шаг. $t_p(1) := 0$;

2 шаг. $t_p(2) := t_{12} = 5$; $E_2^* := \{(1,2)\}$.

3 шаг. $t_p(3) := t_{13} = 7$; $E_3^* := \{(1,3)\}$.

4 шаг. $t_p(4) := t_{14} = 3$; $E_4^* := \{(1,4)\}$.

5 шаг. $t_p(5) := t_p(2) + t_{25} = 5 + 4 = 9$; $E_5^* := \{(2,5)\}$.

6 шаг.

$$t_p(6) := \max\{t_p(2) + t_{26} = 5 + 4 = 9; t_p(3) + t_{36} = 7 + 3 = 10;$$
$$t_p(4) + t_{46} = 3 + 2 + 5\} = 10;\ E_6^* := \{(3,6)\}.$$

7 шаг. $t_p(7) := \max\{t_p(2) + t_{27} = 5 + 3 = 8; t_p(5) + t_{57} = 9 + 2 = 11;$

$$t_p(6) + t_{67} = 10 + 4 = 14\} = 14.\ E_7^* := \{(6,7)\}.$$

8 шаг.

$$t_p(8) := \max\{t_p(4) + t_{48} = 3 + 4 = 7; t_p(6) + t_{68} = 10 + 3 = 13\} = 13$$
$$E_8^* := \{(6,8)\}.$$

9 шаг.

$$t_p(9) := t_p(7) + t_{79} = 14 + 1 = 15;\ E_9^* := \{(7,9)\}.$$

10 шаг.

$$t_p(10) := \max\{t_p(7) + t_{7,10} = 14 + 2 = 16; t_p(6) + t_{6,10} = 10 + 1 = 11;$$
$$t_p(8) + t_{8,10} = 13 + 2 = 15\} = 16;\ E_{10}^* := \{(7,10)\}.$$

11 шаг.

$$t_p(11) := \max\{t_p(4) + t_{4,11} = 3 + 7 = 10; t_p(8) + \ + t_{8,11} = 13 + 4 = 17\} = 13\ ;$$

$$E_{11}^* := \{(8,11)\}.$$

12 шаг.

$$t_p(12) := \max\{t_p(7) + t_{7,12} = 14 + 1 = 15; t_p(9) +$$

$$+ t_{9,12} = 15 + 6 = 21; t_p(10) + t_{10,12} = 16 + 10 = 26\} = 26\ ; E_{12}^* := \{(10,12)\}\ .$$

13 шаг.

$$t_p(13) := \max\{t_p(10) + t_{10,13} = 16 + 5 = 21; t_p(11) +$$

$$+ t_{11,13} = 17 + 2 = 19\} = 21;\ E_{13}^* := \{(10,13)\}.$$

14 шаг.

$$t_p(14) := \max\{t_p(10) + t_{10,14} = 16 + 3 = 19; t_p(12) +$$

$$+ t_{12,14} = 26 + 1 = 27; t_p(13) + t_{13,14} = 21 + 6 = 27\} = 27\ ;$$

$$E_{14}^* := \{(12,14),(13,14)\}.$$

Критическое время выполнения комплекса операций $T_c = t_p(14) = 27$.

2. Найдем критические пути.

Множество $E_{14}^* := \{(12,14),(13,14)\}$ состоит из двух операций. Выберем любую из них, например, $(12,14)$. Она будет входить в первый критический путь. $P_{c1} = \{(12,14)\}$.

Список $E_{12}^* := \{(10,12)\}$ содержит одну дугу. Она также войдет в искомый критический путь. $P_{c1} = \{(10,12);(12,14)\}$.

$E_{10}^* := \{(7,10)\}$. Работа $(7,10)$ является критической. $P_{c1} = \{(7,10);(10,12);(12,14)\}$.

$E_7^* := \{(6,7)\}$; $P_{c1} = \{(6,7);(7,10);(10,12);(12,14)\}$.

$E_6^* := \{(3,6)\}$;

$P_{c1} = \{(3,6);(6,7);(7,10);(10,12);(12,14)\}$.

$E_3^* := \{(1,3)\}$;

$P_{c1} = \{(1,3);(3,6);(6,7);(7,10);(10,12);(12,14)\}$.

Нами построен критический путь. В него вошли перечисленные выше операции или, что равносильно, вершины $(1,3,6,7,10,12,14)$.

Еще один критический путь можно построить, если на первом шаге из списка $E_{14}^* := \{(12,14),(13,14)\}$ выбрать операцию $(13,14)$. Читатель сам может убедиться в том, что второй критический путь P_{c2} пройдет через вершины $(1,3,6,7,10,13,14)$. Оба упомянутых пути выделены на рис. 4.29 жирными стрелками. В скобках около номера вершин указаны два числа. Первое из них – ранний срок наступления события, второе – поздний срок наступления этого события.

Определение 4.5. *Поздний срок* $t_n(i)$ *наступления события - это такой срок, для которого любая задержка наступления события* i *на время* Δ *после этого срока увеличивает все время выполнения комплекса операций на величину* Δ.

Рассмотрим сеть на рис. 4.29. Очевидно $t_n(9)=20$. Действительно, если девятое событие наступит не на двадцатый, а на двадцать первый день, то двенадцатое событие наступит после выполнения работы (9,12) на 27 день, а с ним и конечное четырнадцатое событие произойдет лишь на 28 день. То есть время выполнения проекта T_c увеличится на 1. Очевидно, для всех *критических* событий, лежащих на критическом пути, ранние и поздние сроки совпадают, т.е. наступление этих событий нельзя задержать, не увеличив время выполнения всего проекта. Например, если седьмое событие наступит не на 14, а на 17 день, то десятое событие произойдет лишь на 19 день, тринадцатое – на 24 , а четырнадцатое только на 30 день, и выполнение всего комплекса задержится на 3 дня. Для некритических событий существуют временные резервы, т.е. эти события могут происходить в любой момент из отрезка $[t_p, t_n]$ без изменения времени выполнения проекта.

Пусть L_i длина максимального пути, ведущего из вершины i в конечную вершину n. Поскольку проект нельзя завершить, не выполнив все работы, лежащие на этом пути, то справедливо равенство $t_n(i)+L_i=T_c$, или

$$L_i = T_c - t_n(i). \qquad (3.2)$$

Теорема 4.14. *Для любого события* $i<n$ *справедливо равенство*

$$t_n(i)=\min_{(ij)\in E_i^{(-)}}(t_n(j)-t_{ij}). \qquad (3.3)$$

▼ Рассмотрим вершину i и любую дугу (i,j), выходящую из этой вершины (рис. 4.32). Из максимальности L_i пути, ведущего из i в n, следует, что $L_i \ge t_{ij}+L_j$. Подставим сюда выражения для L из (3.2.) и получаем неравенство $t_n(i)\le t_n(j)-t_{ij}$,

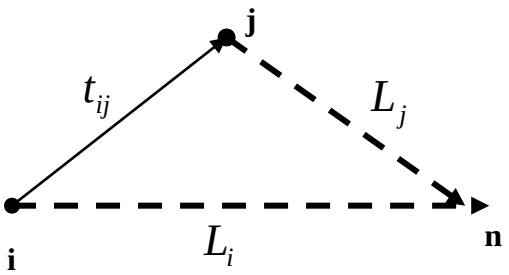

Рис.4.32.

которое справедливо для любой дуги $(i,j)\in E_i^{(-)}$, причем хотя бы для одной дуги оно выполняется как равенство. ▲

Модифицированный алгоритм Беллмана-Калаба вычисления поздних сроков наступления событий

1 шаг.

$t_n(n) := T_c$.

Общий шаг $k = \overline{2,n}$.

Положим $i = n - k + 1$, $t_n(i) := \min\limits_{(ij) \in E_i^{(-)}} (t_n(j) - t_{ij})$.

Алгоритм заканчивает работу после n шагов.

Вновь рассмотрим пример сетевого проекта на рис. 4.29 и вычислим поздние сроки наступления событий.

1 шаг.

$t_n(14) := T_c = 26$.

2 шаг.

$i := 14 - 2 + 1 = 13$, $t_n(13) := t_n(14) - t_{13,14} = 27 - 6 = 21$.

3 шаг.

$i := 14 - 3 + 1 = 12$, $t_n(12) := t_n(14) - t_{12,14} = 27 - 1 = 26$.

4 шаг.

$i := 14 - 4 + 1 = 11$, $t_n(11) := t_n(13) - t_{11,13} = 21 - 2 = 19$.

5 шаг.

$i := 14 - 5 + 1 = 10$, $t_n(10) := \min\{t_n(14) - t_{10,14} = 27 - 3 = 24,$

$t_n(13) - t_{10,13} = 21 - 5 = 16, t_n(12) - t_{10,12} = 26 - 10 = 16\} = 16$.

6 шаг.

$i := 14 - 6 + 1 = 9$, $t_n(9) := t_n(12) - t_{9,12} = 26 - 6 = 20$.

7 шаг.

$i := 14 - 7 + 1 = 8$, $t_n(8) := \min\{t_n(11) - t_{8,11} = 19 - 4 = 15,$

$t_n(10) - t_{8,10} = 16 - 2 = 14\} = 14$.

8 шаг.

$i := 14 - 8 + 1 = 7$, $t_n(7) := \min\{t_n(12) - t_{7,12} = 26 - 1 = 25,$

$t_n(10)-t_{7,10}=16-2=14, t_n(9)-t_{7,9}=20-1=19\}=14.$

9 шаг.

$i:=14-9+1=6,\ t_n(6):=\min\{t_n(10)-t_{6,10}=16-1=15,$

$t_n(8)-t_{6,8}=14-3=11, t_n(7)-t_{6,7}=14-4=10\}=10.$

10 шаг.

$i:=14-10+1=5,\ t_n(5):=t_n(7)-t_{5,7}=14-2=12.$

11 шаг.

$i:=14-11+1=4,\ t_n(4):=\min\{t_n(11)-t_{4,11}=19-7=12,$

$t_n(8)-t_{48}=14-4=10, t_n(6)-t_{46}=10-2=8\}=8.$

12 шаг.

$i:=14-12+1=3,\ t_n(3):=t_n(6)-t_{3,6}=10-3=7.$

13 шаг.

$i:=14-13+1=2,\ t_n(2):=\min\{t_n(7)-t_{27}=14-3=11,$

$t_n(6)-t_{26}=10-4=6, t_n(5)-t_{25}=12-4=8\}=6.$

14 шаг.

$i:=14-14+1=1,\ t_n(1):=\min\{t_n(4)-t_{14}=8-3=5,$

$t_n(3)-t_{13}=7-7=0, t_n(2)-t_{12}=6-5=1\}=0.$

Заключительный 14 шаг можно было бы не совершать, так как начальное событие всегда является критическим, и его ранний и поздний сроки равны нулю.

Начало выполнения операций, лежащих на критическом пути (критические операции), нельзя задерживать или увеличивать их продолжительность без увеличения времени реализации проекта. Например, если операцию $(3,6)$ выполнять не 3 , а 5 дней, то ранние сроки наступления событий 6,7,10,12,13,14 сдвинутся на 2 единицы, а с ними и T_c. А вот для некритических операций существуют временные резервы, основные из которых следующие.

1.) Полный резерв операции (i, j) равен

$$P_n(i, j) = t_n(j) - t_p(i) - t_{ij};$$

2.) Свободный резерв

$$P_c(i, j) = t_p(j) - t_p(i) - t_{ij};$$

3.) Независимый резерв

$$P_H(i, j) = t_p(j) - t_n(i) - t_{ij}.$$

Так как $t_p \le t_n$, то $P_n(i, j) \ge P_c(i, j) \ge P_H(i, j)$. Убедитесь самостоятельно, что для критических операций все три резерва равны нулю.

Смысл введенных величин следующий. Если задержать начало операции (i, j) или увеличить ее продолжительность на величину полного резерва $P_n(i, j)$, то событие j станет критическим, а вместе с ним и все события и работы, принадлежащие максимальному пути из j в n. Действительно, так как работа начнется в $t_p(i) + P_n(i, j)$ день и будет длиться t_{ij} дней, то завершится она в момент $t_p(i, j) + P_n(i, j) + t_{ij}$, который равен $t_n(j)$. Например, полный резерв операции $(4,8)$ равен 14-3-4=7. Предположим, что эта работа началась, как и положено, на 3 день, но длилась не 4, а 11 дней. Тогда событие 8 наступит лишь на 14 день в свой предельно допустимый срок и станет критическим. Вместе с ним станет критической и операция $(8,10)$. Заметим, что хотя 11 событие наступит теперь лишь на 18 день, оно, тем не менее, не станет критическим. Правда, ее временной резерв уменьшится на 1.

Свободный резерв $P_c(i, j)$ указывает на то, что задержка с завершением работы (i, j) на эту величину не вызовет никаких последствий в графике выполнения проекта, так как j событие наступит в свой положенный по расписанию срок. Например, для той же операции $(4,8)$ ее свободный резерв равен 13-3-4=6. Он меньше полного резерва. Видно, что при задержке времени начала выполнения работы $(4,8)$ на 6 дней с момента

$t_p(4)=3$ она закончится на 13 день, т.е. при наступлении раннего срока $t_p(8)=13$ и не повлечет никаких сбоев в графике выполнения последующих операций.

Независимый резерв $P_H(i,j)$ рассчитан на нештатную ситуацию. Если он положителен, то это означает следующее. Если событие i наступит в свой предельный, поздний срок из-за задержек предшествующих работ, у операции (i,j) еще остается временной резерв, на который можно задержать время ее выполнения так, чтобы событие j наступило в свой плановый момент $t_p(j)$. Например, $P_H(4,8)=13-8-4=1$. Однако, в отличие от двух других резервов, величина $P_H(i,j)$ может быть и отрицательной. Например, $P_H(2,5)=9-6-4=-1$. Это сигнал о том, что выполнение проекта идет со сбоем в графике, и данную операцию нельзя задерживать, чтобы не усугубить ситуацию для последующих работ.

Анализ приведенных величин: ранние и поздние сроки наступления событий и резервы времени оказывается эффективным инструментом при диспетчеризации больших проектов.

Упражнения к 4.3.

Упражнение 4.3.1. Докажите, что в графе без контуров всегда существуют висячие вершины.

Упражнение 4.3.2. Доведите самостоятельно до конца алгоритм разбиения графа 4.27 на слои.

Упражнение 4.3.3. Докажите, что если в связном графе без контуров заменить все дуги (i,j) обратными дугами (j,i), то полученный граф будет связным и не будет содержать контуры.

Упражнение 4.3.4. Замените в сети 4.29 все дуги обратными, сохранив их длину, перенумеруйте вершины в обратном порядке и найдите в новой сети ранние и поздние сроки наступления событий, критические пути, а также все резервы операций.