ИЗВЕСТИЯ
ТОМСКОГО ОРДЕНА ОКТЯБРЬСКОЙ РЕВОЛЮЦИИ
И ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ
ПОЛИТЕХНИЧЕСКОГО ИНСТИТУТА имени С. М. КИРОВА

Том 203 1974

АЛГОРИТМ ПОСТРОЕНИЯ СЕТЕВОЙ МОДЕЛИ РАЗРАБОТКИ

Ю. Н. ЕФИМОВ

(Представлена научным семинаром УВЛ ТПИ)

Получившая широкое распространение автоматизированная система сетевого планирования и управления (СПУ) основана на анализе сетевых графиков, являющихся динамическими моделями сложных разработок. Выражение процесса разработки в виде сети предельно формализует оперативный план, что позволяет применить электронные вычислительные машины (ЭВМ) для выполнения процедур по обработке информации о ходе работ и подготовке данных для принятия решений. Однако если анализ сетевых графиков выполняется с помощью ЭВМ, то процесс их построения (сшивания работ) до сих пор остается ручным. Это приводит к тому, что, выражаясь в терминах СПУ, работы по составлению сетевого графика становятся критическими в общем комплексе работ по внедрению системы СПУ для какой-либо разработки. Очевидно, что автоматизация построения сетевых графиков сделает систему СПУ еще более оперативной.

В данной статье приводится алгоритм построения сетевых моделей разработок, описанных с помощью некоторой совокупности работ и связей между ними. Причем предполагается, что соблюдены следующие условия:

1) разработка содержит только одну конечную работу;

2) сетевая модель имеет только одно исходное событие i_0;

3) любая работа (i, j) может быть начата только после выполнения всех предшествующих ей работ, т. е. всякое событие модели реализует элементарную логическую функцию «И» (конъюнкция). Практически для любой одноцелевой разработки, допускающей применение сетевых методов планирования и управления, можно составить совокупность работ, удовлетворяющую вышеперечисленным условиям.

Как известно, исходные данные для построения первичного сетевого графика берутся из перечня работ моделируемой разработки, который составляется ответственными исполнителями работ. Упомянутые работы «сшиваются» событиями одна с другой на основе логической взаимосвязи и технологической последовательности выполнения работ. При этом необходимо учитывать основные правила построения сетей [1], которые сводятся к следующему.

1. Один и тот же номер события нельзя использовать дважды; этим достигается однозначность определения любой работы сети. (Код работы составляется из кодов ее начального и конечного событий).

2. Сеть должна быть простой, без лишних пересечений; направления работ нужно стремиться изображать слева направо.

3. В сети не должно быть ни одного события, кроме начального, в которое не входит ни одна работа; все события, кроме завершающего, должны иметь хотя бы одну последующую работу. Наличие обрывов первого или второго рода [2] указывает на допущенную логическую ошибку при построении сети.

4. В сетевом графике не должно быть замкнутых контуров (циклов), которые также указывают на логическую ошибку.

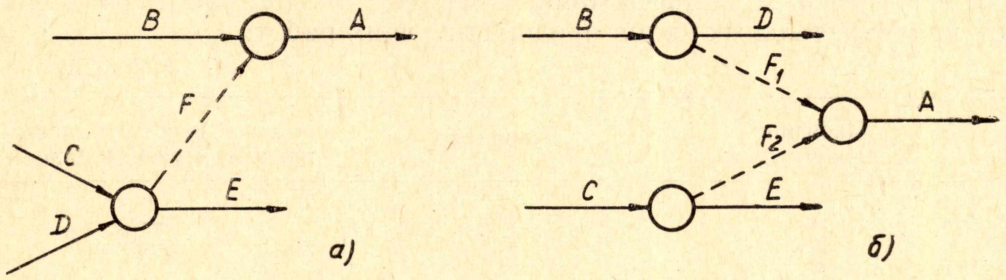

Рис. 1. Примеры введения фиктивных связей в случае дифференцированно-зависимых работ

Составленная сеть, помимо основных работ, указанных в перечне, может, очевидно, содержать некоторое количество так называемых фиктивных связей, необходимых для правильного изображения дифференцированно-зависимых работ. Например, если какая-то работа A может быть начата после выполнения работ B, C и D, а работа E — после выполнения только C и $Д$, то необходимо ввести фиктивную работу F (рис. 1). Если работу A можно начинать после окончания работ B, C и начало работы D зависит только от окончания B, а начало E — от окончания работы C, то в сетевой график необходимо ввести две фиктивные связи F_1 и F_2 (рис. $1^б$).

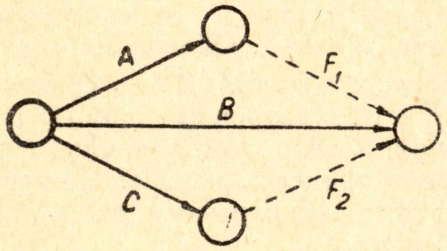

Рис. 2. Пример введения фиктивных связей для параллельных работ

Часто на практике встречаются случаи, когда одно событие служит началом для двух или более работ, которые заканчиваются также в одном событии. (Такие работы будемназывать параллельными). Чтобы показать, что параллельные работы протекают независимо, необходимо вводить дополнительные события и фиктивные работы (рис. 2).

Таким образом, основной задачей при составлении сетевого графика является нахождение необходимого числа событий и фиктивных работ. Причем это число должно быть минимальным в том смысле, что меньшим количеством событий и фиктивных работ невозможно описать моделируемую разработку. Найденные события должны быть пронумерованы в порядке возрастания значений порядковой функции $\varphi(i)$ [3], определенной на этих событиях, т. е. для любой работы (как основной, так и фиктивной) должно выполняться соотношение

$$i < j,$$

где i и j — соответственно номера начального и конечного событий работы (i, j).

Такая нумерация значительно упрощает вычерчивание полученной сети и ускоряет дальнейший процесс ее анализа. К примеру, при применении метода итераций упорядоченная кодировка событий позволяет производить временной анализ сети за два взаимно противоположных просмотра списка работ, в то время как в общем случае требуется $[\varphi(i)_{max}+1]$ число просмотров.

В качестве исходной информации для построения сети используем данные табл. 1, где во втором столбце находится перечень всех работ разработки, в третьем — параметры работ (длительность, стоимость и т. д.), а в четвертом — информация о связях между работами в виде строк предварительных кодов предшествующих работ.

Таблица 1

№ п.п.	Название работы	Параметр	Работы, непосредственно предшествующие данной
1	А	P_1	—
2	Б	P_2	1, 3
3	В	P_3	1
4	Г	P_4	2, 3, 5
. . .	. . .	. . .	. . . . .

За предварительный код работы удобно принять ее порядковый номер в исходном перечне (первый столбец). Для начальных работ разработки в четвертый столбец табл. 1 заносится нуль или прочерк. Очевидно, что исходная информация в такой форме очень легко может быть получена от ответственных исполнителей, знающих после выполнения каких работ они могут начать «свою» работу.

Пусть работы, непосредственно предшествующие какой-либо работе $к$ исходного перечня, составляют подмножество $\Gamma_к^{-1}$ (одна строка четвертого столбца табл. 1) с числом элементов $\omega_к^0$. Обозначим через $\omega_к'$ число еще не рассмотренных элементов подмножества $\Gamma_к^{-1}$. Очевидно, что перед работой алгоритма $\omega_к' = \omega_к^0$. Рассмотренные работы исходного перечня будем отмечать признаком и выписывать в одномерный массив $s[m]$, постоянно пополняя его в процессе решения новыми членами. Тогда работа алгоритма, решающего основные задачи построения сетевого графика, сводится к следующему.

1. Выполнить команды восстановления:

$$C_4 := 1; \quad s[m] := 0.$$

2. Началам всех исходных работ присвоить код, равный значению C_4. Исходные работы разработки отметить признаком и выписать в массив $s[m]$. Выполнить $C_4 := C_4 + 1$.

3. Взять первый элемент массива $s[m]$.

4. Просмотреть все неотмеченные признаком работы в исходной таблице и выполнить $\omega_к' := \omega_к' - 1$, если рассматриваемый элемент массива $s[m]$ является членом подмножества $\Gamma_к^{-1}$.

5. Выяснить, есть ли работы, получившие $\omega_к' = 0$ при выполнении шага 4. Да. Выполнить шаг 6. Нет. Выполнить шаг 19.

6. Работы, получившие $\omega_к' = 0$, расположить в порядке возрастания величины $\omega_к^0$. Работы с одинаковыми $\omega_к^0$ составляют группу работ. Очевидно, что список предшествующих работ для любой работы группы будет одинаковым. Этот список назовем подмножеством (Γ_q^{-1}), предшествующим группе (q).

7. Рассмотреть первую группу работ.

8. Выписать все работы $(i,j) \in \Gamma_q^{-1}$ и расположить в порядке возрастания кодов их начал.

9. Из полученного упорядоченного ряда исключить работы, конечным событиям которых коды (j) уже присвоены. Наибольшую величину из кодов концов исключенных работ принять за код начала очередной фиктивной работы. Оставшиеся в ряду работы разбить на подгруппы работ с одинаковыми кодами (i) их начал. Отметить первые работы всех подгрупп ряда.

10. Взять первую неотмеченную работу ряда.

11. Присвоить концу этой работы код, равный значению C_4. Этот же код взять в качестве начала очередной фиктивной работы.

12. Выполнить $C_4 := C_4 + 1$.

13. Проверить, все ли неотмеченные работы ряда рассмотрены. Да. Выполнить шаг 15. Нет. Выполнить шаг 14.

14. Взять следующую неотмеченную работу ряда и выполнить шаг 11.

15. Присвоить код, равный значению C_4:

а) началам всех работ группы;

б) концам всех работ $(i,j) \in \Gamma_q^{-1}$, которые еще не имеют кода конечного события;

в) концам фиктивных работ (эти работы получили код начала при выполнении шагов 9 и 11).

16. Отметить признаком все работы рассмотренной группы и выписать их в массив $s[m]$, расположив вслед за последним элементом массива.

17. Проверить, все ли группы работ рассмотрены. Да. Выполнить шаг 19. Нет. Выполнить шаг 18.

18. Рассмотреть следующую группу работ и выполнить шаг 8.

19. Проверить, все ли элементы массива $s[m]$ рассмотрены. Да. Выполнить шаг 21. Нет. Выполнить шаг 20.

20. Взять следующий элемент массива $s[m]$ и выполнить шаг 4.

21. Проверить, все ли работы исходного списка отмечены признаком. Да. Выполнить шаг 22. Нет. Выполнить шаг 25.

22. Принять в качестве кода конца завершающей работы значение $C_4 + 1$.

23. Проверить, все ли работы имеют коды концов. Да. Выполнить шаг 24. Нет. Выполнить шаг 26.

24. Упорядочить список реальных и фиктивных работ по коду их начал и окончить вычисления.

Полученный упорядоченный список и является решением поставленной задачи.

25. Прекратить вычисления, так как сетевая модель содержит контур. Необходимо внести соответствующие исправления в исходную информацию.

26. Прекратить вычисления и выписать работы, не имеющие кода конечного события. Результаты этих работ не нужны для выполнения других работ разработки. Следовательно, эти работы либо лишние, либо в исходной информации допущена ошибка.

В качестве примера работы алгоритма приведем построение сетевой модели разработки, исходные данные которой заданы левой частью табл. 2 (здесь названия и параметры работ отсутствуют, так как они не нужны для работы алгоритма). В правой части этой таблицы отражено изменение величины ω_κ' и приведены полученные коды начальных i и конечных j событий работ исходного списка. Внизу дан получен-

Таблица 2

№ п. п.	Предшествующие работы	ω°_k	Значение ω'_k при каждом просмотре исходного списка работ																					i	j
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
1	5, 19	2	2	1				0																3	11
2	1, 4, 16, 17, 10, 12, 14	7	7						6	5	4	3	2	1	0									11	14
3	2, 18, 6, 7, 20	5	5													4	3	2	1	0				14	16
4	5, 19	2	2	1				0																3	10
5		0	0																					1	3
6	1, 4, 16, 17, 10, 12, 14	7	7						6	5	4	3	2	1	0									11	12
7	4, 16, 17, 10, 12, 14	6	6							5	4	3	2	1	0									10	12
8	6, 7, 20	3	3													2	1		0					12	16
9	2, 18, 6, 7, 20	5	5													4	3	2	1	0				14	15
10	5, 19, 11, 13	4	4	3	2	1		0																5	10
11		0	0																					1	5
12	5, 19, 11, 13	4	4	3	2	1		0																5	9
13		0	0																					1	4
14	5, 19, 11, 13, 15 . . .	5	5	4	3	2	1	0																6	10
15		0	0																					1	6
16	5, 19	2	2	1				0																3	7
17	5, 19	2	2	1				0																3	8
18	1, 4, 16, 17, 10, 12, 14	7	7						6	5	4	3	2	1	0									11	13
19		0	0																					1	2
20	5, 19, 11, 13, 15 . . .	5	5	4	3	2	1	0													2	1	0	6	12
21	3, 8, 9	3	3																					16	17

$S[m]$ 5, 11, 13, 15 19, 1, 4, 16, 17, 10, 12, 14, 20, 7, 2, 6, 18, 8, 3, 9.

ный в результате работы алгоритма массив s[m], а в табл. 3 — список фиктивных работ. В табл. 4 приведен общий список реальных и фиктивных работ (фиктивные отмечены значком *), упорядоченных по коду их начал. По этому списку построена сетевая модель разработки, изображенная на рис 3. Нетрудно убедиться, что построенная сетевая модель полностью соответствует исходным данным и всем правилам построения сети.

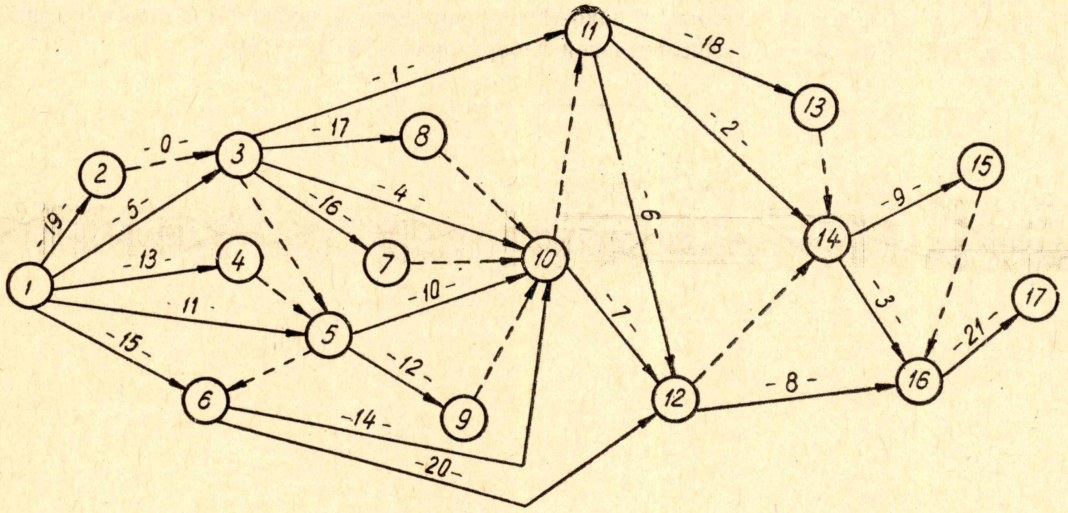

Рис. 3. Сетевая модель, построенная по результатам работы алгоритма

Необходимо отметить, что полученная информация о сетевой модели (табл. 4) обычно используется как исходная для анализа сети. Таким образом, описанный алгоритм позволяет, не разрабатывая полностью сетевой график (не вычерчивая сеть и не шифруя события), приступить к его анализу, основываясь на информации, заданной табл. 1. В исходной информации, полученной после анализа сетевого графика и предназначенной для исполнителей, машинные коды работ могут быть заменены на порядковые номера этих работ в исходном списке или на их действительные названия (при наличии алфавитно-цифрового вывода), взятые из второго столбца табл. 1.

Таблица 3

Список фиктивных работ

i	*j*
2	3
3	5
4	5
5	6
7	10
8	10
9	10
10	11
12	14
13	14
15	16

Таблица 4

Порядковый номер работы	*i*—*j*	Порядковый номер работы	*i*—*j*
19	1—2	20	6—12
5	1—3	*)	7—10
13	1—4	*)	8—10
11	1—5	*)	9—10
15	1—6	*)	10—11
*)	2—3	7	10—12
*)	3—5	6	11—12
16	3—7	18	11—13
17	3—8	2	11—14
4	3—10	*)	12—14
1	3—11	8	12—16
*)	4—5	*)	13—14
*)	5—6	9	14—15
12	5—9	3	14—16
10	5—10	*)	15—16
14	6—10	21	16—17

ЛИТЕРАТУРА

1. Основные положения по разработке и применению СПУ., М., «Экономика», 1965.

2. Ю. Н. Ефимов. Программа предварительного анализа сложного сетевого графика с локализацией логических ошибок. Изв. ТПИ. Сб. «Труды вычислительной лаборатории», № 168, 1969.

3. Ю. Н. Ефимов. Алгоритм нахождения порядковой функции сетевого графика. Изв. ТПИ. Сб. «Труды вычислительной лаборатории», № 168, 1969.