# 3. Сетевые модели планирования и управления проектами

*Проектом* называют совокупность работ, направленных на достижение некоторой цели. Работы проекта, как правило, частично упорядочены. Выполнение работы не может быть начато до завершения всех предшествующих ей работ. Продолжительность выполнения каждой работы известна. Предполагается, что начатая работа продолжается без перерыва до ее завершения. Проект считается выполненным, если выполнены все его работы.

Сетевая модель – это графическое представление проекта. Она позволяет найти минимальные сроки завершения проекта и отдельных работ, а также определить множество *критических* работ, увеличение продолжительности выполнения любой из которых приводит к увеличению времени выполнения всего проекта.

## 3.1. Построение сетевой модели проекта

Пример 3.1. (строительство дома). При строительстве дома необходимо выполнить такие работы, как создание подъездных дорог, подготовка котлована, возведение фундамента, подводка коммуникаций, монтаж здания, крыши, отделочные работы и т. п. Каждую работу проекта можно начать выполнять только после завершения всех предшествующих ей работ. Например, фундамент возводится только после подготовки котлована. В то же время некоторые работы, например подводка коммуникаций и отделочные работы, могут осуществляться параллельно.

*Сетевая модель* проекта представляет собой графическое описание плана работ, показывающее взаимосвязь между всеми работами, выполнение которых необходимо для завершения проекта. В терминах теории графов сетевая модель – это ориентированный граф без контуров и петель с неотрицательными весами вершин или дуг.

При анализе проектов используются два типа сетевых моделей, которые условно можно назвать:

1) «работы-вершины»;
2) «работы-дуги».

В сетевой модели первого типа вершины являются образами работ, а дуги служат для отображения отношения предшествования между работами.

В модели второго типа, наряду с работами-дугами, рассматриваются также новые понятия – *события*, которым соответствуют вершины сети.

Событие отражает результат – завершение одних работ и возможность начала других.

Направление дуги сети задает отношение предшествования работ проекта.

Указанные выше модели являются эквивалентными в том смысле, что для любого проекта можно построить как одну, так и другую модель.

В данном пособии, как и в [1, 3], рассматриваются сетевые модели типа «работы-дуги». В таких моделях для задания соотношения предшествования работ-дуг часто приходится вводить *фиктивные* дуги нулевой длины (продолжительность выполнения соответствующих фиктивных работ равна нулю). Нефиктивные дуги будем называть также *фактическими*.

Определение 3.1. Сети с одним и тем же множеством фактических дуг назовем *эквивалентными*, если они задают одно и то же отношение предшествования дуг-работ.

Вершину, не имеющую входящих в нее дуг, будем называть *входом*, а вершину, из которой не выходит ни одной дуги, – *выходом* сетевой модели. Без ограничения общности можно считать, что сетевая модель имеет один вход и один выход. Если исходная сеть не обладает этим свойством, то можно ввести фиктивный вход (выход) и связать с ним все входы (выходы) фиктивными дугами. Полученная сеть, очевидно, *эквивалентна* исходной сети.

Пример 3.2. Привести сетевую модель, изображенную на рис. 3.1 к эквивалентной сети с одним входом и одним выходом.

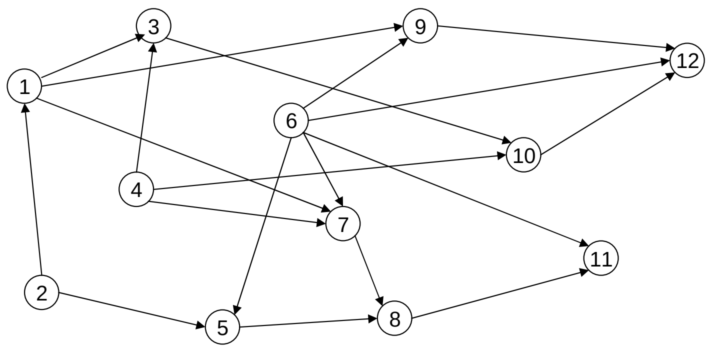


Рис. 3.1

Решение. Входами сети являются вершины 2, 4 и 6. Введем дополнительную вершину 13 и свяжем ее фиктивными дугами (13, 2), (13, 4) и (13, 6) со всеми входами.

Вершины 11 и 12 являются выходами сетевой модели. Добавим вершину 14 и соединим с ней фиктивными дугами (11, 14), и (12, 14) все выходы сетевой модели.

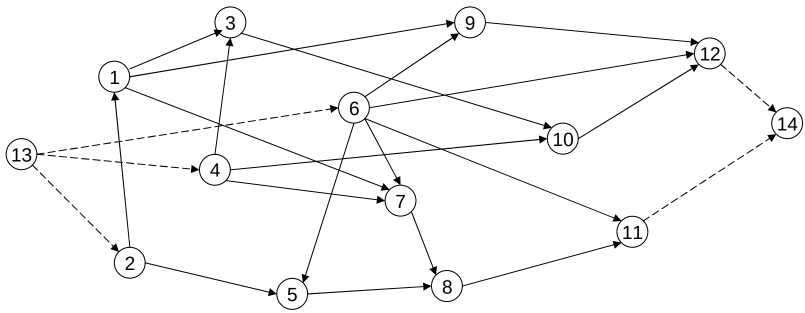


Рис. 3.2

На рис. 3.2 представлена новая сеть с одним источником 13 и одним стоком 14. Фиктивные дуги изображены пунктирными линиями.

### 3.2. Упрощение сетевой модели

В ряде случаев исходную сеть можно упростить, исключив часть вершин и фиктивных дуг. Введем следующие обозначения:

$S = (X, U)$ – сеть с множеством вершин $X$ и множеством дуг $U$;

$i(j)$ – начальная вершина дуги $j$;

$k(j)$ – конечная вершина дуги $j$;

$\overline{U}$ – множество фактических дуг сети;

$U_i$ – множество дуг из $\overline{U}$, принадлежащих объединению всех путей из входа сети в вершину $i$;

$U^i$ – множество дуг из $\overline{U}$, принадлежащих объединению всех путей из вершины $i$ в выход сети.

Зафиксируем пару вершин $i$ и $k$, для которых

1) не существует дуги $j \in \overline{U}$, инцидентной одновременно $i$ и $k$;

2) не существует дуг $p, q \in \overline{U}$, для которых

– $i(p) = i(q)$, где $k(p) = I$ и $k(q) = k$ или
– $k(p) = k(q)$, где $i(p) = i$ и $i(q) = k$.

Результатом операции *склеивания* вершин $i$ и $k$ будет сеть $S'$, полученная из исходной сети $S$ удалением вершины $k$ и замыканием инцидентных ей дуг на вершину $i$. Когда в результате склейки между какой-то парой вершин $i$ и $l$ в $S'$ будет более одной дуги:

– если среди них есть фактическая дуга, то удаляются все параллельные ей фиктивные дуги;

– если $i$ и $l$ в $S'$ связаны только фиктивными дугами, то все они кроме одной удаляются.

Таким образом, если в исходной сети $S$ дуга имела вершину $k$ начальной (конечной), то в полученной сети $S'$ эта дуга имеет начальной (конечной) вершину $i$.

Очевидно, в $S'$ существует путь из одной вершины в другую тогда и только тогда, когда такой путь есть в $S$.

Эквивалентность сетей $S$ и $S'$ проверяется с помощью следующего необходимого и достаточного условия. Пусть вершины $i$ и $k$ удовлетворяют условиям 1–2, описанным выше. Сеть $S'$, полученная из $S$ склеиванием $i$ и $k$ и удалением $k$, эквивалентна $S$ тогда и только тогда, когда выполняется хотя бы одно из условий: $U_i = U_k$ или $U^i = U^k$ .

Пример 3.3. Упростить сетевую модель, представленную на рис. 3.3.

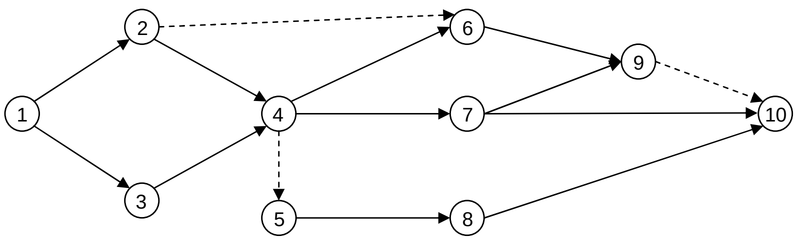


Рис. 3.3.

Решение. Очевидно, для выполнения приведенных выше условий вершины – кандидаты на склеивание должны быть соединены *фиктивной* дугой. Рассмотрим такие пары вершин в данной сети. Вершины 4 и 5 удовлетворяют свойствам 1–2. Очевидно, $U_4 = \{(1, 2),(2, 4),(1, 3),(3, 4)\} = U_5$. Следовательно, вершины 4 и 5 могут быть склеены.

Для вершин 9 и 10 свойство 2 не выполняется, так как вершина 7 связана с обеими вершинами 9 и 10 фактическими дугами. Следовательно, 9 и 10 нельзя склеить.

Для вершин 2 и 6 свойства 1–2 выполняются, но $U_2 = \{(1,2)\} \neq U_6 = \{(1, 2), (2, 4), (1, 3), (3, 4), (4, 6)\}$ и $U^2 = \{(2, 4), (4, 6), (4, 7), (5, 8), (6, 9), (7, 9), (7, 10), (8, 10)\} \neq U^6 = \{(6, 9)\}$. Следовательно, вершины 2 и 6 не могут быть склеены с сохранением эквивалентности сетей.

## 3.3. Вычисление параметров сетевой модели

Пусть задана сеть, вершины которой пронумерованы числами 1, ..., $n$, а дуги – числами 1, ..., $m$. Тогда сетевую модель можно представить списком дуг, каждая из которых закодирована числами:

– $j$ – номер дуги;

– $i(j)$ – номер начальной вершины дуги $j$;

– $k(j)$ – номер конечного вершины дуги $j$;

– $\tau_j$ – длина дуги (длительность выполнения работы) $j$.

Определение 3.2. *Рангом* $R_i$ события $i$ называется число дуг в пути максимальной (по числу дуг) длины, ведущему из входа сети в вершину $i$.

**Алгоритм Форда** вычисляет ранги событий следующим образом:

Шаг 0. Полагаем $R_i = 0$ для всех $i = 1, ..., n$. Пусть в результате $(k - 1)$ шагов найдены ранги $R_i$.

Шаг $k$. Просматривая последовательно дуги (работы) $j = 1, ..., m,$ вычисляем $R_{k(j)} = \max\{R_{k(j)}, R_{i(j)} + 1\}$.

Если на очередном шаге ранги всех вершин сети не изменились, то алгоритм завершает свою работу. Иначе полагаем $k = k + 1$ и повторяем шаг $k$.

Пример 3.4. Найти ранги событий для сетевой модели, изображенной на рис. 3.4, где рядом с каждой дугой указан ее номер.

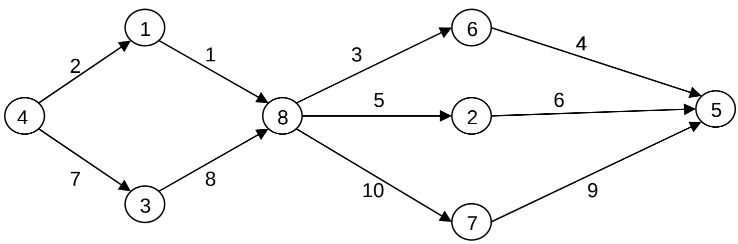


Рис. 3.4

Решение. На нулевом шаге полагаем $R_i = 0$ для всех вершин $i = 1, \ldots, 8$.

На первом шаге алгоритма последовательно просматриваем все дуги и пересчитываем ранги их концевых вершин:

работа 1 = (1, 8) : $R_8 = \max\{R_8, R_1 + 1\} = \max\{0, 0 + 1\} = 1,$

работа 2 = (4, 1) : $R_1 = \max\{R_1, R_4 + 1\} = \max\{0, 0 + 1\} = 1,$

работа 3 = (8, 6) : $R_6 = \max\{R_6, R_8 + 1\} = \max\{0, 1 + 1\} = 2$, и т. д. Продолжая вычисления, получаем табл. 3.1.

*Таблица 3.1*

| Ранг / События | Шаг 0 | Шаг 1 | Шаг 2 | Шаг 3 |
|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 1 |
| 2 | 0 | 2 | 3 | 3 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 0 | 0 | 0 | 0 |
| 5 | 0 | 3 | 4 | 4 |
| 6 | 0 | 2 | 3 | 3 |
| 7 | 0 | 3 | 3 | 3 |
| 8 | 0 | 2 | 2 | 2 |

На третьем шаге ранги событий не изменись. Следовательно, алгоритм завершает свою работу и ранги событий $R_1 = 1$, $R_2 = 3$, $R_3 = 1$, $R_4 = 0$, $R_5 = 4$, $R_6 = 3$, $R_7 = 3$, $R_8 = 2$.

Определение 3.3. Нумерацию событий сетевой модели назовем *правильной*, если $i(j) < k(j)$ для каждой работы $j = 1, \ldots, m$. Правильную нумерацию событий (вершин сети) легко осуществить, зная их ранги. Вершина-вход с рангом 0 получает номер 1. Вершины ранга 1 получают номера 2, …, $n_1$, ранга 2 – номера $n_1 + 1, \ldots, n_2$, и т. д.

Пример 3.5. Осуществить правильную нумерацию событий в сетевой модели примера 3.4.

Решение. Так как ранги событий найдены, то правильная нумерация событий осуществляется следующим образом:

– входу (т. е. событию с рангом 0) приписываем номер 1;

– событиям ранга 1 (событиям 1 и 3) – номера 2 и 3;

– событиям ранга 2 (8) – номер 4;

– событиям ранга 3 (6, 2 и 7) – номера 5, 6 и 7;

– выходу (5) – номер 8.

Определение 3.4. *Длиной пути* $\mu(i, k)$ из вершины $i$ в вершину $k$ во взвешенном графе назовем величину, равную сумме весов входящих в него дуг, которую обозначим $|\mu(i, k)|$. Путь из $i$ в $k$, имеющий максимальную длину, обозначим $\mu^*(i, k)$.

Далее предполагается правильная нумерация вершин, при которой, в частности, вершина 1 – это источник, а $n$ – сток.

Определение 3.5. *Наиболее ранним временем* наступления события $i$ назовем величину $T_i^p = |\mu^*(1, i)|$.

Величина $T_i^p$ соответствует минимально возможному времени выполнения всех работ, предшествующих событию $i$.

Обозначим через $T_{кр} = T_n^p$ *критическое время* проекта (минимальное время выполнения проекта), а через $\mu^*(1, n)$ – *критический путь* сетевой модели. Работы и события, принадлежащие критическому пути, называются *критическими*. Задержка наступления любого критического события приведет к увеличению срока завершения всего проекта $T_{кр}$.

Определение 3.6. *Наиболее поздним* временем наступления события $i$ назовем величину $T_i^n = T_{кр} - |\mu^*(i, n)|$. Величина $T_i^n$ равна максимальному времени свершения события $i$, не приводящему к увеличению времени выполнения всего проекта.

Сегмент $[T_i^p, T_i^n]$ называется *интервалом свободы* события $i$.

Для того чтобы событие $i$ было критическим, необходимо и достаточно $T_i^p = T_i^n$.

Резервы времени выполнения работ – это величины, характеризующие максимально возможное время, на которое можно увеличить продолжительность выполнения работ или задержать начало их выполнения, чтобы время начала выполнения других работ не изменилось.

Определение 3.7. *Полным резервом* времени выполнения работы $j$ называется величина $\Pi_j = T_{k(j)}^n - T_{i(j)}^p - \tau_j$.

Необходимым и достаточным условием принадлежности работы критическому пути является равенство нулю ее полного резерва.

Определение 3.8. *Свободным резервом* времени выполнения работы $j$ называется величина $T_{k(j)}^p - T_{i(j)}^p - \tau_j$.

Свободный резерв может быть использован для выявления работ, задержка которых на эту величину не приводит к изменению полных резервов всех последующих работ.

Наиболее ранние времена наступления событий могут быть найдены при помощи следующего алгоритма Форда:

Шаг 0. Для каждого события $i = 1, \ldots, n$ полагаем $T_i^p = 0$. Пусть в результате $(k - 1)$ шагов найдены времена $T_i^p$.

Шаг $k$. Последовательно просматриваем все работы $j = 1, \ldots, m$ и вычисляем $T_{k(j)}^p = \max\left\{T_{k(j)}^p, T_{i(j)}^p + \tau_j\right\}$.

Если все времена $T_i^p$ не изменились, то алгоритм заканчивает свою работу. Иначе полагаем $k = k + 1$ и выполняем следующий шаг алгоритма.

Список дуг можно упорядочить таким образом, что $T_i^p$ будут найдены за один просмотр работ. Пусть события пронумерованы правильно.

Определение 3.9. Если для каждой пары дуг $p$ и $q$, когда $p \in \mu(1, i(q))$, выполняется неравенство $p < q$, то говорят, что список дуг *правильно упорядочен.*

Правильная упорядоченность дуг согласуется с отношением предшествования работ, т. е. если $p$ предшествует $q$, то имеет место неравенство $p < q$.

Для получения правильной упорядоченности дуг достаточно упорядочить их по номерам конечных вершин (при правильной нумерации последних). В вершине 1 не заканчивается ни одна дуга. Все дуги $I_2$, входящие в вершину 2, получают номера 1, … , $|I_2|$. Дугам $I_3$, входящим в вершину 3, приписываются номера $|I_2| + 1, \dots, |I_2| + |I_3|$, и т. д.

Если дуги правильно упорядочены, то алгоритм Форда найдет $T_i^p$ за один просмотр дуг в порядке их нумерации $j = 1, \dots, m$.

Приведем алгоритм Форда для вычисления наиболее поздних времен наступления событий:

Шаг 0. Для каждого события $i = 1, \dots, n$ полагаем $T_i^n = T_{кр}$. Пусть в результате $(k - 1)$ шагов найдены времена $T_i^n$.

Шаг $k$. Последовательно просматриваем все работы $j = 1, \dots, m$ и вычисляем $T_{i(j)}^n = \min\left\{T_{i(j)}^n,\ T_{k(j)}^n - \tau_j\right\}$.

Если после выполнения очередного шага все времена $T_i^n$ не изменились, то алгоритм заканчивает работу. Иначе полагаем $k = k + 1$ и выполняем следующий шаг алгоритма.

При правильно-упорядоченном списке работ величины $T_i^n$ находятся за один шаг алгоритма, если работы просматривать в обратном порядке $j = m, \dots, 1$.

Когда величины $T_i^p$ и $T_i^n$ известны, остальные характеристики сетевой модели (критические работы, события, резервы и т. п.) могут быть найдены за один просмотр списка работ.

Пример 3.6. Пусть сетевая модель задана табл. 3.2.

*Таблица 3.2*

| Номер дуги | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Дуга | 1–3 | 1–4 | 4–7 | 7–9 | 2–8 | 4–5 | 5–9 | 8–9 | 2–9 | 3–5 | 3–4 | 6–7 | 2–6 | 2–4 | 2–3 | 1–6 | 4–6 | 1–2 |
| Длина дуги | 24 | 52 | 12 | 44 | 26 | 12 | 64 | 69 | 17 | 75 | 3 | 33 | 50 | 5 | 0 | 44 | 29 | 4 |

Требуется найти вход и выход сети, ранние и поздние времена наступления событий, время выполнения проекта, полные резервы времени для каждой работы-дуги, критические работы и пути. Нумерацию вершин и дуг не менять.

Решение. Просматривая список дуг, выясняем, что вершина 1 не фигурирует в качестве концевой вершины ни одной дуги. Это вход сети.

Аналогично для нахождения выходов сети определяем, какие из вершин не являются начальными вершинами дуг. Такая вершина одна – 9. Она является выходом сетевой модели.

Найдем ранние времена наступления событий, используя алгоритм Форда.

Полагаем $T_i^p = 0$ для всех $i = 1, \dots, 9$.

На первом шаге алгоритма просматриваем все работы и пересчитываем ранние времена наступления их концевых событий в соответствии с рекуррентными соотношениями алгоритма Форда. В частности, для первых двух работ имеем:

работа 1 = (1, 3) : $T_3^p = \max\{ T_3^p,\ T_1^p + \tau_1\} = \max\{0, 0 + 24\} = 24,$

работа 2 = (1, 4) : $T_4^p = \max\{ T_4^p,\ T_1^p + \tau_2\} = \max\{0, 0 + 52\} = 52.$

Результаты вычислений ранних времен помещены в табл. 3.3, где в первом столбце находятся номера событий, а в верхней строке – номера шагов алгоритма. После третьего шага ранние времена наступления событий не изменились ни для одного из них, поэтому алгоритм завершает свою работу.

Время выполнения проекта соответствует раннему времени наступления события 9 и равно $T_{кр} = T_9^p = 163$.

*Таблица 3.3*

| $i / k$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 4 | 4 | 4 |
| 3 | 0 | 24 | 24 | 24 |
| 4 | 0 | 52 | 52 | 52 |
| 5 | 0 | 99 | 99 | 99 |
| 6 | 0 | 81 | 81 | 81 |
| 7 | 0 | 64 | 114 | 114 |
| 8 | 0 | 26 | 30 | 30 |
| 9 | 0 | 128 | 163 | 163 |

Найдем поздние времена наступления событий.

Положим $T_i^n = T_{кр} = 163$ для всех вершин $i = 1, \dots, 9$.

Просматриваем последовательно все работы и пересчитываем поздние времена наступления событий, соответствующих концевым вершинам работ-дуг. Так для первых двух работ имеем:

работа 1 = (1, 3): $T_1^n = \min\{T_1^n, T_3^n - \tau_1\} = \min\{163, 163 - 24\} = 139$,

работа 2 = (1, 4): $T_1^n = \min\{T_1^n, T_4^n - \tau_2\} = \min\{139, 163 - 52\} = 111$.

Результаты вычислений помещены в табл. 3.4.

*Таблица 3.4*

| i/k | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 1 | 163 | 20 | 0 | 0 |
| 2 | 163 | 24 | 24 | 24 |
| 3 | 163 | 24 | 24 | 24 |
| 4 | 163 | 57 | 57 | 57 |
| 5 | 163 | 99 | 99 | 99 |
| 6 | 163 | 86 | 86 | 86 |
| 7 | 163 | 119 | 119 | 119 |
| 8 | 163 | 94 | 94 | 94 |
| 9 | 163 | 163 | 163 | 163 |

Найдем полные резервы времени для каждой работы и поместим их в табл. 3.5.

*Таблица 3.5*

| Дуга $j$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $П_j$ | 0 | 5 | 55 | 5 | 64 | 35 | 0 | 64 | 142 | 0 | 30 | 5 | 32 | 48 | 20 | 42 | 5 | 20 |

Для примера приведем вычисления данной характеристики для первых двух работ:

работа 1 = (1, 3) : $П_1 = T_3^n - T_1^p - \tau_1 = 24 - 0 - 24 = 0$,

работа 2 = (1, 4) : $П_2 = T_4^n - T_1^p - \tau_2 = 57 - 0 - 52 = 5$.

Критическими работами являются те, полный резерв времени которых равен нулю. В данном случае это работы 1, 7 и 10.

Критическому пути принадлежат события, у которых ранние и поздние времена их свершения совпадают. В нашем примере это события 1, 3, 5, 9.

## Упражнения

1. Упростить сетевую модель, изображенную на рис. 3.5.

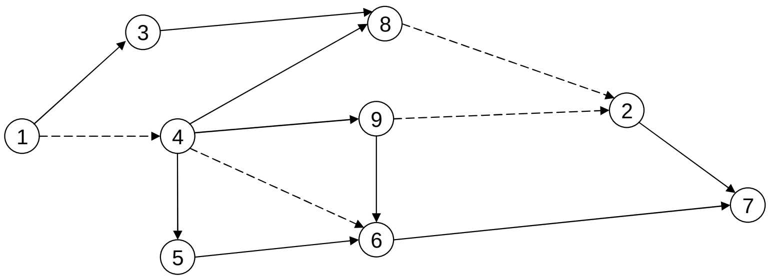


Рис. 3.5

2. Для сетевой модели из примера 3.6 вычислить ранги событий.
3. Пусть сетевая модель задана табл. 3.6.

*Таблица 3.6*

| Номер дуги | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Дуга | 4–9 | 4–7 | 6–7 | 6–8 | 2–4 | 2–3 | 1–6 | 1–3 | 3–7 | 1–2 | 1–5 | 5–6 | 1–4 | 1–9 | 3–5 | 2–7 | 3–4 | 8–9 |
| Длина дуги | 1 | 52 | 6 | 13 | 65 | 52 | 47 | 20 | 34 | 30 | 33 | 64 | 47 | 60 | 30 | 66 | 36 | 44 |

Найти вход и выход сети, ранние и поздние времена наступления событий, время выполнения проекта, полный резерв времени для каждой работы-дуги, критические работы и пути. Нумерацию вершин и дуг не менять.