

*П.С. Баркалов, И.В. Буркова, А.В. Глаголев, В.Н. Колпачев*

## **ЗАДАЧИ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В УПРАВЛЕНИИ ПРОЕКТАМИ**

**Москва: ИПУ РАН, 2002. – 65 с.**

# СОДЕРЖАНИЕ

|  |           |
|--|-----------|
| <b><u>ВВЕДЕНИЕ</u></b> .....   | <b>4</b>  |
| <b><u>ГЛАВА 1. ЗАДАЧИ РЕСУРСНОГО ПЛАНИРОВАНИЯ КОМПЛЕКСОВ РАБОТ</u></b> .....                       | <b>5</b>  |
| 1.1. <u>ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ</u> .....   | 5         |
| 1.2. <u>ОПТИМИЗАЦИЯ ПО СТОИМОСТИ</u> .....   | 16        |
| <b><u>ГЛАВА 2. ДВОЙНАЯ СЕТЕВАЯ МОДЕЛЬ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ</u></b> .....                         | <b>20</b> |
| 2.1. <u>ОПРЕДЕЛЕНИЕ МОМЕНТОВ ОКОНЧАНИЯ ОПЕРАЦИЙ ПРИ ЗАДАННОМ ПОТОКЕ РЕСУРСОВ ПО ГРАФУ ПР</u> ..... | 25        |
| 2.2. <u>ПОСТАНОВКА ЗАДАЧИ И ОБСУЖДЕНИЕ МОДЕЛИ</u> .....  | 34        |
| 2.3. <u>ОПТИМИЗАЦИЯ КАЛЕНДАРНОГО ГРАФИКА С УЧЁТОМ ВРЕМЕНИ ПЕРЕМЕЩЕНИЙ БРИГАД</u> .....             | 35        |
| 2.4. <u>ОПТИМИЗАЦИЯ КАЛЕНДАРНОГО ГРАФИКА ДЛЯ РАДИАЛЬНОЙ ТРАНСПОРТНОЙ СХЕМЫ</u> .....               | 45        |
| 2.5. <u>ПРОИЗВОЛЬНАЯ ТРАНСПОРТНАЯ СХЕМА</u> .....  | 51        |
| <b><u>ЗАКЛЮЧЕНИЕ</u></b> .....   | <b>65</b> |
| <b><u>ЛИТЕРАТУРА</u></b> .....   | <b>65</b> |

## ВВЕДЕНИЕ

Управление проектами представляет собой совокупность методологии, методик моделей, методов, технических и программных средств, применяемых при разработке и реализации проектов, то есть, уникальных процессов, ограниченных во времени и требующих затрат ресурсов. Существенную часть моделей и методов управления проектами составляют задачи построения календарных планов реализации проекта, связанных в основном с распределением ограниченных ресурсов. Задачи распределения ресурсов на сетевых графиках относятся к сложным многоэкстремальным задачам.

Существует небольшое число частных постановок, для которых предложены точные методы решения. В общем случае применяются приближенные и эвристические алгоритмы. Сложность задач еще более возрастает, если учитывать времена перемещения ресурсов между работами (достаточно сказать, что простейшая задача определения очередности выполнения работ одной бригадой при учете времени перемещения бригады с работы на работу превращается в сложнейшую задачу коммивояжера).

В предлагаемой работе рассмотрен ряд задач оптимального распределения ресурсов при учете времен их перемещения (двойная сетевая модель) и предложены достаточно эффективные методы их решения.

# ГЛАВА 1. Задачи ресурсного планирования комплексов работ

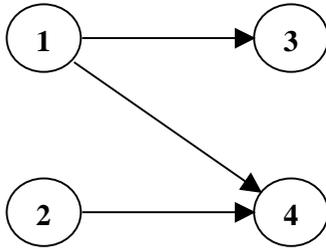
## 1.1. Основные понятия и определения

При постановке задач ресурсного планирования предполагается, что проект описан в виде *комплекса работ* в определенными зависимостями между ними. Зависимости между работами отображаются в виде *сетевого графика* (сети).

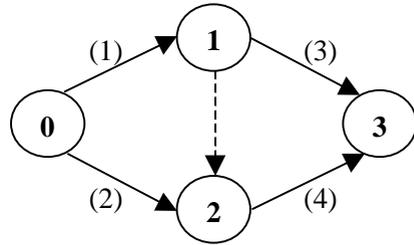
Существуют два способа изображения работ в сетевом графике. В первом способе работы изображаются в виде вершин сети, а зависимости между работами – в виде дуг сети. Во втором способе вершины сети соответствуют событиям сети, то есть моментам завершения одной или нескольких работ, а дуги – работам сети, при этом, для отображения всех требуемых взаимосвязей иногда приходится вводить дуги специального вида – фиктивные работы (можно считать, что такие дуги соответствуют работам нулевой продолжительности, не требующим ресурсов). На рис. 1.1.а приведен комплекс работ в виде «вершина – работа», а на рис. 1.1.б этот же комплекс представлен в виде «вершина – событие» (зависимость, соответствующая фиктивной работе показана пунктиром). В скобках на рис. 1.1.б указаны номера работ рис. 1.1.а, которым соответствуют дуги рис. 1.1.б.

Как правило, будем применять изображение комплекса работ в виде «вершина – работа».

Для полного описания комплекса работ необходимо задать описание каждой работы.



a)



**Ошибка! Ошибка связи.б)**

Рис. 1.1.

Важной характеристикой работы является ее *объем*  $W$ . Он определяется на основе нормативов, экспертных оценок или имеющегося опыта. Объем может измеряться в единицах трудоемкости, стоимости и т.д.

Следующей характеристикой работы является ее продолжительность (время выполнения). В простейшем случае работа описывается величиной продолжительности и количеством требуемых для ее выполнения ресурсов. В этом случае будем говорить, что работа выполняется с *фиксированной* интенсивностью. Тогда объем работы для решения задачи ресурсного планирования не нужен, он используется при контроле хода реализации проекта. Если количество ресурсов на работу может принимать различные значения, то и продолжительность работы тоже может быть разной. При этом, если количество ресурсов в процессе выполнения работы не меняется, то будем говорить, что работа выполняется с *постоянной* интенсивностью. Для описания работы, выполняемой с постоянной интенсивностью достаточно задать продолжительность работы при различных уровнях ресурсов, то есть зависимость  $\tau(u)$ , где  $u$  – количество ресурсов, выполняющих работу. Отношение

$$w(u) = \frac{W}{\tau(u)} \quad (1.1.1)$$

определяет интенсивность выполнения работы (производительность участвующих в работе ресурсов), которую мы будем называть *скоростью* выполнения работы (или просто – скоростью работы). Из выражения (1.1.1) видно, что скорость измеряется объемом работ, выполняемым в единицу времени. Без ограничения общности можно принять, что скорость работы является неубывающей функцией количества ресурсов.

Заметим, что одновременное увеличение (уменьшение) и объема, и скорости работы в одно и то же число раз не изменяет ее продолжительности. Следовательно, и величина объема, и его единица измерения могут быть выбраны произвольно. Как правило, единица измерения объема выбирается из содержательного смысла.

Наиболее сложным является случай, когда работа может выполняться с переменной интенсивностью, то есть количество ресурсов на работе может меняться в процессе ее выполнения. Для описания работы в этом случае необходимо задать ее объем  $W$  и зависимость  $w = f(u)$  скорости работы от количества выполняющих ее ресурсов. Обозначим через  $u(t)$  количество ресурсов на работе в момент времени  $t$ ,  $t_n$  – момент начала работы,  $t_k$  – момент ее окончания. Имеет место соотношение:

$$\int_{t_n}^{t_k} f[u(t)]d\tau = W \quad (1.1.2)$$

Типичный вид зависимости скорости операции от количества ресурсов приведен на рис. 1.2. Сначала с ростом количества ресурсов средняя производительность растет, а затем она начинает падать.

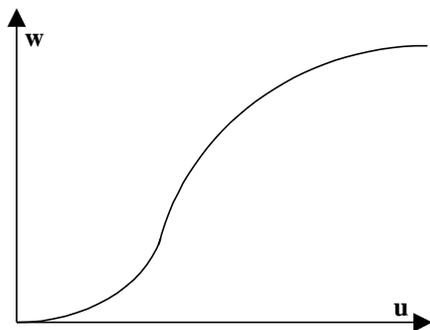


Рис. 1.2.

На практике применяются более простые – либо линейные зависимости вида

$$f(u) = \begin{cases} 0, & u < a \\ a, & a \leq u \leq b, \\ b, & b < u \end{cases} \quad (1.1.3)$$

либо степенные вида  $f(u) = u^\alpha$  (как правило,  $\alpha < 1$ ).

Важной характеристикой работы являются затраты ресурсов

$$S = \int_{t_n}^k u(\tau) d\tau \quad (1.1.4)$$

(прямые затраты сырья, материалов, трудозатраты, финансовые и т.д.). В ряде случаев ограничения наложены на затраты ресурсов на работу. Очевидно, что с ростом затрат продолжительность работ не увеличивается при разумном использовании ресурсов. Определим зависимость продолжительности работы  $\tau$  от затрат на ее выполнение при заданной зависимости скорости работы от количества ресурсов, предполагая, что ресурсы распределяются оптимально.

Примем сначала, что зависимость  $f(u)$  является вогнутой дифференцируемой функцией, то есть, для любого  $0 < \alpha < 1$  и любых  $u_1$  и  $u_2$

$$f(\alpha u_1 + (1-\alpha)u_2) \geq \alpha f(u_1) + (1-\alpha)f(u_2).$$

**Теорема 1.1.** Оптимальному распределению ресурсов соответствует выполнение работы с постоянной интенсивностью.

*Доказательство.* Пусть работа выполняется за время  $\tau$  периодов. Поставим задачу распределить затраты по периодам так, чтобы объем выполненной работы был максимальным, то есть

$$\sum_{k=1}^{\tau} f(u_k) \rightarrow \max$$

при ограничении

$$\sum_{k=1}^{\tau} u_k = S,$$

где  $u_k$  – количество ресурсов в периоде  $k$ . Применяя метод множителей Лагранжа получим необходимое условие оптимальности:

$$f'(u_k) = \lambda, \quad k = 1 \div T.$$

Следовательно,  $u_k = u$  для всех  $k$ . Учитывая, что  $u\tau = S$  и

$$f(u) \times \tau = W,$$

получаем:

$$f\left(\frac{S}{\tau}\right) \times \tau = W. \quad (1.1.5)$$

Из этого уравнения определяется зависимость  $\tau(S)$  либо  $S(\tau)$ .

**Пример 1.1.** Пусть  $f(u) = u^{1/\alpha}$ ,  $\alpha \geq 1$ . Имеем:

$$\left(\frac{S}{\tau}\right)^{1/\alpha} \times \tau = W.$$

Из этого уравнения получаем

$$S(\tau) = \frac{W^\alpha}{\tau^{\alpha-1}}.$$

В случае  $\alpha = 2$ ,  $S(\tau) = \frac{W^2}{\tau}$ .

Заметим, что в случае линейной зависимости (1.1.3) затраты равны объему работы  $W$ . Если величину затрат умножить на стоимость единицы ресурса, то получим стоимость работы, которая является основой формирования сметы и бюджета проекта. Если зависимость  $f(u)$  имеет произвольный вид (например, задана в конечном числе точек), то строим вогнутую зависимость, максимально близкую к заданной. Способ построения ясен из рис. 1.3.

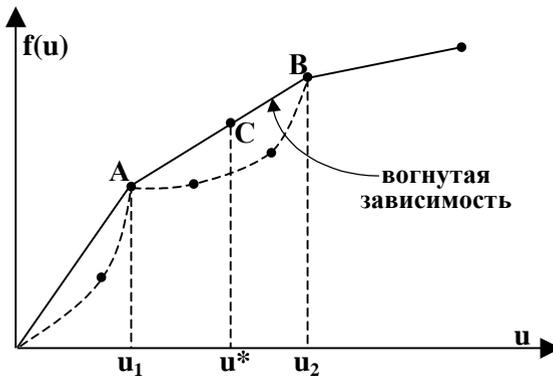


Рис. 1.3.

Далее для заданной продолжительности  $\tau$  определяем  $u^*$  и соответственно  $S = u^*\tau$ , принимая полученную вогнутую зависимость за истинную. представим точку  $u^*$  как выпуклую линейную комбинацию ближайших точек  $u_1$  и  $u_2$ :

$$u^* = \alpha u_1 + (1-\alpha)u_2, \quad 0 < \alpha < 1.$$

Очевидно, что

$$f(u^*) = \alpha f(u_1) + (1-\alpha)f(u_2). \quad (1.1.6)$$

Разделим интервал времени  $\tau$  на два:  $\tau_1 = \alpha\tau$  и  $\tau_2 = (1-\alpha)\tau$ . Примем  $u = u_1$  в интервале  $\tau_1$  и  $u = u_2$  в интервале  $\tau_2$ . Из условия (1.1.6). Имеем

$$\begin{aligned} \tau f(u^*) &= \tau_1 f(u_1) + \tau_2 f(u_2) = W, \\ u_1 \tau_1 + u_2 \tau_2 &= [\alpha u_1 + (1-\alpha)u_2]\tau = u^* \tau, \end{aligned}$$

то есть полученное допустимое распределение ресурсов позволяет выполнить весь объем работ с затратами ресурсов  $S$ .

Из полученного результата следует важный вывод: для любых зависимостей  $f(u)$  и затрат  $S$  существует оптимальное распределение затрат во времени с не более чем двумя интервалами постоянства уровней ресурсов. Типичными зависимостями затрат от времени, применяемыми на практике, являются кусочно-линейные (рис. 1.4) и степенные:

**Ошибка! Объект не может быть создан из кодов полей редактирования.**

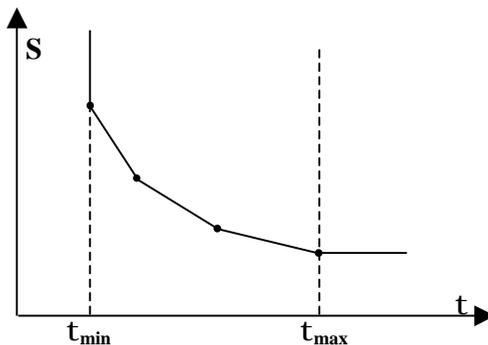


Рис. 1.4.

В выполнении работ проекта, как правило, участвуют различные ресурсы. Можно выделить две взаимосвязанные группы ресурсов. Материально-технические ресурсы, т.е. сырье, материалы, конструкции, комплектующие, энергетические ресурсы, топливо, т.е. ресурсы типа «мощность» или технологические ресурсы, т.е. машины, механизмы для выполнения работ проекта, устанавливаемое оборудование и пр. трудовые ресурсы, осуществляющие непосредственную работу с материально-техническими ресурсами (например, строители, водители машин, монтажники оборудования и пр.).

Это многообразие сводится к *двум основным типам*:

***Невоспроизводимые, складуемые, накапливаемые*** ресурсы в процессе выполнения работ расходуются полностью, не допуская повторного использования. Не использованные в данный отрезок времени, они могут использоваться в дальнейшем. Иными словами, такие ресурсы можно накапливать с последующим расходом запасов. Поэтому их часто называют ресурсами типа «энергия». Примерами таких ресурсов являются топливо, предметы, средства труда однократного применения, а также – финансовые средства.

***Воспроизводимые, нескладуемые, ненакапливаемые*** ресурсы сохраняют свою натурально-вещественную форму и по мере высвобождения могут использоваться на других работах. Если эти ресурсы простаивают, то их неиспользование в данный отрезок времени не компенсируется в будущем, т.е. они не накапливаются, поэтому ресурсы второго типа называют еще ресурсами типа «мощности». Примерами ресурсов типа «мощности» являются люди и средства труда многократного использования (машины, станки, механизмы и т.д.).

В дальнейшем будем предполагать, что ресурсы участвуют в работе в определенном соотношении, образуя набор ресурсов. Как правило, один из видов ресурсов является определяющим (например, на один токарный станок нужен один токарь, инструменты, детали для обработки и т.д.). Параметры набора показывают количество ресурса данного вида, требуемого на единицу определяющего ресурса. Скорость операции задается в этом случае как функция количества определяющего ресурса.

Ограничения на ресурсы, которые можно использовать на работах проекта, определяются функцией наличия (доступности) ресурсов. Если  $N(t)$  – функция наличия определяющего ресурса, то ограничения на распределение ресурсов по работам проекта имеет вид:

$$\sum_{i=1}^n u_i(t) \leq N(t),$$

где  $u_i(t)$  – количество ресурса на работе  $i$ ,  $n$  – число работ проекта. Довольно часто работы проекта разбиваются на классы, так что работы одного класса выполняются ресурсами одного вида. Если обозначить  $P_j$  – множество работ, выполняемых ресурсами  $j$ -го вида,  $N_j(t)$  – функцию наличия ресурсов  $j$ -го вида, то ограничения на распределение ресурсов записываются следующим образом:

$$\sum_{i \in P_j} u_i(t) \leq N_j(t), \quad j = \overline{1, m}.$$

Для накапливаемых ресурсов ограничения задаются в интегральном виде:

$$\sum_{i=1}^n \int_0^t u_i(\tau) d\tau \leq S(t).$$

Если ограничены общие затраты  $S$  на проект (или ограничена стоимость проекта), то ограничения на ресурсы определяются следующим неравенством:

$$\sum_{i=1}^n s_i \leq S,$$

где  $s_i$  – затраты ресурса на  $i$ -й работе (или стоимость  $i$ -й работы). Задача оптимального распределения ресурсов заключается в определении допустимого по ограничениям распределения ресурсов, минимизирующего заданный критерий оптимальности. Если ограничены ресурсы, то, как правило, ставится задача минимизации продолжительности  $T$  проекта, либо задача минимизации упущенной выгоды:

$$\Phi = \sum_{i=1}^n c_i t_i,$$

где  $t_i$  – момент окончания  $i$ -й работы,  $c_i$  – коэффициент упущенной выгоды. Если задан срок завершения проекта, либо допустимая величина упущенной выгоды, то решается в определенном смысле обратная задача – минимизации ресурсов либо затрат.

Поставленные задачи достаточно сложны и, как правило, не имеют эффективных методов решения. В общем случае для их решения применяются приближенные и эвристические алгоритмы. Точные методы получены для ряда частных случаев, которые рассматриваются ниже. В первую очередь выделяются различные виды сетевых графиков.

Будем рассматривать два частных вида сетевых графиков: независимые операции и сети с упорядоченными событиями. Случай независимых операций соответствует ситуации, когда все работы

могут выполняться одновременно, то есть отсутствуют логические (технологические) зависимости между работами.

Случай сети с упорядоченными событиями соответствует ситуации, когда задана некоторая очередность свершения событий сети. В сетях с упорядоченными событиями естественно использовать представление сети в виде «вершина – событие», рис. 1.5.

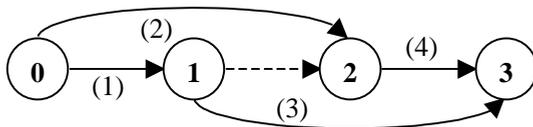


Рис. 1.5.

Однако, можно определить аналог таких сетей и в представлении «вершина – работа», рис. 1.6.

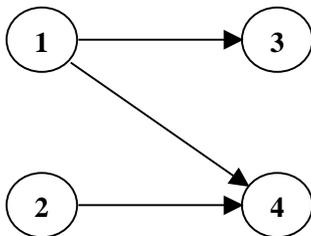


Рис. 1.6.

Для этого определим понятие «фронта работ», как максимального множества независимых работ, то есть таких, которые могут выполняться одновременно.

В сети рис. 1.6. можно выделить три различных фронта работ: (1;2), (3;2), (3;4). Заметим, что эти фронты в определенном смысле

упорядочены, а именно, фронт (1;2) расположен «левее» фронта (3;2), а последний – «левее» фронта (3;4). Другими словами, для любых двух фронтов работы одного из них либо совпадают, либо предшествуют работам другого. Таким образом, сетям с упорядоченными событиями соответствуют сети с упорядоченными фронтами.

## 1.2. Оптимизация по стоимости

Задачи оптимизации комплексов работ по стоимости относятся к классу задач, для которых существуют достаточно эффективные алгоритмы. Сначала рассмотрим простой случай, когда сетевой график представляет собой последовательную цепочку работ. Примем, что зависимость стоимости от продолжительности является линейной для каждой работы:

$$S_i(\tau_i) = a_i - k_i\tau_i, \quad d_i \leq \tau_i \leq D_i, \quad i = \overline{1, n},$$

где  $d_i$  – минимально возможная продолжительность работы,  $D_i$  – максимальная. Примем продолжительности всех работ равными максимальным  $\tau_i = D_i$ . При этом продолжительность проекта

$$T = \sum_{i=1}^n D_i.$$

Если мы хотим сократить продолжительность проекта с минимальным увеличением стоимости, то очевидно, что в первую очередь необходимо сокращать продолжительность работы, имеющей минимальную величину коэффициента  $k_i$ . Действительно, величина  $k_i$  определяет увеличение стоимости проекта при уменьшении продолжительности  $i$ -й работы на единицу. Продолжая таким образом, получим зависимость стоимости проекта от его продолжи-

тельности. Рассмотрим на примере обобщение этого алгоритма на случай произвольного сетевого графика.

**Пример 1.2.** Пусть сетевой график («вершина – событие») имеет вид рис. 1.7.

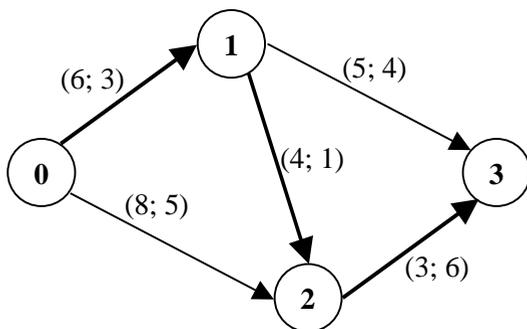


Рис. 1.7.

Величины  $a_i$ ,  $k_i$ ,  $d_i$  и  $D_i$  для всех работ приведены в таблице.

Таблица 1.1.

|               | $a_i$ | $k_i$ | $d_i$ | $D_i$ |
|---------------|-------|-------|-------|-------|
| <b>(0, 1)</b> | 20    | 3     | 2     | 6     |
| <b>(0, 2)</b> | 45    | 5     | 5     | 8     |
| <b>(1, 2)</b> | 7     | 1     | 1     | 4     |
| <b>(1, 3)</b> | 23    | 4     | 1     | 5     |
| <b>(2, 3)</b> | 20    | 6     | 1     | 3     |

**1 шаг.** Полагаем  $\tau_i = D_i$  для всех работ и определяем критический путь в сети. На рис. 1.7. в скобках у дуг указаны значения  $D_i$  (первые числа) и  $k_i$  (вторые числа) соответствующих работ. Критический путь выделен толстыми дугами. Для критического пути

$T_0 = 13$ , стоимость проекта  $S_0 = 15$ . Очевидно, что сокращать следует работу (1, 2). При сокращении работы (1, 2) на две единицы, критическими становятся работы (0, 2) и (1, 3), рис. 1.8, длина критического пути  $T_1 = 11$ , стоимость проекта  $S_1 = 17$ .

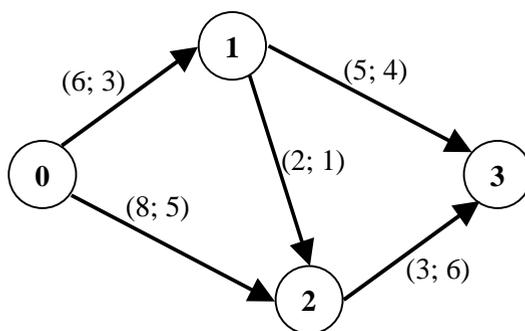


Рис. 1.8.

**2 шаг.** Чтобы сократить продолжительность проекта теперь следует сократить продолжительность всех критических путей. Для этого необходимо определить множество работ, таких что каждый критический путь содержит хотя бы одну работу из этого множества и сумма коэффициентов  $k_i$  является минимальной. Это задача эквивалентна задаче определения разреза в сети, имеющего минимальную пропускную способность, которая является двойственной к задаче о потоке максимальной величины (коэффициенты  $k_i$  определяют пропускные способности дуг, [1]). В нашем примере непосредственным перебором можно убедиться, что уменьшение продолжительностей работ (0, 1) и (0, 2) дает минимальное увеличение стоимости проекта (8 единиц на каждую единицу уменьшения продолжительности. Уменьшаем продолжительности работ (0, 1) и

(0, 2) на 3 единицы. Больше нельзя, т.к. минимальная продолжительность работы (0, 2) равна 5. Длина критического пути становится равной  $T_2 = 8$ , стоимость проекта  $S_2 = 41$ .

**3 шаг.** Теперь минимальное увеличение стоимости обеспечивается при уменьшении продолжительностей работ (0, 1) и (2, 3). Уменьшаем продолжительности работ (0, 1) и (2, 3) на единицу (при этом продолжительность работы (0, 1) становится минимальной). Длина критического пути  $T_3 = 7$ , стоимость проекта  $S_3 = 50$ .

**4 шаг.** Заметим, что в сети имеются всего 2 критических пути (рис. 1.9).

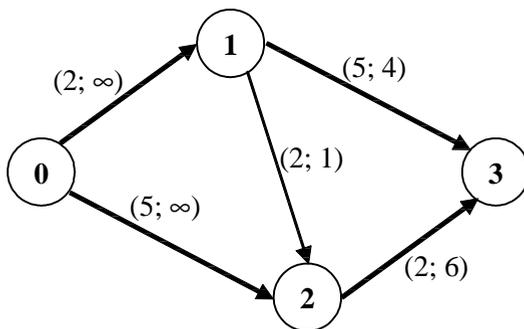


Рис. 1.9.

Сокращаем продолжительности работ (1, 3) и (2, 3) на 1. Продолжительность проекта становится равной  $T_4 = 6$ , стоимость проекта  $S_4 = 60$ .

## ГЛАВА 2. Двойная сетевая модель распределения ресурсов

В практических задачах обычно предполагается, что  $u(\tau)$  кусочно-постоянная непрерывная справа функция (рис. 2.1,а). В этом случае, интегрирование в (1.1.2) заменяется суммированием (рис. 2.1,б).

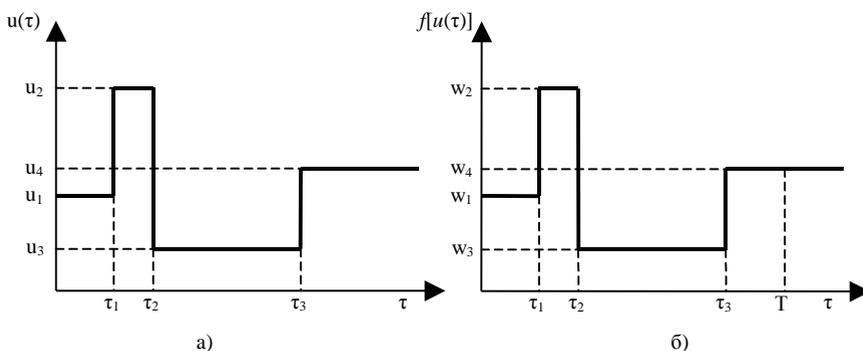


Рис. 2.1

Пусть  $k$  такое, что

$$\sum_{i=1}^k w_i \Delta \tau_i < W \leq \sum_{i=1}^{k+1} w_i \Delta \tau_i,$$

где  $w_i = f[u(\tau_i)]$ ;  $\Delta \tau_i = \tau_i - \tau_{i-1}$ . Тогда, момент окончания операции определяется по формуле

$$t = \frac{W - \sum_{i=1}^k w_i \Delta \tau_i}{w_{k+1}} + \tau_k \quad (2.1)$$

Выбор единицы ресурсов произволен (мы можем принять за единицу ресурсов одного человека или 10 человек, или два станка и т.д.). Важным является понятие ресурсов одного вида. Так называются ресурсы, которые неразличимы в условиях данной задачи по их влиянию на скорость выполнения операции (при соответствующем выборе единицы измерения). При этом физическая природа ресурсов может быть различной, хотя чаще всего ресурсы различной физической природы принадлежат к разным видам. Мы будем рассматривать задачи, в которых каждая операция может выполняться только ресурсами одного вида. Таким образом, множество операций комплекса разбивается на классы. Операции одного класса могут выполняться только ресурсами соответствующего вида. Рассмотрим подробнее процесс выполнения комплекса операций [2]. После выполнения одной операции ресурсы перемещаются на другие операции своего класса (образуют поток по множеству операций). Может случиться, что перемещение ресурсов с одной операции на другую недопустимо по тем или иным причинам (отсутствие транспортных средств, высокая стоимость или невозможность перемещения данного вида ресурсов и т.д.). Определим граф перемещений ресурсов. Он состоит из  $k$  компонент (по числу классов операций). Вершины графа соответствуют операциям, от вершины  $i$  идет дуга к вершине  $j$ , если возможно перемещение ресурсов от  $i$ -й операции на  $j$ -ю операцию. Кроме того, каждой дуге  $(i, j)$  ставят в соответствие время  $\vartheta_{ij}$  перемещение ресурсов от  $i$ -й операции на  $j$ -ю. На рис. 2.2 показан граф перемещений ресурсов (будем писать в дальнейшем граф ПР) для сети рис. 2.3. Он состоит из двух компонент, так как в комплекс входят работы двух классов. Отметим, что сеть рис. 2.3 является сопряженной, т. е. операциям комплекса со-

ответствуют вершины сети, а дуги отражают зависимости между операциями. Такое изображение более удобно, так как в графе ПР

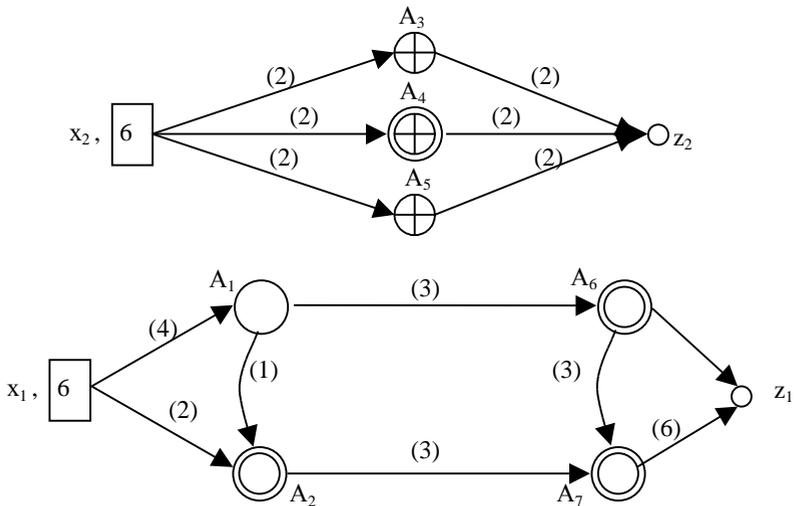


Рис. 2.2

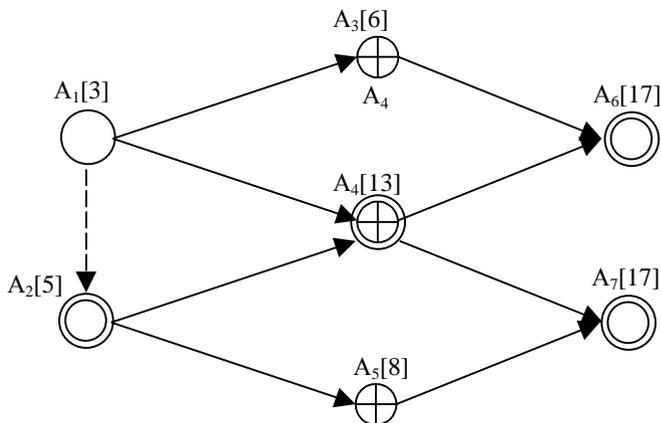


Рис. 2.3

вершины также соответствуют операциям. К первому классу относятся операции  $A_1, A_2, A_6, A_7$ , ко второму —  $A_3, A_4, A_5$ . Числа в скобках равны потоку ресурсов по соответствующей дуге.

В квадратах  $x_1, x_2$  каждой компоненты графа ПР пишется количество ресурсов  $N_1, N_2$ , предназначенных для выполнения операций соответствующего класса ( $N_1 = N_2 = 6$  на рис. 2.2). Фиктивные вершины  $x_1, x_2$  могут соответствовать некоторым пунктам, в которых находятся ресурсы. В свою очередь,  $z_1, z_2$  могут соответствовать пунктам, в которые нужно собрать ресурсы после выполнения комплекса. Определив некоторый поток ресурсов по графу ПР, можно найти момент окончания каждой операции и, следовательно, время выполнения всего комплекса.

Фронтом операций в момент  $i$  называется множество  $F(i)$  операций, которые выполняются или могут выполняться в этот момент. Основная группа алгоритмов для решения задач распределения ресурсов основана на последовательном получении решения путем

распределения ресурсов по операциям фронта. Для этого определяется некоторое правило (или несколько правил), позволяющее в любой момент времени принимать решение о распределении ресурсов по операциям фронта. В дальнейшем будем называть момент перераспределения ресурсов конфликтной ситуацией. Процесс разрешения конфликтных ситуаций удобно изображать графически в виде дерева. Вместо таких правил можно определять некоторую функцию (функция предпочтения) и выбирать распределение ресурсов, при котором эта функция принимает минимальное (максимальное) значение. Приведем два простых правила, которые часто применяются в алгоритмах такого типа.

**Правило I.** В первую очередь выполняются операции с меньшим полным резервом времени (резерв времени определяется при условии достаточного количества ресурсов).

**Правило II.** В первую очередь выполняются операции с меньшей длительностью.

Если для получения решения используются эвристические правила (взятые из интуитивных соображений или на основании опыта), то целесообразно испробовать различные правила (или системы правил), выбирая затем наилучшее решение.

Отметим, что большинство правил эквивалентно заданию некоторой функции предпочтения. Например, распределение ресурсов, полученное по правилу I, минимизирует функцию

$$\sum_{i \in R(t)} u_i(t) \Delta \tau_i(t), \text{ где } R(t) \text{ — множество номеров операций фронта,}$$

$\Delta \tau_i(t)$  – полный резерв  $i$ -й операции в момент  $t$ ,  $u_i(t)$  – количество ресурсов расходуемых в  $i$ -й операции. Распределение, полученное

по правилу II, минимизирует функцию  $\sum_{i \in R(t)} u_i(t) \tau_i$ , где  $\tau_i$  — время выполнения операции и т.д.

В некоторых случаях удобно в качестве функции предпочтения взять нижнюю границу времени выполнения комплекса при выбранном распределении ресурсов по операциям фронта. При этом если мы уже получили какое-либо решение, а значение функции предпочтения на остальных вершинах дерева решений больше или равно времени выполнения комплекса для полученного решения, то, очевидно, полученное решение оптимально.

### **2.1. Определение моментов окончания операций при заданном потоке ресурсов по графу ПР**

Рассмотрим на примере определение моментов окончания операций при заданном потоке ресурсов. Примем, что время перемещения ресурсов с операции на операцию равно нулю. Кроме того, примем, что не разрешается снимать ресурсы с операции, пока она не закончена (отказ от этих предположений несущественно меняет методику расчета). Пусть скорость выполнения операции прямо пропорциональна количеству ресурсов, т. е. примем  $w_i = u_i$ ,  $i=1, 2, 3, 4, 5, 6, 7$  (рис. 2.2, 2.3). Объемы операции следующие:

|                |    |    |   |    |   |    |    |
|----------------|----|----|---|----|---|----|----|
| Номер операции | 1  | 2  | 3 | 4  | 5 | 6  | 7  |
| $W_i$          | 12 | 12 | 6 | 16 | 6 | 12 | 12 |

Определение моментов окончания операций производится последовательным просчетом сети. Пусть  $A_{i_1}, A_{i_2}, A_{i_3}$  – операции, непосредственно предшествующие операции  $A_i$ ;  $t_{i_1}^1, t_{i_2}^1, t_{i_3}^1$  – моменты окончания этих операций. Тогда  $t_i^0 = \max(t_{i_1}^1, t_{i_2}^1, t_{i_3}^1)$  – возможный момент начала  $i$ -й операции. Далее, пусть  $A_{i_4}, A_{i_5}, A_{i_6}$  – операции, с которых перемещаются ресурсы на операцию  $A_i$ ;  $t_{i_4}^1, t_{i_5}^1, t_{i_6}^1$  – моменты окончания этих операций (соответственно моменты прихода ресурсов на  $i$ -ю операцию, если времена перемещения равны нулю). Для определения момента окончания  $i$ -й операции применяем формулу (2.1), отсчитывая интервалы  $\Delta\tau_i$ , с момента  $t_i^0$ .

На рис. 2.2 показан некоторый поток ресурсов по графу ПР. Будем обозначать  $Q_i$  множество операций, непосредственно предшествующих операции  $A_i$ ;  $P_i$  – множество операций, с которых перемещаются ресурсы на операцию  $A_i$ .

1) Операция  $A_1$ .  $Q_1 = \emptyset, P_1 = \emptyset, u_1(t) = 4,$

$$t_1^1 = \frac{W_1}{u_1} = 3.$$

2) Операция  $A_2$ .  $Q_2 = \emptyset, P_2 = \{A_1\}.$

Применяя формулу (2.1), получаем

$$t_2^1 = t_1^1 + \frac{W_2 - 2(t_1^1 - t_2^0)}{3} = 3 + \frac{12 - 6}{3} = 5$$

3) Операция  $A_3$ .  $Q_3 = \{A_1\}, P_3 = \emptyset, t_3^0 = t_1^1 = 3,$

$$u_3(t) = 2, \quad t_3^1 = t_3^0 + \frac{W_3}{2} = 6$$

4) Операция  $A_4$ .  $Q_4 = \{A_1, A_2\}$ ,  $P_4 = \emptyset$ ,  $t_4^0 = \max(t_1^1, t_2^1) = t_2^1 = 5$ ,

$$u_4(t) = 2, \quad t_4^1 = t_4^0 + \frac{W_4}{2} = 13,$$

5) Операция  $A_5$ .  $Q_5 = \{A_2\}$ ,  $P_5 = \emptyset$ ,  $t_5^0 = t_2^1 = 5$ ,

$$u_5(t) = 2, \quad t_5^1 = t_5^0 + \frac{W_5}{2} = 8.$$

6) Операция  $A_6$ .  $Q_6 = \{A_3, A_4\}$ ,  $P_6 = \{A_1\}$ ,  $t_6^0 = \max(t_3^1, t_4^1) = 13$ ,

$$u_6(t) = \begin{cases} 0, & t < 3 \\ 3, & t \geq 3 \end{cases}, \quad t_6^1 = t_6^0 + \frac{W_6}{3} = 17,$$

7) Операция  $A_7$ .  $Q_7 = \{A_4, A_5\}$ ,  $P_7 = \{A_2, A_6\}$ ,  
 $t_7^0 = \max(t_4^1, t_5^1) = 13$ ,

$$\text{Имеем } t_7^1 = t_7^0 + \Delta\tau_1 + \frac{W_7 - 3\Delta\tau_1}{6} = 17,$$

Моменты окончания операции указаны в скобках на рис. 2.3.

Время выполнения комплекса  $T = \max_i t_i^1 = 17$ .

Теперь появляется возможность улучшить решение, изменив поток ресурсов (ресурсы с операций, имеющих большие резервы, перебрасываются на критические или близкие к ним операции). Уменьшим, например, потоки ресурсов через вершины  $A_3$ ,  $A_5$  графа ПР на единицу и увеличим поток через вершину  $A_4$  на две единицы (рис. 2.4). При этом время выполнения комплекса уменьшилось до  $T = 14$  (рис. 2.5).

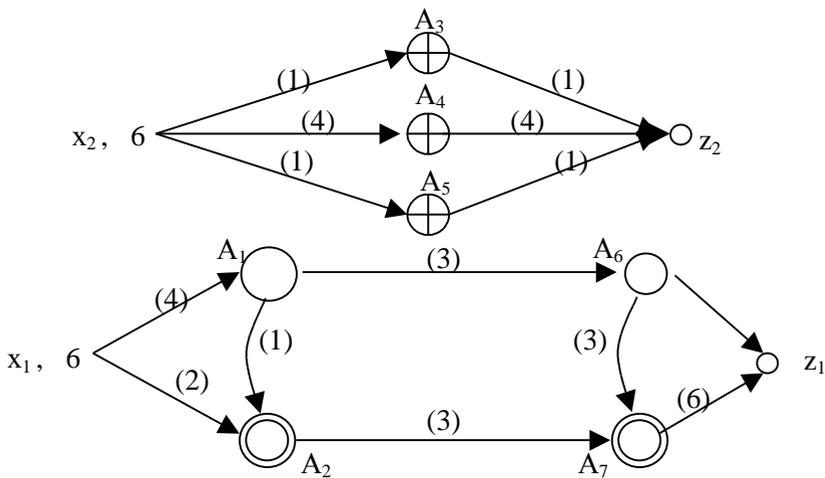


Рис. 2.4

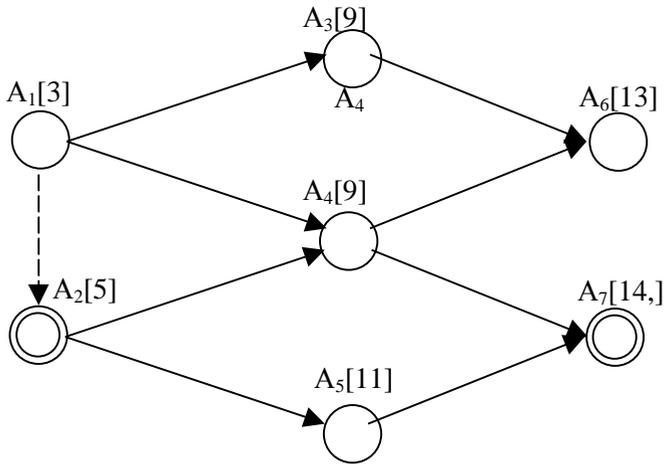


Рис. 2.5

Действительно, последовательно определяем

$$t_1^1 = \frac{W_1}{u_1} = \frac{12}{4} = 3$$

$$t_2^1 = t_1^1 + \frac{W_2 - 2 \cdot t_1^1}{3} = 5$$

$$t_3^1 = t_1^1 + \frac{W_3}{u_3} = 3 + \frac{6}{1} = 9$$

$$t_4^1 = \max(t_1^1; t_2^1) + \frac{W_4}{u_4} = 5 + \frac{16}{4} = 9$$

$$t_5^1 = t_2^1 + \frac{W_5}{u_5} = 5 + \frac{6}{1} = 11$$

$$t_6^1 = \max(t_3^1; t_4^1) + \frac{W_6}{u_6} = 9 + \frac{12}{3} = 13$$

$$t_7^1 = t_6^1 + \frac{W_7 - 3 \cdot 2}{6} = 14$$

Иногда решение задачи должно удовлетворять дополнительному условию: количество ресурсов на операции не меняется в процессе ее выполнения. Такое условие позволяет упростить процедуру.

Действительно, в этом случае время выполнения операции определяется по формуле

$$\tau = \frac{W}{w(u)},$$

где  $u$  – поток ресурсов, входящий в соответствующую вершину.

Теперь достаточно дополнить сетевой график недостающими дугами, по которым проходит ненулевой поток, и применить обычные алгоритмы определения критического пути. Для нашего примера (рис. 2.2) имеем (табл. 2.1):

Таблица 2.1

| Номер операции | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----------------|---|---|---|---|---|---|---|
| $u_i$          | 4 | 3 | 2 | 2 | 2 | 3 | 6 |
| $\tau_i$       | 3 | 4 | 3 | 8 | 3 | 4 | 2 |

Добавляя в сетевой график рис. 2.3 дуги  $(A_1, A_2)$  и  $(A_6, A_7)$ , получаем сеть (рис. 2.6), просчитывая которую обычным способом, определяем  $T=21$ . Увеличение времени выполнения комплекса по сравнению с предыдущим случаем ( $T=17$ ) вызвано запрещением изменять количество ресурсов в процессе выполнения операции. Определим критический путь в случае потока ресурсов, изображенного на рис. 2.4.

Продолжительности операций указаны в табл. 2.2.

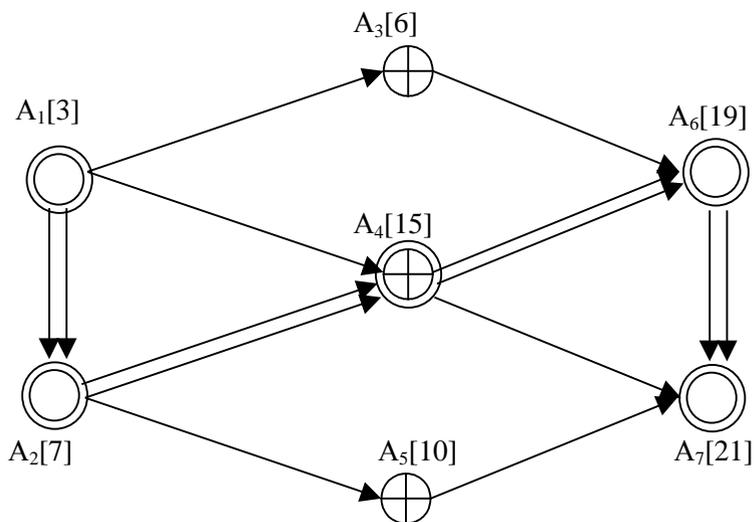


Рис. 2.6

Таблица 2.2

|          |   |   |   |   |   |   |   |
|----------|---|---|---|---|---|---|---|
| $i$      | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $u_i$    | 4 | 3 | 1 | 4 | 1 | 3 | 6 |
| $\tau_i$ | 3 | 4 | 6 | 4 | 6 | 4 | 2 |

Сетевой график с поздними моментами окончания операций приведен на рис. 2.7.

Можно предложить также другой способ определения времени выполнения комплекса в случае запрещения изменять количество ресурсов на операции в процессе ее выполнения.

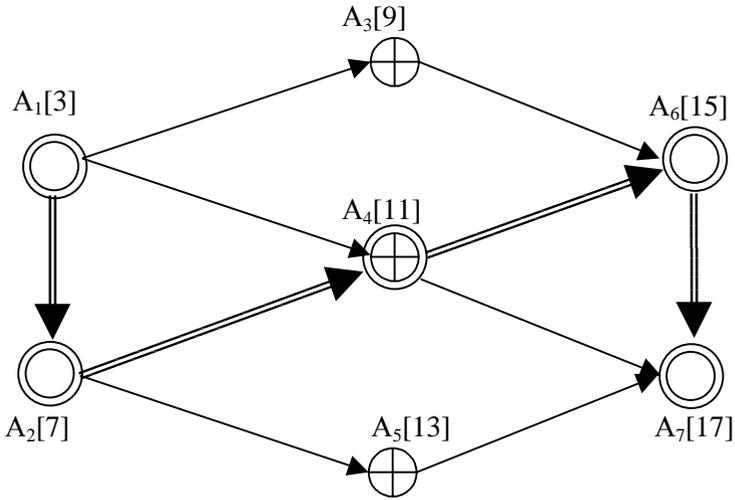


Рис. 2.7

Рассмотрим зависимость  $w(t)$  (рис. 2.8), начиная с момента возможного начала операции  $\tau$  (в момент  $\tau$  мы могли бы начать операцию, если бы имелось достаточное количество ресурсов). Если начать операцию в момент  $t_j$ , то она будет выполняться со скоростью  $w_j$  и момент окончания  $t_i = t_j + \frac{W_i}{w_j}$ . Естественно определить

момент начала операции так, чтобы момент окончания был минимальным, т. е. момент окончания определяется по формуле

$$t_i = \min_j \left( t_j + \frac{W_i}{w_j} \right) = t_{j_0} + \frac{W_i}{w_{j_0}}.$$

Для примера рис. 2.8 имеем в случае  $W_i = 36$

$$t_i = \min (36, 15, 13, 14 \cdot 1/7, 191/2) = 13.$$

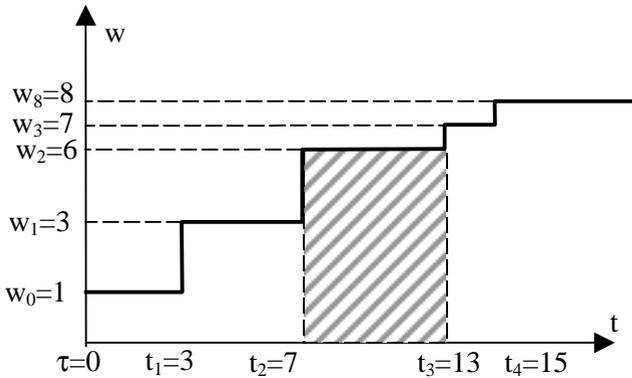


Рис. 2.8

При таком способе определения  $t_i$  добавочные дуги следует проводить в сети только от тех операций, ресурсы с которых перемещаются на  $i$ -ю операцию в моменты времени не позднее  $t_{j_0}$ . Ресурсы, приходящие позже, не принимают участия в выполнении операции и сразу перемещаются на следующие операции. Заметим, что этот способ определения моментов окончания операций можно рассматривать как простейший алгоритм оптимизации распределения ресурсов. Рассмотрим, например, поток рис. 2.4. Для операции  $A_2$  имеем

$$t_2 = \min\left(0 + \frac{12}{2}; 3 + \frac{12}{3}\right) = 6$$

т. е. более выгодно выполнять операцию  $A_2$  двумя единицами ресурсов, не ожидая прихода третьей. Более того, эту третью единицу можно направить теперь на операцию  $A_6$  уменьшив время ее выполнения до  $\tau_6=3$ . Такое преобразование потока ресурсов уменьшает время выполнения комплекса в нашем примере с 17 до 15.

## 2.2. Постановка задачи и обсуждение модели

После введения основных понятий и определения графа ПР мы в состоянии четко поставить задачу оптимального распределения ресурсов.

**Задача.** Задана сетевая модель комплекса из  $n$  операций, в которую входит:

- 1) сетевой график;
- 2) граф ПР;
- 3) матрица  $\|\vartheta_{ij}\|$ , где  $\vartheta_{ij}$  — время перемещения ресурсов с  $i$ -й операции на  $j$ -ю;
- 4) зависимость  $w_i=f_i(u_i)$  скорости выполнения  $i$ -ой операции от количества ресурсов соответствующего вида (предполагаем, что  $f_i$  — неубывающие функции  $u_i$ );
- 5) количество ресурсов  $N_j$   $j$ -го вида ( $j=1, 2, \dots, k$ ) где  $k$  — число классов операций.

Требуется определить поток ресурсов по графу ПР, минимизирующий время выполнения комплекса, либо упущенную выгоду.

Описанная модель охватывает довольно большой круг практических задач. Отметим лишь, что ее частными случаями являются такие известные задачи, как «задача коммивояжера», задача определения оптимального порядка обработки деталей на станках и др. Однако следует отметить, что некоторые черты практических задач не отражены в нашей модели. В свое оправдание скажем лишь, что, во-первых, «нельзя объять необъятное» (особенно в такой маленькой книге), во-вторых, в такой упрощенной форме модель позволяет решать; практически интересные задачи и, в-третьих, методы,

которые мы будем рассматривать, могут служить основой более сложных постановок.

В следующих параграфах мы рассмотрим некоторые алгоритмы распределения ресурсов для частных случаев.

### **2.3. Оптимизация календарного графика с учётом времени перемещений бригад**

Рассмотрим комплекс из  $n$  работ, выполнение которых происходит в пунктах, расположенных друг от друга на заданных расстояниях. Все работы выполняются одной бригадой. Обозначим через  $\tau_i$  – продолжительность  $i$ -ой работы,  $D_i$  – заданный срок завершения  $i$ -ой работы,  $\ell_{ij}$  – время перемещения бригады из пункта  $i$  в пункт  $j$  ( $\ell_{0i}$  – время перемещения бригады от места её расположения в пункт  $i$ ).

**Постановка задачи.** Определить очерёдность выполнения работ, обеспечивающую их завершение не позже заданных сроков. Если это невозможно, то минимизировать максимальное запаздывание сверх заданных сроков, то есть минимизировать

$$\max_i (t_i - D_i),$$

где  $t_i$  – момент завершения  $i$ -ой работы. Задача является NP-трудной, поскольку её частным случаем является известная задача Коммивожера.

Рассмотрим частный случай задачи, когда все пункты расположены в линию (например, вдоль железнодорожного пути или автострады) (рис. 2.9). В этом случае

$$\ell_{ij} = |q_j - q_i|,$$

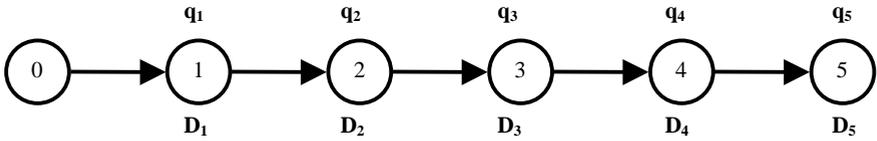


Рис. 2.9.

где  $q_j$  – время переезда бригады из начального пункта 0 в пункт  $j$ .

Получим оценку снизу  $C_{in}$  момента завершения работы  $i$  при условии, что она выполняется в последнюю очередь. Для этого необходимо определить длину кратчайшего пути из пункта 0 в пункт  $i$ , проходящего через все остальные пункты. Эта длина равна

$$L_i(n) = 2q_n - q_i. \quad (2.2)$$

Оценка снизу

$$C_{in} = \sum_{i=1}^n \tau_i + L_i(n). \quad (2.3)$$

Определим множество  $Q$  работ, для которых  $C_{in} \leq D_i$ . Если это множество пустое, то определяем

$$\varepsilon = \min_i (C_{in} - D_i) \quad (2.4)$$

и множество  $Q$  работ, для которых  $C_{in} - D_i = \varepsilon$ .

Выбираем любую работу  $k$  из множества  $Q$ . Для каждой из оставшихся работ  $i$  определяем оценку снизу  $C_{i, n-1}$  её завершения при условии, что эта работа выполняется предпоследней. Для этого определяем кратчайший путь  $L_i(n-1)$  из пункта 0 в пункт  $i$ , проходящей через все пункты за исключением  $k$ -го. Оценка снизу

$$C_{i, n-1} = L_i(n-1) + \sum_{i \neq k} \tau_i \quad (2.5)$$

Снова определяем множество  $Q$  и выбираем любую работу из этого множества и т.д. пока не получаем допустимого решения.

**Пример 2.1.** Пусть имеются пять работ. Величины  $q_i$ ,  $\tau_i$  и  $D_i$  приведены в табл. 2.3.

Таблица 2.3.

| $i$      | 1 | 2 | 3  | 4  | 5  |
|----------|---|---|----|----|----|
| $q_i$    | 1 | 2 | 3  | 4  | 5  |
| $\tau_i$ | 3 | 2 | 1  | 2  | 4  |
| $D_i$    | 6 | 7 | 22 | 13 | 16 |

*1 шаг.* Определяем

$$C_{1n} = 9 + 12 = 21, D_1 = 6,$$

$$C_{2n} = 8 + 12 = 20, D_2 = 7,$$

$$C_{3n} = 7 + 12 = \underline{19}, D_3 = \underline{22},$$

$$C_{4n} = 6 + 12 = 18, D_4 = 13,$$

$$C_{5n} = 5 + 12 = 17, D_5 = 16.$$

*2 шаг.* Множество  $Q$  состоит из одной вершины **3**. Исключаем эту вершину и определяем

$$C_{1n} = 9 + 11 = 20, D_1 = 6,$$

$$C_{2n} = 8 + 11 = 19, D_2 = 7,$$

$$C_{4n} = 6 + 11 = 17, D_4 = 13,$$

$$C_{5n} = 5 + 11 = \underline{16}, D_5 = \underline{16}.$$

*3 шаг.* Множество  $Q$  состоит из одной вершины **5**. Исключаем эту вершину и определяем

$$C_{1n} = 7 + 7 = 14, D_1 = 6,$$

$$C_{2n} = 6 + 7 = 13, D_2 = 7,$$

$$C_{4n} = 4 + 7 = \underline{11}, D_4 = \underline{13}.$$

4 шаг. Множество Q состоит из одной вершины 4. Исключаем эту вершину и определяем

$$C_{12} = 3 + 5 = 8, D_1 = 6,$$

$$C_{22} = 2 + 5 = 7, D_2 = 7.$$

Множество Q состоит из двух одной вершины 2.

Окончательно получаем допустимое решение:

$$\pi = (1, 2, 4, 5, 3).$$

Описанный алгоритм может не дать оптимального решения.

Однако, оценки можно применить в методе ветвей и границ для получения оптимального решения задачи. Рассмотрим метод ветвей и границ на данных предыдущего примера, изменив значения сроков  $D_i$ , а именно, примем значения  $D_i$ , указанные в таблице 2.4.

Таблица 2.4

| i     | 1  | 2  | 3  | 4  | 5  |
|-------|----|----|----|----|----|
| $D_i$ | 11 | 16 | 18 | 13 | 10 |

1 шаг. Определяем

$$C_{15} = 9 + 12 = 21, \quad D_1 = 11$$

$$C_{25} = 8 + 12 = 20, \quad D_2 = 16$$

$$C_{35} = 7 + 12 = 19, \quad D_3 = 18$$

$$C_{45} = 6 + 12 = 18, \quad D_4 = 13$$

$$C_{55} = 5 + 12 = 17, \quad D_5 = 10$$

Минимальная величина  $C_{i5} - D_i = \min(10; 4; 1; 5; 7) = 1$  и достигается на третьей операции. Выберем подмножество решений, в котором последней выполняется третья операция.

2 шаг. Определяем

$$C_{14}=9+11=20, \quad D_1=11$$

$$C_{24}=8+11=19, \quad D_2=16$$

$$C_{44}=6+11=17, \quad D_4=13$$

$$C_{54}=5+11=16, \quad D_5=10$$

$$\min_{i \neq 3} (C_{i4} - D_i) = \min(9; 3; 4; 6) = C_{24} - D_2 = 3$$

Выбираем подмножество, в котором предпоследней выполняется вторая операция.

*3 шаг.* Определяем

$$C_{13}=9+9=18, \quad D_1=11$$

$$C_{43}=6+9=15, \quad D_4=13$$

$$C_{53}=5+9=14, \quad D_5=10$$

$$\min_{i \neq 3,2} (C_{i3} - D_i) = \min(7; 2; 4) = C_{43} - D_4 = +2$$

Выбираем подмножество, в котором третьей выполняется четвертая операция.

*4 шаг.* Определяем

$$C_{12}=9+7=16, \quad D_1=11$$

$$C_{52}=5+7=12, \quad D_5=10$$

$$\min_i (C_{i2} - D_i; C_{52} - D_5) = \min(5; 2) = C_{52} - D_5 = 2$$

Выбираем подмножество, в котором второй выполняется пятая операция. Дерево ветвлений приведено на рис. 2.10.

Полученное решение  $\pi=(1; 5; 4; 2; 3)$  является оптимальным, поскольку оценки снизу всех остальных подмножеств больше 3. Для приближенного решения задачи можно применить и метод локальной оптимизации.

Сначала, пользуясь каким либо эвристическим правилом, получаем допустимое решение.

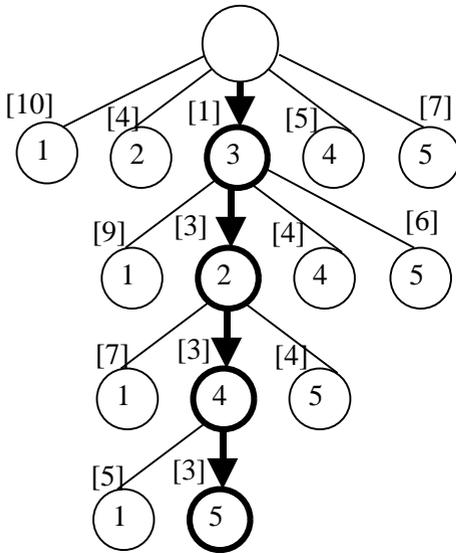


Рис. 2.10

Так, например, хорошим эвристическим правилом, как показало решение большого числа примеров, является выполнение операции в очередности возрастания  $D_i$ . В нашем примере это правило дает следующее решение

$$\pi_0 = (5; 1; 4; 2; 3)$$

со значением целевой функции

$$F = \max(9 - 10; 16 - 11; 21 - 13; 25 - 16; 27 - 18) = 9$$

Рассмотрим множество соседних перестановок, полученных транспозицией начальной перестановки  $\pi_0$ . Имеем

$$\pi_1 = (1; 5; 4; 2; 3), \quad F_1 = 3$$

$$\pi_2 = (5; 4; 1; 2; 3), \quad F_2 = 7$$

$$\pi_3=(5; 1; 2; 4; 3), \quad F_3=11$$

$$\pi_4=(5; 1; 4; 3; 2), \quad F_4=10$$

Лучшее решение  $\pi_1$  является оптимальным.

Описанный подход можно применить и к ряду других схем расположения операций.

Пусть, например, все операции которые предстоит выполнить, расположены вдоль кольцевой дороги (см. рис. 2.11)

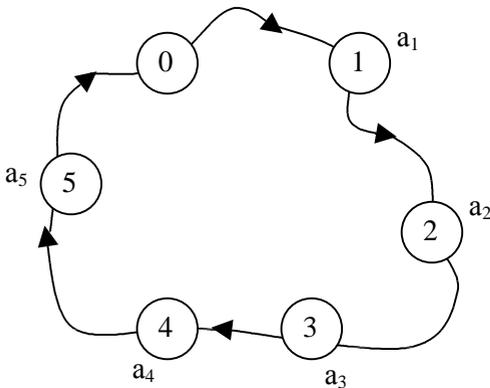


Рис. 2.11.

В случае одностороннего движения оценка  $C_{in}$  определяется следующим выражением

$$L_i(n) = L + q_i, \quad i \neq n$$

$$L_n(n) = q_{ni}$$

где  $L$  – длина кольцевой дороги.

В случае двустороннего движения оценка  $C_{in}$  получается более сложным образом, поскольку возможны различные варианты

выполнения всех операций так чтобы операция  $i$  выполнялась последней.

*1 вариант.* Выполняем последовательно операции с 1 по  $n$  (исключая операцию  $i$ ), а затем выполняем операцию  $i$ . В этом случае оценка снизу будет равна

$$L_i(n) = 2q_n - q_i$$

*2 вариант.* Выполняем последовательно операции с 1 по  $(i-1)$ , а затем операции с  $n$  по  $i$  (в обратном порядке). Оценка снизу будет равна

$$L_i(n) = 2q_{i-1} + L - q_i$$

Если варианты 1 и 2 проделать в обратном порядке, то получим еще две оценки

$$L_i(n) = L + q_i - 2q_i$$

$$L_i(n) = 2(L - q_{i+1}) + q_i$$

Окончательная оценка снизу равна минимальному из полученных чисел

Рассмотрим решение примера 2.1 полагая, что пункт 5 соединен с начальным пунктом 0 путем длины 1.

Сначала рассмотрим случай одностороннего движения.

*1 шаг.* Вычисляем ( $L=6$ )

$$C_{15}=7+12=19, \quad D_1=11, \quad F_1=8$$

$$C_{25}=8+12=20, \quad D_2=16, \quad F_2=4$$

$$C_{35}=9+12=21, \quad D_3=18, \quad F_3=3$$

$$C_{45}=10+12=22, \quad D_4=13, \quad F_4=9$$

$$C_{55}=5+12=17, \quad D_5=10, \quad F_5=7$$

Выполним последней операцией 3

*2 шаг.* Вычисляем

$$C_{14}=7+11=18, \quad D_1=11, \quad F_1=7$$

$$C_{24}=8+11=19, \quad D_2=16, \quad F_2=3$$

$$C_{44}=10+11=21, \quad D_4=13, \quad F_4=8$$

Здесь следует учесть, что выполняется операция, которая расположена ближе к начальному пункту, чем операция 3. Поэтому момент завершения операции 3 увеличится до  $2L + q_3 + T = 27$ , и следовательно оценка снизу этого варианта  $F_4=9$ . Для операции 5 имеем

$$C_{54}=5+11=16, \quad D_5=10, \quad F_5=6$$

Выполняем предпоследней операцию 2.

*3 шаг.* Вычисляем

$$C_{13}=14+9=16, \quad D_1=11, \quad F_1=5$$

$$C_{43}=15+9=19, \quad D_4=13, \quad F_4=6$$

Однако при этом

$$C_{24}=14+11=25, \quad D_2=16, \quad F_2=9$$

$$C_{35}=15+12=27, \quad D_3=18, \quad F_3=9$$

Так, что оценка снизу этого варианта равна 9.

$$C_{53}=5+9=14, \quad D_5=10, \quad F_5=4$$

Выполняем третьей операцией 5

*4 шаг.* Вычисляем

$$C_{12}=7+5=12, \quad D_1=11, \quad F_1=1$$

при этом,  $C_{53}$ ,  $C_{24}$  и  $C_{35}$  увеличиваются на длину кольца, то есть на шесть, что дает соответствующее увеличение оценки снизу до 10.

$$C_{42}=4+5=9, \quad D_4=13, \quad F_4=-4$$

Выбираем вариант, в котором второй выполняется операция 4, а первой операция 1. Окончательно получаем решение  $\pi=(1; 4; 5; 2; 3)$  со значением целевой функции

$$F = \max(4 - 11; 9 - 13; 14 - 10; 19 - 16; 21 - 18) = 4$$

Дерево ветвлений приведено на рис. 2.12. Полученное решение является оптимальным, так как нижние оценки остальных подмножеств не менее 4.

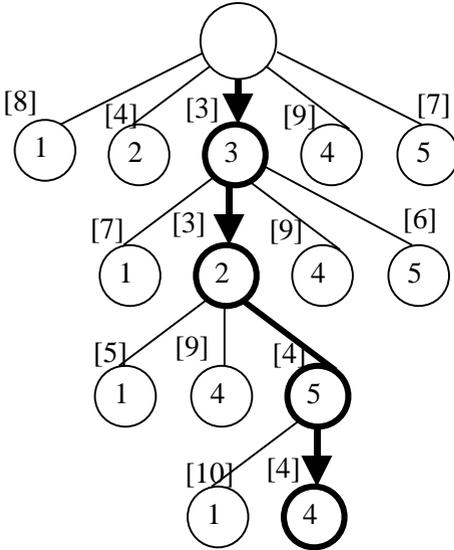


Рис. 2.12

Рассмотрим теперь случай двустороннего движения.

1 шаг. Вычисляем ( $L=6$ )

|             |                   |           |         |
|-------------|-------------------|-----------|---------|
| $L_1(5)=5,$ | $C_{15}=5+12=17,$ | $D_1=11,$ | $F_1=6$ |
| $L_2(5)=6,$ | $C_{25}=6+12=18,$ | $D_2=16,$ | $F_2=2$ |
| $L_3(5)=7,$ | $C_{35}=7+12=19,$ | $D_3=18,$ | $F_3=1$ |
| $L_4(5)=6,$ | $C_{45}=6+12=18,$ | $D_4=13,$ | $F_4=5$ |
| $L_5(5)=5,$ | $C_{55}=5+12=17,$ | $D_5=10,$ | $F_5=7$ |

Выполним последней операцию 3

2 шаг. Вычисляем

$$\begin{array}{llll} L_1(4)=5, & C_{14}=5+11=16, & D_1=11, & F_1=5 \\ L_2(4)=6, & C_{24}=6+11=17, & D_2=16, & F_2=1 \\ L_4(4)=6, & C_{44}=6+11=17, & D_4=13, & F_4=4 \\ L_5(4)=5, & C_{54}=5+11=16, & D_5=10, & F_5=6 \end{array}$$

Четвертой выполним операцию 2

3 шаг. Вычисляем

$$\begin{array}{llll} L_1(3)=5, & C_{13}=5+9=14, & D_1=11, & F_1=3 \\ L_4(3)=4, & C_{43}=4+9=13, & D_4=13, & F_4=0 \\ L_5(3)=5, & C_{53}=5+9=14, & D_5=10, & F_5=4 \end{array}$$

при этом, увеличив значения  $L_2(4)=8$  и  $L_3(5)=9$ , что приводит к увеличению оценок  $C_{24}$  и  $C_{35}$ . Выполняем третьей операцию 4

4 шаг. Вычисляем

$$L_1(2)=3, \quad C_{12}=3+7=10, \quad D_1=11, \quad F_1=-1$$

при этом, увеличивается на 2 единицы оценки  $L_4(3)$ ,  $L_2(4)$  и  $L_3(5)$ , что дает оценку снизу  $F_1=3$ .

$$L_5(2)=3, \quad C_{52}=3+7=10, \quad D_5=10, \quad F_5=0$$

Окончательно получаем вариант  $\pi=(1; 5; 4; 2; 3)$  со значением критерия

$$F = \max(4 - 11; 10 - 10; 13 - 13; 17 - 16; 19 - 18) = 1$$

Это решения является оптимальным.

## 2.4. Оптимизация календарного графика для радиальной транспортной схемы

Выше мы рассмотрели алгоритм построения оптимальных планов перемещения бригады для линейной и кольцевой транс-

портных схем. Рассмотрим еще один частный случай, когда транспортная схема является радиальной (см. рис. 2.13.)

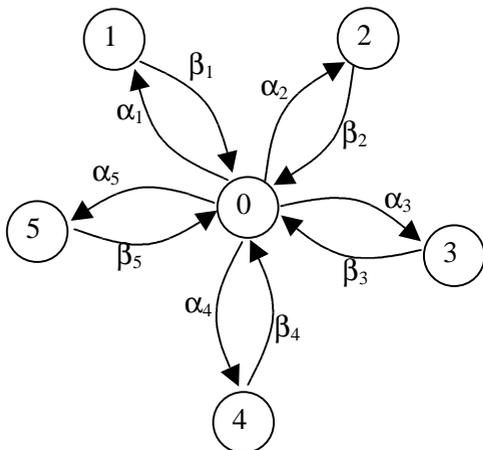


Рис. 2.13.

Заметим, что время  $\alpha_i$  перемещения из начального пункта в пункт  $i$ , где выполняется работа  $i$ , в общем случае не равно времени  $\beta_i$  возвращения в начальный пункт. Дело в том, что  $\alpha_i$  может включать время на подготовительные работы, подбор инструмента и т.д., а  $\beta_i$  может включать время на подготовку техники и инструмента к отъезду. Таким образом, время перехода бригады от операции  $i$  к операции  $j$  равно

$$I_{ij} = \beta_i + \alpha_j \quad (2.6)$$

Такой граф называется псевдопотенциальным [1]. Любой его гамильтонов контур имеет одну и ту же длину

$$L = \sum_i (\alpha_i + \beta_i)$$

Таким образом, продолжительность выполнения всех операций одной бригадой равна

$$T = \sum_i (\alpha_i + \beta_i + \tau_i) \quad (2.7)$$

с учетом времени возвращения бригады в начальный пункт. Рассмотрим задачу определения очередности выполнения операций минимизирующей

$$\eta = \max_i (t_i - D_i) \quad (2.6)$$

где  $t_i$  – момент завершения  $i$ -ой операции;  $D_i$  – желательный срок завершения  $i$ -ой операции. Пусть  $\pi=(i_1, i_2, \dots, i_n)$  очередность выполнения операций. Тогда

$$t_{i_k} = \sum_{j=1}^{k-1} (\alpha_{i_j} + \beta_{i_j} + \tau_{i_j}) + \alpha_{i_k} + \tau_{i_k} = \sum_{j=1}^k (\alpha_{i_j} + \beta_{i_j} + \tau_{i_j}) - \beta_{i_k} \quad (2.7)$$

Из (2.6) и (2.7) получаем

$$\sum_{j=1}^k (\alpha_{i_j} + \beta_{i_j} + \tau_{i_j}) \leq \beta_{i_k} + D_{i_k} + \eta, \quad k = \overline{1, n} \quad (2.8)$$

Покажем, что оптимальным является выполнение операций в очередности возрастания величин  $(\beta_i + D_i)$ .

Пусть в решении  $\pi$  имеет место

$$\beta_{i_q} + D_{i_q} > \beta_{i_{q+1}} + D_{i_{q+1}}$$

Поменяем очередность выполнения операций  $i_q$  и  $i_{q+1}$ , то есть сначала выполняем операцию  $i_{q+1}$ , а затем  $i_q$ .

Покажем, что в новом решении неравенства (2.8) будут выполняться при той же величине  $\eta$ .

Имеем

$$\sum_{j=1}^{q-1} (\alpha_{i_j} + \beta_{i_j} + \tau_{i_j}) + (\alpha_{i_{q+1}} + \beta_{i_{q+1}} + \tau_{i_{q+1}}) \leq \sum_{j=1}^{q+1} (\alpha_{i_j} + \beta_{i_j} + \tau_{i_j}) \leq \beta_{i_{q+1}} + D_{i_{q+1}} + \eta$$

$$\sum_{j=1}^{q+1} (\alpha_{i_j} + \beta_{i_j} + \tau_{i_j}) \leq \beta_{i_{q+1}} + D_{i_{q+1}} + \eta < \beta_{i_q} + D_{i_q} + \eta$$

Таким образом, всегда существует оптимальное решение, в котором операции выполняются в очередности возрастания (неубывания)  $p_i = \beta_i + D_i$ .

**Пример 2.2.** Данные об операциях приведены в табл. 2.5.

Таблица 2.5

| i          | 1  | 2 | 3  | 4  | 5  | 6  |
|------------|----|---|----|----|----|----|
| $\alpha_i$ | 3  | 4 | 6  | 5  | 1  | 2  |
| $\beta_i$  | 5  | 2 | 4  | 8  | 3  | 1  |
| $D_i$      | 10 | 4 | 23 | 34 | 50 | 57 |
| $p_i$      | 15 | 6 | 27 | 42 | 53 | 58 |
| $\tau_i$   | 3  | 2 | 1  | 4  | 5  | 3  |

Оптимальная очередь выполнения операций  $\pi=(2; 1; 3; 4; 5; 6)$ . Величина задержки сверх желательных сроков

$$\eta = \max(6 - 4; 14 - 10; 26 - 23; 39 - 34; 53 - 50; 61 - 57) =$$

$$= \max(2; 4; 3; 5; 3; 4) = 5$$

Пусть теперь число бригад равно  $m > 1$ . Рассмотрим задачу минимизации времени выполнения всех операций. Обозначим через  $Q_k$  множество операций, выполняемых  $k$ -ой бригадой. Время выполнения  $T_k$  согласно (2.7) составит

$$T_k = \sum_{i \in Q_k} (\alpha_i + \beta_i + \tau_i) = \sum_{i \in Q} C_i \quad (2.7)$$

где  $C_i = \alpha_i + \beta_i + \tau_i$ . Время выполнения всех операций равно

$$T = \max_k T_k \quad (2.9)$$

Задача заключается в разбиении всех операций на  $m$  групп, так чтобы величина критерия (2.9) была минимальной. Это известная «задача о камнях», которая относится к сложным, комбинаторным задачам.

Рассмотрим методы ее решения для крайних случаев, когда число бригад «велико» и когда число бригад «мало»..

Пусть число бригад равно 2. В этом случае эффективным является метод динамического программирования. Рассмотрим его на примере.

***Пример 2.3.***

|            |   |   |   |   |   |
|------------|---|---|---|---|---|
| $i$        | 1 | 2 | 3 | 4 | 5 |
| $\alpha_i$ | 8 | 5 | 9 | 4 | 6 |

Заметим, что  $\sum_i C_i = 32$ . Следовательно, необходимо определить множество работ,  $Q$ , выполняемых первой бригадой, такое чтобы величина  $\sum_{i \in Q} C_i$  была возможно ближе к 16. Построим систему координат, на одной оси которой отметим номера операций, а на другой время работы первой бригады (рис. 2.14).

Рассматриваем первую операцию. Если она выполняется первой бригадой, то проводим наклонную линию в точку с координатами

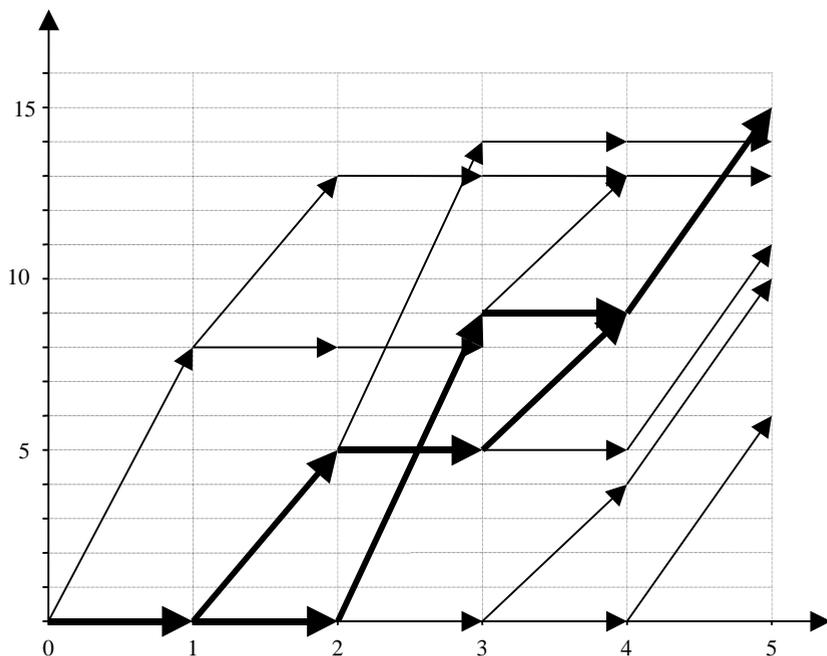


Рис. 2.14

(1;  $c_1$ )=(1; 8). Если она выполняется второй бригадой, то проводим горизонтальную линию в точку с координатами (1; 0). Из каждой полученной точки проводим две линии (наклонную и горизонтальную) в зависимости от того какая бригада выполняет вторую операцию, и т.д. Получаем сеть, приведенную на рис. 2.14. Положим длины горизонтальных дуг равными 0, а длины наклонных величинам  $C_i$  соответствующих операций. Задача свелась к определению пути, соединяющего начало координат с одной из конечных вершин и имеющего максимальную длину среди всех путей, длина которых не превышает 16. Таких путей два, и каждый имеет длину 15. Соответственно получаем два оптимальных решения. В первом первая

бригада выполняет операции 2, 4 и 5 (время работы равно 15), а вторая – операции 1,3 (время работы равно 17). Во втором – первая бригада выполняет операции 3, 5 (время работы равно 15), вторая – операции 1, 2, 4 ( время работы равно 17).

Перейдем к рассмотрению случаев, когда число бригад «велико», Далее точно примем, что каждой бригаде назначается не более двух операций. Следовательно  $n \leq 2m$ . Пусть  $n=m+k$ , где  $k \leq m$ . Пусть далее операции пронумерованы по возрастанию  $C_i$ , то есть  $C_1 \leq C_2 \leq \dots \leq C_n$ . Оптимальное решение получается по следующему правилу.  $(m-k)$  операций с номерами  $(2k+1), \dots, (m+k)$  выполняются по одной  $(m-k)$  бригадами, а  $2k$  операций выполняются по две  $k$  бригадами, причем первая бригада выполняет операции 1 и  $2k$ , вторая – 2 и  $2k-1$ , третья – 3 и  $2k-2$  и т.д.

**Пример 2.4.** Пусть  $n=9$ ,  $m=6$ .

|       |   |   |   |   |   |   |    |    |    |
|-------|---|---|---|---|---|---|----|----|----|
| $i$   | 1 | 2 | 3 | 4 | 5 | 6 | 7  | 8  | 9  |
| $C_i$ | 2 | 3 | 5 | 6 | 8 | 9 | 10 | 11 | 11 |

Так как  $n=m+3$ , то  $k=3$ . Следовательно операции 7,8 и 9 выполняются по одной. Далее одна из бригад выполняет операции 1 и 6, вторая – 2 и 5 и третья – 3 и 4. Время выполнения всех операций

$$T = \max(10; 11; 11; 2 + 9; 3 + 8; 5 + 6) = 11$$

## 2.5. Произвольная транспортная схема

В случае достаточно большого числа бригад, когда каждой бригаде достается не более двух операций, удается эффективно ре-

шать задачи календарного планирования и для произвольных транспортных схем.

Пусть задана матрица  $\| \mathbf{I}_{ij} \|$  времен перехода с одной операции  $i$  на другую  $j$ . Без ограничения общности можно принять, что  $n=2m$  (этого всегда можно добиться, вводя фиктивные операции). Пусть операции  $i$  и  $j$  выполняются одной бригадой. Если операция  $i$  делается первой, то время выполнения двух операций составит

$$T_{ij} = \mathbf{I}_{0i} + \tau_i + \mathbf{I}_{ij} + \tau_j .$$

В противном случае

$$T_{ji} = \mathbf{I}_{0j} + \tau_j + \mathbf{I}_{ji} + \tau_i .$$

Обозначим через  $C_{ij}$  следующую величину

$$C_{ij} = \min(T_{ij}; T_{ji})$$

Рассмотрим симметрический граф с длинами ребер  $C_{ij}$ . Задача свелась к выделению в этом графе  $m$  ребер (по числу бригад), никакие два из которых не имеют общих вершин. Такое множество ребер называется паросочетанием графа. Таким образом, необходимо найти паросочетание  $Q$ , для которого величина

$$\eta = \max_{(i,j) \in Q} C_{ij}$$

минимальна. В основе метода решения задачи лежит алгоритм определения паросочетанием в графе. Опишем этот алгоритм.

Предварительно получим необходимые и достаточные условия оптимальности для более общей задачи. Обозначим  $q_{ij}$  – вес ребра  $(i, j)$ . Будем рассматривать полные графе с четным числом вершин. Поставим задачу определения паросочетания с максимальным суммарным весом ребер.

Пусть  $Q$  – произвольное паросочетание.

**Определение 2.1.** Чередующимся циклом называется цикл, в котором из любых двух смежных ребер одно принадлежит циклу, а другое не принадлежит.

**Определение 2.2.** Длиной чередующегося цикла  $M$  называется разность суммы весов ребер паросочетания и суммы весов ребер, не принадлежащих паросочетанию, то есть

$$L(M) = \sum_{(i,j) \in M-Q} q_{ij} - \sum_{(i,j) \in M \setminus Q} q_{ij}$$

**Теорема 2.1.** Для того, чтобы паросочетание  $Q$  – было оптимальным, необходимо и достаточно, чтобы длина любого чередующегося цикла была неположительной.

**Необходимость.** Пусть  $Q$  – оптимальное сочетание и пусть нашелся чередующийся цикл  $M$ , такой что,  $L(M) > 0$ . Но в этом случае ребра  $M-Q$  и ребра  $Q-M$  образуют паросочетания с большим весом.

**Достаточность.** Пусть  $Q$  – паросочетание такое, что любой чередующейся цепи имеет неположительную длину. Пусть  $Q_0$  – оптимальное паросочетание. Заметим, что если  $Q \neq Q_0$ , то ребра  $Q-Q_0$  и  $Q_0-Q$  образуют чередующийся цикл. Заменяя ребра  $Q_0-Q$  на ребра  $Q-Q_0$ , то мы получим паросочетание  $Q_0$ . При этом вес паросочетания  $Q_0$  будет не больше чем  $Q$ , поскольку чередующийся цикл имеет неположительную длину. Поэтому  $Q$  – оптимальное паросочетание. На основе доказанной теоремы можно предложить алгоритм определения оптимального паросочетания, в основе которого лежит процедура поиска чередующихся циклов с положительной длиной.

Описание алгоритма проведем на примере.

**Пример 2.5.** На рис. 2.15. приведен граф из 6 вершин. Веса ребер указаны в скобках.

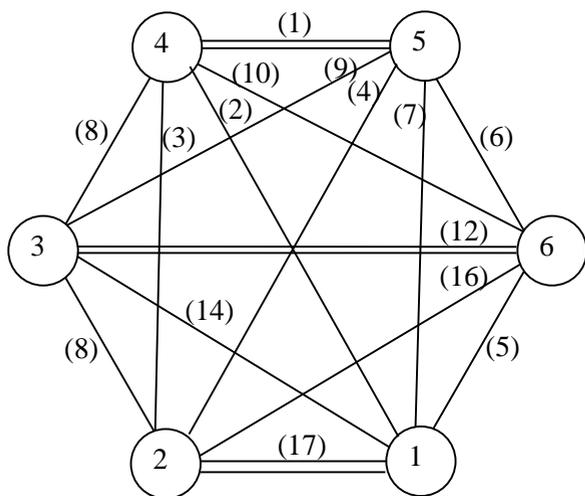


Рис. 2.15.

*Предварительный шаг.* Получение начального решения. Начальное решение можно получить, применяя любое эвристическое правило, приведем одно из них.

Выбираем ребро с максимальным весом. Удаляем все смежные с ним ребра, снова выбираем ребро с максимальным весом и т. д. Пока не получим паросочетание. В нашем примере выбираем ребро (1, 2) с весом 17, затем ребро (3, 6) с весом 12 и наконец ребро (4, 5) с весом 1. (Эти ребра выделены на рис. двойными линиями). Получаем паросочетание  $Q_0 = \{(1, 2); (3, 6); (4, 5)\}$  с весом  $L(Q_0) = 30$ .

*2 этап.* Улучшение начального решения.

*1 шаг.* Выбираем любые два ребра паросочетания и рассматриваем подграф из четырех вершин. Возьмем, например, ребра (1, 2) и (3, 6) и соответствующий подграф из четырех вершин (1, 2, 3, 6) Рис. 2.16.

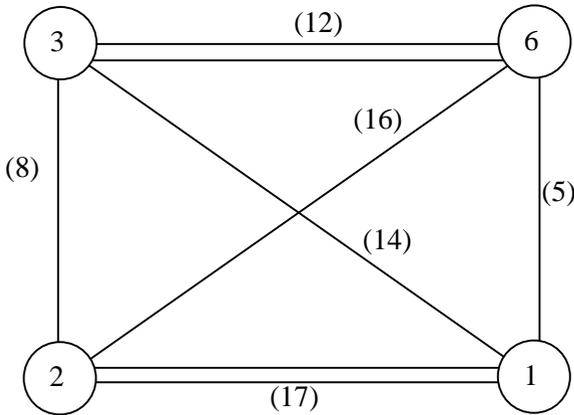


Рис. 2.16.

Наша задача определить в этом подграфе чередующийся цикл максимальной длины. Для этого перейдем к другому графу следующим образом. Выделяем вершины 1 и 2 и построим сеть в которой вершина 1 является входом, а вершина 2 выходом сети (Рис. 2.17).

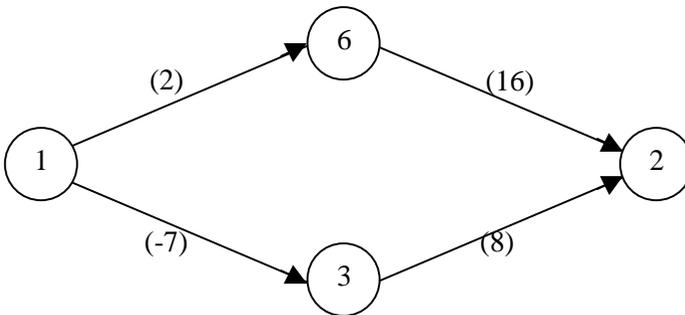


Рис. 2.17.

Длины дуг определим следующим образом:

$$l_{16} = q_{13} - q_{36} = 14 - 12 = 2$$

$$l_{13}=q_{16}-q_{36}=5-12=-7$$

$$l_{32}=q_{32}=8$$

$$l_{62}=q_{62}=16$$

Определим в этой сети путь максимальной длины, соединяющий вход сети с выходом. Задача имеет решение, так как сеть не имеет контуров. В нашем примере это путь (1, 6, 2), длина которого  $L(1, 6, 2)=18$ . Если длина пути превышает вес ребра (1, 2), то начальное решение можно улучшить.

Действительно, в этом случае существует чередующийся цикл  $M(1, 3, 6, 2, 1)$  положительной длины

$$L(M_1)=18-17=1>0$$

Этому циклу соответствует паросочетание  $Q_1=\{(1, 3), (2, 6), (4, 5)\}$  с большим весом  $L(Q_1)=31$ .

*2 шаг.* Выбираем третье ребро из оставшихся ребер паросочетания. В нашем примере это ребро (4, 5). Принимаем вершину 4 за вход сети, а вершину 5 за выход сети и строим сеть из шести вершин (рис 2.18).

Определяем длины дуг:

$$l_{41}=q_{43}-q_{31}=8-14=-6$$

$$l_{42}=q_{46}-q_{26}=10-16=-6$$

$$l_{43}=q_{41}-q_{13}=12-14=-2$$

$$l_{46}=q_{42}-q_{26}=3-16=-13$$

$$l_{12}=q_{16}-q_{26}=5-16=-11$$

$$l_{16}=q_{12}-q_{26}=17-16=1$$

$$l_{21}=q_{23}-q_{31}=8-14=-6$$

$$l_{23}=q_{12}-q_{13}=17-14=3$$

$$l_{32}=q_{36}-q_{62}=12-16=-4$$

$$l_{36}=q_{32}-q_{62}=8-16=-8$$

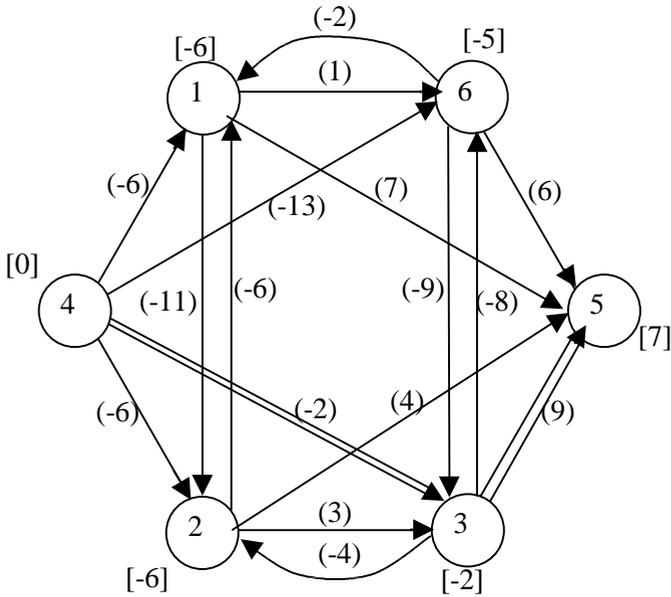


Рис. 2.18.

$$l_{61} = q_{63} - q_{31} = 12 - 14 = -2$$

$$l_{63} = q_{61} - q_{31} = 5 - 14 = -9$$

$$l_{15} = q_{15} = 7$$

$$l_{25} = q_{25} = 4$$

$$l_{35} = q_{35} = 9$$

$$l_{65} = q_{65} = 6$$

Определяем путь максимальной длины, соединяющий вход с выходом. Эта задача всегда имеет решение, поскольку в полученной сети отсутствуют контуры положительной длины. Действительно, если бы такой контур существовал, то в подграфе, состоящем из вершин (1, 2, 3, 6) нашлась бы чередующаяся цепь положительной длины, что невозможно. В нашем примере это путь

(4, 3, 5), имеющий длину  $L(4, 3, 5)=7$ . Так как  $L(4, 3, 5)>q_{45}=1$ , то существует чередующийся цикл

$$M_2=(4. 1. 3. 5. 4)$$

положительной длины

$$L(M_2)=21-15=6.$$

следовательно, можно построить следующее паросочетание

$$Q_2=\{(1, 4); (3, 5); (2, 6)\}$$

имеющее вес  $P(Q_2)=37$ .

Это паросочетание является оптимальным, то есть имеет максимальный вес.

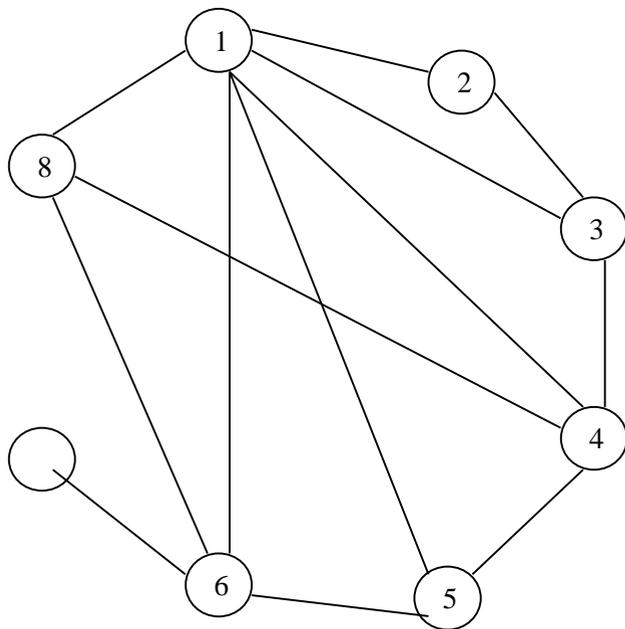
Описанный алгоритм можно применить для определения паросочетания произвольного графа. Для этого достаточно положить веса ребер равными 1, а веса отсутствующих ребер – равными 0 и решить задачу построения паросочетания с максимальным весом для полученного полного графа. Если вес оптимального паросочетания равен  $n/2$ , то исходный граф имеет паросочетание.

*Лемма.* Если в графе имеется висячее ребро, то есть ребро, одно из граничных вершин которого имеет единичную степень, то существует паросочетание, с максимальным весом, содержащее это ребро.

Доказательство очевидно. Если  $Q$  – паросочетание с максимальным весом, не содержащее висячего ребра  $(i, j)$ , где  $j$  – висячая вершина, то удаляя ребро паросочетания с единичным весом инцидентное вершине  $i$  и добавляя ребро  $(i, j)$  также с единичным весом мы получаем паросочетание, содержащее ребро  $(i, j)$  и имеющее такой же вес.

Лемма позволяет сократить размерность задачи, если имеются висячие ребра.

**Пример 2.6.** Рассмотрим граф из 8 вершин, приведенный на рис 2.19.



*Рис.2.19.*

Вершина 7 является висячей. Поэтому сразу включаем ребро (7, 6) в паросочетание и удаляем вершины 6, 7 и инцидентные им ребра.

Получаем граф, приведенный на рис. 2.20.

Применяя любое эвристическое правило, пытаемся получить паросочетание для оставшегося графа. Так, например, берем ребра (1, 2) и (3, 4) с единичными весами и ребро (5, 8) с нулевым весом.

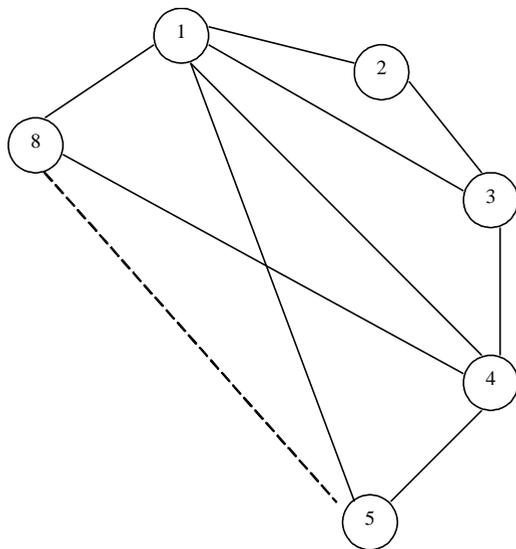
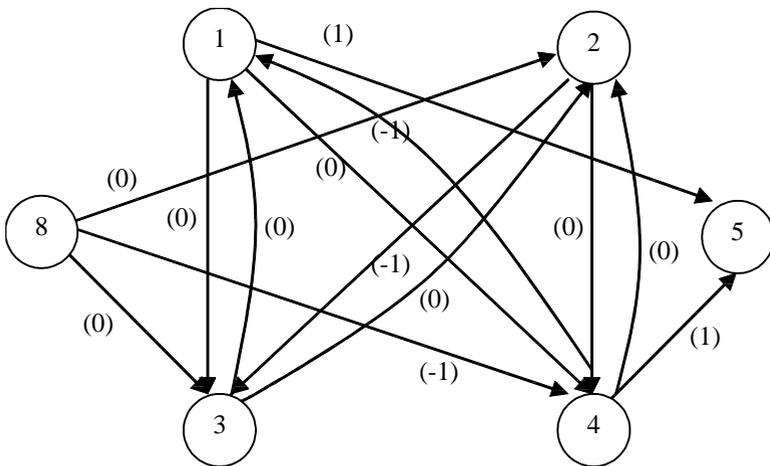


Рис.2.20.

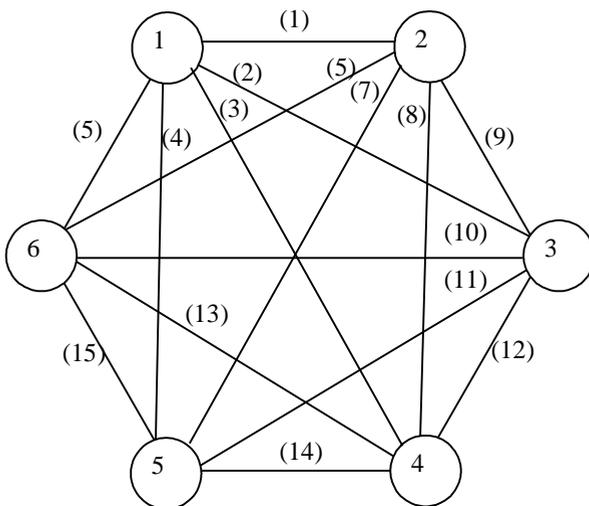
Заметим, что ребра (1, 2) и (3, 4) образуют паросочетание с максимальным весом в подграфе, состоящем из вершин 1, 2, 3, 4. Поэтому определяем чередующуюся цепь максимальной длины из вершины 8 в вершину 5. Соответствующая сеть с длинами дуг приведена на рис. 2.21.

Путей максимальной длины несколько. Один из них это путь  $\mu=(8, 3, 1, 5)$  длины 1. Этому пути соответствует чередующийся цикл  $(8, 4, 3, 2, 1, 5)$  длины 1. Соответственно получаем паросочетание включающее ребра (8, 4), (3, 2) и (1, 5) с максимальным весом 3.

Применим описанный алгоритм для решения задачи назначения бригад. Опишем алгоритм решения задачи. Рассмотрим его на примере. На рис. 2.22 приведен граф из 6 вершин.



*Puc.2.21.*



*Puc. 2.22.*

1 шаг. Для каждой вершины графа определяем ребро с минимальной длиной. Обозначим через  $a_1$  – максимальную длину полученных ребер и добавляем в граф все ребра, длины которых не превышает  $a_1$ . В результате получаем граф  $G_1$  приведенный на рис. 2.23.

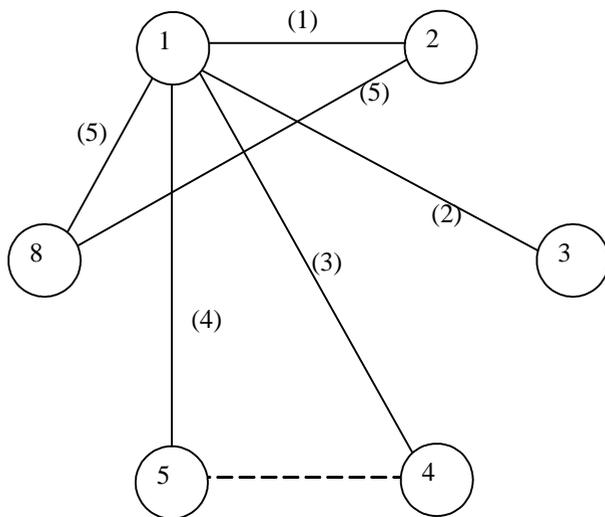


Рис. 2.23.

Поскольку в графе имеются висячие вершины, то берем любую из них, например, вершину 3 и включаем ребро (1, 3) в паросочетание. Удаляя вершины 1, 3 и все инцидентные им ребра, видим, что остается Единственное ребро (2, 6). Итак, мы получаем паросочетание, содержащее ребра (1, 3), (2, 6) с единичными весами и ребро (4, 5) с нулевым весом.

2 шаг. Находим ребра с минимальным весом, среди ребер, не вошедших в граф  $G_1$  (см. рис. 2.24) и повторяем алгоритм

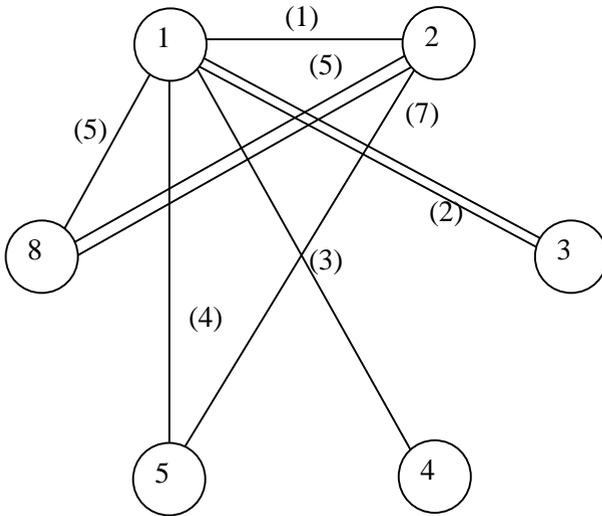
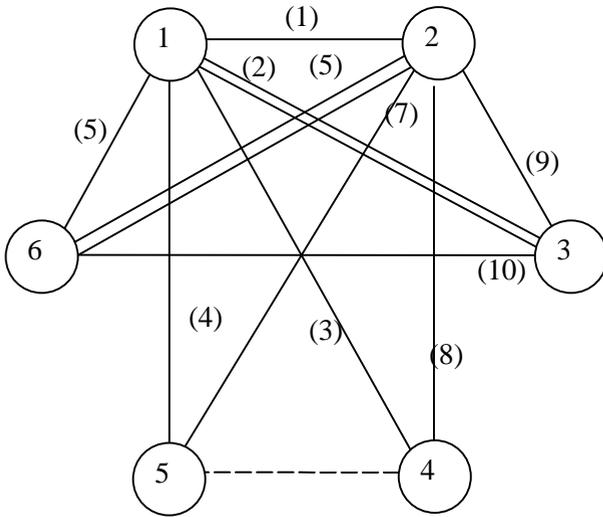


Рис. 2.24.

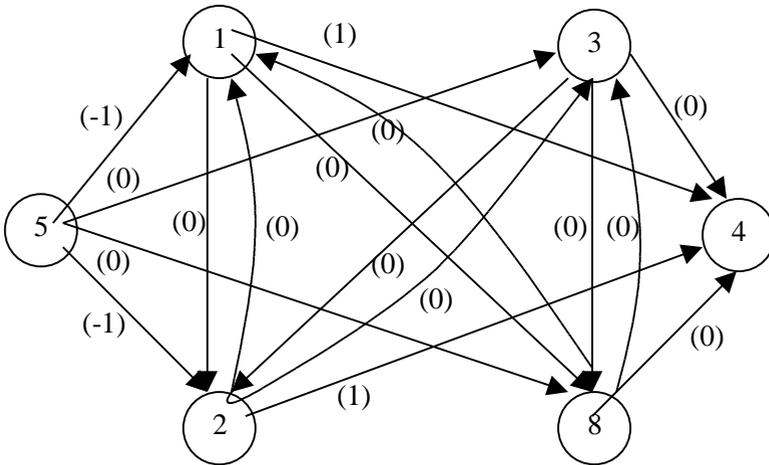
определения паросочетания с максимальным весом для нового графа  $G_2$ . Легко убедиться, что максимальный вес остается равным 2. Поэтому снова добавляем ребра с минимальным весом из числа ребер, не вошедших в граф  $G_2$  и т.д., пока не получим паросочетание, имеющее вес 3. В нашем примере это происходит на пятом шаге, когда в граф добавляются все ребра с весом, не превышающим 10 (см. рис. 2. 25).

На основе графа 2.25 строим сеть 2.26 и определяем путь максимальной длины из вершины 5 в вершину 4.

Существует несколько путей максимальной длины 1. Один из них (5, 3, 2, 4). Этому пути соответствует чередующий цикл  $M=(5, 1, 3, 6, 2, 4, 5)$  единичной длины, что позволяет получить паросочетание  $Q=\{(1, 5), (2, 4), (3, 6)\}$  с максимальным весом 3. Это



*Puc. 2.25.*



*Puc.2.26.*

значит, что мы получили паросочетание для исходного графа (рис.2.23). Таким образом, получено оптимальное решение задачи назначения бригад, в котором одна из бригад выполняет работы 1 и 5, вторая – работы 2 и 4, а третья работы 3 и 6.

## **ЗАКЛЮЧЕНИЕ**

Рассмотренные в работе постановки задач определения календарного графика работ бригад при различных транспортных схемах являются первыми попытками предложить методы оптимального распределения ресурсов на двойной сетевой модели. Развитие этого направления требует дальнейших исследований.

## **ЛИТЕРАТУРА**

1. Бурков В.Н., Новиков Д.А. Теория графов в управлении организационными системами. – М.: СИНТЕГ, 2001.
2. Бурков В.Н. и др. Сетевые модели и задачи управления. – М.: Советское радио, 1966.