Тема 2 Динамическое программирование

2.3 Задача сетевого планирования

2.3.1. Постановка задачи

В *сетевом планировании* исследуются проблемы реализации сложных проектов (комплексов работ), состоящих из большого количества отдельных работ, которые должны выполняться в определенной технологической

Составим сетевую модель задачи. Факт (явление) начала (или окончания) какого-либо множества работ из заданной совокупности работ проекта назовем *событием* и поставим ему в соответствие узел $i \in I$. Работу, которая может начаться после события $i \in I$ и которая предшествует событию *j*, обозначим дугой $(i, j) \in U$. Ни одна работа $(i, j) \in U$ не может начаться, если не завершены все работы

$$(k, i) \in U,\ k \in I_i^- = \{k \in I:\ \exists\, (k, i) \in U\},$$

т.е. ни одна работа $(i, j) \in U$ не может начаться до наступления события *i*; момент наступления события *i* определяется моментом завершения **всех** работ $(k, i),\ k \in I_i^-$.

Определения основных понятий на сети $S = \{I, U\}$ приведены в разделе [Задача о потоке минимальной стоимости].

На сети $S = \{I, U\}$ выделим два узла: *s* s} --- начальное событие (начало выполнения проекта) и *t* --- конечное событие (завершение проекта). По построению верны соотношения $I_s^- = \emptyset$, $I_t^+ = \emptyset$, где

$$I_i^+ = \{j \in I: \exists\, (i, j) \in U\}.$$

Каждой дуге $(i, j) \in U$ припишем одну характеристику $c_{ij} > 0$ --- время выполнения работы (i, j) . Обозначим через $x_i, i \in I,$ момент наступления события *i*; $x_s = 0.$

Из технологических требований, имеющихся в проекте и отраженных в структуре сети *S*, следуют неравенства

$$x_i + c_{ij} \le x_j,\ i \in I_j^-,\ j \in I, \tag{9}$$

которые означают, что событие *j* наступает не раньше, чем будут завершены все работы $(i, j),\ i \in I_j^-$, которые предшествуют этому событию *j*.

Из (9) следует, что в сети *S* нет контуров. Действительно, предположим, что в *S* существует контур и обозначим через $U_k \subset U$ дуги этого контура. Просуммируем неравенства (9) по дугам $(i,j) \in U_k$ и получим

$$\sum_{(i,j)\in U_k} c_{ij} \le 0.$$

Однако это противоречит предположению $c_{ij} > 0,\ (i,j) \in U$.

Далее будем предполагать, что

$$I_i^+ \ne \emptyset,\ i \in I \setminus t;\ \ I_i^- \ne \emptyset, i \in I \setminus s.$$

ибо этого всегда можно добиться, модифицируя сеть *S* и не нарушая технологической последовательности выполнения работ.

Минимальное время выполнения проекта определяется наименьшим числом x_t^0, которое в совокупности с числами

$$x_i^0 \ge 0,\ i \in I\backslash t;\ \ x_s^0 = 0 \tag{10}$$

удовлетворяет неравенствам (9).

Поскольку для окончания проекта необходимо, чтобы все работы были завершены, то длина каждого пути из *s* в *t*, равная сумме $\sum c_{ij}$, вычисленной вдоль дуг пути, не меньше чем x_t^0. Чтобы убедиться в этом, достаточно сложить неравенства (9) вдоль этого пути.

С другой стороны, очевидно, что найдется такая последовательность работ, составляющая путь из *s* в *t*, общая продолжительность которых равна x_t^0.

Таким образом, задача вычисления x_t^0 сводится к поиску пути из *s* в *t* с максимальной длиной $\sum c_{ij}$. Такой путь принято называть *критическим*.

Заметим, что сформулированная задача: найти

$$x_t - x_s \to \min \tag{11}$$

при условиях (9), (10)

является двойственной к задаче о потоке минимальной стоимости (см. раздел [Задача о потоке минимальной стоимости].) на сети *S* с дуговыми стоимостями $-c_{ij}, (i,j) \in U$, и интенсивностями узлов

$$a_s = 1,\ a_t = -1,\ a_i = 0, i \in I \setminus \{s,t\}.$$

При этом числа $x_i^0, i \in I$ -- оптимальные потенциалы в задаче о потоке минимальной стоимости. Следовательно, задачу (11) можно решать *методом потенциалов*, который был рассмотрен в разделе [Задача о потоке минимальной стоимости].

Используя специфику задачи (11), решим её методом динамического программирования.

2.3.2 Метод динамического программирования

Итак, мы решаем задачу о построении критического (максимального) пути из s в t.

Согласно общей схеме динамического программирования, осуществим **первый этап** -- инвариантное погружение в семейство задач.

Общая задача семейства состоит в построении максимального пути из c в любой узел $j \in I$. Длина B_j этого пути называется *функцией Беллмана*.

Осуществим **второй этап** -- составим уравнение Беллмана, которому удовлетворяет функция Беллмана B_j. Для этого поступим следующим образом. Будем исследовать пути из s в j.

Вначале предположим, что последней дугой пути из s в j является дуга $(i, j) \in U$, где $i \in I_j^-$ и что в узел i из s мы попали вдоль пути максимальной длины, т. е. вдоль пути длины B_i. Тогда длина рассматриваемого пути из s в j через i равна

$$B_i + c_{ij}. \tag{12}$$

Переберем все узлы $i \in I_j^-$ и найдем максимальное среди чисел (12):

$$\max_{i \in I_j^-}(B_i + c_{ij}). \tag{13}$$

Рассуждая от противного, легко показать, что число (13) равно длине максимального пути из s в j. По определению длина максимального пути из s в j равна B_j, следовательно,

$$B_j = \max_{i \in I_j^-}(B_i + c_{ij}). \tag{14}$$

Уравнение (14) -- это **уравнение Беллмана**.

Значение функции Беллмана для узла s известно:

$$B_s = 0. \tag{15}$$

Равенство (15) -- краевое (начальное) условие для уравнения (14).

В отличие от рассмотренного ранее примера уравнение Беллмана (14) не является явно рекуррентным.

Для решения уравнения (14) с начальным условием (15) разработаем специальный метод, называемый методом пометок.

2.3.3 Алгоритм метода пометок

Обозначим через I_* множество узлов $i \in I$, для которых известна функция Беллмана B_i . Множество I_* не пусто, так как, согласно (7), $s \in I_*$.

Если $t \in I_*$, то задача решена, B_t -- длина максимального пути из *s* в *t*. Для восстановления этого пути надо осуществить *«обратный ход»* решения уравнения Беллмана. Эту процедуру рассмотрим ниже. Для того чтобы легче осуществить «обратный ход», будем для узлов $i \in I_*$ кроме функции Беллмана B_i задавать еще функцию $f(i), i \in I_*$.

На первом шаге имеем

$$I_* = \{s\}, \;\; B_s = 0, \; f(s) = 0.$$

Пусть $t \notin I_*$. В сети *S* построим множество узлов

$$w(I_*) = \{j \in I : \exists \; (i,j) \in U, \; i \in I_*, \; j \notin I_*\},$$

соседних с множеством I_* . В силу того, что в сети *S* нет контуров, среди узлов $j \in w(I_*)$ обязательно найдется узел j_* , для которого

$$I^-_{j_*} \subset I_*. \tag{16}$$

Поскольку для узлов $i \in I_*$ значения B_i функции Беллмана уже известны, то из уравнения (14) можно найти

$$B_{j_*} = \max_{i \in I^-_{j_*}} (c_{ij_*} + B_i) = c_{i_* j_*} + B_{i*}. \tag{17}$$

Полагаем $I_* = I_* \cup j_*, \; f(j_*) = i_*$ и переходим к следующей итерации.

Замечание. Узлов типа j_* , для которых верно (16), может оказаться несколько. Для всех них по правилам (17) находим функцию Беллмана и функцию $f(j_*)$ и все эти узлы добавляем к множеству I_*.

Очевидно, что через конечное число (меньшее либо равное |*I*|) шагов мы придем к ситуации, когда $t \in I_*$. Это означает, что задача решена: B_t -- длина максимального пути из *s* в *t*.

Осуществим «обратный ход» для нахождения максимального пути:

$\{t,\ i_1,\ i_2, ..., i_k,\ s\}$ по правилу:

$$i_1 = f(t),\ i_2 = f(i_1),\ ...,\ i_k = f(i_{k-1}),\ s = f(i_k). \tag{18}$$

Пример.

Рассмотрим сеть, изображенную рис. 2.1,, где *s*= 1, *t* = 4.

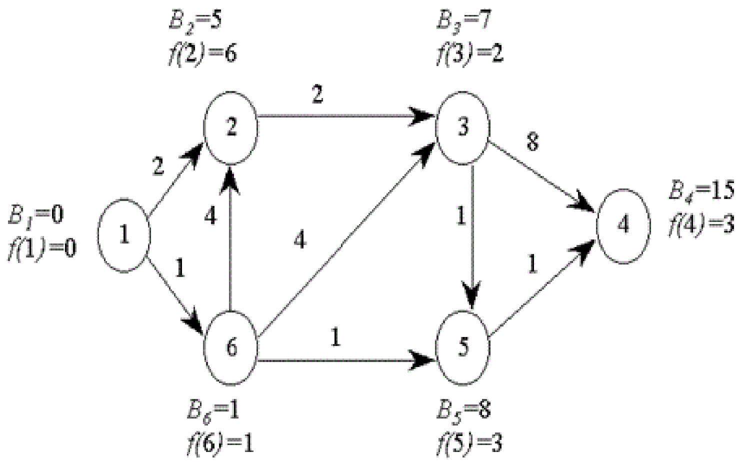

Рис. 2.1

Числа, указанные на дугах, равны длине соответствующей дуги. На этом же рисунке приведены значения функции Беллмана $B_j,\quad j = \overline{1,6}$, определенные по методу пометок. Вторые метки $f(j),\quad j = \overline{1,6}$, позволяют восстановить критический путь из *s* в *t* по правилу (18):

$$i_1 := f(t) = 3;\quad i_2 := f(3) = 2;\quad i_3 := f(2) = 6;\quad i_4 := f(1).$$

Следовательно, искомый путь имеет вид

$$s = 1 \to 6 \to 2 \to 3 \to 4 = t.$$

Замечание. Работы, входящие в критический путь, должны начинаться в строго фиксированные моменты времени: например, работа (i, j) , вошедшая в критический путь, не может (чисто физически) начаться раньше момента времени x_i^0 ; если же она начнется позже момента x_i^0 , то это приведет к увеличению времени выполнения всего проекта.

Для работ, не принадлежащих критическому пути, есть возможность несколько варьировать момент их начала без увеличения общей продолжительности выполнения всего проекта. Пусть дуга (i, j) не принадлежит критическому пути, тогда работа (i, j) чисто физически не может начаться раньше момента x_i^0 . Но мы можем начать работу эту (i, j) в любой момент вида $x_i^0 + \Delta t_{ij}$, где $\Delta t_{ij} \in [\,0,\ x_j^0\ - c_{ij}\ -\ x_i^0]$, без ущерба для общего времени выполнения проекта.

Эту возможность «сдвига» работ некритического пути используют при сетевом планировании распределения ресурсов (например трудовых ресурсов), при решении следующей задачи: определить такие $\Delta t_{ij}, \ (i,j) \in U,$ чтобы в каждый момент времени для выполнения текущих работ требовалось минимальное количество рабочих. Моменты времени $x_i^0, \ i \in I,$ считаются уже найденными и фиксированными.