Тема 4 Кратчайшие пути

Рассмотрим *сеть (ориентированную)* $S=\{I,U\}$ со множеством узлов I и дуг U . Каждой ориентированной дуге $(i,j)\in U$ поставим в соответствие пару точек $\{i,j\}$, которую назовем **ребром** с граничными узлами *i* и *j*.

Последовательность различных ребер

$$\{i_1,i_2\},\ \ \{i_2,i_3\},\ \ \ldots,\ \ \ \{i_{k-1},i_k\}$$

называется (простой) цепью, соединяющей узлы i_1 и i_k.

Пусть данная цепь рассматривается в направлении от узла i_1 к узлу i_k. Если это направление совпадает с направлением $i\to j$ дуги (i,j) в этой цепи, то дуга (i,j) называется *прямой*. Дуга с противоположным направлением называется *обратной.*

Путем из узла $s\in I$ в узел $t\in I$ называется простая цепь, соединяющая *s* и *t*, при этом все дуги цепи являются прямыми при движении из *s* в *t*.

Цепь (путь)

$$\{i_1,i_2\},\ \ \{i_2,i_3\},\ \ \ldots,\ \ \ \{i_{k-1},i_k\}$$

с совпадающими узлами i_1 и i_k : $i_1=i_k$ называется *циклом (контуром)*.

Каждой дуге $(i,j)\in U$ припишем некоторый параметр c_{ij} («стоимость»), который определяет длину дуги (i,j) .

В этом случае под длиной пути будем понимать сумму длин дуг, образующих данный путь.

В данной главы изучается задача о нахождении кратчайшего (минимального) пути из фиксированного узла *s* в другой заданный узел *t* или во все другие узлы сети. В последнем случае говорят о задаче построения дерева кратчайших путей. Следует различать три случая:

1. Все дуги сети имеют неотрицательную длину.
2. Некоторые дуги имеют отрицательную длину, но в сети не существует контуров с суммарной отрицательной длиной.
3. В сети существует один или несколько контуров с отрицательной суммарной длиной.

В третьем случае задача о построении кратчайшего пути из *s* в *t* может не иметь решения. (Задача будет иметь решение только в том случае, если нет ни одного пути из *s* в узел *i*, принадлежащий контуру с отрицательной длиной.) В таких ситуациях используются алгоритмы, позволяющие обнаружить и указать контуры с отрицательной длиной.

Заметим, что отсутствие в сети $S=\{I,U\}$ контуров отрицательной длины и существование пути из i в j для любых $i,\ j\in I$, являются необходимыми и достаточными условиями существования кратчайшего пути из любого узла $i \in I$ в любой узел $j \in I$.

Второй параграф данной главы посвящен построению кратчайших путей между всеми парами узлов заданной сети.

4.1 Задача о кратчайшем пути

4.1.1. Поиск кратчайшего пути и задача о потоке минимальной стоимости

Покажем, что задача построения кратчайших путей из заданного узла s во все другие узлы $i \in I \setminus s$ сети $S = \{I, U\}$ является частным случаем задачи о потоке минимальной стоимости (см. раздел [Задача о потоке минимальной стоимости]). , которая, в свою очередь, является частным случаем задачи линейного программирования.

Рассмотрим задачу построения кратчайших путей из узла s в узлы $i \in I \setminus s$.

Покажем, что эта задача эквивалентна следующей задаче о потоке минимальной стоимости на сети $S = \{I, U\}$:

$$\begin{gathered} \sum_{(i,j)\in U} c_{ij}x_{ij} \to \min, \\ \sum_{j\in I_i^+} x_{ij} - \sum_{j\in I_i^-} x_{ji} = a_i, \quad i \in I, \\ x_{ij} \geq 0, \quad (i,j) \in U, \end{gathered} \tag{1}$$

где

$$\begin{gathered} I_i^+ = \{j \in I : (i,j) \in U\}; \quad I_i^- = \{j \in I : (j,i) \in U\}; \\ a_s = n-1; \;\; a_i = -1, \;\; i \in I \setminus \;\; n = |I|\,. \end{gathered} \tag{2}$$

С учетом того, что все числа $a_i, i \in I$, -- целые, можно показать, что задача (1) имеет оптимальный целочисленный поток.

Ранее в курсе методов оптимизации было доказано, что если в задаче (1) существует оптимальный поток, то для нее найдется и оптимальный **базисный** поток, который определяется следующими свойствами: пусть $x^0 = (x^0_{ij}, (i,j) \in U)$ -- оптимальный базисный поток, тогда существует такое базисное множество (являющееся деревом) $U^0_B \subset U$, что

$$x^0_{ij} \geq 0, \;\; (i,j) \in U^0_B, \;\; x^0_{ij} = 0, \;\; (i,j) \in U \setminus U^0_B. \tag{3}$$

В силу специального подбора чисел $a_i, i \in I$, можно утверждать, что $x^0_{ij} > 0$ (и даже

$x^0_{ij} \geq 1$), если $(i,j) \in U^0_B$.

Покажем, что базисное множество U^0_B и будет деревом кратчайших путей из s в $j \in I \setminus s$.

Действительно, выше отмечалось, что множество U^0_B является деревом. Из свойств дерева следует, что для любого узла $j_* \in I \setminus s$ существует единственная цепь, принадлежащая U^0_B и соединяющая s и j_* .

В силу (3) ненулевые потоки могут быть только на дугах дерева U^0_B . У нас только один узел-источник (с $a_i > 0$) -- это узел s .

Значит, единичный поток, который попадает в узел j_* с $a_{j_*} = -1 < 0$, приходит в узел j_* из узла s . Следовательно, все дуги упомянутой цепи должны быть прямыми. Значит, эта цепь является путем из s в j_* .

Обозначим дуги этого пути через $U(j_*) \subset U^0_B$. Так как из s в j_* вдоль пути $U(j_*)$ идет единица потока, то по построению $x^0_{ij} \geq 1, \;\; (i,j) \in U(j_*)$.

Предположим, что существует другой путь из s в j_* меньшей длины, т.е. существует такой путь $\bar{U}(j_*)$, что

$$\sum_{(i,j)\in \bar{U}(j_{*})} c_{ij} < \sum_{(i,j)\in U(j_{*})} c_{ij} . \tag{4}$$

Построим новый поток $x^* = (x^*_{ij}, (i,j) \in U)$ в сети S :

$$x^*_{ij} = x^0_{ij} + 1, \text{ если } (i,j) \in \bar{U}(j_*), \;\; (i,j) \notin U(j_*),$$

$$x^*_{ij} = x^0_{ij} - 1, \text{ если } (i,j) \notin \bar{U}(j_*), \;\; (i,j) \in U(j_*),$$

$$x^*_{ij} = x^0_{ij} \text{ для остальных дуг } (i,j) \in U.$$

Нетрудно проверить, что x^* -- допустимый поток в сети S , причем

$$\sum_{(i,j)\in U} c_{ij} (x_{ij}^{*} - x_{ij}^{0}) = \sum_{(i,j)\in \bar{U}^{j_{*}}} c_{ij} - \sum_{(i,j)\in U^{j_{*}}} c_{ij} < 0.$$

Однако последнее неравенство противоречит оптимальности потока x^0 в сети S .

Таким образом, мы показали, что, решив задачу (1) , (2) , мы находим дерево U^0_B (оптимальный базис, соответствующий потоку минимальной стоимости), которое является деревом кратчайших путей из s в узлы $j \in I \setminus s$.

Следовательно, для построения дерева кратчайших путей можно использовать метод потенциалов, рассмотренный ранее в разделе [Задача о потоке минимальной стоимости].

Заметим, что задача (1), (2) является частным случаем общей задачи о потоке минимальной стоимости. Специфика задачи (1) , (2) состоит в специальной структуре интенсивностей $a_i, i \in I$.

Кроме того, мы получим дополнительную специфику, если, например, предположим, что $c_{ij} \geq 0, \quad (i,j) \in U$. Учет этой специфики позволяет разработать специальные методы, учитывающие особенности данной задачи. Рассмотрим некоторые из таких методов.

4.1.2 Построение кратчайшего пути на сети с неотрицательными длинами дуг

Пусть задана сеть $S = \{I, U\}$ со множеством узлов I и множеством дуг U . На дугах $(i,j) \in U$ заданы характеристики $c_{ij} \geq 0, \ (i,j) \in U$, где c_{ij} -- длина дуги (i,j) . Требуется для двух фиксированных узлов $s, t \in I$ найти путь из s в t минимальной длины.

Используем метод динамического программирования для построения и обоснования специального алгоритма нахождения кратчайших путей.

Вложим задачу построения кратчайшего пути из узла s в узел t в семейство аналогичных задач. Общая задача семейства состоит в построении кратчайшего пути из s в произвольный узел $j \in I \setminus s$.

Обозначим через B_j функцию Беллмана -- длину кратчайшего пути из s в j.

Для составления уравнения, которому удовлетворяет функция Беллмана B_j , в пути из s в j в качестве последней дуги выберем произвольную дугу $(i,j) \in U, i \in I_j^-$, и предположим, что в узел i мы пришли кратчайшим путем, т.е. длина пути из s в i равна B_i . Здесь

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

Очевидно, что длина минимального пути из s в j через узел $I \in I_j^-$ равна

$$c_{ij} + B_i, i \in I_j^-. \tag{5}$$

Ясно, что в узел j из s мы можем попасть, только пройдя через какой-либо узел $i \in I_j^-$. Нас интересует кратчайший путь, поэтому в (5) найдем минимум:

$$\min_{i \in I_j^-} (c_{ij} + B_i). \tag{6}$$

Ясно, что (6) -- длина кратчайшего пути из s в j , т.е.

$$B_j = \min_{i \in I_j^-} (c_{ij} + B_i). \tag{7}$$

Краевое условие для уравнения Беллмана (7) имеет вид

$$B_s = 0. \tag{8}$$

Уравнение Беллмана (7) не является рекуррентным. Однако этому уравнению можно придать рекуррентный характер.

Обозначим через I_* множество узлов сети S , для которых функция Беллмана B_i уже построена. Очевидно, что $I_* \neq \emptyset$, так как по построению $s \in I_*$.

Если $t \in I_*$, то исходная задача решена, B_t -- длина минимального пути из s в t . Сам путь можно восстановить «обратным ходом» алгоритма (см. ниже).

Пусть $t \notin I_*$. В сети S по множеству I_* построим разрез

$$U(I_*) = \{(i, j) \in U : i \in I_*, j \notin I_*\}.$$

Предположим, что $U(I_*) \neq \emptyset$. Ясно, что каждый путь из s в узел $k \notin I_*$ содержит хотя бы одну дугу из множества $U(I_*)$.

Следовательно, в силу того, что $c_{ij} \geq 0, (i, j) \in U$, для любого $k \notin I_*$ справедливо неравенство

$$B_k \geq \min_{(i,j) \in U(I_*)} (B_i + c_{ij}) = B_{i_*} + c_{i_* j_*}, \quad k \notin I_*. \tag{9}$$

Поскольку (i_*, j_*) -- дуга разреза $U(I_*)$, то $i_* \in I_*, j_* \notin I_*$.

В (9) положим $k = j_*$. Тогда, согласно (9) , имеем

$$B_{j_*} \geq B_{i_*} + c_{i_* j_*}. \tag{10}$$

С другой стороны, так как $i_* \in I_{j_*}^-$, согласно (7) , получаем

$$B_{j_*} = \min_{i \in I_{j_*}^-} (B_i + c_{ij_*}) \leq B_{i_*} + c_{i_* j_*}. \tag{11}$$

Из (10) и (11) следует, что

$$B_{j_*} = B_{i_*} + c_{i_* j_*}.$$

Узел j_* добавим ко множеству I_* и с новым множеством I_* повторяем описанные выше операции.

Очевидно, что через конечное число шагов придем к одной из следующих ситуаций: либо $t \in I_*$, либо $U(I_*) = \emptyset$. Первая ситуация означает, что минимальный путь из s в t найден. Вторая ситуация означает, что в сети S нет ни одного пути из s в t.

Описанную выше схему решения уравнения Беллмана можно реализовать с помощью *метода пометок.*

4.1.3 Метод пометок

Перед первой итерацией алгоритма полагаем

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

Пусть перед началом текущей итерации известно множество узлов I_* сети, для которых найдены значения функции Беллмана $B_i, i \in I_*$, а также некоторой функции $f(i) \in I_*, i \in I_*$.

Обозначим через $\omega(I_*) = \{j \in I : (i,j) \in U(I_*)\}$ множество узлов, соседних со множеством I_*.

Если $\omega(I_*) = \emptyset$, то в сети S нет путей из s в t.

Пусть $\omega(I_*) \neq \emptyset$. Подсчитаем числа (временные метки):

$$B'_j = \min_{i \in I_* \bigcap I_j^-} (B_i + c_{ij}), \quad j \in \omega(I_*), \tag{12}$$

и найдем минимальное среди них

$$B'_{j_*} = \min_B {}'_j, j \in \omega(I_*). \tag{13}$$

Узлу j_*, на котором реализовался минимум в (13), приписываем постоянную метку

$$B_{j_*} = B'_{j_*},$$

полагаем $f(j_*) = i_*$, где $i_* \in I_*$ -- такой узел, что существует дуга $(i_*, j_*) \in U$ и $B_{j_*} = B_{i_*} + c_{i_* j_*}$. Узел j_* добавляем к узлам I_*. Переходим к следующей итерации.

На каждой итерации алгоритма количество узлов, имеющих постоянные метки, увеличивается на единицу. Следовательно, через конечное число шагов придем к одной из следующих ситуаций:

а) $t \in I_*$;   б) $\omega(I_*) = \emptyset$.

Случай «б» означает, что в сети $S = \{I, U\}$ нет путей из s в t.

Случай «а» означает, что B_t -- длина кратчайшего пути из s в t.

Для построения самого кратчайшего пути из s в t осуществим «обратный ход», используя функцию $f(i), i \in I_*$. С этой целью определим узлы по правилу

$$i_0 := f(t), \quad i_1 := f(i_0), \quad i_2 := f(i_1), \quad \ldots, \tag{14}$$

$$i_k := f(i_{k-1}), \quad \ldots, \quad i_m := f(i_{m-1}) = s.$$

Тогда путь из s в t имеет вид

$$s \to i_{m-1} \to \ldots \to i_k \to i_{k-1} \to \ldots \to i_1 \to i_0 \to t. \tag{15}$$

Пример, иллюстрирующий работу метода пометок, приведен на рис. 3.1.

Дуги кратчайшего пути из узла $s = 1$ в узел $t = 4$ выделены жирными линиями.

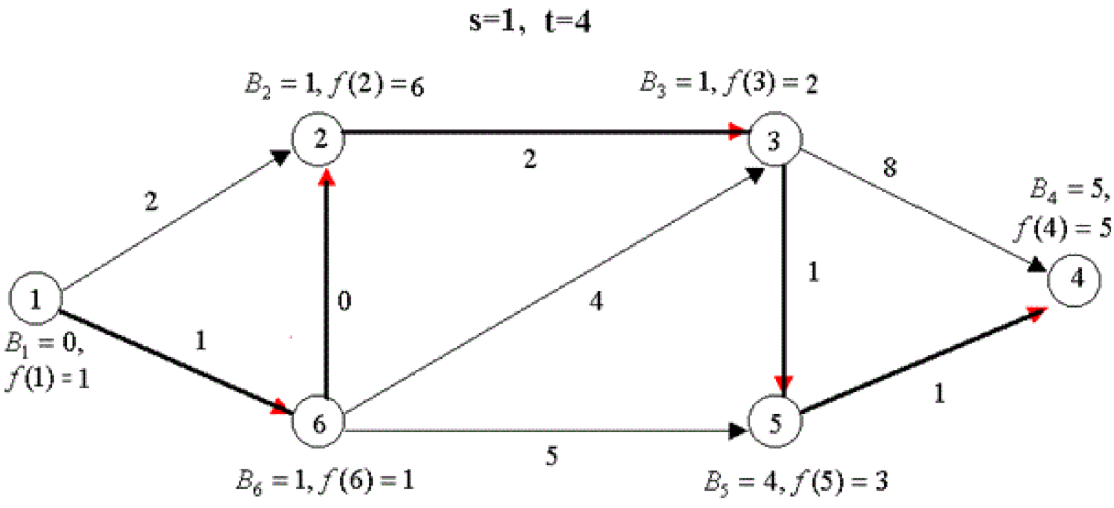

Рис. 3.1

4.1.4 Алгоритм Дейкстры

Описанный метод пометок можно сделать еще более детальным. Приводимая ниже модификация называется «**жадным**» алгоритмом, или алгоритмом Дейкстры. Опишем этот алгоритм.

Перед началом работы алгоритма полагаем

$$I_* = \{s\},\ B_s = 0,\ f(s) = s,\ B_j' = \infty,\ f'(j) = 0,\ j \in I \setminus s,\ i_* = s.$$

Пусть заданы множество и числа

$$I_*, B_i, f(i), i \in I_*, B_j', f'(j), j \in I \setminus I_*,$$

а также индекс $i_* \in I_*$.

Опишем алгоритм Дейкстры по шагам.

Шаг 1. Рассмотрим узлы

$$\tilde{I}_{i_*}^+ = \{j \in I \backslash I_* : \exists\ (i_*, j) \in U\} = I_{i_*}^+ \bigcap (I \backslash I_*).$$

Для любого $j \in \tilde{I}_{i_*}^+$ при $B_j' > B_{i_*} + c_{i_* j}$ положим

$$B_j' := B_{i_*} + c_{i_* j}, f'(j) := i_*,$$

в противном случае (т.е. при $B_j' \le B_{i_*} + c_{i_* j}$) временные метки узла j не меняем : $B_j' := B_j',\ \ f'(j) = f'(j)$. Переходим к шагу 2.

Шаг 2. Среди всех узлов $j \in I \backslash I_*$ (т.е. узлов с временными метками) находим узел j_* с минимальной временной меткой:

$$B_{j_*}' = \min_{j \in I \backslash I_*} B_j'.$$

Если $B_{j_*}' = \infty$, то STOP -- алгоритм заканчивает свою работу, так как в сети нет путей из s в t.

Если $B_{j_*}' < \infty$, то переходим к шагу 3.

Шаг 3. Полагаем $B_{j_*} := B'_{j_*}, \quad f(j_*) = f'(j_*)$, т.е. узлу j_* приписываем постоянные метки и относим узел j_* ко множеству I_* , заменяя I_* на $I_* :== I_* \bigcup j_*$.

Полагаем $i_* := j_*$ и переходим к шагу 1, если $t \notin I_*$.

Случай $t \in I_*$ означает, что кратчайший путь из s в t найден. Восстанавливаем этот путь по вторым постоянным меткам $f(i), i \in I_*$, согласно правилу (14) , (15) . Останавливаем работу алгоритма.

Заметим, что описанные алгоритмы обладают достоинством, о котором уже говорилось при изучении метода динамического программирования: мы можем изменить условия задачи и быстро найти решение новой задачи. Например, мы можем легко найти кратчайший путь из s в другой узел $j_0 \in I$, отличный от t .

Действительно, если узел j_0 уже принадлежит множеству помеченных узлов I_* , то кратчайший путь из s в j_0 можно сразу восстановить по вторым постоянным меткам $f(i), i \in I_*$. Если же данный узел $j_0 \notin I_*$, т.е. не имеет еще постоянной метки, то работу алгоритма надо продолжить до тех пор, пока ни реализуется одна из ситуаций:

$$j_0 \in I_* \text{ либо } B'_{j_*} = \min_{j \in I \setminus I_*} B'_j = \infty.$$

4.1.5 Алгоритм построения дерева кратчайших путей

Предположим теперь, что в сети $S = \{I, U\}$ могут быть дуги с отрицательной длиной, но нет контуров с отрицательной суммарной длиной. В этом случае описанные выше алгоритмы, основанные на идеях динамического программирования, не применимы. В данной ситуации разумно использовать метод потенциалов (см. в разделе [Задача о потоке минимальной стоимости]). для решения задачи (1), (2) . Отметим, что в силу специфики данной задачи (см. условие (2)) метод потенциалов можно упростить.

Приведем этот упрощенный вариант метода потенциалов, предназначенный для решения задачи (1), (2).

Опишем вначале **алгоритм построения начального базиса** U_B , который является деревом путей (необязательно кратчайших) из s в произвольный узел $j \in I$. Это -- аналог алгоритма первой фазы метода потенциалов.

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

Пусть на некотором шаге известны множество I_*, числа $B_j, f(j), j \in I_*.$

Если $I_* = I$, то начальный базис-дерево U_B построен. Его легко восстановить по вторым меткам $f(i), i \in I$, согласно правилам (14), (15).

Пусть $I_* \neq I$. Если существует узел $j \in I \setminus I_*$, для которого найдется дуга $(i, j) \in U$ с $i \in I_*$, то полагаем

$$B_j = B_i + c_{ij}, \quad f(j) = i, \quad I_* := I_* \bigcup j.$$

Повторяем описанные выше операции с новым множеством I_*.

Если $I_* \neq I$ и не существует узла $j \in I \backslash I_*$, для которого найдется дуга $(i,j) \in U$ с $i \in I_*$, то в заданной сети $S = \{I, U\}$ нельзя построить дерево кратчайших путей, поскольку не во все узлы есть пути из s.

Пусть начальный базис-дерево U_B построен. При этом будут построены и числа $B_j, j \in I$, соответствующие U_B. Очевидно, что B_j -- длина пути из s в j вдоль дуг дерева U_B.

Общий вид множества дуг U_B приведен на рис. 3.2.

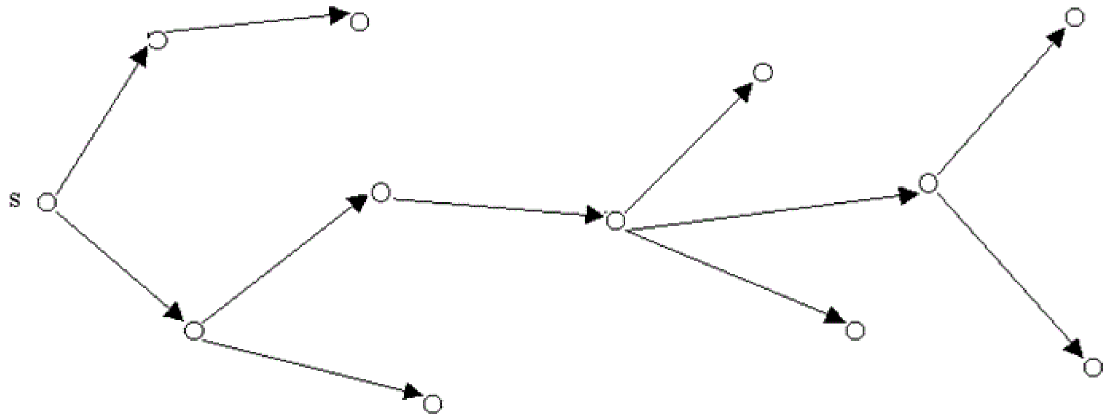

Рис. 3.2

Опишем шаги **алгоритма построения дерева кратчайших путей**

Шаг 1. Для небазисных дуг $U_H = U \backslash U_B$ проверяем условия оптимальности

$$B_i + c_{ij} \geq B_j, (i,j) \in U_H. \tag{16}$$

Если соотношения (16) выполняются, то текущее дерево U_B является деревом кратчайших путей. Алгоритм прекращает работу.

Если найдется дуга $(i_0, j_0) \in U_H$, для которой

$$B_{i_0} + c_{i_0 j_0} < B_{j_0},$$

то зафиксируем эту дугу и перейдем к шагу 2.

Шаг 2. Рассмотрим множество дуг $U_B \bigcup (i_0, j_0)$.

В разделе [Задача о потоке минимальной стоимости] было показано, что добавление дуги $(i_0, j_0) \in U_H$ к дереву U_B приводит к образованию единственного цикла $U_* \subset U_B \bigcup (i_0, j_0)$. В нашем случае этот цикл обладает спецификой: цикл состоит как бы из двух частей. В одну часть входят дуги, направление которых совпадает с направлением дуги (i_0, j_0) , другую часть цикла образуют дуги, имеющие обратное направление (см. рис. 3.3). Отметим, что «прямые» дуги идут подряд, затем подряд идут «обратные» дуги, т.е. нет чередования прямых и обратных дуг (что могло иметь место в общей задаче о потоке минимальной стоимости).

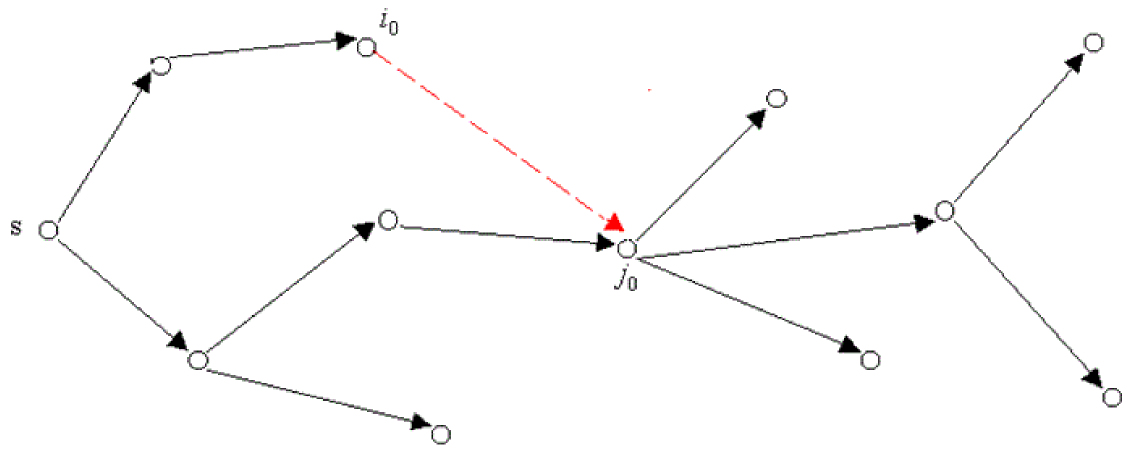

Рис. 3.3

Шаг 3. Далее, согласно методу потенциалов, надо вычислить шаг $\theta_0 = \min_{U_*^-} x_{ij}$, где U_*^- -- обратные дуги цикла. В общей задаче шаг θ_0 мог реализоваться на любой дуге $(i, j) \in U_*^-$.

В силу специфики (2) видно, что в нашем случае шаг θ_0 обязательно реализуется на обратной дуге $(i_*, j_0) \in U_*^-$, ведущей в узел j_0 . Удалим дугу (i_*, j_0) из множества U_B , получим две компоненты связности $\{I_1, U_1\}, \{I_2, U_2\}$, одна из которых содержит узел s , другая -- узел j_0 ,

$$s \in I_1, j_0 \in I_2, U_1 \bigcup U_2 = U_B \setminus (i_*, j_0).$$

(см. рис. 3.4).

Вычислим новые потенциалы $B_j, j \in I$, по правилу

$$\begin{gathered} \bar{B}_j = B_j - (B_{j_0} - c_{i_0 j_0} - B_{i_0}),\ \ j \in I_2; \\ \bar{B}_j = B_j,\ j \in I_1. \end{gathered} \tag{17}$$

Шаг 4. Построим новое базисное дерево (см. рис. 3.5)

$$\bar{U}_B = (U_B \setminus (i_*, j_0)) \bigcup (i_0, j_0).$$

Перейдем к шагу 1 с новым множеством $\bar{U}_B$ и соответствующими ему потенциалами (17).

Приведенный алгоритм решает задачу за конечное число итераций.

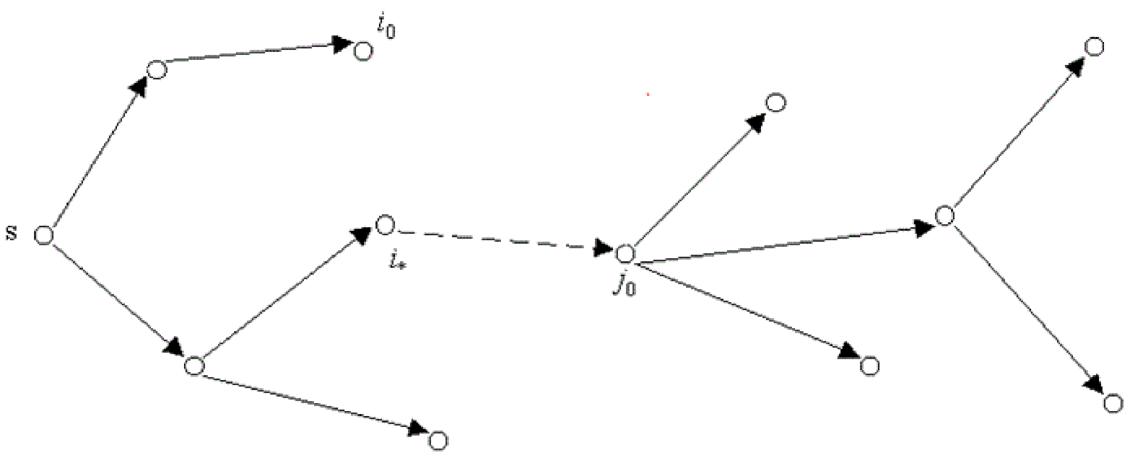

Рис. 3.4

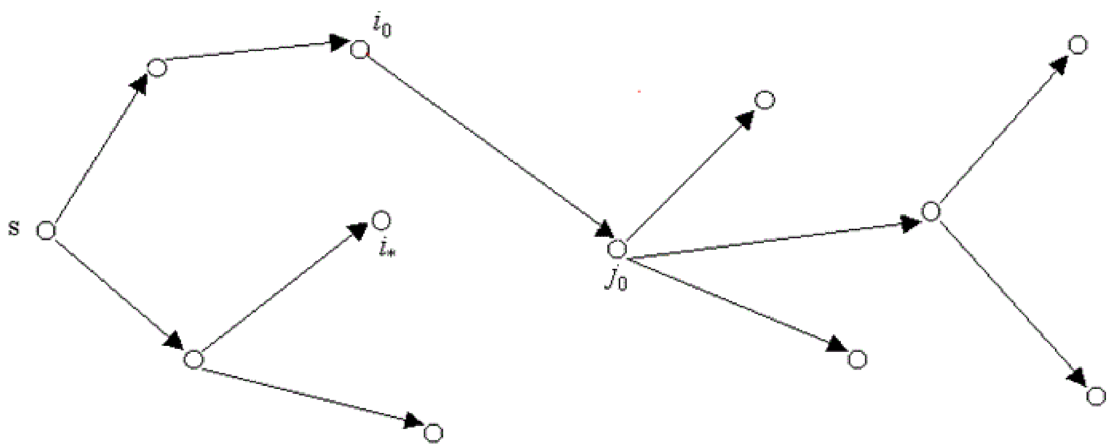

Рис. 3.5

Описанный выше алгоритм, базирующийся на методе потенциалов, можно использовать и для выявления отрицательных контуров в сети. Для этого на шаге 2 надо рассмотреть цикл $U_* \subset U_B \cup (i_0, j_0)$.

Если в этом цикле не все дуги имеют одно направление, то действуем как и раньше.

Если в цикле U_* все дуги имеют одно направление, то STOP -- в сети есть отрицательные контуры. В этом случае задача построения дерева кратчайших путей не имеет решения.