АКАДЕМИЯ НАУК СССР

ЭКОНОМИКА И МАТЕМАТИЧЕСКИЕ МЕТОДЫ

(ОТДЕЛЬНЫЙ ОТТИСК)

4

МОСКВА · 1975

ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

ЭКОНОМНЫЙ АЛГОРИТМ НАХОЖДЕНИЯ МАКСИМАЛЬНОГО ПОТОКА В СЕТИ

А. В. КАРЗАНОВ

(*Москва*)

Излагается алгоритм нахождения максимального потока в сети с верхней оценкой числа действий — $O(n^3)$, где n — число вершин сети. Из известных автору лучшую оценку имеет алгоритм Е. А. Диница [1] — $O(n^2p)$, где p — число дуг сети.

1. ТУПИКОВЫЙ ПОТОК

Потоковой сетью называется ориентированный граф $G=[V;\ U]$ с множеством вершин V, $|V|=n$, множеством дуг U, $|U|=p$, действительной функцией «пропускной способности дуги» — $c(u)\geqslant 0$, $u\in U$, и выделенными вершинами — «источником» s и «стоком» t. Функция $f(u)$, $u\in U$, называется потоком, если: 1) $0\leqslant f(u)\leqslant c(u)$, $u\in U$ (условие допустимости),

$$2)\ \sum_{(x,y)\in U} f((x,y))-\sum_{(y,x)\in U} f((y,x))=\begin{cases} v(f), & x=s,\\ -v(f), & x=t,\\ 0, & x\in V\setminus\{s,t\},\end{cases}\qquad \text{(условие баланса).}$$

Максимальным называется поток с наибольшей мощностью $v(f)$ [2].

В [1] задача о максимальном потоке фактически сводится к решению не более чем $n-1$ следующих задач.

Дана сеть $S_k=[V_k;\ U_k]$, $|V_k|\leqslant n$, $|U_k|\leqslant p$, $1\leqslant k\leqslant n-1$, с «источником» s, «стоком» t и пропускными способностями дуг $c_k(u)$, $u\in U_k$ (S_k называется справочной кратчайших путей) такая, что: 1) длина кратчайшего по числу дуг (ориентированного) пути с началом в s и концом в t равна k, 2) любая дуга $u\in U_k$, а также любая вершина $x\in V_k$ принадлежат некоторым кратчайшим путям из s в t. Требуется найти поток $f_k(u)$, $u\in U_k$ (называемый в дальнейшем тупиковым) такой, что для любого пути s, u_0, x_1, u_1, $x_2,\ldots$, x_{k-1}, u_{k-1}, t имеется дуга u_j, $0\leqslant j\leqslant k-1$, для которой $f_k(u_j)=c_k(u_j)$ (u_j — *насыщенная дуга*).

В [1] каждая справочная S_k строится за $O(p)$ действий, тупиковый поток в S_k — за $O(np)$, следовательно, максимальный поток находится за $O(n^2p)$ действий (в общем случае — $O(n^4)$ действий). В предлагаемом алгоритме тупиковый поток в справочной строится за $O(n^2)$ действий.

2. ЗАБЛОКИРОВАННЫЕ ПУТИ, ВЕРШИНЫ И ДУГИ

В справочной $S=[V;\ U]$ (индекс k для краткости будем опускать) i-м слоем называется множество $O_i=\{x\colon x\in V,\ x$ находится на расстоянии i от s и $k-i$ от $t\}$, $i=0,\ldots,k$. По условию $O_0=\{s\}$, $O_k=\{t\}$. Любая дуга тем самым ведет из вершины одного слоя в вершину следующего. Обозначим через $\alpha(x)$, $x\in V$, множество исходящих из вершины x дуг, $\beta(x)$ — входящих в x.

Пусть функция $\rho(u)$, $u\in U$, удовлетворяет условию допустимости: $0\leqslant\rho(u)\leqslant c(u)$, $\forall u\in U$. Скажем, что путь ξ является *ρ-заблокированным*, если найдется ρ-насыщенная дуга $u\in\xi$, т. е. $\rho(u)=c(u)$. Дуга $u=(x,\ y)$ (соответственно вершина x') называется ρ-заблокированной, если любой путь $\xi=x,\ u,\ y,\ldots,t$ (соответственно любой путь $\xi'=x',\ldots,t$) — ρ-заблокированный. Таким образом, поток $f(u)$, $u\in U$, является тупиковым тогда и только тогда, когда вершина s — f-заблокирована.

В дальнейшем будем пользоваться обозначениями: $\bar V=V\setminus\{s,\ t\}$, $\rho(\alpha(x))$ и $\rho(\beta(x))$ — для сужения $\rho(u)$ на подмножества $\alpha(x)$ и $\beta(x)$, $|\rho(\alpha(x))|$ и $|\rho(\beta(x))|$ для $\sum\limits_{u\in\alpha(x)}\rho(u)$ и $\sum\limits_{u\in\beta(x)}\rho(u)$, $x\in V$.

3. СОДЕРЖАТЕЛЬНОЕ ОПИСАНИЕ АЛГОРИТМА

Функция $g(u)$, $u\in U$, называется *предпотоком*, если выполняются свойства

Р1. $0\leqslant g(u)\leqslant c(u)$.

Р2. $|g(\beta(x))|\geqslant|g(\alpha(x))|$, $\forall x\in\bar V$.

Р3. Если $|g|\beta(x))|>g(\alpha(x))|$, $x\in\bar V$

(x называется в этом случае дефицитной вершиной), то x — g-заблокирована.

Р4. Вершина s — g-заблокирована.

Заметим, что если $f(u)$, $u\in U$,— тупиковый поток, то f является предпотоком, если же у предпотока g нет дефицитных вершин, то g — тупиковый поток.

В алгоритме строится некоторый (начальный) предпоток. Если он имеет дефицитные вершины, то производится перестройка с получением нового предпотока и так до тех пор, пока не будет получен предпоток, у которого нет дефицитных вершин.

В основе построения начального предпотока g_0 лежит *прямой послойный обход* вершин сети: сначала рассматривается вершина s (и производятся некоторые действия с ней и инцидентными ей дугами), затем в не-котором порядке рассматриваются все вершины слоя O_1, затем — O_2 и т. д. до $O_k=\{t\}$.

Положим сначала $g_0(\alpha(s))=c(\alpha(s))$. Пусть в процессе послойного обхода мы уже побывали в вершинах $s, x_1, \ldots, x_N$ и назначили $g_0(\alpha(s)), \ldots, g_0(\alpha(x_N))$. Перейдя к x_{N+1}, назначаем $g_0(\alpha(x_{N+1}))=c(\alpha(x_{N+1}))$, если $|c(\alpha(x_{N+1}))|<|g_0(\beta(x_{N+1}))|$, либо $g_0(\alpha(x_{N+1}))$ такой, что $|g_0(\alpha(x_{N+1}))|=|g_0(\beta(x_{N+1}))|$, если $|c(\alpha(x_{N+1}))|\geqslant|g_0(\beta(x_{N+1}))|$ * (по условию послойного обхода $g_0(\beta(x_{N+1}))$ уже определено). В первом случае вершина $x_{N+1}\neq\neq s, t$ будет дефицитной. Если x — дефицитная вершина построенного предпотока g_0, то любая дуга $u\in\alpha(x)$ является насыщенной и, следовательно, x—g_0-заблокированная вершина. По построению вершина s — g_0-заблокирована. Следовательно, g_0 — предпоток.

Пусть O_{d_0} ($d_0>0$) — слой, в котором есть дефицитная вершина x, а в слоях $O_{d_0+1}, \ldots, O_{k-1}$ дефицитных вершин нет.

В x проведем операцию *балансирования*: зададим $g_0'(u)$, $\forall u\in\beta(x)$ так, чтобы: 1) $g_0'(u)\leqslant g_0(u)$, 2) $|g_0(\alpha(x))|=|g_0'(\beta(x))|$, что, учитывая P2, всегда можно сделать.

Вершина x и инцидентные ей дуги объявляются *закрытыми* (при всех дальнейших перестройках значения предпотоков в $\alpha(x)$ и $\beta(x)$ меняться не будут). Если в O_{d_0} несколько дефицитных вершин, проведем балансирование для каждой. Во всех остальных дугах сети положим $g_0'=g_0$. Очевидно, закрытая дуга является g_0'-заблокированной. Если $d_0=1$, то построение заканчивается (g_0', как будет показано,— тупиковый поток).

В слоях $O_{d_0}, \ldots, O_{k-1}$ у g_0' не будет дефицитных вершин. Назовем эти слои и их вершины *собственными* для предпотока g_0. Дуга $u=(x, y)\in U$ объявляется собственной, если вершина y собственная. В слое O_{d_0-1} могут появиться новые, по сравнению с g_0, дефицитные вершины; g_0' уже не является, вообще говоря, предпотоком, так как новая дефицитная вершина $x\in O_{d_0-1}$ не является g_0'-заблокированной, если она не была g_0-заблокированной. Слой O_{d_0-1} назовем *источниковым*.

Будем строить следующий предпоток g_1, если у какой-либо дефицитной вершины слоя O_{d_0-1} есть незакрытая ненасыщенная исходящая дуга, иначе сразу же положим $g_1=g_0'$ и перейдем к балансированию.

1) Для каждой несобственной дуги $u\in U$ автоматически положим $g_1(u)=g_0'(u)$ $(=g_0(u))$.

2) Производим послойный обход, начиная с источникового слоя, при этом рассматриваем только незакрытые вершины. Для каждой закрытой дуги $u\in U$ автоматически полагаем $g_1(u)=g_0'(u)$.

Перейдя в обходе к вершине $x\neq t$, назначаем $g_1(\alpha_0(x))$, где $\alpha_0(x)$ — множество незакрытых дуг в $\alpha(x)$, так, чтобы выполнялось: а) $g_1(u)\geqslant g_0'(u)$, $\forall u\in\alpha_0(x)$, б) $|g_1(\alpha(x))|\leqslant|g_1(\beta(x))|$, в) если $|g_1(\alpha(x))|<|g_1(\beta(x))|$, то $g_1(u)=c(u)$, $\forall u\in\alpha_0(x)$ (если $|g_1(\alpha(x))|>|g_0'(\alpha(x))|$, то скажем, что в вершине x было произведено активное назначение); g_1 является предпотоком. Действительно, P1, P2, P4, очевидно, выполняются. P3 следует из того, если $x\in\bigcup\limits_{l=d_0-1}^{k-1}O_l$ — дефицитная вершина, то любая дуга $u\in\alpha(x)$ либо закрытая (тогда она g_0'-заблокирована и, следовательно, g_1-заблокирована), либо насыщенная g_1. Построив g_1, находим слой O_{d_1}, содержащий дефицитные вершины (слои $O_{d_1+1}, \ldots, O_{k-1}$ не имеют дефицитных вершин).

---

* Это можно сделать различными способами. Для получения нужной оценки используется метод, описанный в разд. 4, п. 3.

Операция балансирования для g_1 (и всех последующих предпотоков) производится некоторым специальным образом, так что в любой ранее закрытой дуге $u\in U$ будет обеспечено выполнение $g_1'(u)=g_1(u)$ $(=g_0'(u))$. Для этого используется техника, описанная в разд. 4. Вершины, в которых производилось балансирование, и инцидентные им дуги объявляем закрытыми. Если $d_1\neq 1$, то переходим к построению предпотока g_2 и т. д.

На каждой итерации алгоритма появляется по крайней мере одна новая закрытая вершина (в старых закрытых вершинах предпотоки не изменяются и дефицит отсутствует). Следовательно, число итераций не свыше $n-2$. У последнего построения предпотока g_m либо нет дефицита вершин, т. е. g_m — поток, либо $d_m=1$, тогда g_m' — поток. Покажем, что g_m (или g_m') — тупиковый.

Лемма. *У g_j и у g_j', $0\leqslant j\leqslant m$, одно и то же множество заблокированных вершин.*

Доказательство. Покажем, что если вершина $x\in V$ — g_j-заблокированная, то x — g_j'-заблокирована. Если x — собственная вершина для g_j, то это очевидно. Пусть $x\in \bigcup\limits_{i=0}^{d_j-1} O_i$. Любой путь ξ из x в t либо не проходит через какую-нибудь вершину $y\in O_{d_j}$, подвергшуюся балансированию, тогда для любой дуги $u\in\xi$ справедливо: $g_j'(u)=g_j(u)$ и путь ξ — g_j'-заблокирован, либо проходит, и так как y — g_j'-заблокирована, то ξ — g_j'-заблокирован.

Доказательство обратного утверждения очевидно. Так как $g_{j+1}\geqslant g_j'$, $j=0,\dots,m-1$, а также в силу леммы, вершина s, будучи g_0-заблокированной, является g_m-заблокированной (или g_m'-заблокированной), следовательно, g_m (или g_m') — тупиковый поток.

4. ОРГАНИЗАЦИЯ ИНФОРМАЦИИ

1) Вместо набора величин $g_0(u)$, $g_0'(u),\dots,g_m(u)$, $g_m'(u)$ в дуге $u\in U$ поддерживается число $g(u)$ — значение последнего предпотока (или результата балансирования), построенного в дуге u к данному моменту (индексы при предпотоках будем употреблять при объяснении, когда будет нужно подчеркнуть, о какой по счету итерации идет речь).

2) Скажем, что множество $M=\{m_1, m_2,\dots,m_p\}$ задано в виде списка, если фиксировано некоторое упорядочение $m_{i_1},\dots,m_{i_p}$, которое задается указанием начала $(m_{i_1}=m_{\text{нач}})$, конца $(m_{i_p}=m_{\text{кон}})$ и для каждого m_{i_j}, $j=2,\dots,p-1$,— предыдущего $(m_{i_{j-1}})$ и последующего $(m_{i_{j+1}})$ элементов (у m_{i_1} только последующего, а у m_{i_p} только предыдущего). Если в M добавляется или из M вычеркивается один элемент, то в списке соответствующие изменения производятся за $O(1)$ действий.

В виде списков задаются: а) множества $\bar\alpha(x)=\{u$: $u\in\alpha(x)$, $g(u)<c(u)$ & u — незакрытая$\}$, $\forall x\in V\setminus\{t\}$. Вначале множества $\bar\alpha(x)$ совпадают с $\alpha(x)$, б) множества $\beta_\Delta(x)\subseteq\beta(x)$, $\forall x\in V\setminus\{s\}$, определяемые ниже. Для каждой дуги $u\in\beta_\Delta(x)$ задается число $\Delta(u)>0$ («добавка»).

3) Уточним правила алгоритма. а) При активном назначении $g_i(\alpha(x))$, $x\in\bar V$, последовательно идем по списку $\bar\alpha(x)$ с начала и насыщаем дуги, т. е., дойдя до дуги $u\in\bar\alpha(x)$, полагаем $g_i(u)=\min\,\{c(u),\ |g_i(\beta(x))|-|g_i(\hat\alpha(x))|\}$, где $\hat\alpha(x)\subset\alpha(x)$ — множество дуг, в которых g_i уже построен. Если дуга u насыщается, она вычеркивается из списка (тем самым сдвигается начало списка). Когда $|g_i(\hat\alpha(x))|$ станет равным $|g_i(\beta(x))|$ либо когда исчерпается весь $\bar\alpha(x)$ (x станет тогда, вообще говоря, дефицитной вершиной), построение $g_i(\alpha(x))$ заканчивается. Очевидно правила 2 а), б), в) разд. 3

будут соблюдены. Одновременно с активным назначением g_i в $u=(x, y)\in\bar\alpha(x)$ дуга u вносится в конец списка $\beta_\Delta(y)$ и вычисляется «добавка» $\Delta(u)=g_i(u)-g_{i-1}'(u)>0$. Для каждого списка β_Δ запоминается номер итерации, на которой он возник. Если на i-й итерации нужно внести, быть может заново, в β_Δ дугу $u\in U$, а β_Δ имеет номер j, $j<i$, то «старый» список β_Δ отменяется и вместо него заводится новый, начинающийся с дуги u и имеющий номер i.

б) Если дуга $u=(x, y)\in U$ закрывается при балансировании x или y, то она вычеркивается из списков $\beta_\Delta(y)$ или $\bar\alpha(x)$ (если присутствовала там) и уничтожается добавка $\Delta(u)$.

в) Пусть на i-й итерации производится балансирование вершины $x\in O_d{}^i$. Тогда $g_i'(\beta_0(x))$, где $\beta_0(x)\subseteq\beta(x)$ — множество незакрытых дуг, назначается так, что $g_i'(u)\geqslant g_i(u)-\Delta(u)$, $\forall u\in\beta_0(x)$ (т. е. уменьшение предпотока делается за счет добавок). В разд. 6 доказывается, что такое балансирование всегда возможно, т. е. $|\Delta(\beta(x))|\geqslant|g_i(\beta(x))|-|g_i(\alpha(x))|$. (Технически при назначении $g_i'(\beta_0(x))$ идем последовательно по списку $\beta_\Delta(x)$ и назначаем для очередной дуги $u\in\beta_\Delta(x)$: $g_i'(u)=\max\{g_i(u)-\Delta(u), |g_i(\alpha(x))|-|g_i'(\hat\beta_\Delta(x))|-|g_i(\beta(x)\setminus\hat\beta_\Delta(x)\setminus u)|\}$, где $\hat\beta_\Delta(x)\subset\beta_\Delta(x)$ — множество дуг с уже назначенным g_i'.)

5. ОЦЕНКА ЧИСЛА ДЕЙСТВИЙ

1) Число итераций алгоритма, как показано в разд. 3, меньше n.

2) В течение одной итерации число «действий с вершинами» (т. е. переход от вершины к вершине, нахождение слоя O_{d_i} и дефицитных вершин в нем) — $O(n)$ и в течение всей работы — $O(n^2)$.

3) Действия с дугой после ее закрытия не производятся.

4) При активном назначении предпотока в дуге $u=(x, y)\in U$ она либо насыщается (событие $A(u)$), либо нет (событие $B(u)$).

5) После того как произошло событие $A(u)$, число действий с дугой u до конца работы алгоритма — $O(1)$.

6) Пусть $r(A)$ и $r(B)$ — число событий A и B за всю работу алгоритма. $r(A)=O(p)$.

7) В течение одной итерации в каждом множестве $\alpha(x)$, $x\in V$, происходит не более одного события B (см. п. 3а), разд. 4), следовательно, число событий B за одну итерацию — $O(n)$, за всю работу — $O(n^2)$.

8) После события $B(u)$ до следующего события ($B(u)$ или $A(u)$) действия с дугой u не производятся.

9) Пока в дуге $u\in U$ сохраняется $g(u)=0$, действия с дугой u не производятся.

Из вышеизложенного следует, что число «действий с дугами» за всю работу алгоритма — $O(r(A)+r(B))$ или $O(p+n^2)$. Следовательно, число действий на построение тупикового потока в справочной — $O(n^2)$, максимального потока в сети — $O(n^3)$.

6. КОРРЕКТНОСТЬ АЛГОРИТМА

Осталось доказать, что правило 3в) из разд. 4 всегда корректно, т. е. к моменту, когда требуется произвести балансирование в вершине $x\in\bar V$, в ней выполняется $|\Delta(\beta(x))|\geqslant|g(\beta(x))|-|g(\alpha(x))|$. Докажем это неравенство вообще для произвольной вершины $x\in\bar V$ и момента окончания построения очередного g_i, $i=0, 1, \ldots, m$ (перед балансированием). Для краткости назовем его «i-й момент».

Будем приписывать добавке $\Delta(u)$, $u \in U$ (если она существует) индекс $\delta(u)$ — номер той итерации, на которой она возникла. Определим в i-й момент, $i=0, 1, \ldots, m$, функцию

$$\tilde{\Delta}_i(u) = \begin{cases} \Delta(u), \text{ если дуга } u \text{ — несобственная для итерации } i-1, \\ \Delta(u), \text{ если дуга } u \text{ — собственная для итерации } i-1 \text{ и} \\ \qquad \delta(u) = i, \\ 0, \text{ в остальных случаях.} \end{cases}$$

В частности, если дуга u — собственная, то $\tilde{\Delta}_i(u) = g_i(u) - g'_{i-1}(u)$.

Лемма. *Если $x \in \bar{V}$ — дефицитная вершина в момент i, $0 \leqslant i \leqslant m$, то для любой дуги $u \in \beta(x)$ выполняется*: $\tilde{\Delta}_i(u) = \Delta(u)$.

Доказательство. Пусть по индукции лемма верна в моменты $0, 1, \ldots, i-1$, $i \leqslant m$. Для первого момента это очевидно. Рассмотрим произвольную дефицитную вершину $x \in \bar{V}$ в момент i: 1) x — несобственная вершина на итерации $i-1$, тогда с момента $i-1$ до i действия с $\beta(x)$ не производятся и $\tilde{\Delta}_i(u) = \tilde{\Delta}_{i-1}(u) = \Delta(u)$, $u \in \beta(x)$; 2) x — собственная вершина на итерации $i-1$, тогда выполняется $|g'_{i-1}(\beta(x))| = |g'_{r-1}(\alpha(x))|$, следовательно, $|g_i(\beta(x))| > |g_i(\alpha(x))| \geqslant |g'_{i-1}(\beta(x))|$, т. е. в вершине x производилось активное назначение, и по правилу 3а) из разд. 4 на $i-1$-й итерации заводился новый список $\beta_\Delta(x)$, откуда $\tilde{\Delta}_i(\beta(x)) = \tilde{\Delta}(\beta(x))$. Лемма доказана.

Теорема. *В момент i, $0 \leqslant i \leqslant m$, выполняется*

$$|g_i(\beta(x))| - |\tilde{\Delta}_i(\beta(x))| \leqslant |g_i(\alpha(x))| - |\tilde{\Delta}_i(\alpha(x))|. \tag{1}$$

Доказательство. Теорема верна для момента 1, поскольку $g_1(u) = \Delta(u) = \tilde{\Delta}_1(u)$, $\forall u \in U$. Пусть она справедлива для моментов $0, 1, \ldots, i-1$ (и, значит, в эти моменты правила балансирования п. 3в) разд. 4 были разрешимы). Докажем ее для момента i.

Рассмотрим произвольную вершину $x \in V$. Возможны следующие случаи.

1) x — собственная вершина на итерации $i-1$. Поскольку $g_i(u) = g'_{i-1}(u) + \tilde{\Delta}_i(u)$, $\forall u \in \alpha(x) \cup \beta(x)$, то $|g_i(\beta(x))| - |\tilde{\Delta}_i(\beta(x))| = |g'_{i-1}(\beta(x))| = |g'_{i-1}(\alpha(x))| = |g_i(\alpha(x))| - \tilde{\Delta}_i(\alpha(x))|$, что и требовалось.

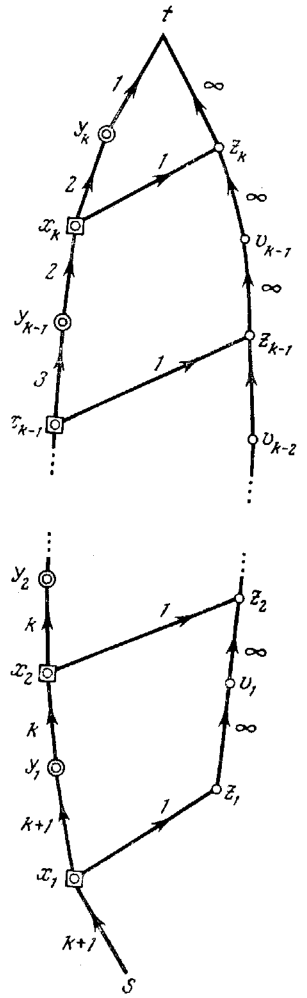
2) $x \in \bigcup\limits_{l-1}^{d_{l-1}-2} O_l$. Тогда в множествах $\alpha(x)$ и $\beta(x)$ с момента $i-1$ до i действия не производятся и соотношение (1) сохраняется.

3) $x \in O_{d\,{}_{-1}}{}^{i}{}_{-1}$. Рассмотрим отдельные операции, при которых величины, входящие в соотношение (1), могли измениться. Заметим, что левая часть (1) с момента $i-1$ до i не изменилась, так как $\beta(x)$ — множество несобственных дуг для $i-1$-й итерации.

а) Величины $g(u)$ и $\Delta(u)$ для некоторых $u \in \alpha(x)$ изменились при балансировании вершин слоя $O_{d\,-1}$. Рассмотрим момент окончания балансирования (оно, по предположению индукции, корректно): $g'_{i-1}(u) \geqslant g_{i-1}(u) - \tilde{\Delta}_{i-1}(u)$, $\forall u \in \alpha(x)$, или $|g'_{i-1}(\check{\alpha}(x))| \geqslant |g_{i-1}(\alpha(x))| - |\tilde{\Delta}_{i-1}(\check{\alpha}(x))| \geqslant |g_{i-1}(\alpha(x))| - |\tilde{\Delta}_{i-1}(\alpha(x))|$, где $\check{\alpha}(x) \subseteq \alpha(x)$ — множество дуг, подвергшихся закрытию (в результате балансирования) на $i-1$-й итерации. Отсюда

$$|g'_{i-1}(\alpha(x))|\geqslant|g_{i-1}(\beta(x))|-$$
$$-|\tilde{\Delta}_{i-1}(\beta(x))|. \qquad (2)$$

б) Так как дуги множества $\alpha(x)$ — собственные на $i-1$-й итерации, то в результате достройки на i-й итерации выполняется: $|g_i(\alpha(x))|=|g'_{i-1}(\alpha(x))|+|\tilde{\Delta}_i(\alpha(x))|$. Отсюда в силу (2) получаем требуемое соотношение: $|g_i(\alpha(x))|-|\tilde{\Delta}_i(\alpha(x))| = |g'_{i-1}(\alpha(x))|\geqslant|g_{i-1}\cdot(\beta(x))| - |\Delta_{i-1}(\beta(x))| = |g_i(\beta(x))| - |\Delta_i(\beta(x))|$.
Теорема доказана.



7. ЭКСТРЕМАЛЬНЫЙ ПРИМЕР

Приведем пример справочной S_{2k+1}, число вершин и дуг в которой имеет порядок k и тупиковый поток находится не менее чем за Ck^2 действий, где $C>0$ — константа, не зависящая от k. Тем самым будет доказана точность установленной оценки для метода предпотоков. Вершинами в S_{2k+1} будут $s, t, x_1, \ldots, x_k, y_1, \ldots, y_k, z_1, \ldots, z_k, v_1, \ldots, v_{k-1}$. Вершины соединены дугами, как показано на рисунке. Числа на дугах равны их пропускным способностям. Для вершин $x_1, \ldots, x_k$ списки $\alpha(x_1), \ldots, \alpha(x_k)$ устроены так, что первыми являются дуги вида (x_i, y_i). Нетрудно убедиться, что начальный предпоток отличен от 0 только на дугах вида (s, x_1), (y_k, t), (x_i, y_i), (x_i, y_{i+1}), причем на них он равен пропускным способностям. Все вершины y_i будут дефицитными. Балансирование на первой итерации произойдет в вершине y_k, при этом слой, содержащий вершину x_k, станет источниковым. На второй итерации предпоток будет построен путем увеличения функции на дугах пути, проходящего через вершины x_k, z_k, t.

Вообще, на i-й итерации, $i=1, 2, \ldots, k$, будет производиться балансирование в вершине y_{k-i+1}. Источниковым слоем при этом становится слой, содержащий вершину x_{k-i+1}. При построении предпотока на $i+1$-й итерации происходит возрастание (на единицу) функции на дугах пути, проходящего через вершины $x_{k-i+1}, z_{k-i+1}, v_{k-i+1}, \ldots, z_k, t$ (всего в $2i+2$ дугах). Таким образом, число действий на $i+1$-й итерации — $O(i)$ и на построение тупикового потока — $O\left(\sum_{i=1}^{k} i\right)$, или $O(k^2)$.

Отметим, что, исходя из рассмотренного примера, можно построить пример сети с n вершинами для алгоритма в целом, для которого число действий будет не менее $C'n^3$, где $C'>0$ — не зависящая от n константа.

ЛИТЕРАТУРА

1. *Е. А. Диниц.* Алгоритм решения задачи о максимальном потоке в сети со степенной оценкой. Докл. АН СССР, 1970, т. 194, № 4.
2. *Л. Форд, Д. Фалкерсон.* Потоки в сетях. М., «Мир», 1966.

Поступила в редакцию
25 VI 1973