Лекция 16: Потоки

Б.М.Верников, А.М.Шур

Уральский федеральный университет,
Институт математики и компьютерных наук,
кафедра алгебры и дискретной математики

Пусть дана сеть, имеющая ровно один источник («пункт А») и ровно один сток («пункт Б»). Предположим, что эта сеть представляет собой схему транспортных потоков. Дуги сети изображают допустимые направления движения, вершины — пункты, в которых можно сменить направление движения, а числа, поставленные в соответствие дугам, интерпретируются как *пропускные способности* дуг, т. е. как количество единиц груза, которые можно пропустить по данной дуге за фиксированную единицу времени. Предполагая, что из пункта А в пункт Б происходит непрерывный грузопоток (т. е. в каждую единицу времени сеть «заполнена» грузами, отправляющимися из А, прибывающими в Б и находящимися в пути), рассмотрим два естественных вопроса.

- Какое максимальное число единиц груза можно доставлять в Б за единицу времени и как нужно для этого распределить грузы между направлениями движения?
- Какие участки сети являются ее «узким местом»? Иначе говоря, пропускную способность какой дуги (дуг) надо увеличить, чтобы количество доставляемых грузов увеличилось?

Данные вопросы и составляют неформальную постановку *задачи о максимальном потоке*, которой посвящена данная лекция.

В дальнейшем мы рассматриваем только те сети (G, α), в которых

- имеется ровно один источник s и ровно один сток t;
- любая вершина достижима из источника и из любой вершины достижим сток (если в сети есть вершины, не удовлетворяющие какому-то из этих условий, то они бесполезны с точки зрения доставки грузов из источника в сток, а значит, в рамках рассматриваемой задачи можно считать, что они удалены из сети).

Число $\alpha(u, v)$ будем называть *пропускной способностью* дуги (u, v).

Определение

Функция φ с неотрицательными значениями, заданная на дугах сети (G, α), называется *потоком*, если
(i) для всякой дуги (u, v) выполнено неравенство $\varphi(u, v) \leqslant \alpha(u, v)$;
(ii) выполнено *условие сбалансированности потока*: для всякой вершины сети, отличной от источника и стока, сумма значений потока на входящих дугах равна сумме значений потока на исходящих дугах.

Отметим, что

- вместо «значение потока на дуге» обычно говорят «поток вдоль дуги»;
- в любой сети можно задать *нулевой* поток, равный нулю вдоль всех дуг.

Пример потока

Изображая поток в сети на рисунках, мы будем указывать рядом с каждой дугой два числа в формате $x(y)$. Число x обозначает пропускную способность дуги, а y — поток вдоль дуги. Пример потока в сети приведен на рис. 1.

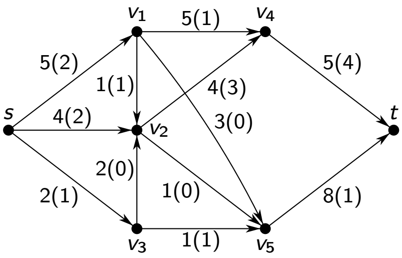


Рис. 1. Поток в сети

Определение

Сумма потоков вдоль дуг, исходящих из источника, называется *величиной потока* и обозначается $|\varphi|$ для потока φ. Поток называется *максимальным*, если его величина максимальна среди всех потоков в данной сети.

Лемма о величине потока

В любой сети сумма потоков вдоль дуг, исходящих из источника, равна сумме потоков вдоль дуг, входящих в сток.

Доказательство. Пусть G — сеть с множеством вершин $\{v_1, v_2, \ldots, v_n\}$, источником $v_1 = s$ и стоком $v_n = t$. Для любого $i = 1, \ldots, n$ обозначим через $\varphi^+(v_i)$ сумму потоков вдоль всех дуг, входящих в v_i, а через $\varphi^-(v_i)$ — сумму потоков вдоль всех дуг, выходящих из v_i. Обозначим через S сумму потоков вдоль всех дуг сети. Ясно, что $\sum\limits_{i=1}^{n} \varphi^-(v_i) = S = \sum\limits_{i=1}^{n} \varphi^+(v_i)$.

По определению потока $\varphi^-(v_i) = \varphi^+(v_i)$ для всех $i = 2, \ldots, n-1$. Так как $\varphi^-(v_n) = \varphi^+(v_1) = 0$, получаем $\varphi^-(v_1) = \varphi^+(v_n)$. □

- Ввиду доказанной леммы, первый из вопросов, поставленных в начале лекции, на языке орграфов звучит как «найти величину максимального потока в сети и сам максимальный поток».

Введем еще одно важное понятие.

Определение

Разрезом сети (G, α) с источником s и стоком t называется множество $\{e_1, \ldots, e_k\}$ дуг сети, обладающее следующими свойствами:
(i) в орграфе $G - \{e_1, \ldots, e_k\}$ нет (s, t)-цепей;
(ii) если любую из дуг $e_1, \ldots, e_k$ добавить к орграфу $G - \{e_1, \ldots, e_k\}$, то в полученном орграфе найдется (s, t)-цепь.

Например, множество всех дуг, исходящих из источника, является разрезом в сети.

Определение

Пропускной способностью разреза называется сумма пропускных способностей дуг, входящих в разрез. Разрез называется *минимальным*, если его пропускная способность минимальна среди пропускных способностей всех разрезов сети.

Следующий основной результат о связи потоков и разрезов мы приводим без доказательства.

Теорема Форда–Фалкерсона

В любой сети величина максимального потока равна пропускной способности минимального разреза. $\square$

- Теорема Форда–Фалкерсона показывает, что с содержательной точки зрения минимальный разрез — это и есть упомянутое в начале лекции «узкое место» сети: увеличив пропускную способность любой дуги, входящей в минимальный разрез, мы увеличим максимальный поток в сети.

Теперь мы можем сформулировать задачу о максимальном потоке на языке орграфов:

Задача о максимальном потоке

В заданной сети найти максимальный поток и минимальный разрез.

Увеличение потока

Для построения максимального потока в сети нужно вначале понять, как увеличить уже имеющийся поток. Основная хитрость состоит в том, что

- чтобы увеличить поток в сети, может потребоваться поток вдоль некоторых дуг *уменьшить*.

Рассмотрим сеть на рис. 2. Поток, указанный на рисунке слева, не является максимальным. Если мы хотим его увеличить, пустив поток вдоль дуги (s, y) (а других вариантов нет), то вершина y «переполнится», ведь пропускная способность единственной исходящей из y дуги равна 1. Справиться с этим переполнением можно только одним способом — уменьшить поток вдоль дуги (x, y), перенаправив его вдоль дуги (x, t). В результате этого действия мы и получим поток, изображенный на рис. 2 справа — очевидно, максимальный для данной сети.

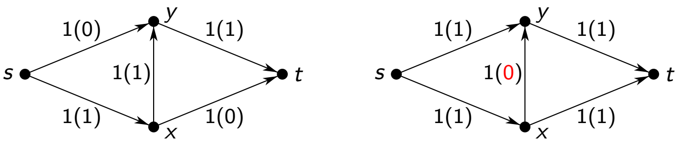


Рис. 2

- Похожая ситуация возникала при поиске максимального паросочетания (лекция 8): чтобы увеличить размер паросочетания, некоторые ребра из него приходилось выбрасывать.

Чтобы находить возможности увеличения потока, подобные разобранной на предыдущем слайде, вводится важная вспомогательная конструкция.

Определение

Остаточной сетью $(\hat{G}, \alpha_\varphi)$ называется сеть, полученная из сети (G, α) с заданным потоком φ применением двух преобразований:
(i) для каждой пары вершин $u, v \in V(G)$ таких, что $(u, v) \in E(G)$ и $(v, u) \notin E(G)$, добавить к орграфу G дугу (v, u) пропускной способности 0;
(ii) для каждой дуги $(u, v) \in E(G)$ такой, что $\varphi(u, v) > 0$, уменьшить пропускную способность дуги (u, v) на $\varphi(u, v)$ и увеличить пропускную способность дуги (v, u) на $\varphi(u, v)$.

Остаточная сеть для примера с предыдущего слайда выглядит как на рис. 3 (красным выделены добавленные дуги). Заметим, что в ней есть цепь $s \xrightarrow{1} y \xrightarrow{1} x \xrightarrow{1} t$.

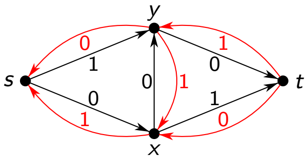


Рис. 3. Остаточная сеть

Определение

Пропускной способностью цепи в сети называется минимум из пропускных способностей входящих в нее дуг.

Лемма об остаточной сети

Если в остаточной сети $(\hat{G}, \alpha_\varphi)$ существует простая (s,t)-цепь пропускной способности $\delta{>}0$, то величину потока в сети (G,α) можно увеличить на δ.

Доказательство. Пусть C — упомянутая (s,t)-цепь. Изменим значения потока на дугах сети (G,α) по следующим правилам:

(1) $\varphi'(u,v) = \varphi(u,v) + \delta$, если $(u,v) \in C$;

(2) $\varphi'(u,v) = \varphi(u,v) - \delta$, если $(v,u) \in C$;

(3) $\varphi'(u,v) = \varphi(u,v)$ в остальных случаях.

Докажем, что φ' — поток и $|\varphi'| = |\varphi| + \delta$. Так как $\delta \leqslant \alpha_\varphi(u,v)$ для любой дуги $(u,v) \in C$, то по определению остаточной сети для правила (1) имеем $\varphi(u,v) + \delta \leqslant \alpha(u,v)$, а для правила (2), соответственно, $\varphi(u,v) - \delta \geqslant 0$. С учетом того, что φ — поток, получаем, что функция φ' неотрицательна и не превосходит α.

Лемма об остаточной сети: окончание доказательства

Проверим выполнение условия сбалансированности потока для φ'. Так как это условие выполнено для φ, равенство $\varphi^+(v) = \varphi^-(v)$ выполнено для любой вершины $v \in V(G)$, отличной от s и t. Если v не входит в цепь C, то поток вдоль дуг, инцидентных v, не изменится и равенство сохранится. В противном случае в C есть пара дуг (u, v) и (v, w). При изменении потока вдоль дуги (u, v) по правилу (1) левая часть равенства увеличивается на δ, а при изменении по правилу (2) правая часть уменьшается на δ. Аналогично, при изменении потока вдоль (v, w) либо правая часть увеличивается на δ, либо левая часть уменьшается на δ; в любом случае, равенство будет восстановлено, т. е. условие сбалансированности выполнено. Итак, φ' — поток по определению.

Поскольку C — простая цепь, в ней ровно одна исходящая из s дуга и нет дуг, входящих в s. Увеличение потока вдоль дуги, исходящей из s, происходит согласно правилу (1) (исходящие из s дуги в сетях $\hat{G}$ и G одни и те же, поскольку в G нет входящих в s дуг). Значит,

$$|\varphi'| = \varphi'^-(s) = \varphi^-(s) + \delta = |\varphi| + \delta. \qquad \square$$

Применив доказанную лемму к нашему примеру, возьмем в остаточной сети на рис. 3 цепь $s \xrightarrow{1} y \xrightarrow{1} x \xrightarrow{1} t$, увеличим на единицу поток вдоль дуг (s, y) и (x, t), уменьшим на единицу поток вдоль (x, y), и в результате получим максимальный поток в исходной сети (см. рис. 2 справа).

Изложим алгоритм решения задачи о максимальном потоке.

Алгоритм Форда–Фалкерсона

Вход: сеть (G, α) с источником s и стоком t.

Выход: максимальный поток φ и минимальный разрез R в сети (G, α).

1. Для всякой дуги (u, v) положить $\varphi(u, v) = 0$.
2. Построить остаточную сеть $(\hat{G}, \alpha_\varphi)$.
3. Если в $(\hat{G}, \alpha_\varphi)$ найдется (s, t)-цепь C пропускной способности $\delta > 0$, изменить поток φ вдоль C по правилам (1)–(3) и вернуться на шаг 2.
4. Найти множество S, состоящее из всех вершин v сети, для которых в $(\hat{G}, \alpha_\varphi)$ существует (s, v)-цепь ненулевой пропускной способности; положить $R = \{(u, v) \in E(G) \mid u \in S, v \notin S\}$.

По лемме об остаточной сети, на каждой итерации шагов 2 и 3 алгоритм строит поток строго большей величины, чем предыдущий. Если все пропускные способности — целые числа, то число итераций не превосходит суммы пропускных способностей дуг, исходящих из источника. В частности, алгоритм рано или поздно выполнит шаг 4 и остановится. (Существуют примеры сетей с иррациональными пропускными способностями, для которых алгоритм может зациклиться, но мы не будем на этом останавливаться.)

Докажем, что выходом алгоритма Форда–Фалкерсона являются именно максимальный поток и минимальный разрез. Заметим, что $t \notin S$, поскольку иначе алгоритм перейдет на шаг 2 вместо шага 4; отсюда следует, что $R \neq \varnothing$. Пусть $T = V(G) \setminus S$.

Если $(u, v) \in E(G)$, $u \in T$ и $v \in S$, то $\varphi(u, v) = 0$ (иначе у «обратной» дуги (v, u) в сети $(\hat{G}, \alpha_\varphi)$ была бы ненулевая пропускная способность, что вместе с наличием (s, v)-цепи ненулевой пропускной способности означало бы, что $u \in S$). В то же время, $\varphi(u, v) = \alpha(u, v)$ для каждой из дуг множества R, так как в остаточной сети эти дуги имеют нулевую пропускную способность по определению множества R. Следовательно, для любой дуги $(u, v) \in R$ из вершины v достижим сток t. Действительно, $\varphi(u, v) > 0$, а значит, существует (v, t)-цепь, вдоль всех дуг которой поток положителен; по доказанному выше, в этой цепи нет вершин из S. В итоге, R — разрез по определению, так как в орграфе $G - R$ нет (s, t)-цепей, а добавление к нему дуги (u, v) такую цепь обеспечивает.

Мы доказали, что из T в S «ничего не течет». Тогда ввиду свойства сбалансированности суммарный поток на дугах из S в T должен равняться $|\varphi|$. Но эта же сумма является, по определению, пропускной способностью разреза R; ввиду теоремы Форда–Фалкерсона, поток должен быть максимальным, а разрез — минимальным.

Важным элементом реализации алгоритма Форда–Фалкерсона является проверка условия на шаге 3, т. е. поиск в остаточной сети (s, t)-цепи ненулевой пропускной способности. Следующая процедура находит кратчайшую по числу дуг цепь с требуемыми свойствами. Для каждой вершины v вычисляются $\delta(v)$ — пропускная способность найденной (s, v)-цепи — и $P(v)$ — предшествующая v вершина в этой цепи.

Процедура поиска (s, t)-цепи ненулевой пропускной способности

1. Положить $\delta(s) = \infty$, $\delta(v) = 0$ для остальных вершин, а также $P(v) = \varnothing$ для всех вершин; создать очередь из одной вершины s.
2. Пока очередь не пуста, повторять
   - извлечь первый элемент из очереди (обозначим его через v);
   - для каждой дуги (v, w) такой, что $\alpha_\varphi(v, w) > 0$ и $\delta(w) = 0$, положить $\delta(w) = \min\{\delta(v), \alpha_\varphi(v, w)\}$; если $w = t$, закончить работу, иначе поместить w в конец очереди.

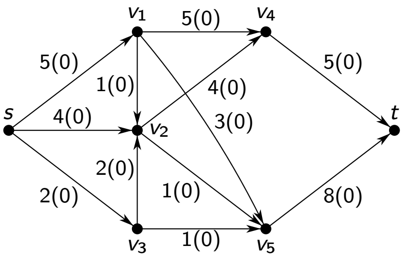
Если процедура завершила работу по условию $w = t$, то значения $P(t)$, $P(P(t))$, . . . определяют (s, t)-цепь с пропускной способностью $\delta(t) > 0$. Если же процедура завершилась ввиду исчерпания очереди, то такой цепи не существует, а вершины с условием $\delta(v) > 0$ образуют множество S, необходимое на шаге 4 алгоритма Форда–Фалкерсона.

Найдем максимальный поток и минимальный разрез в сети, изображенной на рис. 1.

Эта же сеть с нулевым потоком изображена на рис. 4 слева. Остаточную сеть будем представлять без рисования в явном виде, запомнив, что

- по дугам можно двигаться и против стрелок;
- пропускная способность дуги «против стрелки» равна второму числу (в скобках), а «по стрелке» — разности первого и второго чисел.

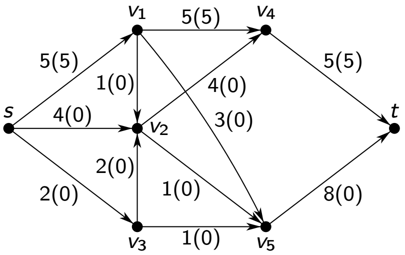
Справа на рис. 4 приведен «протокол» работы процедуры поиска (s, t)-цепи в остаточной сети (порядок строк соответствует порядку присвоения значений). В результате работы найдена цепь $s \to v_1 \to v_4 \to t$ пропускной способности 5 (см. выделенные значения в протоколе).



v	$P(v)$	$\delta(v)$
s	$\varnothing$	∞
v_1	s	5
v_2	s	4
v_3	s	2
v_4	v_1	5
v_5	v_1	3
t	v_4	5

Рис. 4

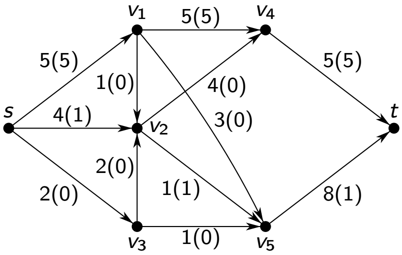
Увеличим поток вдоль дуг (s, v_1), (v_1, v_4) и (v_4, t) на $\delta(t) = 5$ в соответствии с правилом 1 увеличения потока. Получим поток, изображенный на рис. 5 слева. Протокол работы процедуры поиска (s, t)-цепи в получившейся остаточной сети приведен справа. Обратите внимание, что вершина v_1 была достигнута из v_4 «против стрелки». В результате найдена цепь $s \to v_2 \to v_5 \to t$ пропускной способности 1.



v	$P(v)$	$\delta(v)$
s	$\varnothing$	∞
v_2	s	4
v_3	s	2
v_4	v_2	4
v_5	v_2	1
v_1	v_4	4
t	v_5	1

Рис. 5

Увеличим поток вдоль дуг (s, v_2), (v_2, v_5) и (v_5, t) на $\delta(t) = 1$ в соответствии с правилом 1 увеличения потока. На рис. 6 слева приведен полученный поток, справа — протокол работы процедуры поиска (s, t)-цепи. В результате найдена цепь $s \to v_3 \to v_5 \to t$ пропускной способности 1.

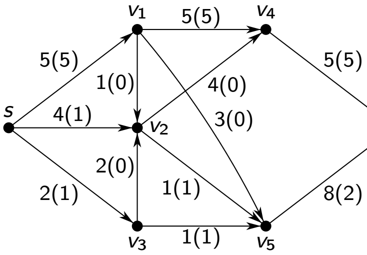


v	$P(v)$	$\delta(v)$
s	$\varnothing$	∞
v_2	s	3
v_3	s	2
v_4	v_2	3
v_5	v_3	1
v_1	v_4	3
t	v_5	1

Рис. 6

Увеличим поток вдоль дуг (s, v_3), (v_3, v_5) и (v_5, t) на $\delta(t) = 1$ в соответствии с правилом 1 увеличения потока. На рис. 7 слева приведен полученный поток, справа — протокол работы процедуры поиска (s, t)-цепи. В результате найдена цепь $s \to v_2 \to v_4 \to v_1 \to v_5 \to t$ пропускной способности 3.

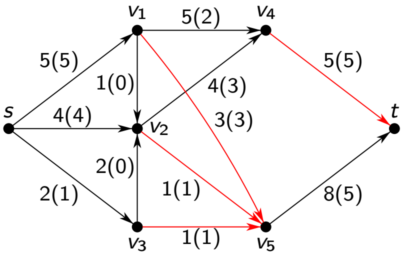
Найденная (s, t)-цепь содержит «обратную» дугу (v_4, v_1); это значит, что поток вдоль дуги (v_1, v_4) нужно уменьшить на 3 в соответствии с правилом 2 увеличения потока, а поток вдоль дуг (s, v_2), (v_2, v_4), (v_1, v_5) и (v_5, t) нужно увеличить на 3 в соответствии с правилом 1. Результат приведен на рис. 8 на следующем слайде.



v	$P(v)$	$\delta(v)$
s	$\varnothing$	∞
v_2	s	3
v_3	s	1
v_4	v_2	3
v_1	v_4	3
v_5	v_1	3
t	v_5	3

Рис. 7

Протокол работы процедуры поиска (s, t)-цепи в очередной остаточной сети показывает, что таких цепей ненулевой пропускной способности нет и находит множество $S = \{s, v_1, v_2, v_3, v_4\}$. После этого на шаге 4 находим разрез $R = \{(v_1, v_5), (v_2, v_5), (v_3, v_5), (v_4, t)\}$ (выделен красным). Пропускная способность R и величина максимального потока φ равны 10.



v	$P(v)$	$\delta(v)$
s	$\varnothing$	∞
v_3	s	1
v_2	v_3	1
v_4	v_2	1
v_1	v_4	1
очередь пуста		

Рис. 8