Потоковые Алгоритмы

Лекция 7

Единичные пропускные способности

- Пусть пропускные способности всех дуг равны между собой и равны 1.
- Тогда целочисленный s-t-поток можно рассматривать как набор непересекающихся по дугам (ребрам) s-t-путей.

Первая Теорема Менгера

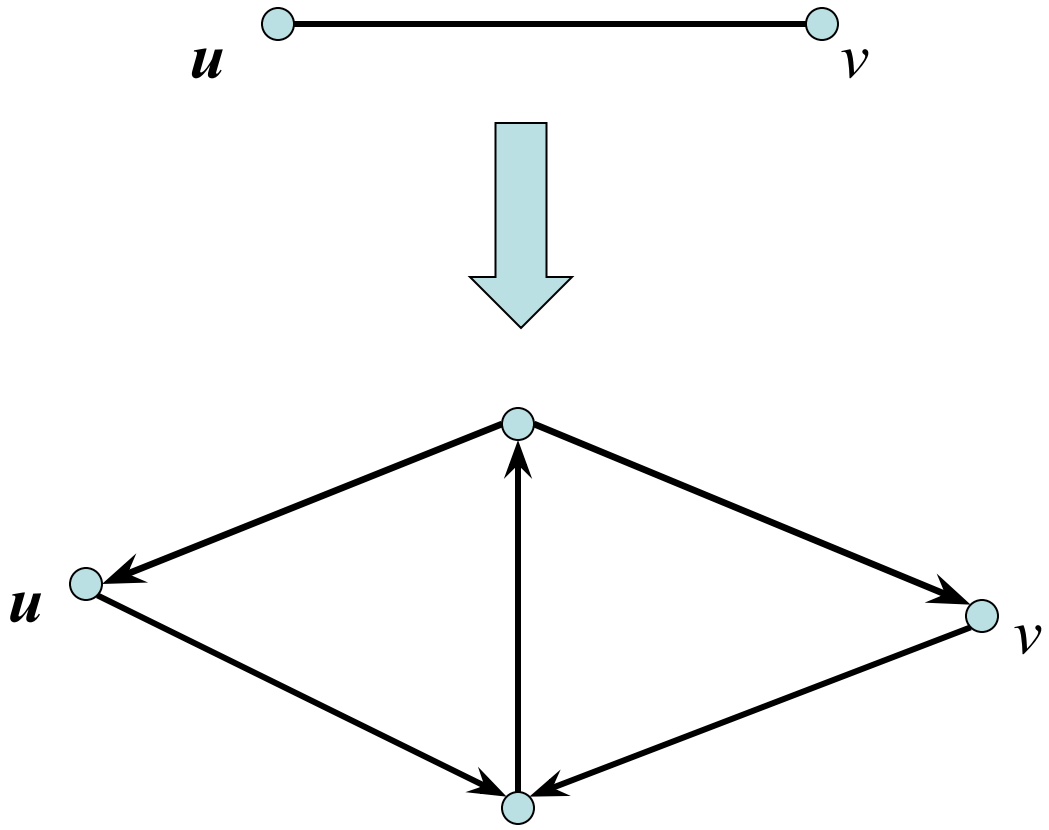
Theorem 7.1 (Menger [1927])

Пусть G — граф (ориентированный или неориентированный), пусть s и t две вершины и $k \in \mathbf{N}$. Тогда существует k реберно-непересекающихся s-t-путей, тогда и только тогда, когда после удаления любых $k-1$ ребер t остается достижима из s.

Доказательство (*для орграфа*)

- Пусть (G, u, s, t) — сеть с $u \equiv 1$, такая что t достижима после удаления любых $k - 1$ дуг.
- Тогда минимальным s-t-разрез имеет пропускную способность по крайней мере k.
- По теореме 6.5 существует целочисленный поток величины по крайней мере k.
- По теореме 6.7 этот поток можно разложить в целочисленные потоки на s-t-путях.
- Так как $u \equiv 1$ получаем по крайней мере k s-t-путей.

Доказательство (для графа)



Пути, непересекающиеся по вершинам

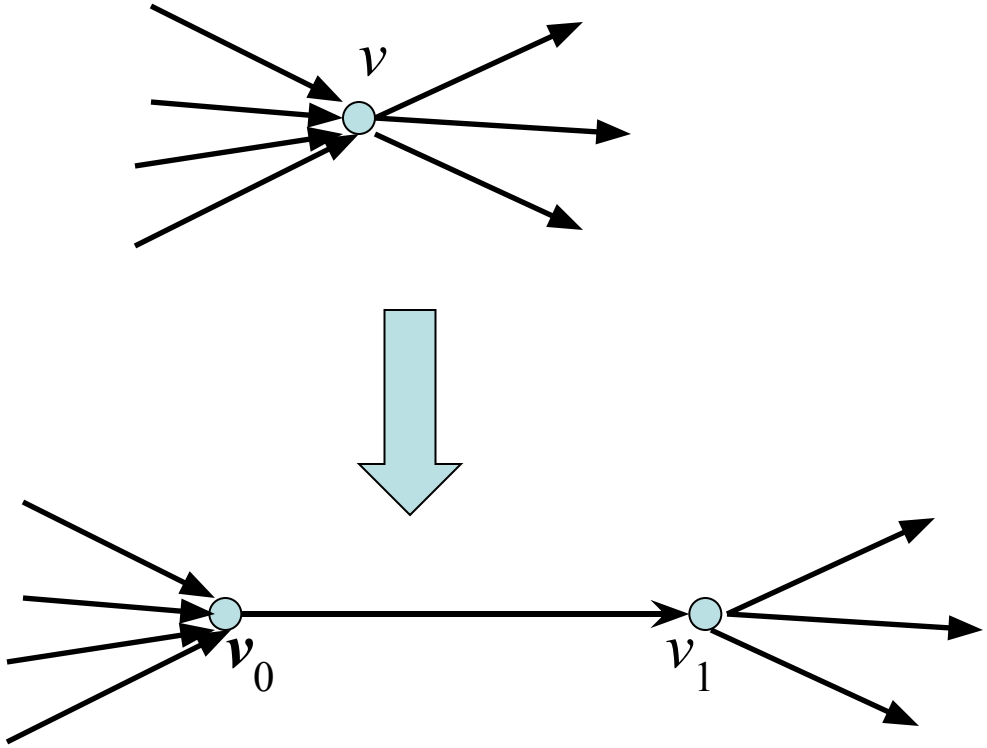
- Будем говорить, что два пути **вершинно-непересекающиеся** если они не имеют общих ребер и общих внутренних вершин (они могут иметь одну или две общих граничных точки).

Вторая Теорема Менгера

Theorem 7.2 (Menger [1927])

Пусть G — граф (ориентированный или неориентированный), пусть s и t две несмежные вершины и $k \in \mathbf{N}$. Тогда существует k вершинно-непересекающихся s-t-путей, тогда и только тогда, когда после удаления любых $k - 1$ вершин (отличных от s и t) t остается достижима из s.

Доказательство



k-связные графы

- Граф с более чем k вершинами и свойством, что он остается связным после удаления любого множества из $k-1$ вершины называется ***k*-связным**.
- Граф с не менее чем двумя вершинами называется ***k*-реберно-связным**, если он остается связным после удаления любого множества из $k-1$ ребра.

Характеризация k-связных графов

Следствие 7.3 (Уитни [1932])

Граф G с не менее чем двумя вершинами является k-реберно-связным тогда и только тогда, когда для каждой пары $s, t \in V(G)$ с $s \neq t$ существует k реберно-непересекающихся s-t-путей.

Граф G с не менее чем k вершинами является k-связным тогда и только тогда, когда для каждой пары $s, t \in V(G)$ с $s \neq t$ существует k вершинно-непересекающихся s-t-путей.

Доказательство

- Первое утверждение прямо следует из теоремы 7.1.
- Если граф не является k-связным, то существуют вершины s и t и множество X из $k-1$ вершины, такие, что в графе нет s-t-пути после удаления множества X.
- $\Rightarrow$ в графе нет k вершинно-непересекающихся s-t-путей.
- Пусть в G есть две вершины s и t для которых нет k вершинно-непересекающихся s-t-путей.
- Если s и t не смежны, то теорема 7.2 $\Rightarrow$ существует множество X из $k-1$ вершины, такое, что после его удаления в G нет s-t-пути.
- $\Rightarrow$ G не является k-связным.

Доказательство (s и t смежны)

- Пусть s *и* t соединено множеством F параллельных ребер.
- Тогда в $G - F$ нет $k - |F|$ вершинно-непересекающихся s-t-путей. ($|F| \geq 1$)
- Теорема 7.2 $\Rightarrow$ существует множество X из $k - |F| - 1$ вершины, такое, что после его удаления в G нет s-t-пути.
- Существует вершина v, которая не достижима в $G - F - X$, либо из s, либо из t.
- Пусть из t. Добавляя s к X, получаем разделяющее множество вершин мощности не более $k - 1$.
- $\Rightarrow$ G не является k-связным.

Задача «Ориентированные реберно-непересекающиеся пути»

- *Дано*: Два орграфа (G, H) на одном множестве вершин.
- *Найти* семейство $(P_f)_{f \in E(H)}$ реберно-непересекающихся путей в G таких, что для каждого ребра(дуги) $f = (t, s)$ в H, P_f — s-t-путь.

Такое семейство называется **решением** (G, H).

Разрешимый случай

Предложение 7.4

Пусть (G, H) пример задачи «Ориентированные реберно-непересекающиеся пути» такой, что H является множеством параллельных ребер и $G + H$ — эйлеров граф. Тогда (G, H) имеет решение.

Алгоритм Эдмонса-Карпа

Input: Сеть (G, u, s, t).
Output: s-t-поток f максимального значения.

1. **Set** $f(e) = 0$ для всех $e \in E(G)$.
2. Найти кратчайший по числу ребер f-увеличивающий путь P.
   If такого пути нет **then stop**.
3. Вычислить $\gamma := \min_{e \in E(P)} u_f(e)$.
   Увеличить f вдоль P на γ и **go to** 2.

Лемма

Лемма 7.5

Пусть f_1, f_2, ... последовательность потоков, таких что f_{i+1} получается из f_i увеличением потока вдоль P_i, где P_i — кратчайший f_i-увеличивающий путь. Тогда

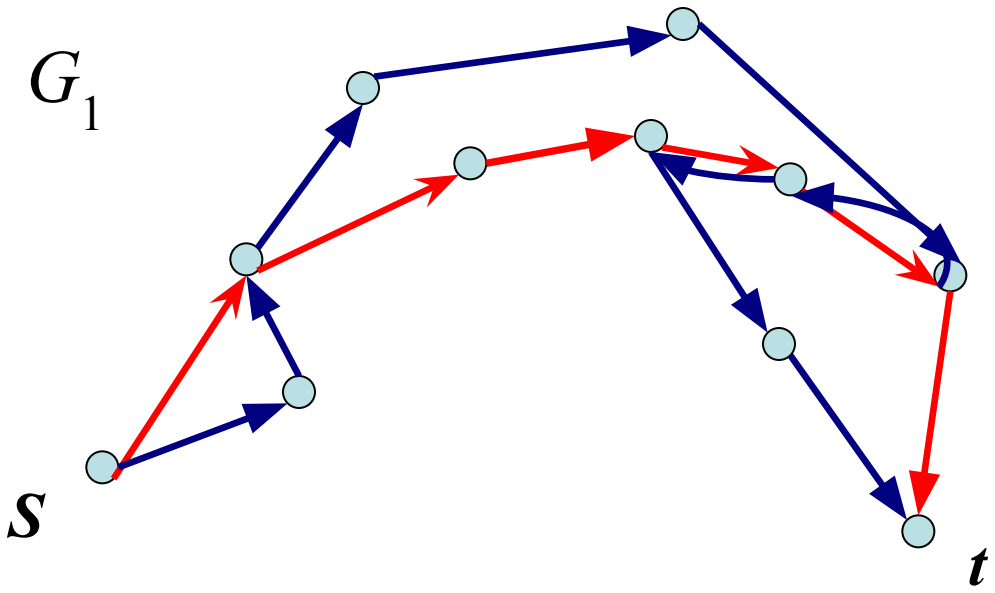
a) $|E(P_k)| \leq |E(P_{k+1})|$ для всех k.

b) $|E(P_k)| + 2 \leq |E(P_l)|$ для всех $k < l$ таких, что $P_k \cup P_l$ содержит пару обратных дуг.

$|E(P_k)| \leq |E(P_{k+1})|$ для всех k

- Рассмотрим граф G_1, который получается из $P_k \cup P_{k+1}$ удалением обратных дуг. (Дуги, появляющиеся в обоих путях, берутся дважды).

Граф G_1

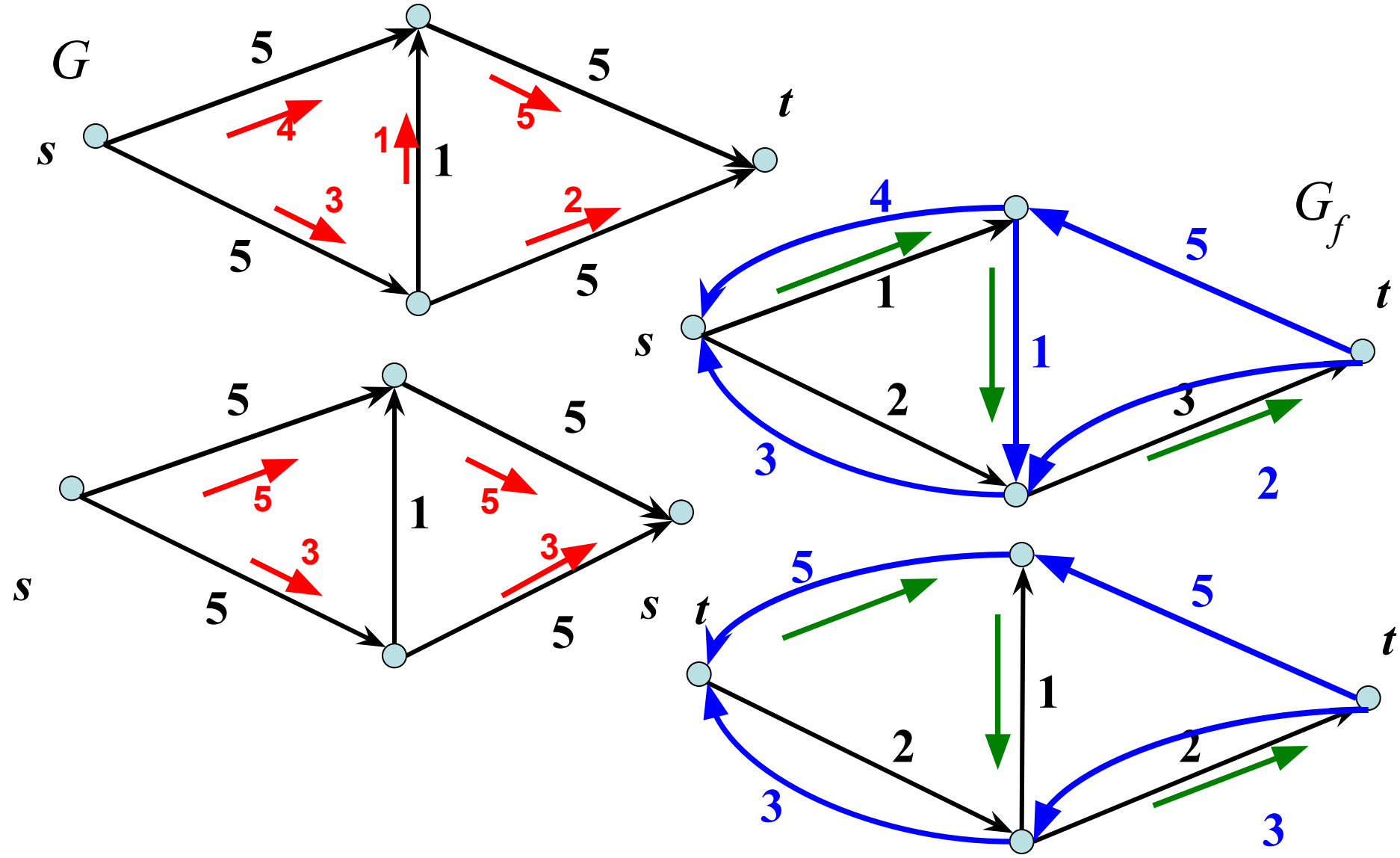


$\boldsymbol{G_1 = P_k \cup P_{k+1}}$ – обратные дуги

Доказательство а)

- Рассмотрим граф G_1, который получается из $P_k \cup P_{k+1}$ удалением обратных дуг. (Дуги, появляющиеся в обоих путях, берутся дважды).
- Так как для любой дуги в $E(G_{fk}) \backslash E(G_{fk+1})$ путь P_k должен содержать обратную ей дугу, то $E(G_1) \subseteq E(G_{fk})$.

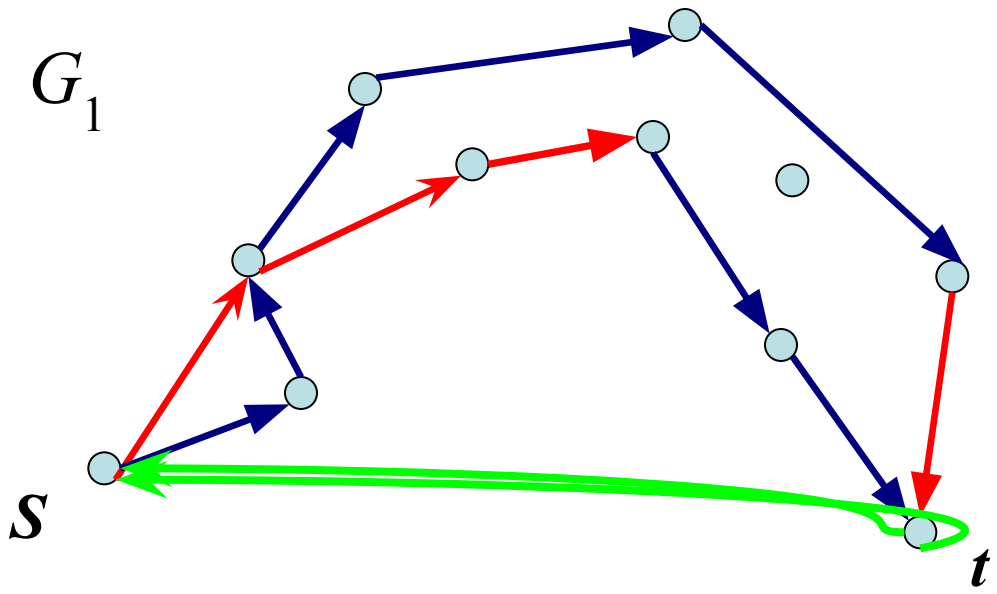
Остаточные графы



$|E(P_k)| \leq |E(P_{k+1})|$ для всех k

- Рассмотрим граф G_1, который получается из $P_k \cup P_{k+1}$ удалением обратных дуг. (Дуги, появляющиеся в обоих путях, берутся дважды).
- Так как для любой дуги в $E(G_{fk}) \backslash E(G_{fk+1})$ путь P_k должен содержать обратную ей дугу, то $E(G_1) \subseteq E(G_{fk})$.
- Пусть H_1 состоит из двух копий (t,s). Тогда $G_1 + H_1$ Эйлеров.

$G_1 + H_1$



$G_1 = P_k \cup P_{k+1}$ – обратные дуги

H_1 – две дуги (t,s)

$|E(P_k)| \leq |E(P_{k+1})|$ для всех k

- Рассмотрим граф G_1, который получается из $P_k \cup P_{k+1}$ удалением обратных дуг. (Дуги, появляющиеся в обоих путях, берутся дважды).
- Так как для любой дуги в $E(G_{fk}) \backslash E(G_{fk+1})$ путь P_k должен содержать обратную ей дугу, то $E(G_1) \subseteq E(G_{fk})$.
- Пусть H_1 состоит из двух копий (t,s). Тогда $G_1 + H_1$ Эйлеров.
- Предложение 7.4 $\Rightarrow \exists$ два реберно-непересекающихся пути Q_1 и Q_2.
- $E(G_1) \subseteq E(G_{fk}) \Rightarrow Q_1$ и Q_2 увеличивающие пути в $G_{fk.}$

$|E(P_k)| \leq |E(P_{k+1})|$ для всех k

- P_k был выбран кратчайшим путем.
- $|E(P_k)| \leq |E(Q_1)|$ и $|E(P_k)| \leq |E(Q_2)|$.
- $2|E(P_k)| \leq |E(Q_1)| + |E(Q_2)| \leq |E(G_1)| \leq$
  $\leq |E(P_k)| + |E(P_{k+1})|$.
- $|E(P_k)| \leq |E(P_{k+1})|$.

$|E(P_k)| + 2 \le |E(P_l)|$ для всех $k < l$ таких, что $P_k \cup P_l$ содержит пару обратных дуг

- Пусть $k < l$ такие, что для любого i, $k < i < l$ $P_i \cup P_l$ не содержит обратных дуг.
- Рассмотрим граф G_1, который получается из $P_k \cup P_l$ удалением обратных дуг.
- $E(G_1) \subseteq E(G_{fk})$
  - $E(P_k) \subseteq E(G_{fk})$, $E(P_l) \subseteq E(G_{fl})$
  - Любая дуга в $E(G_{fl}) \backslash E(G_{fk})$ должна быть обратной дуге в одном из путей $P_k, P_{k+1}, \ldots, P_{l-1}$.
  - По выбору k и l только P_k содержит обратные дуги.
- $2|E(P_k)| \le |E(Q_1)| + |E(Q_2)| \le |E(G_1)| \le$
  $\le |E(P_k)| + |E(P_{k+1})| - 2.$

Число увеличений

Теорема 7.6 (Edmonds, Karp [1972])

Алгоритм Эдмондса-Карпа остановится, сделав не более чем $mn/2$ увеличений, где m — число ребер и n — число вершин.

Доказательство

- Пусть P_1, P_2,… увеличивающие пути, выбранные в алгоритме Эдмонса-Карпа.
- По выбору γ на шаге 3 алгоритма, каждый увеличивающий путь содержит по крайней мере одну узкую дугу e: $u_f(e) = \gamma$.
- Пусть P_{i1}, P_{i2} ,… последовательность увеличивающих путей, в которых дуга e узкая.
- Тогда между P_{ij}, P_{ij+1} должен быть увеличивающий путь $P_к$ с обратной дугой к e.

Доказательство

- Лемма 7.5 b) $\Rightarrow$

$$|E(P_{ij})| + 4 \leq |E(P_k)| + 2 \leq |E(P_{ij+1})| \text{ для всех } j.$$

- $1 \leq |E(P_{ij})| \leq n - 1 \Rightarrow$

$\exists$ не более $n/4$ увеличивающих путей, в которых дуга e узкая.

- Так как любой увеличивающий путь содержит по крайней мере одну дугу из $\breve{G}$, как узкую, то $\exists$ не более $(n|E(\breve{G})|)/4 = (mn)/2$ увеличивающих путей.

Время работы Алгоритма Эдмондса-Карпа

Следствие 7.7

Алгоритм Эдмондса-Карпа решает Задачу «Максимальный Поток» за $O(m^2n)$ элементарных операций.

Три Условия на Максимальный s-t-Поток

Функция $f : E(G) \to \mathbf{R}_+$ является максимальным *s*-*t*-потоком тогда и только тогда, когда выполнены следующие три условия:

- $f(e) \le u(e) \qquad \forall\, e \in E(G),$
- $\sum_{e \in \delta^-(v)} f(e) = \sum_{e \in \delta^+(v)} f(e) \quad \forall v \in V(G) \setminus \{s,t\},$
- в G_f не существует *f*-увеличивающего пути.

s-t-Предпоток

- Определение 7.8

  Дана сеть (G, u, s, t), функция $f : E(G) \to \mathbf{R}_+$, удовлетворяющая $f(e) \le u(e)$ для всех $e \in E(G)$ и

$$ex_f(v) := \sum_{e \in \delta^-(v)} f(e) - \sum_{e \in \delta^+(v)} f(e) \ge 0 \quad \forall v \in V(G) \setminus \{s\}.$$

  Назовем вершину $v \in V(G)\setminus\{s,t\}$ **активной,** если $ex_f(v) > 0$.

  s-t-Предпоток является s-t-потоком тогда и только тогда, когда нет активных вершин.

Функция расстояний

- Определение 7.9

Даны сеть (G, u, s, t) и s-t-предпоток f.
Функцией расстояния называется функция $\psi : V(G) \to \mathbf{Z}_+$ такая, что $\psi(t) = 0$, $\psi(s) = n$ и $\psi(v) \le \psi(w) + 1$ для всех $(v, w) \in E(G_f)$.

Дуга $e = (v, w) \in E(\check{G})$ называется **допустимой** если

$$e \in E(G_f) \text{ и } \psi(v) = \psi(w) + 1.$$

Идея алгоритма

- В процессе работы алгоритм строит последовательность предпотоков и задает функцию расстояния на них.
- Алгоритм стартует с предпотока, который вдоль дуг, выходящих из s, равен их пропускным способностям и 0 вдоль остальных дуг и с функции расстояния $\psi(s) = n$ и $\psi(v) = 0$ для всех $v \in V(G) \setminus \{s\}$.
- Алгоритм останавливается, когда в сети нет активных вершин.

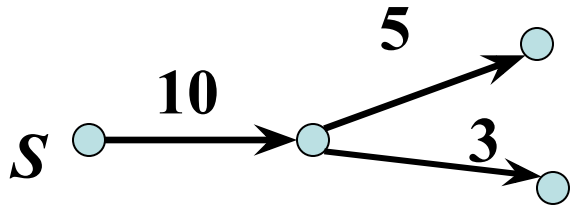
Алгоритм Проталкивания Предпотока

Input: Сеть (G, u, s, t).
Output: максимальный s-t-поток f.

1. **Set** $\psi(s) = n$ и $\psi(v) = 0$ для всех $v \in V(G) \setminus \{s\}$.
2. **Set** $f(e) := u(e)$ для каждого $e \in \delta^+(s)$.
Set $f(e) := 0$ для каждого $e \in E(G) \setminus \delta^+(s)$.

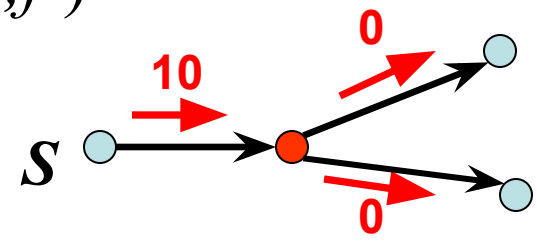
- **While** существуют активные вершины **do:**
  Пусть v — активная вершина
  If нет допустимой дуги $e \in \delta^+(v)$
  then Relabel(v)
  else Push(e) ($e \in \delta^+(v)$ произвольная допустимая дуга)

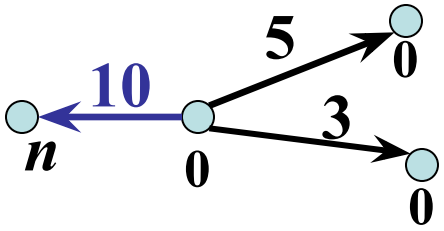
Push(e): 1) **Set** $\gamma := \min\{ex_f(v), u_f(e)\}$, где v — вершина с $e \in \delta^+(v)$.
2) Увеличим f вдоль e на γ.
Relabel(v): **Set** $\psi(v) := \min\{\psi(w)+1 : e = (v, w) \in E(G_f)\}$. ($\psi(v) \uparrow$)

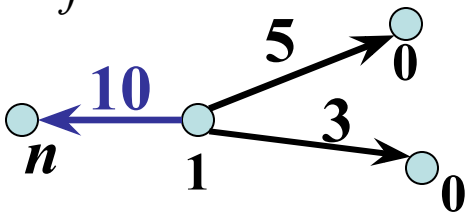

10
5
3
s

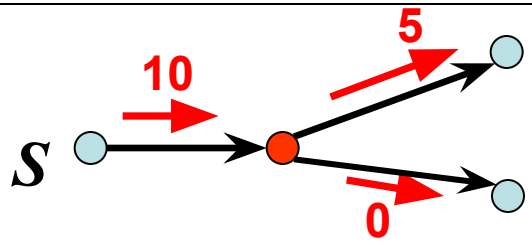
(G,f)

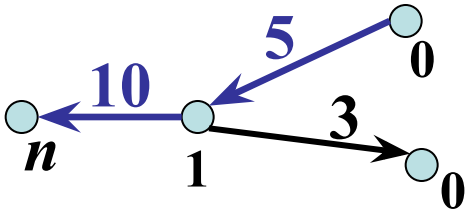
G_f

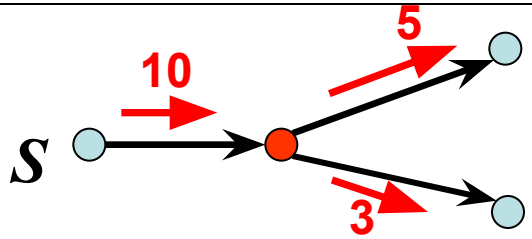

10
0
0
s

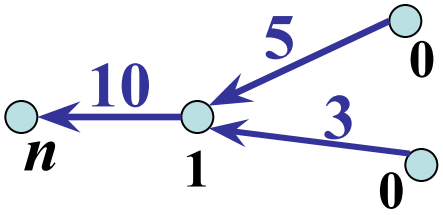

10
5
0
3
0
n
0

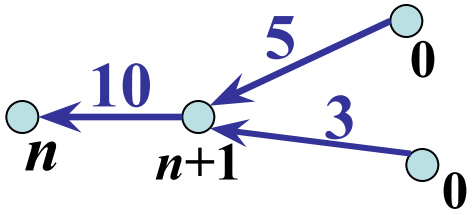

10
5
0
3
0
n
1

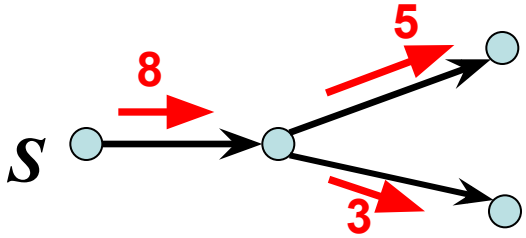

10
5
0
s


10
5
0
3
0
n
1


10
5
3
s


10
5
0
3
0
n
1


10
5
0
3
0
n
n+1


8
5
3
s

Алгоритм Проталкивания Предпотока (2)

Предложение 7.10

В процессе работы Алгоритма Проталкивания Предпотока f всегда является s-t-предпотоком и ψ всегда является функцией расстояния относительно f.

Доказательство

- Предпоток изменяется только в процедуре Push.
- Так как $\gamma := \min\{ex_f(v), u_f(e)\}$, то после процедуры Push f остается предпотоком.
- Так как $\psi(v) := \min\{\psi(w) + 1: e = (v, w) \in E(G_f)\}$, то ψ остается функцией расстояния после процедуры Relable.
- Покажем, что после процедуры Push(e) ψ также остается функцией расстояния относительно нового предпотока.

Доказательство

- Необходимо показать, что $\psi(a) \leq \psi(b) + 1$ для всех новых дуг (a, b) в G_f.
- Поскольку в процедуре Push(e) поток изменяется только вдоль дуги $e = (v, w)$, то новой в G_f может быть только одна дуга (w, v), обратная к e.
- Так как e была допустимой дугой, то $\psi(w) = \psi(v) - 1 \leq \psi(v) + 1$.

Алгоритм проталкивания предпотока (3)

Лемма 7.11

Если f — s-t-предпоток и ψ функция расстояния относительно f, то

a) s достижима из любой активной вершины в G_f.

b) t не достижима из s в G_f.

Доказательство а)

- Пусть v активная вершина, и пусть R множество вершин достижимых из v в G_f.
- Тогда $f(e) = 0$ для всех $e \in \delta^{-}(R)$ в G.
- $$\sum_{w \in R} ex_f(w) = \sum_{e \in \delta_G^-(R)} f(e) - \sum_{e \in \delta_G^+(R)} f(e) \le 0$$
- v – активная $\Rightarrow ex_f(v) > 0$
- $\Rightarrow \exists w \in R$, такая что $ex_f(w) < 0$
- f – предпоток, то такая вершина может быть только s.

Доказательство b)

- Пусть существует s-t-путь в G_f, например $s = v_0, v_1, \ldots, v_k = t$.
- $\psi(v_i) \leq \psi(v_{i+1}) + 1, i = 0,\ldots k - 1.$
- $\psi(s) \leq \psi(t) + k.$
- Но $\psi(s) = n$, $\psi(t) = 0$ и $k \leq n - 1.$

Алгоритм Проталкивания Предпотока (4)

Теорема 7.12

Когда Алгоритм Проталкивания Предпотока останавливается, f является максимальным s-t-потоком.

Число вызовов процедуры Relabel

Лемма 7.13

a) Для каждого $v \in V(G)$, $\psi(v)$ строго возрастает при каждом выполнении процедуры Relabel(v), и никогда не убывает.

b) На любом шаге алгоритма, $\psi(v) \leq 2n - 1$ для всех $v \in V(G)$.

c) Общее число вызовов процедуры Relabel не превышает $2n^2$.

Процедура Push

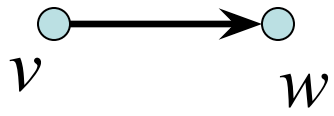
- Процедура Push(e) называется **проталкиванием**, примененным к вершине v. Проталкивание называется **насыщающим**, если в результате ребро e становится **насыщенным**, то есть если $u_f(e)$ обращается в нуль (ребро исчезает из остаточного графа); в противном случае проталкивание считают **ненасыщающим**.

Число насыщающих проталкиваний

Лемма 7.14

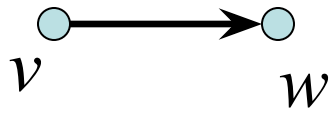
Число насыщающих проталкиваний не превышает mn.

Доказательство



$\psi(v) = k + 1$, $\psi(w) = k$, $k \leq 2n - 1$.

$\psi(v) \geq k + 1$, $\psi(w) \geq k + 2$, $k \leq 2n - 1$.



$\psi(v) \geq k + 3$, $\psi(w) \geq k + 2$, $k \leq 2n - 1$.

Возможно не более n насыщающих проталкиваний вдоль одного ребра.

list(*v*) и *curr*(*v*)

- Для получения оценки $O(n^3)$ на число ненасыщающих проталкиваний мы должны выбрать порядок в котором применяются процедуры Push и Relabel.
- Как обычно, предположим, что орграф G задан листом смежности, то есть для каждой вершины v указан список *list*(v) дуг выходящих из v. При этом указатель *curr*(v) указывает на один элемент в списке *list*(v) (вначале на первый элемент в списке).

Алгоритм Голдберга-Тарьяна

Input: Сеть (G, u, s, t).
Output: Максимальный s-t-поток f.

1. **Set** $\psi(s) = n$ и $\psi(v) = 0$ для всех $v \in V(G) \setminus \{s\}$.
2. **Set** $f(e) := u(e)$ для каждого $e \in \delta^+(s)$.
   Set $f(e) := 0$ для каждого $e \in E(G) \setminus \delta^+(s)$.
- **For** всех $v \in V(G)$ **do:** $curr(v)$:= первый элемент $list(v)$
- **Set** Q:=$\{v \in V(G)$: v **—** активная $\}$. **If** $Q = \varnothing$ **then stop.**

1. **For** всех $v \in Q$ **do:** Discharge(v).
   Go to 4.

Discharge(v)

1. **Set** e:= $curr(v)$.
2. **If** e допустимо **then** Push(e) **else do:**
   If e последнее ребро в $list(v)$ **then** Relabel(v),
   $curr(v)$:= первый элемент $list(v)$, **return,**
   else $curr(v)$:= следующий элемент $list(v)$.
3. **If** $ex_f(v) > 0$ **then go to 1.**

Процедура Разгрузки

Лемма 7.15

Процедура Discharge вызывает процедуру Relabel только, если v активна и нет допустимых ребер в $\delta^+(v)$.

Процедура Разгрузки

Discharge(v)

1. **Set** e:= *curr*(v).
2. **If** e допустимо **then** Push(e) **else do:**
   If e последнее ребро в *list*(v) **then** Relabel(v),
   curr(v):= первый элемент *list*(v), **return,**
   else *curr*(v):= следующий элемент *list*(v).
3. **If** $ex_f(v) > 0$ **then go to 1.**

Доказательство

- Вершина v всегда активна перед выполнением шага 2 процедуры Discharge(v).
- $\Rightarrow$ Процедура Discharge вызывает процедуру Relabel только, если v активна.
- Осталось показать, что в момент вызова Relabel $\psi(v) \leq \psi(w)$.
- Рассмотрим произвольную дугу $e = (v,w) \in E(G_f)$.
- С момента предыдущего вызова Relabel для вершины v весь список ее дуг был просмотрен. В частности, указатель *curr* (v) указывал и на дугу e. Поскольку, он ее покинул, то
  - либо $\psi(v) < \psi(w) + 1$
  - либо $e \notin E(G_f)$ и появилась в G_f позднее, когда проталкивался поток по дуге $=(w,v)$ и $\psi(w) = \psi(v) + 1 > \psi(v)$.

Число ненасыщающих проталкиваний

Лемма 7.16

Число ненасыщающих проталкиваний не превышает $4n^3$.

Доказательство

- Так как $\gamma := \min\{ex_f(v), u_f(e)\}$, то на каждой итерации шага 5 может быть не более одного ненасыщающего проталкивания для каждой вершины.
- Покажем, что число итераций шага $5 \leq 4n^2$.
- Разделим все итерации на итерации с запуском Relabel и без запуска.
- Лемма 7.13 c) $\Rightarrow$ не более $2n^2$ итераций с Relabel

Число итераций шага 5 без Relabel

- Пусть $\Psi = \max\{\psi(v) : v \text{ - активная}\}$
- $\Psi = 0$, если нет активных вершин.
- На каждой итерации без Relabel Ψ уменьшается минимум на 1.
  - Пусть w – активная вершина после шага 5 без Relabel. Так как шаг 5 выполнялся для всех активных на тот момент вершин, то w стала активной в процессе этой итерации шага 5 по причине проталкивания потока по дуге (v,w).
  - v была активной и $\psi(v) = \psi(w) + 1$
- В начале и в конце алгоритма $\Psi = 0 \Rightarrow$ число итераций без Relabel не больше суммарной величины Δ на которое Ψ увеличивается в течение работы алгоритма.
- Так как увеличение Ψ соответствует увеличению $\psi(v)$ в результате Relabel, то $\Delta \leq 2n^2$ (по Лемме 7.13).

Алгоритм Голдберга-Тарьяна

Теорема 7.17 (Goldberg, Tarjan [1988])

Алгоритм Голдберга-Тарьяна определяет максимальный s-t-поток за время $O(n^3)$.

Задача «Разрез с минимальной пропускной способностью»

- *Дано*: Сеть (G, u, s, t).
- *Найти* s-t-разрез в G с минимальной пропускной способностью.

Минимальный разрез

Предложение 7.18

Задача «Разрез с минимальной пропускной способностью» может быть решена за то же самое время как и задача «Максимальный Поток», в частности за время $O(n^3)$.

Упражнение 7.1

- Построить линейный алгоритм для задачи «*Максимальный Поток*», для случая когда $G - t$ является ордеревом с корнем в s.

Упражнение 7.2

- Задан ациклический орграф с весами $c : E(G) \to \mathbf{R}_+$, найти максимальный взвешенный ориентированный разрез в G.
- Показать как эта проблема может быть сведена к задаче «Разрез с минимальной пропускной способностью» и решена за время $O(n^3)$.