Алгоритм Форда-Фалкерсона (АФФ)

ВХОД. Сеть $G = (V, E, s, t, c(e))$.

ВЫХОД. Поток f наибольшей мощности в сети G.

Begin $f = (0, \ldots, 0)$; $H = G$;

While в H есть путь L из s в t **do**

begin $\rho := \min \{c(e) \mid e \in E(L)\}$;

получить поток $\varphi_L(\rho)$ мощности ρ вдоль L;

перестроить сеть $H := H_{\varphi_L(\rho)}$;

пересчитать поток $f := \varphi_L(\rho) \oplus f$

end;

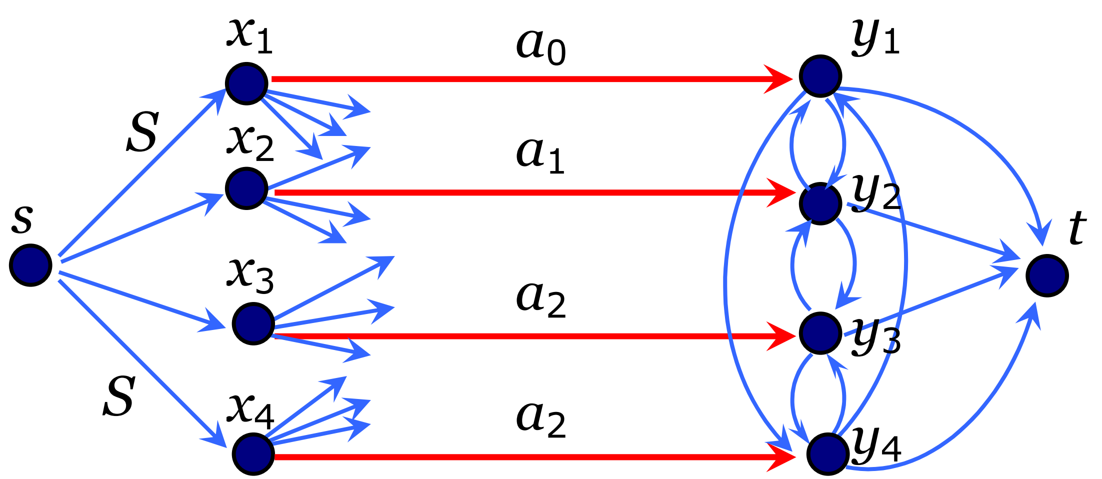
End;

Контрпример

Рассмотрим последовательность $\{a_n\}_{n=0}^{\infty}$, $a_n = r^n$, где $r = \frac{\sqrt{5}-1}{2} < 1$.

При таком выборе r получаем: $a_{n+2} = a_n - a_{n+1}$ и $\sum_{n=0}^{\infty} a_n = S < \infty$.

Рассмотрим сеть на 10 вершинах: $s, t, x_1, \dots, x_4, y_1, \dots, y_4$.



Дуги: 1) (s, x_i), $i=1,\dots,4$ пропускная способность S

2) (y_i, t), $i=1,\dots,4$ пропускная способность S

3) (x_i, y_i), $i=1,\dots,4$ пропускная способность $\{a_0, a_1, a_2, a_2\}$

4) (y_i, y_j), $i, j = 1,\dots,4$ пропускная способность S

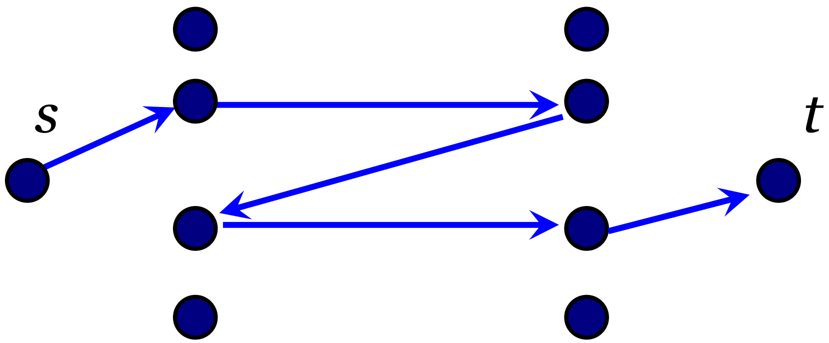
5) (x_i, y_j) (y_i, x_j), $i \neq j$ пропускная способность S

Максимальный поток $M^*(f) = 4S$

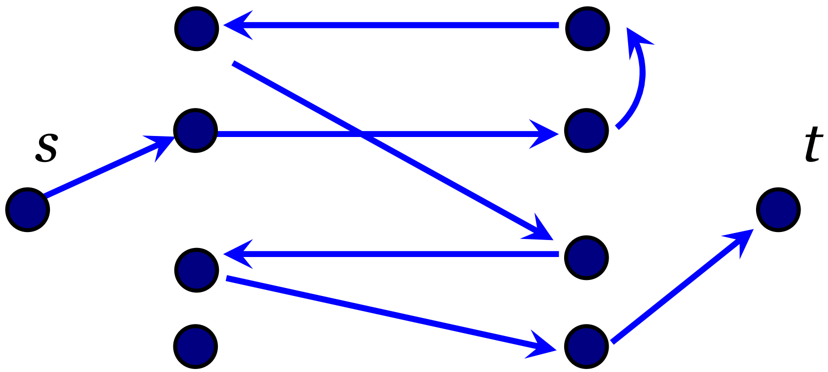
Шаг 1. $L_1 = \{s\ x_1\ y_1\ t\}$ поток a_0; $\quad 0 \quad a_1 \quad a_2 \quad a_2$

Шаг 2. $L_2 = \{s\ x_2\ y_2\ x_3\ y_3\ t\}$ поток a_2; $\quad 0 \quad a_1 - a_2 \quad 0 \quad a_2$

a_3



Шаг 3. $L_3 = \{s\ x_2\ y_2\ y_1\ x_1\ y_3\ x_3\ y_4\ t\}$ поток a_3; $\quad a_3 \quad 0 \quad a_3 \quad a_2$



Шаги 2 и 3 дают поток $a_1 = a_2 + a_3$ и возвращают к состоянию $\{0;\ a_{n+1};\ a_{n+1};\ a_n\}$. В пределе получаем S.

Алгоритм кратчайших путей

Назовем *рангом* вершины v в сети G расстояние (по числу дуг) в G от s до v.

Лемма 1. Пусть f — поток в сети G, L — кратчайший по числу дуг путь из s в t в сети G_f, φ_L — поток вдоль пути φ_L в сети G_f, $g = \varphi_L \oplus f$. Тогда ранг $r_g(v)$ любой вершины v в сети G_g не меньше ее ранга в сети G_f.

Доказательство. Если $r_g(v) = \infty$, то для v утверждение леммы выполнено. Пусть $r_g(v) = k$ и $L_v = (v_0, v_1, \ldots, v_k)$ — кратчайший по числу дуг путь из $v_0 = s$ в $v_k = v$ в сети G_g.

Рассмотрим произвольную дугу $e = (v_i, v_{i+1})$ пути L_v. Если $e \in E(G_f)$, то $r_f(v_{i+1}) - r_f(v_i) \le 1$. Если $e \notin E(G_f)$, то $\bar{e} = (v_{i+1}, v_i) \in E(G_f) \cap E(L)$.

Поскольку L — кратчайший по числу дуг путь из s в t в сети G_f и $\bar{e} \in E(L)$, то $r_f(v_{i+1}) + 1 = r_f(v_i)$ и $r_f(v_{i+1}) - r_f(v_i)) = -1$. Следовательно,

$$r_f(v_k) = \sum_{i=1}^{k-1} \left(r_f(v_{i+1}) - r_f(v_i) \right) \le k. \quad \blacksquare$$

Назовем *t-рангом* вершины v в сети G расстояние (по числу дуг) в G от v до t. Аналогично лемме 1 доказывается

Лемма 2. Пусть f — поток в сети G, L — кратчайший по числу дуг путь- из s в t в сети G_f, φ_L — поток вдоль пути φ_L в сети G_f, $g= f \oplus \varphi_L$. Тогда t ранг $tr_g(v)$ любой вершины v в сети G_f не меньше ее t -ранга в сети G_f.

Алгоритм кратчайших путей (АКП) отличается от алгоритма Форда-Фалкерсона лишь тем, что увеличивающий путь L ищется с помощью поиска в ширину (и тем самым оказывается кратчайшим в G_f по числу дуг).

Трудоемкость АКП

Поиск в ширину требует $O(|E|)$ элементарных операций.

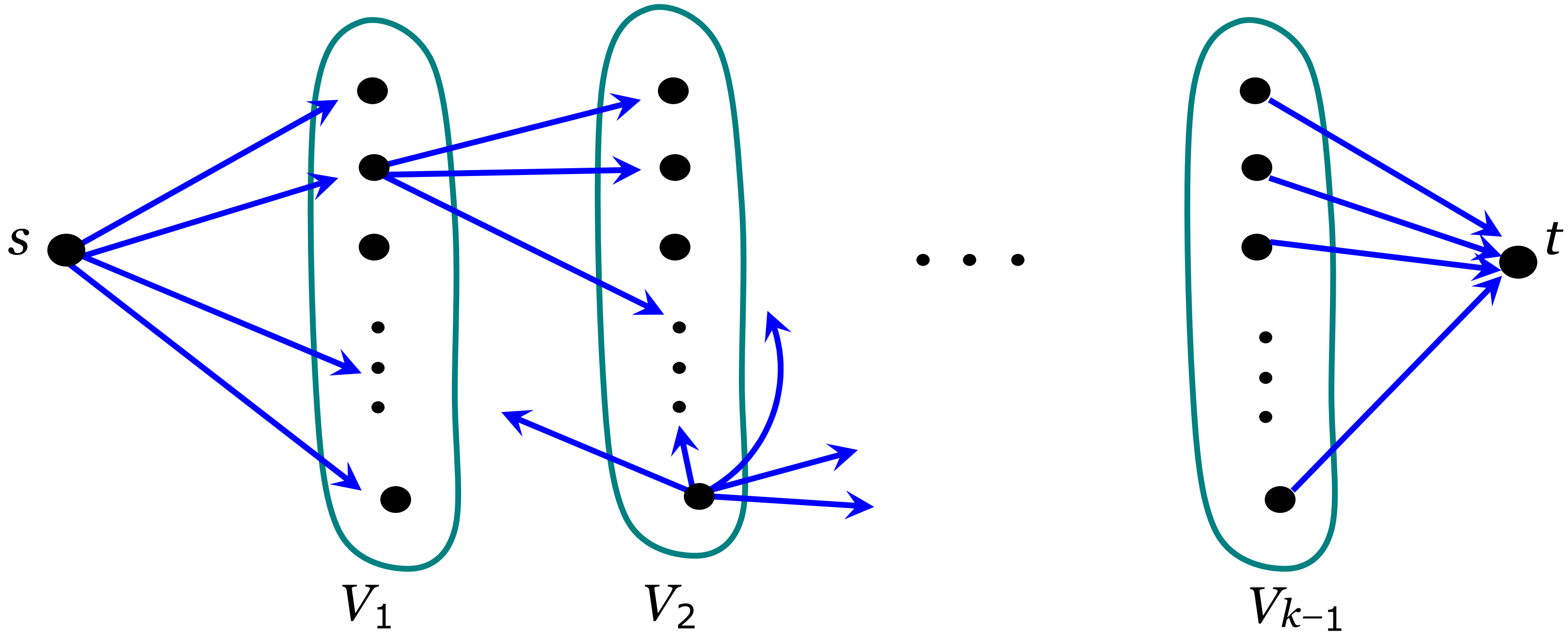
Для вычисления ρ и пересчета f и H достаточно $O(|V|)$ элементарных операций, поскольку пересчет происходит лишь для дуг L и обратных к ним. Значит, каждая итерация требует не более $O(|E|)$ элементарных операций.

Чтобы оценить число итераций, разобьем их на группы.

В k -ю группу попадут те итерации, на которых ранг вершины t равен k. По лемме 2 все итерации из группы k идут подряд. Предположим, что перед началом итераций k -й группы мы имели сеть G_g.

Лемма 3. Все дуги увеличивающих путей k-й группы итераций АКП принадлежат G_g и для каждой используемой дуги ранг в G_g ее конца на единицу больше ранга ее начала.

Доказательство. Обозначим через V_j, j = 0,1,...$|V|$ множество вершин ранга j в G_g. Покажем, что любая дуга любого из увеличивающих путей k-й группы итераций ведет из V_j в V_{j+1} для некоторого $j \in \{0, 1,\ldots, k-1\}$.



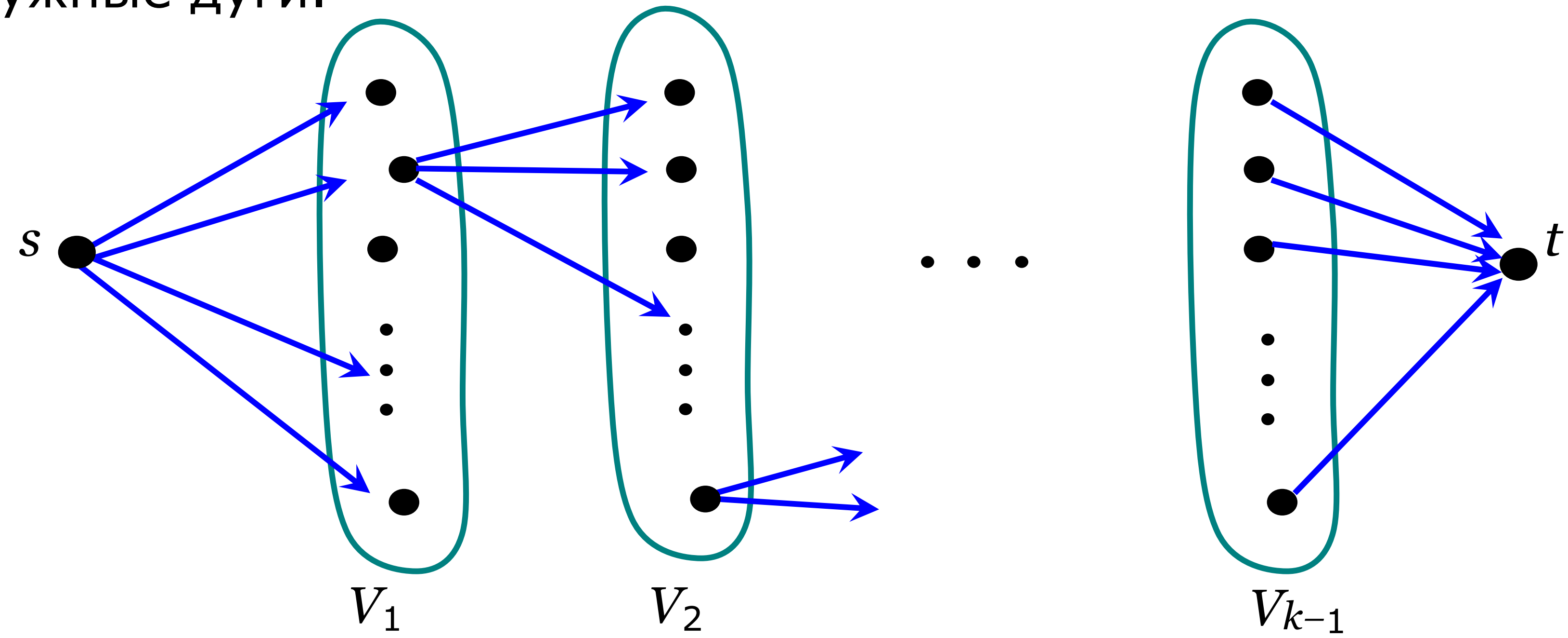
По леммам 1 и 2 для каждой вершины v, лежащей на каком-нибудь увеличивающем пути, имеем $r(v) + tr(v) = k$. Пусть L — первый из увеличивающих путей k-й группы итераций, в котором есть такая дуга (v, u), $v \in V_i$, $u \in V_j$, что $j \neq i+1$. Поскольку L — первый такой путь, то «новые» (по сравнению с G_g) дуги могут вести лишь из V_i в V_{i-1} для некоторого $i \in \{1, \ldots, k\}$. Значит, $j \leq i$. Но тогда L должен содержать более чем k дуг.

■

Поскольку на каждой итерации из k-й группы хотя бы одна из «старых» дуг исчезает (поворачивается), то эта группа состоит из не более чем $|E|$ итераций. Отсюда следует оценка $O(|E|^2\,|V|)$ элементарных операций для АКП.

Модифицированный АКП

На итерациях k-й группы участвовали лишь такие вершины ранга j, $j \in \{0, 1, \ldots, k\}$ в G_g, расстояние от которых до t равно $k - j$. Дуги ведут из V_j в V_{j+1} для некоторого $j \in \{0, 1, \ldots, k-1\}$. Построить такую подсеть можно за $O(|E|)$ элементарных операций: сначала поиском в ширину найти множества V_j, $j = 0, 1, \ldots, k$, затем обратным поиском в ширину из t выделить те из них, расстояние от которых до t равно $k - j$, и, наконец, оставить только нужные дуги.



Модификация АКП для построения путей длины k в полученной *k-слойной сети*.

На каждой итерации последовательно для $i = 0, 1,\ldots, k-1$ проделываем следующее: уже имеется путь $(v_0, v_1,\ldots, v_i)$ от $v_0 = s$ до v_i и мы, если $E^-(v_i) = \varnothing$, удаляем v_i из сети вместе со всеми инцидентными ей дугами и начинаем новую итерацию; а если $E^-(v_i) \neq \varnothing$, то добавляем к пути $(v_0, v_1,\ldots, v_i)$ первую дугу (v_i, v_{i+1}) из списка $E^-(v_i)$. Если мы дошли до t, то понижаем пропускные способности дуг (v_i, v_{i+1}) для $i = 0, 1,\ldots, k-1$ на величину $\rho = \min\{c\,(v_i, v_{i+1}) \mid i \in \{0, 1,\ldots, k-1\}\}$ и удаляем дуги с новыми пропускными способностями, равными нулю.

На удаление из подсети одной дуги тратится $O(k)$ операций.

На k-ю группу — $O(|E||V|)$ операций.

Общая трудоемкость — $O(|E||V|^2)$ операций.

Алгоритмы нахождения максимального потока

$O(n^2 mC)$	Dantzig (1951) *simplex method*
$O(nmC)$	Ford and Fulkerson (1955, 1957) *augmenting path*
$O(nm^2)$	Dinits (1970), Edmonds and Karp (1972) *shortest augmenting path*
$O(n^2 m \log nC)$	Edmonds and Karp (1972) *fattest augmenting path*
$O(n^2 m)$	Dinits (1970) *shortest augmenting path, layered network*
$O(m^2 \log C)$	Edmonds and Karp (1972) *capacity–scaling*
$O(nm \log C)$	Dinits (1973), Gabov (1983, 1985) *capacity–scaling*
$O(n^3)$	Karzanov (1974) *preflow push*, Malhotra, Kumar and Maheshwari (1978), Tarjan (1984)
$O(n^2 \sqrt{m})$	Cherkasskii (1977) *blocking preflow with long pushes*
$O(n\, m \log^2 n)$	Shiloach (1978), Galil and Naamad (1979, 1980)
$O(n^{5/3} m^{2/3})$	Galil (1978, 1980)

$O(nm\log n)$	Sleator (1980), Sleator and Tarjan (1981,1983) *dynamic trees*
$O(nm\log(n^2/m))$	Goldberg and Tarjan (1986, 1988) *push-relabel+ dynamic trees*
$O(nm + n^2\log C)$	Ahuja and Orlin (1989) *push-relabel+excess scaling*
$O(nm + n^2\sqrt{\log C})$	Ahuja, Orlin, and Tarjan (1989) *Ahuja–Orlin improved*
$O(n\log((n/m)n^2\sqrt{\log C}+2))$	Ahuja, Orlin, and Tarjan (1989) *Ahuja–Orlin improved+ dynamic trees*
$O(n^3/\log n)$	Cheriyan, Hagerup and Mehlhorn (1990, 1996)
$O(n(m + n^{5/3}\log n))$	Alon (1990) (*derandomization of Cherian and Hagerup (1989, 1995)*)
$O(nm + n^{2+\varepsilon})$	(for each ε >0) King, Rao, and Tarjan (1992)
$O(nm\log_{m/n}n+n^2\log^{2+\varepsilon}n)$	(for each ε >0) Phillips and Westbrook (1993, 1998)
$O(nm\log_{\frac{m}{n\log n}}n)$	King, Rao, and Tarjan (1994)
$O(m^{3/2}\log(n^2/m)\log C)$	Goldberg and Rao (1997, 1998)
$O(n^{2/3}m\log(n^2/m)\log C)$	Goldberg and Rao (1997, 1998)

Использование сетевых моделей

Пусть $G = (V,E)$ — орграф, $B, C \subset V$, $B \cap C = \varnothing$.

Подмножество X дуг (вершин, не принадлежащих $B \cap C$) в G называется *(BC)-разделяющим*, если в графе $G–X$ все вершины C недостижимы из B.

Утверждение 1. Для любого орграфа $G = (V,E)$ и любых B, $C \subset V$, $B \cap C = \varnothing$ мощность наименьшего (B, C)–разделяющего множества дуг равна наибольшему количеству попарно непересекающихся по дугам путей, ведущих из B в C.

Доказательство. Построим сеть H по следующим правилам. Добавим к G вершины s и t и дуги (s, b), $b \in B$ и (c, t), $c \in C$. Положим пропускные способности дуг (s, b), $b \in B$ и (c, t), $c \in C$ равными $|E|+1$, а пропускные способности остальных дуг равными 1. Поскольку число дуг, выходящих из B, не больше $|E|$, то минимальный разрез $R = (X, \overline{X})$ в H не содержит дуг вида (s, b), $b \in B$ и (c, t), $c \in C$. Пропускная способность R равна числу дуг, ведущих из X в $\overline{X}$, а множество A этих дуг является (B, C)–

разделяющим. По теореме о максимальном потоке и минимальном разрезе в H существует поток f мощности $|A|$ из s в t.

Так как пропускные способности всех дуг H целочислены, то, следуя АФФ, можно выбрать f целочисленным. По теореме о разложении потоков, f можно представить в виде суммы целочисленных положительных потоков вдоль путей и циклов. Удалив потоки вдоль циклов, получим поток той же величины, являющийся суммой потоков вдоль путей. По определению пропускных способностей дуг из G, поток вдоль каждого из этих путей равен 1 и, следовательно, их число равно $|A|$. По той же причине эти пути не имеют общих дуг, кроме инцидентных s или t. Тогда части этих путей, полученные после удаления вершин s и t, удовлетворяют требованиям нашего утверждения. ■

Утверждение 2. Для любого орграфа $G = (V,E)$ и любых $B,\ C \subset V$, $B \cap C = \varnothing$ таких, что нет дуг, ведущих из B в C, мощность наименьшего (B, C)-разделяющего множества вершин равна

наибольшему количеству попарно непересекающихся по вершинам путей, ведущих из B в C.

Доказательство. Построим по следующим правилам вспомогательный орграф G'. Пусть V' — множество копий (дублей) элементов множества V, то есть $V'=\{v'|v\in V\}$.

Положим $V(G') = V \cup V'$, $E(G') = \{(v, v')|v\in V\} \cup \{(v', u)|(v, u) \in E\}$. Обозначим $B'=\{v'|v \in B\}$. По утверждению 1 мощность наименьшего (B, C)-разделяющего множества дуг равна наибольшему количеству попарно непересекающихся по дугам путей, ведущих из B' в C. Заметим, что в G' пути длины два и более, непересекающиеся по дугам, не пересекаются и по вершинам, а каждому пути в G' из B' в C взаимнооднозначно соответствует путь в G из B в C. ■

Пусть $G = (V, E)$ — граф, B, $C \subset V$, $B \cap C = \varnothing$. Подмножество X ребер (вершин, не принадлежащих $B \cup C$) в G называется *(B,C)-разделяющим*, если в графе $G–X$ все вершины C недостижимы из B.

Теорема Менгера. Пусть $G = (V, E)$ — граф, $\{v, u\} \subset V$, $(v, u) \notin E$. Тогда мощность наименьшего $(\{v\}, \{u\})$-разделяющего множества вершин в G равна наибольшему количеству попарно непересекающихся по внутренним вершинам путей, ведущих из v в u.

Доказательство. Построим вспомогательный орграф G' по следующим правилам:

$$V(G') = V, \qquad E(G') = \bigcup_{(v,u)\in E} \{(v,u),(u,v)\}$$

Применение утверждения 2 с $B = \{v\}$, $C = \{u\}$ к орграфу G' завершает доказательство. ■

Аналогично доказывается реберный вариант теоремы Менгера.

Утверждение 3. Пусть $G = (V, E)$ — граф, $\{v, u\} \subset V$. Тогда мощность наименьшего $(\{v\}, \{u\})$-разделяющего множества ребер в G равна наибольшему количеству попарно непересекающихся по ребрам путей, ведущих из v в u.

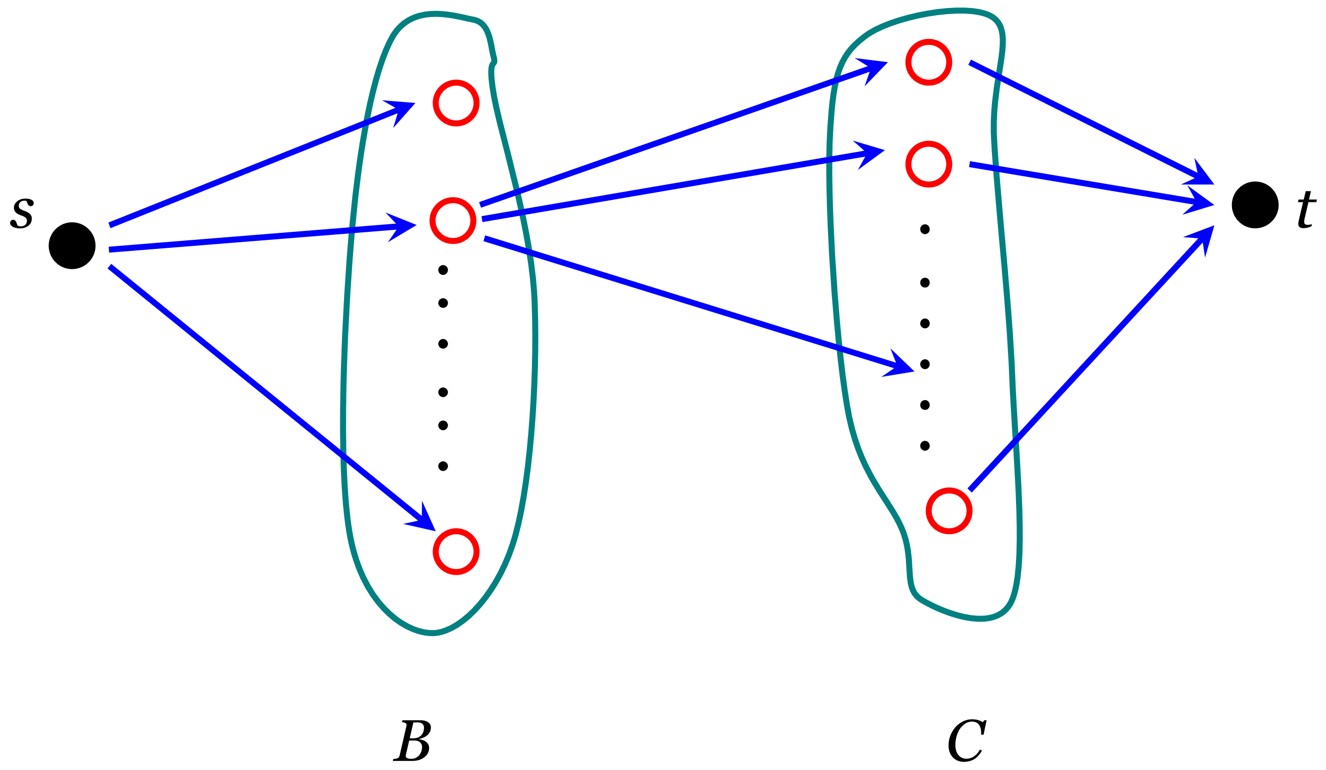
Реберной (вершинной) связностью $\lambda(G)$ $(\kappa(G))$ графа G называется наименьшее число ребер (вершин), после удаления которых граф становится несвязным или одновершинным.
Говорят, что граф G k-связен, если $\kappa(G) \geq k$.

Из теоремы Менгера легко следует

Теорема Уитни. Граф G является k-связным тогда и только тогда, когда любые две его вершины соединяют не менее k непересекающихся по внутренним вершинам путей.

Паросочетания в двудольном графе

Любой целочисленный поток в сети взаимно–однозначно соответствует паросочетанию в G, то есть $M(f) = \pi(G)$



$H = (V, E)$

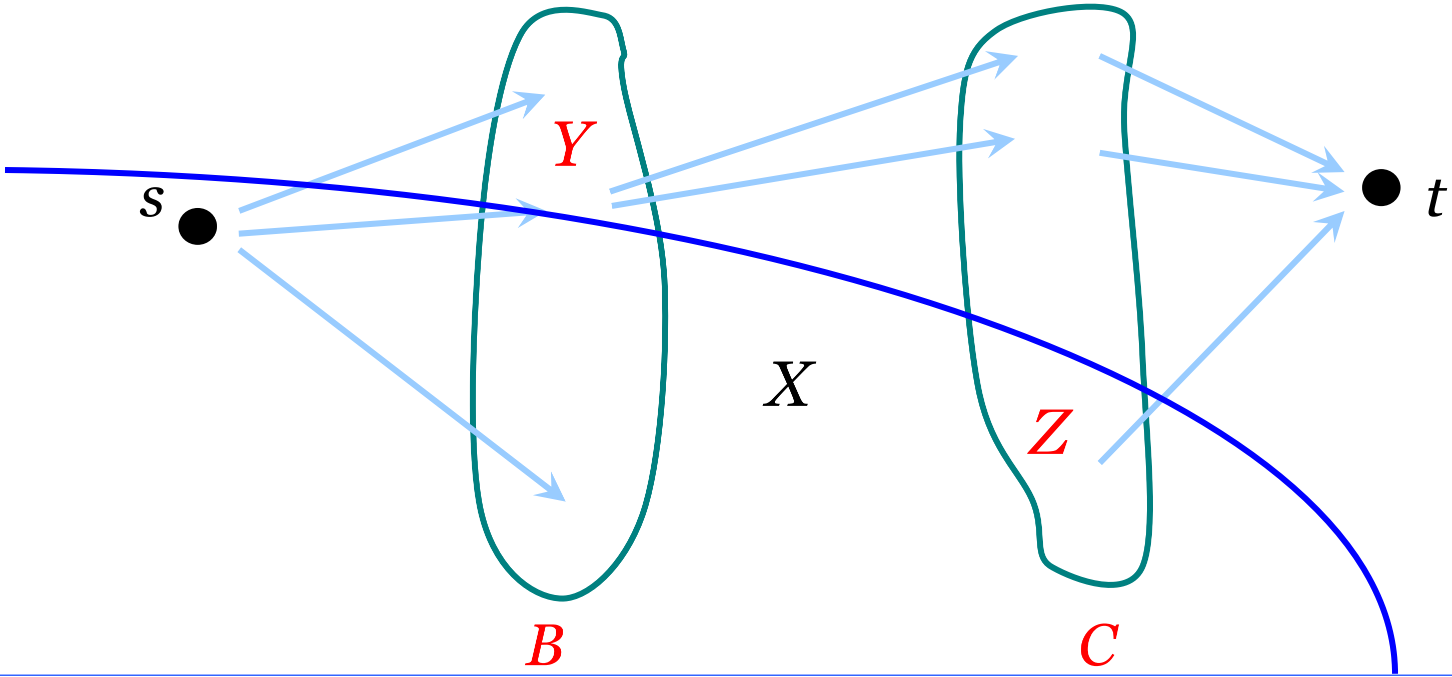
$V = B \cup C \cup \{s, t\}$

$c(e) = 1$, для всех дуг сети

Теорема Кёнига. Для любого двудольного мультиграфа G мощность $\pi(G)$ наибольшего паросочетания равна мощности $\tau(G)$ наименьшего множества вершин, покрывающего все ребра.

Доказательство. На паросочетания и покрытия наличие кратных ребер не влияет. Поэтому можно считать, что G — двудольный граф с долями B и C. Построим сеть H и поток f. Имеем $M(f) = \pi(G) \le \tau(G)$.

Пусть $R = (X, \overline{X})$ — минимальный разрез в сети H, $Y = \overline{X} \cap B$, $Z = X \cap C$.



Можно считать, что среди всех минимальных разрезов в сети H на разрезе R минимизируется $|X|$. Допустим, что в H есть дуга (b, c), $b \in B \setminus Y = X \cap B$, $c \in C \setminus Z = \overline{X} \cap C$. Тогда, положив $X' = X \setminus \{b\}$, получим разрез R' не большей пропускной способности с $|X'| = |X| - 1$. Это противоречит выбору R.

Следовательно, $Y \cup Z$ покрывает все ребра G, и $\tau(G) \le |Y \cup Z|$.

С другой стороны, по определению, $E^{-}(R) \supset \{(s,y) \mid y \in Y\} \cup \{(z,t) \mid z \in Z\}$, откуда $|Y \cup Z| \le |E^{-}(R)|$. Но $|E^{-}(R)| = M(f)$. ■

Составление расписаний на параллельных машинах

Имеется m одинаковых машин и n работ. Для каждой работы заданы время выполнения $p_i > 0$, время поступления на обслуживание $r_i \geq 0$ и директивный срок окончания $d_i \geq r_i$. Требуется найти расписание с минимальной задержкой $L_{\max} = \max\limits_{i=1,\dots,n} (c_i - d_i)$,

где c_i — время окончания работы i при возможности прерываний.

Для решения этой задачи сначала научимся отвечать на вопрос: *$\exists$ ли расписание с $L_{max} \leq L$?* А затем методом дихотомии найдем минимальное значение L, для которого такое расписание существует.

Пусть L задано. Если расписание существует, то работа i выполняется в интервале $[r_i, c_i]$ и $c_i - d_i \leq L$, то есть $c_i \leq L + d_i = d_i^L$ и можно считать, что работа i выполняется в интервале $[r_i, d_i^L]$.

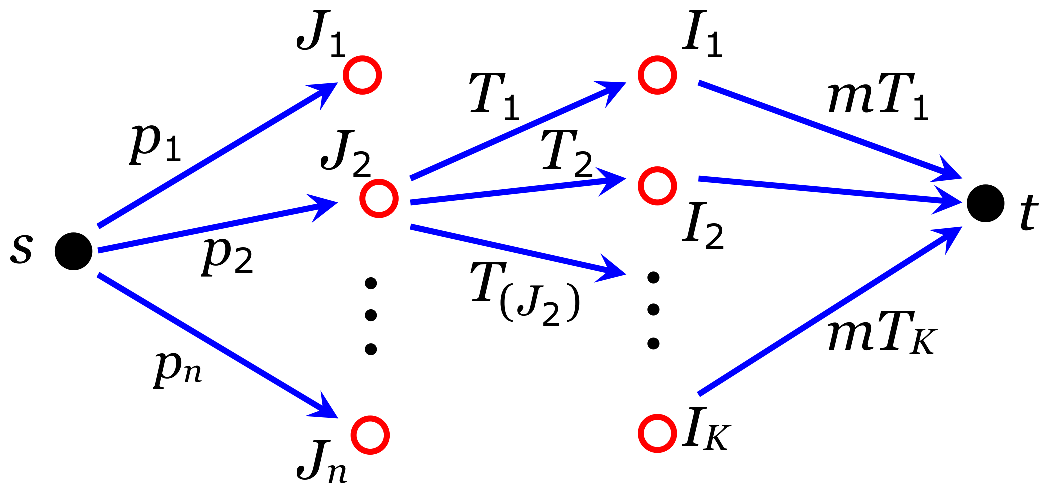
Чтобы ответить на вопрос: *$\exists$ ли расписание с прерываниями, где каждая работа выполняется в своем временном окне?*, используем задачу о потоке в сети.

Сольем два массива r_i, d_i^L и упорядочим их значения

$$t_1 < t_2 < \ldots < t_k, \quad k \leq 2n.$$

Рассматриваем только различные значения r и d.

Построим интервалы $I_k = [t_k, t_{k+1}]$, длины $T_k = t_{k+1} - t_k$ и рассмотрим сеть $G = (V, E, s, t)$:



Дуга (i, k) принадлежит E, если работа J_i может выполняться в интервале I_k, т.е. $I_k \subseteq [r_i, d_i^L]$. Каждой дуге приписан вес, как показано на рисунке.

Решим задачу о максимальном потоке в этой сети. Получим $M^*(f)$.

Сравним эту величину с $\sum_{i=1}^{n} p_i$. Если они равны, то искомое расписание существует, если нет, то есть $\sum p_i > M^*(f)$, то такого расписания не существует.

Пусть имеет место равенство. Тогда сохранение потока в каждой вершине J_i дает:

$$\sum_k f_{ik} = p_i, \quad \forall i = 1,\ldots,n$$

и величины f_{ik} определяют расписание работы J_i.

Сохранение потока в вершине I_k: $\sum_i f_{ik} \le mT_k, \quad \forall k = 1,\ldots,K$, гарантирует, что m машин справятся со всеми работами в интервале T_k.