Комбинаторные алгоритмы

Паросочетания в двудольных графах

Гальперин Александр Леонидович

2018 г.

Определение

Паросочетанием в графе называется произвольное множество его ребер такое, что каждая вершина графа инцидентна не более чем одному ребру из этого множества.

Определение

Паросочетанием в графе называется произвольное множество его ребер такое, что каждая вершина графа инцидентна не более чем одному ребру из этого множества.

$\mathcal{NB}$

- Рассматривая различные задачи о паросочетаниях, мы ограничимся случаем двудольных графов.

Паросочетания в двудольных графах

Основные понятия

Определение

Паросочетанием в графе называется произвольное множество его ребер такое, что каждая вершина графа инцидентна не более чем одному ребру из этого множества.

$\mathcal{NB}$

- Рассматривая различные задачи о паросочетаниях, мы ограничимся случаем двудольных графов.
- Для решения аналогичных задач в произвольных графах используются те же идеи, но реализация существенно усложняется.

Паросочетания в двудольных графах

Основные понятия

Определение

Граф $G = (V, E)$ называется *двудольным*, если множество его вершин V можно разбить на два непересекающихся подмножества X и Y так, что каждое ребро $e \in E$ имеет вид $e = xy$, где $x \in X, y \in Y$.

Определение

Граф $G = (V, E)$ называется *двудольным*, если множество его вершин V можно разбить на два непересекающихся подмножества X и Y так, что каждое ребро $e \in E$ имеет вид $e = xy$, где $x \in X, y \in Y$.

$\mathcal{NB}$

Двудольных граф будем обозначать (X, Y, E), если граф не является взвешенным, и (X, Y, E, c), если ребрам $e \in E$ приписаны веса $c(e)$.

Паросочетания в двудольных графах

Основные понятия

$\mathcal{NB}$

Всюду в дальнейшем, говоря о двудольном графе, мы предполагаем, что разбиение множества V на подмножества X и Y зафиксировано.

Самые разные практические задачи связаны с построением тех или иных паросочетаний в двудольных графах. Рассмотрим несколько примеров.

Пример 1

Пусть имеется n рабочих, каждый из которых может выполнить один или несколько из m видов работ. При этом каждый из m видов работ должен быть выполнен одним рабочим. Требуется так распределить работы среди рабочих, чтобы наибольшее количество работ оказалось выполненными.

Пример 2

Пусть в предыдущей задаче количество рабочих n будет равно числу работ m. Спрашивается, можно ли так распределить работы между рабочими, чтобы были выполнены все виды работ?

Паросочетания в двудольных графах

Основные понятия

Пример 3

Пусть сверх условий второй задачи для каждой пары рабочий–работа известна стоимость $c(x, y)$ выполнения рабочим x работы y. Требуется так подобрать каждому рабочему определенный вид работы, чтобы суммарная стоимость выполнения всех работ была минимальна.

Математическая модель всех приведенных выше примеров строится очевидным образом.

- Определим двудольный граф $G = (X, Y, E)$, в котором в качестве X выберем рабочих, а в качестве Y — множество работ.
- Множество ребер E определим как множество всех пар (x, y) таких, что рабочий x может выполнить работу y.

Пусть $M \subseteq E$ — паросочетание в построенном графе G. Тогда каждое ребро $e = xy$ можно интерпретировать как назначение рабочему x работы y.

Действительно, по определению паросочетания никакие два ребра из M не могут иметь общих вершин. Следовательно, на каждую работу назначается не более одного рабочего, и каждый рабочий получает не более одной работы.

Паросочетания в двудольных графах

Основные понятия

В этой модели

- первая задача означает, что в графе требуется найти паросочетание с наибольшим количеством ребер;
- вторая задача — выяснить, существует ли паросочетание, состоящее из n ребер;
- третья задача — найти паросочетание из n ребер с минимальным суммарным весом.

В этой модели

- первая задача означает, что в графе требуется найти паросочетание с наибольшим количеством ребер;
- вторая задача — выяснить, существует ли паросочетание, состоящее из n ребер;
- третья задача — найти паросочетание из n ребер с минимальным суммарным весом.

В этой теме будут рассмотрены все три вида задач.

Пусть M — паросочетание в графе $G = (X, Y, E)$. Говорят, что M *сочетает* x *с* y *и* y *с* x, если $xy \in M$.

Пусть M — паросочетание в графе $G = (X, Y, E)$. Говорят, что M *сочетает* x *с* y *и* y *с* x, если $xy \in M$.

Вершины, не принадлежащие ни одному ребру из паросочетания, называют *свободными относительно* M или просто *свободными*, а все прочие — *насыщенными* в M или просто *насыщенными*.

Паросочетания в двудольных графах

Основные понятия

Пусть M — паросочетание в графе $G = (X, Y, E)$. Говорят, что M *сочетает x с y и y с x*, если $xy \in M$.

Вершины, не принадлежащие ни одному ребру из паросочетания, называют *свободными относительно M* или просто *свободными*, а все прочие — *насыщенными* в M или просто *насыщенными*.

Удобно также все ребра, входящие в паросочетание M называть *M–темными* или просто *темными*, а все прочие — *M–светлыми* или просто *светлыми*.

Паросочетание, содержащее наибольшее число ребер, называется *наибольшим*. Паросочетание, не содержащееся ни в каком другом паросочетании, называется *максимальным* (по включению).

Паросочетания в двудольных графах

Основные понятия

Паросочетание, содержащее наибольшее число ребер, называется *наибольшим*. Паросочетание, не содержащееся ни в каком другом паросочетании, называется *максимальным* (по включению).

$\mathcal{NB}$

Любое наибольшее паросочетание является максимальным. Обратное неверно.

Паросочетание, содержащее наибольшее число ребер, называется *наибольшим*. Паросочетание, не содержащееся ни в каком другом паросочетании, называется *максимальным* (по включению).

$\mathcal{NB}$

Любое наибольшее паросочетание является максимальным. Обратное неверно.

Упражнение

Постройте пример максимального паросочетания, не являющегося наибольшим.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании
Алгоритм Хопкрофта–Карпа

Задача о наибольшем паросочетании

Задача о наибольшем паросочетании состоит в следующем: *в заданном двудольном графе найти наибольшее паросочетание.*

Задача о наибольшем паросочетании

Задача о наибольшем паросочетании состоит в следующем: *в заданном двудольном графе найти наибольшее паросочетание.*

Оказывается, эту задачу можно свести к задаче построения максимального потока в некоторой сети.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Пусть $G = (X, Y, E)$ — произвольный двудольный граф и $s, t \notin X \cup Y$.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Пусть $G = (X, Y, E)$ — произвольный двудольный граф и $s, t \notin X \cup Y$.

- Построим сеть $G^* = (V^*, E^*, c)$ с источником s и стоком t.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Пусть $G = (X, Y, E)$ — произвольный двудольный граф и $s, t \notin X \cup Y$.

- Построим сеть $G^* = (V^*, E^*, c)$ с источником s и стоком t.
- В качестве множества вершин сети G^* возьмем множество $V^* = X \cup Y \cup \{s\} \cup \{t\}$.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Пусть $G = (X, Y, E)$ — произвольный двудольный граф и $s, t \notin X \cup Y$.

- Построим сеть $G^* = (V^*, E^*, c)$ с источником s и стоком t.
- В качестве множества вершин сети G^* возьмем множество $V^* = X \cup Y \cup \{s\} \cup \{t\}$.
- Множество дуг E^* определим следующим образом: каждое ребро $e = xy \in E$, где $x \in X, y \in Y$, превращаем в дугу xy, исходящую из x и входящую в y.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

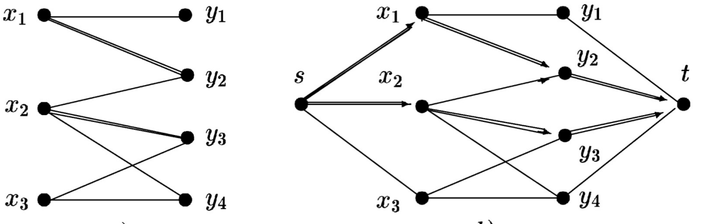
Пусть $G = (X, Y, E)$ — произвольный двудольный граф и $s, t \notin X \cup Y$.

- Построим сеть $G^* = (V^*, E^*, c)$ с источником s и стоком t.
- В качестве множества вершин сети G^* возьмем множество $V^* = X \cup Y \cup \{s\} \cup \{t\}$.
- Множество дуг E^* определим следующим образом: каждое ребро $e = xy \in E$, где $x \in X, y \in Y$, превращаем в дугу xy, исходящую из x и входящую в y.
- Добавим к полученному множеству все дуги вида sx, yt, где $x \in X, y \in Y$.

Пусть $G = (X, Y, E)$ — произвольный двудольный граф и $s, t \notin X \cup Y$.

- Построим сеть $G^* = (V^*, E^*, c)$ с источником s и стоком t.
- В качестве множества вершин сети G^* возьмем множество $V^* = X \cup Y \cup \{s\} \cup \{t\}$.
- Множество дуг E^* определим следующим образом: каждое ребро $e = xy \in E$, где $x \in X, y \in Y$, превращаем в дугу xy, исходящую из x и входящую в y.
- Добавим к полученному множеству все дуги вида sx, yt, где $x \in X, y \in Y$.
- Пропускную способность каждой дуги положим равной единице.

На рисунке показан пример графа G и соответствующей ему сети G^*.



$a)$ $b)$

$\mathcal{NB}$

Заметим, что:

- если f — целочисленный поток в сети G^*, то $f(e) = 0$ или $f(e) = 1$ для любой дуги $e \in E$;
- среди таких 0–1 потоков существует максимальный поток.

Оказывается, что каждому паросочетанию в графе G однозначно соответствует какой-нибудь 0–1 поток в сети G^*.

Пусть $\mathcal{P}$ — множество всех паросочетаний в графе G, а $\mathcal{F}$ — множество всех 0–1 потоков в сети G^*.

Теорема 1

Существует взаимно–однозначное отображение ϕ множества $\mathcal{P}$ на множество $\mathcal{F}$, причем $|M| = ||\phi(M)||$ для любого паросочетания M (здесь $||\phi(M)||$ — величина потока $\phi(M)$).

Доказательство. Для произвольного паросочетания M в графе G определим поток $f_M = \phi(M)$ в сети G^* по формулам:

$$f_M(s,x) = f_M(x,y) = f_M(y,t) = 1 \text{ для любого ребра } xy \in M \text{ и}$$
$$f_M(x,y) = 0 \text{ для остальных дуг } e \in E^*.$$

Докажем, что f_M — поток в сети G^*.

Поскольку $0 \leqslant f_M(e) \leqslant 1$, условие ограничения по пропускной способности выполняется. Остается проверить условие сохранения потока в вершинах.

- Пусть x — насыщенная в M вершина из X паросочетания M. Тогда $f_M(s, x) = 1$ и, следовательно, $f_M(x-) = 1$, т.е. поток, втекающий в вершину x, равен 1.

- Пусть x — насыщенная в M вершина из X паросочетания M. Тогда $f_M(s, x) = 1$ и, следовательно, $f_M(x-) = 1$, т.е. поток, втекающий в вершину x, равен 1.
- Поскольку вершина x насыщена в M, то существует равно одно ребро $xy \in M$, инцидентное x.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Пусть x — насыщенная в M вершина из X паросочетания M. Тогда $f_M(s, x) = 1$ и, следовательно, $f_M(x-) = 1$, т.е. поток, втекающий в вершину x, равен 1.
- Поскольку вершина x насыщена в M, то существует равно одно ребро $xy \in M$, инцидентное x.
- Для этого ребра $f_M(x, y) = 1$ на соответствующей дуге xy.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Пусть x — насыщенная в M вершина из X паросочетания M. Тогда $f_M(s, x) = 1$ и, следовательно, $f_M(x-) = 1$, т.е. поток, втекающий в вершину x, равен 1.
- Поскольку вершина x насыщена в M, то существует равно одно ребро $xy \in M$, инцидентное x.
- Для этого ребра $f_M(x, y) = 1$ на соответствующей дуге xy.
- Все остальные ребра графа, инцидентные вершине x, являются светлыми, т.е. не входят в паросочетание M. Поэтому $f_M(e) = 0$ для всех соответствующих дуг.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Пусть x — насыщенная в M вершина из X паросочетания M. Тогда $f_M(s,x) = 1$ и, следовательно, $f_M(x-) = 1$, т.е. поток, втекающий в вершину x, равен 1.
- Поскольку вершина x насыщена в M, то существует равно одно ребро $xy \in M$, инцидентное x.
- Для этого ребра $f_M(x,y) = 1$ на соответствующей дуге xy.
- Все остальные ребра графа, инцидентные вершине x, являются светлыми, т.е. не входят в паросочетание M. Поэтому $f_M(e) = 0$ для всех соответствующих дуг.
- Следовательно, $f_M(x+) = 1$.

- Пусть x — насыщенная в M вершина из X паросочетания M. Тогда $f_M(s, x) = 1$ и, следовательно, $f_M(x-) = 1$, т.е. поток, втекающий в вершину x, равен 1.
- Поскольку вершина x насыщена в M, то существует равно одно ребро $xy \in M$, инцидентное x.
- Для этого ребра $f_M(x, y) = 1$ на соответствующей дуге xy.
- Все остальные ребра графа, инцидентные вершине x, являются светлыми, т.е. не входят в паросочетание M. Поэтому $f_M(e) = 0$ для всех соответствующих дуг.
- Следовательно, $f_M(x+) = 1$.
- Тем самым, равенство $f_M(x-) = f_M(x+)$ выполняется для всех насыщенных вершин.

- Если же x — не насыщена, т.е. x — свободная вершина, то как входящий в x, так и выходящий из x потоки равны нулю.

- Если же x — не насыщена, т.е. x — свободная вершина, то как входящий в x, так и выходящий из x потоки равны нулю.
- Следовательно, Условие сохранения потока в вершинах, лежащих в X, выполняется.

- Если же x — не насыщена, т.е. x — свободная вершина, то как входящий в x, так и выходящий из x потоки равны нулю.
- Следовательно, Условие сохранения потока в вершинах, лежащих в X, выполняется.
- Для вершин $y \in Y$ условие проверяется аналогично.

- Далее, поскольку количеству насыщенных в M вершин $x \in X$ в точности равно $|M|$, то $f_M(s+) = |M|$, т.е. $||\phi(M)|| = |M|$.

- Далее, поскольку количеству насыщенных в M вершин $x \in X$ в точности равно $|M|$, то $f_M(s+) = |M|$, т.е. $||\phi(M)|| = |M|$.
- Легко видеть, что разным паросочетаниям соответствуют разные потоки. Следовательно, отображение $\phi : \mathcal{P} \longrightarrow \mathcal{F}$, определенное равенством $\phi(M) = f_M$, инъективно.

- Обратно, пусть f — 0–1 поток в сети G^*. Положим

$$M_f = \{xy | f(x, y) = 1, x \in X, y \in Y\}.$$

- Обратно, пусть f — 0–1 поток в сети G^*. Положим

$$M_f = \{xy | f(x,y) = 1, x \in X, y \in Y\}.$$

- Поскольку в каждую вершину $x \in X$ входит ровно одна дуга (это дуга вида sx), то имеется не более одной дуги вида xy, для которой $f(x,y) = 1$.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Обратно, пусть f — 0–1 поток в сети G^*. Положим

$$M_f = \{xy | f(x, y) = 1, x \in X, y \in Y\}.$$

- Поскольку в каждую вершину $x \in X$ входит ровно одна дуга (это дуга вида sx), то имеется не более одной дуги вида xy, для которой $f(x, y) = 1$.
- Следовательно, каждая вершина $x \in X$ инцидентна не более чем одному ребру из M_f.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Обратно, пусть f — 0–1 поток в сети G^*. Положим

$$M_f = \{xy | f(x,y) = 1, x \in X, y \in Y\}.$$

- Поскольку в каждую вершину $x \in X$ входит ровно одна дуга (это дуга вида sx), то имеется не более одной дуги вида xy, для которой $f(x,y) = 1$.
- Следовательно, каждая вершина $x \in X$ инцидентна не более чем одному ребру из M_f.
- Аналогично, каждая вершина $y \in Y$ инцидентна не более чем одному ребру из M_f.

- Обратно, пусть f — 0–1 поток в сети G^*. Положим

$$M_f = \{xy | f(x, y) = 1, x \in X, y \in Y\}.$$

- Поскольку в каждую вершину $x \in X$ входит ровно одна дуга (это дуга вида sx), то имеется не более одной дуги вида xy, для которой $f(x, y) = 1$.
- Следовательно, каждая вершина $x \in X$ инцидентна не более чем одному ребру из M_f.
- Аналогично, каждая вершина $y \in Y$ инцидентна не более чем одному ребру из M_f.
- Отсюда следует, что M_f является паросочетанием в графе G.

- Легко видеть, что для паросочетания M_f справедливы равенства

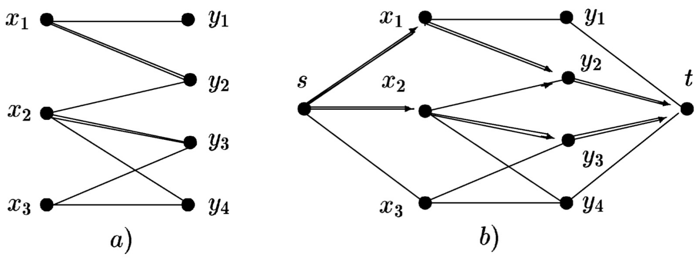
$$|M_f| = ||f|| \quad \text{и} \quad \phi(M_f) = f,$$

что завершает доказательство теоремы. ■

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

На рисунке изображены двудольный граф G, сеть G^*, паросочетание $M = \{x_1y_2, x_2y_3\}$ и соответствующий ему поток f_M:



- Пусть f — произвольный 0–1–поток в сети G^* и P — f–дополняющая (s, t)–цепь.Тогда цепь P содержит нечетное количество дуг.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Пусть f — произвольный 0–1–поток в сети G^* и P — f–дополняющая (s,t)–цепь. Тогда цепь P содержит нечетное количество дуг.
- Удалим из этой цепи первую дугу (она обязательно имеет вид $sx, x \in X$) и последнюю дугу (она имеет вид $yt, y \in Y$).

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Пусть f — произвольный 0–1–поток в сети G^* и P — f–дополняющая (s, t)–цепь.Тогда цепь P содержит нечетное количество дуг.
- Удалим из этой цепи первую дугу (она обязательно имеет вид $sx, x \in X$) и последнюю дугу (она имеет вид $yt, y \in Y$).
- Оставшиеся дуги цепи P чередуются следующим образом:
  - первой идет дуга вида xy, для которой $f(x, y) = 0$, поскольку эта дуга прямая в цепи;

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Пусть f — произвольный 0–1–поток в сети G^* и P — f–дополняющая (s, t)–цепь.Тогда цепь P содержит нечетное количество дуг.
- Удалим из этой цепи первую дугу (она обязательно имеет вид $sx, x \in X$) и последнюю дугу (она имеет вид $yt, y \in Y$).
- Оставшиеся дуги цепи P чередуются следующим образом:
  - первой идет дуга вида xy, для которой $f(x, y) = 0$, поскольку эта дуга прямая в цепи;
  - второй — дуга вида $x_! y$, причем — это дуга обратная в цепи P; поэтому $f(x_1, y) = 1$.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Пусть f — произвольный 0–1–поток в сети G^* и P — f–дополняющая (s,t)–цепь.Тогда цепь P содержит нечетное количество дуг.
- Удалим из этой цепи первую дугу (она обязательно имеет вид $sx, x \in X$) и последнюю дугу (она имеет вид $yt, y \in Y$).
- Оставшиеся дуги цепи P чередуются следующим образом:
  - первой идет дуга вида xy, для которой $f(x,y) = 0$, поскольку эта дуга прямая в цепи;
  - второй — дуга вида $x_!y$, причем — это дуга обратная в цепи P; поэтому $f(x_1, y) = 1$.
  - затем снова прямая дуга; потом — обратная, и т.д.

- Увеличение потока вдоль этой цепи по правилам алгоритма Форда–Фалкерсона приводит к тому, что новый поток становится равным единице на всех нечетных (т.е. прямых) дугах цепи P, и равным нулю на всех четных (т.е. обратных) дугах цепи P.

- Увеличение потока вдоль этой цепи по правилам алгоритма Форда–Фалкерсона приводит к тому, что новый поток становится равным единице на всех нечетных (т.е. прямых) дугах цепи P, и равным нулю на всех четных (т.е. обратных) дугах цепи P.
- Величина потока при этом возрастает на единицу.

Понятие f–дополняющей цепи для потока f в сети G^* естественным образом соответствует понятию *M–чередующейся цепи* для паросочетания M в графе G.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Пусть M — паросочетание в графе G. M– *чередующейся цепью* называется такая последовательность вершин и ребер вида

$$x_0, \quad x_0 y_1, \quad y_1, \quad y_1 x_2, \quad x_2, \quad \dots, \quad x_k, \quad x_k y_{k+1},$$

где $k > 0$, что все вершины этой цепи различны, x_0 и y_{k+1} — свободные, а все остальные вершины насыщенные в паросочетании M, причем каждое второе ребро принадлежит M (т.е. ребра $y_i x_{i+1}$ входят в M), а остальные ребра в M не входят.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Пусть M — паросочетание в графе G. M– *чередующейся цепью* называется такая последовательность вершин и ребер вида

$$x_0, \quad x_0y_1, \quad y_1, \quad y_1x_2, \quad x_2, \quad \dots, \quad x_k, \quad x_ky_{k+1},$$

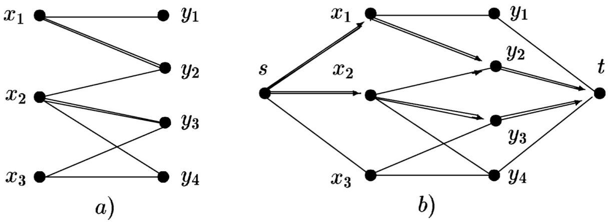
где $k > 0$, что все вершины этой цепи различны, x_0 и y_{k+1} — свободные, а все остальные вершины насыщенные в паросочетании M, причем каждое второе ребро принадлежит M (т.е. ребра y_ix_{i+1} входят в M), а остальные ребра в M не входят.

Другими словами, в M–чередующейся цепи цвета ребер чередуются по правилу *светлое–темное* и наоборот, причем первое и последнее ребра — *светлые*.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

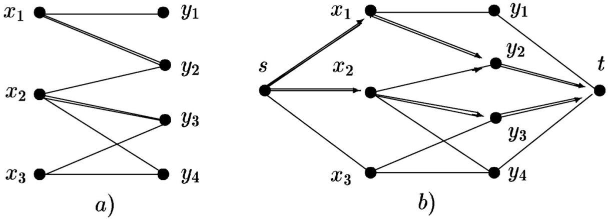
Ясно, что чередующаяся цепь однозначно определяется как последовательностью ее вершин, так и последовательностью ее ребер.



Например, здесь M–чередующуюся цепь можно задать последовательностью вершин $x_3, y_3, x_2, y_2, x_1, y_1$. Эта цепь содержит два темных ребра и три светлых.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа



Соответствующая f–дополняющая цепь в сети G^*, где $f = f_M$, задается последовательностью вершин $s, x_3, y_3, x_2, y_2, x_1, y_1, t$.

В этой цепи дуги x_2y_3 являются обратными, а остальные дуги — прямыми.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Увеличение потока вдоль f–дополняющей цепи соответствует увеличению количества ребер количеству ребер в паросочетании M вдоль M–чередующейся цепи.

- Увеличение потока вдоль f–дополняющей цепи соответствует увеличению количества ребер количеству ребер в паросочетании M вдоль M–чередующейся цепи.
- Для этого в M–чередующейся цепи нечетные ребра, не входившие в M, объявляются элементами M, а все четные, входившие в M, из M удаляются.

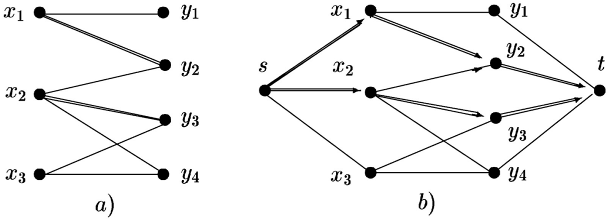
Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Увеличение потока вдоль f–дополняющей цепи соответствует увеличению количества ребер количеству ребер в паросочетании M вдоль M–чередующейся цепи.
- Для этого в M–чередующейся цепи нечетные ребра, не входившие в M, объявляются элементами M, а все четные, входившие в M, из M удаляются.
- Иначе говоря, все темные ребра становятся светлыми, а все светлые — темными. Такая операция приводит к увеличению количества ребер в паросочетании на единицу.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа



Например паросочетание M на этом рисунке заменится на паросочетание $\{x_3y_3, x_2y_2, x_1y_1\}$.

Процесс получения нового паросочетания M_1 из паросочетания M с помощью M–чередующейся цепи P можно выразить равенством

$$M_1 = M \oplus P = (M \backslash P) \cup (P \backslash M)$$

(здесь и далее под M–чередующейся цепью понимается последовательность ребер).

Процесс получения нового паросочетания M_1 из паросочетания M с помощью M–чередующейся цепи P можно выразить равенством

$$M_1 = M \oplus P = (M \backslash P) \cup (P \backslash M)$$

(здесь и далее под M–чередующейся цепью понимается последовательность ребер).

Для паросочетания M_1 справедливо равенство $|M_1| = |M| + 1$, поскольку в цепи P светлых ребер на одно больше, чем темных.

Из приведенных рассуждений вытекает следующий классический результат.

Теорема 2 (Берж, 1957)

Паросочетание M в двудольном графе G является наибольшим тогда и только тогда, когда в G не существует M–чередующейся цепи.

Из приведенных рассуждений вытекает следующий классический результат.

Теорема 2 (Берж, 1957)

Паросочетание M в двудольном графе G является наибольшим тогда и только тогда, когда в G не существует M–чередующейся цепи.

Ради полноты изложения приведем прямое доказательство этой теоремы, не опирающееся на теорему Форда–Фалкерсона.

Доказательство. $\boxed{\Rightarrow}$ Очевидно, что если M — наибольшее паросочетание, то в графе $G = (X, Y, E)$ не существует M–чередующейся цепи. $\square$

Доказательство. $\boxed{\Rightarrow}$ Очевидно, что если M — наибольшее паросочетание, то в графе $G = (X, Y, E)$ не существует M–чередующейся цепи. $\square$

$\boxed{\Leftarrow}$ Докажем обратное. Предположим, что для паросочетания M не сущетвует M–чередующейся цепи. Рассмотрим произвольное паросочетание N и убедимся, что $|N| \leqslant |M|$.

Заметим, что в графе $G_1 = (X, Y, N \cup M)$ степень каждой вершины не превосходит двух. Следовательно, каждая компонента связности графа G_1 может быть одного из следующих типов:

1. изолированная вершина;
2. цепь четной длины;
3. цепь нечетной длины;
4. цикл.

- В случаях 1, 2 и 4 компонента связности содержит одинаковое количество ребер из M и N.

- В случаях 1, 2 и 4 компонента связности содержит одинаковое количество ребер из M и N.
- Если же компонента связности — цепь нечетной длины (случай 3), то она является либо M–чередующейся, либо N–чередую цепью.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- В случаях 1, 2 и 4 компонента связности содержит одинаковое количество ребер из M и N.
- Если же компонента связности — цепь нечетной длины (случай 3), то она является либо M–чередующейся, либо N–чередую цепью.
- По предположению, M–чередующихся цепей в графе нет. Следовательно, могут быть только N–чередующиеся цепи.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- В случаях 1, 2 и 4 компонента связности содержит одинаковое количество ребер из M и N.
- Если же компонента связности — цепь нечетной длины (случай 3), то она является либо M–чередующейся, либо N–чередую цепью.
- По предположению, M–чередующихся цепей в графе нет. Следовательно, могут быть только N–чередующиеся цепи.
- Но в каждой из этих цепей ребер из M на одно больше, чем ребер из N. Отсюда $|N| \leqslant |M|$. Т.е. M — наибольшее паросочетание. □ ■

Теорема Бержа подсказывает следующий алгоритм построения наибольшего паросочетания.

1. Пустое паросочетание M объявить текущим.

Теорема Бержа подсказывает следующий алгоритм построения наибольшего паросочетания.

1. Пустое паросочетание M объявить текущим.
2. Искать M–чередующуюся цепь.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Теорема Бержа подсказывает следующий алгоритм построения наибольшего паросочетания.

1. Пустое паросочетание M объявить текущим.
2. Искать M–чередующуюся цепь.
3. Если такая цепь P найдена, то положить $M = M \oplus P$ и вернуться на шаг 2.

Теорема Бержа подсказывает следующий алгоритм построения наибольшего паросочетания.

1. Пустое паросочетание M объявить текущим.
2. Искать M–чередующуюся цепь.
3. Если такая цепь P найдена, то положить $M = M \oplus P$ и вернуться на шаг 2.
4. Иначе СТОП (текущее паросочетание M — наибольшее).

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

$\mathcal{NB}$

1. Предложенный алгоритм является лишь легкой модификацией алгоритма Форда–Фалкерсона.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

$\mathcal{NB}$

1. Предложенный алгоритм является лишь легкой модификацией алгоритма Форда–Фалкерсона.
2. Поскольку каждый раз текущее паросочетание увеличивается ровно на единицу, то алгоритм завершит работу не более, чем за n итераций, где $n = \min\{|X|, |Y|\}$ (здесь использован тот факт, что наибольшее паросочетание содердит не более n ребер).

Разберем подробнее процесс поиска M–чередующейся цепи. Можо использовать любую схему: и ПВГ, и ПВШ. Чуть удобнее ПВШ. Сформулируем правило поиска в виде «волнового» алгоритма.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

1. В нулевой фронт распространения волны включаем все M–свободные вершины $x \in X$.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

1. В нулевой фронт распространения волны включаем все M–свободные вершины $x \in X$.
2. Пусть фронт с номером k построен.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

1. В нулевой фронт распространения волны включаем все M–свободные вершины $x \in X$.
2. Пусть фронт с номером k построен.
   - Если k — четно, то фронт $k+1$ включает все вершины $y \in Y$, не содержащиеся ни в каком из предыдущих фронтов, которые можно пометить из вершин предыдущего фронта с помощью светлых ребер.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

1. В нулевой фронт распространения волны включаем все M–свободные вершины $x \in X$.
2. Пусть фронт с номером k построен.
   - Если k — четно, то фронт $k+1$ включает все вершины $y \in Y$, не содержащиеся ни в каком из предыдущих фронтов, которые можно пометить из вершин предыдущего фронта с помощью светлых ребер.
   - Если k нечетно, то во фронт $k+1$ включаем вершины $x \in X$, не содержащиеся ни в каком из предыдущих фронтов, которые можно пометить с помощью темных ребер из вершин предыдущего фронта.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

1. В нулевой фронт распространения волны включаем все M–свободные вершины $x \in X$.
2. Пусть фронт с номером k построен.
   - Если k — четно, то фронт $k+1$ включает все вершины $y \in Y$, не содержащиеся ни в каком из предыдущих фронтов, которые можно пометить из вершин предыдущего фронта с помощью светлых ребер.
   - Если k нечетно, то во фронт $k+1$ включаем вершины $x \in X$, не содержащиеся ни в каком из предыдущих фронтов, которые можно пометить с помощью темных ребер из вершин предыдущего фронта.
3. Поиск завершается, как только будет помечена свободная вершина $y \in Y$, или очередной фронт окажется пустым.

1. В первом случае окончание поиска M–чередующаяся цепь существует и может быть легко восстановлена с помощью стандартных меток *Previous*.

1. В первом случае окончание поиска M–чередующаяся цепь существует и может быть легко восстановлена с помощью стандартных меток *Previous*.
2. Во втором случае M–чередующейся цепи в графе не существует и, следовательно, текущее паросочетание M — наибольшее.

Теорема 3

Модифицированный алгоритм Форда–Фалкерсона построения наибольшего паросочетания в двудольном графе $G = (X, Y, E)$ *имеет сложность* $O(pqn)$, *где* $p = |X|, q = |Y|, n = \min\{p, q\}$ *и граф* G *представлен матрицей* $A[1 \dots p, 1 \dots q]$.

Теорема 3

Модифицированный алгоритм Форда–Фалкерсона построения наибольшего паросочетания в двудольном графе $G = (X, Y, E)$ *имеет сложность* $O(pqn)$, *где* $p = |X|, q = |Y|, n = \min\{p, q\}$ *и граф* G *представлен матрицей* $A[1 \dots p, 1 \dots q]$.

$\mathcal{NB}$

Заметим, что матрица A — подматрица матрицы смежности двудольного графа G.

Доказательство. Выше уже отмечалось, что процесс завершится не более, чем за n итераций. Сложность каждой итерации есть $O(pq)$. Поэтому модифицированный алгоритм Форда–Фалкерсона для построения наибольшего паросочетания имеет сложность $O(pqn)$. ∎

Доказательство. Выше уже отмечалось, что процесс завершится не более, чем за n итераций. Сложность каждой итерации есть $O(pq)$. Поэтому модифицированный алгоритм Форда–Фалкерсона для построения наибольшего паросочетания имеет сложность $O(pqn)$. ∎

$\mathcal{NB}$

Отметим, что в частном случае, когда $n = |X| = |Y|$, модифицированный алгоритм Форда–Фалкерсона имеет сложность $O(n^3)$.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

В 1973 году Хопкрофт и Карп предложили более эффективный алгоритм построения наибольшего паросочетания в двудольном графе.

В 1973 году Хопкрофт и Карп предложили более эффективный алгоритм построения наибольшего паросочетания в двудольном графе.

Пусть M — паросочетание в графе G. *Цепь P назовем M–цепью, если она начинается в свободной вершине $x \in X$, имеет нечетную длину, и цвета ребер чередуются по правилу светлое–темное и наоборот*.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

В 1973 году Хопкрофт и Карп предложили более эффективный алгоритм построения наибольшего паросочетания в двудольном графе.

Пусть M — паросочетание в графе G. *Цепь P назовем M–цепью, если она начинается в свободной вершине $x \in X$, имеет нечетную длину, и цвета ребер чередуются по правилу светлое–темное и наоборот*.

Другими словами, M–цепь —это почти M–чередующаяся цепь, только M–цепь может заканчиваться в насыщенной вершине $y \in Y$.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

$\mathcal{NB}$

Каждая M–чередующаяся цепь является M–цепью. Обратное неверно.

Пусть r — длина кратчайшей M–чередующейся цепи. Через $G(M)$ обозначим граф кратчайших M–цепей, т.е. граф, состоящий из всех вершин и ребер таких, что каждое ребро и каждая вершина входят в некоторую M–цепь длины r.

Общую схему алгоритма Хопкрофта–Карпа построения наибольшего паросочетания можно описать следующим образом.

1. начать с произвольного паросочетания M в графе G;
2. построить граф $G(M)$ кратчайших M–цепей;
3. построить максимальное по включению множество $\{P_1, \dots, P_k\}$ вершинно–непересекающихся M–чередующихся цепей в $G(M)$. Увеличить паросочетание M вдоль всех цепей из построенного множества по формулам

$$M_1 = M \oplus P_1, \quad M_2 = M_1 \oplus P_2, \quad \dots, \quad M_k = M_{k-1} \oplus P_k.$$

   Объявить паросочетание M_k текущим, т.е. $M := M_k$.

$\mathcal{NB}$

Заметим, что

$$M_k = M \oplus P_1 \oplus P_2 \oplus \ldots \oplus P_k, \qquad |M_k| = |M| + k.$$

4. повторять шаги 2 и 3 до тех пор, пока в сети G существует хотя бы одна M–чередующаяся цепь. Если такой цепи не существует, то текущее паросочетание M является наибольшим (см. теорему 2).

Перейдем к формальному описанию алгоритма Хопкрофта–Карпа.

Двудольный граф $G = (X, Y, E)$ будем задавать матрицей A размера pq, где $p = |X|$, $q = |Y|$, в которой

$$A[x, y] = \begin{cases} 1, & xy \in E \\ 0, & xy \notin E. \end{cases}$$

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Паросочетание M в графе G можно описывать с помощью двух массивов $Xdouble$ длины p, и $Ydouble$ длины q, считая, что $Xdouble[x] = y$, если x сочетается с y, и $Xdouble[x] = nil$, если вершина x свободна. Аналогично определяется массив $Ydouble$.

Паросочетание M в графе G можно описывать с помощью двух массивов $Xdouble$ длины p, и $Ydouble$ длины q, считая, что $Xdouble[x] = y$, если x сочетается с y, и $Xdouble[x] = nil$, если вершина x свободна. Аналогично определяется массив $Ydouble$.

Таким образом, пустое паросочетание определяется равенствами $Xdouble[x] = Ydouble[y] = nil$ для всех $x \in X, y \in Y$.

Вспомогательный граф $G(M)$ кратчайших M–цепей несложно построить, используя поиск в ширину. Напомним, что каждая M–цепь начинается в свободной вершине $x \in X$. Поэтому во вспомогательный граф $G(M)$ следует отнести все свободные вершины $x \in X$ и начать поиск в них. Поскольку нас интересуют чередующиеся цепи, нужно различать шаги от X к Y и от Y к X.

- В первом случае переход осуществляется по светлым ребрам., а во втором — по темным.

- В первом случае переход осуществляется по светлым ребрам., а во втором — по темным.
- В первом случае, находясь в светлой вершине $x \in X$ следует отнести в $G(M)$ все вершины, смежные с x, и все ребра, инцидентные x.

- В первом случае переход осуществляется по светлым ребрам., а во втором — по темным.
- В первом случае, находясь в светлой вершине $x \in X$ следует отнести в $G(M)$ все вершины, смежные с x, и все ребра, инцидентные x.
- Во втором случае выбор однозначен: из вершины $y \in Y$ можно шагнуть только в вершину $x = Ydouble[y]$, используя ребро $xy \in M$. При этом вершину x и ребро xy следует добавить к $G(M)$.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Процесс поиска завершается либо полным построением того фронта, в котором в первый раз встретится свободная вершина $y \in Y$, либо тогда, когда s граф $G(M)$ нельзя отнести ни одной новой вершины и ни одного нового ребра, но свободных вершин $y \in Y$ достичь не удалось. Последний случай означает, что M–чередующихся цепей в графе G не существует.

Через $DX \cup DY$, где $DX \subseteq X$, $DY \subseteq Y$, будем обозначать множество вершин графа $G(M)$. Множество ребер этого графа описывается матрицей DA размера pq, где $p = |X|$, $q = |Y|$. При этом лишние строки и столбцы матрицы DA, соответствующие элементам $x \in X$, $y \in Y$, не попавшим в DX и DY, будут игнорироваться.

- Построение вспомогательного графа $G(M)$ представлено процедурой $Graph(M)$. В ней используются две очереди. В очереди Q_1 хранится последний построенный фронт, а в Q_2 накапливаются вершины нового строящегося фронта. При этом в очередной фронт распространения волны относятся лишь вершины из множества X, ибо переход от Y к X однозначен, т.е. за один шаг строим сразу два фронта.
- Все достигнутые свободные вершины $y \in Y$ заносятся в $Yfree$. $Xfree$ — множество всех свободных вершин $x \in X$.
- Через $front[v]$, $v \in X \cup Y$, обозначается номер фронта, в который попадает вершина v. При этом для всех непомещенных еще в какой–то фронт вершин (в традиционной терминологии *непомеченных*) выполняется равенство $front[v] = \infty$.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

```
procedure Graph(M)
begin
  DX := DY := ∅;
  for x ∈ X  do
    for y ∈ Y  do  DA[x, y] := 0;
  for v ∈ X ∪ Y  do  front[v] := ∞;
  Q1 := Q2 := Xfree := Yfree := nil;
  for x ∈ X  do
    if Xdouble[x] = nil  then
      begin
        Q2 ⇐ x; Xfree ⇐ x;
        DX := DX ∪ {x}; front[x] := 0;
      end;
```

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

```
repeat
  Q1 := Q2; Q2 := nil;
  while Q1 ≠ nil do
    begin
      x ⇐ Q1;
      for y ∈ Y  do
        if (A[x, y] = 1) and (front[x] < front[y])
        then
          begin
            DA[x, y] := 1;
            if front[y] = ∞ then
              begin
                DY:=DY ∪ {y}; z := Ydouble[y];
                front[y] := front[x] + 1;
                if z ≠ nil then
                  begin
                    DA[z, y] := 1; DX := DX ∪ z;
                    front[z] := front[y] + 1; Q2 ⇐ z;
                  end
```

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

```
                    else Yfree ⇐ y;
                    end
                  end
              end
          until (Yfree ≠ nil) or (Q₂ = nil);
      end.
```

Прокомментируем работу процедуры $Graph(M)$.

- В строках 3–5 инициализируется пустой граф $G(M)$.

Прокомментируем работу процедуры $Graph(M)$.

- В строках 3–5 инициализируется пустой граф $G(M)$.
- Цикл 8–12 означает, что все свободные вершины включаются в граф $G(M)$, и поиск в ширину начинается с них.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Прокомментируем работу процедуры $Graph(M)$.

- В строках 3–5 инициализируется пустой граф $G(M)$.
- Цикл 8–12 означает, что все свободные вершины включаются в граф $G(M)$, и поиск в ширину начинается с них.
- В строке 15 инициализируется последний построенный фронт.

Прокомментируем работу процедуры $Graph(M)$.

- В строках 3–5 инициализируется пустой граф $G(M)$.
- Цикл 8–12 означает, что все свободные вершины включаются в граф $G(M)$, и поиск в ширину начинается с них.
- В строке 15 инициализируется последний построенный фронт.
- В строке 16 начинается основной цикл поиска в ширину. При этом последний построенный фронт используется полностью.

Прокомментируем работу процедуры $Graph(M)$.

- В строках 3–5 инициализируется пустой граф $G(M)$.
- Цикл 8–12 означает, что все свободные вершины включаются в граф $G(M)$, и поиск в ширину начинается с них.
- В строке 15 инициализируется последний построенный фронт.
- В строке 16 начинается основной цикл поиска в ширину. При этом последний построенный фронт используется полностью.
- В цикле 19–35 анализируются все $y \in Y$, смежные с очередной вершиной $x \in X$ и удовлетворяющие условию $front[x] < front[y]$. Здесь возможны лишь два случая.

1. В первом случае $\mathit{front}[y] = \infty$, т.е. вершина y ранее не посещалась.
2. Во втором — $\mathit{front}[y] = \mathit{front}[x] + 1$ те вершина y уже помечена, но из вершины того же последнего построенного фронта. В этом случае в граф $G(M)$ нужно лишь добавить одно ребро xy (строка 23).

- Выполнение условия в строке 28 означает, что вершина y насыщена в паросочетании M. В этом случае вершина $z = Ydouble[y]$ и ребро zy включаются в граф $G(M)$, причем z включается и в новый фронт (строка 31).
- В противном случае вершина y является свободной и включается в $Yfree$.
- В строке 37 даны условия прекращения поиска. Случай $Yfree \neq \varnothing$ свидетельствует, что найдется хотя бы одна свободная вершина $y \in Y$ и, следовательно, в графе существует M–чередующаяся цепь. Поиск на этом прекращается и в граф $G(M$ (благодаря свойствам поиска в ширину) попадут все кратчайшие M–цепи.

Разберем теперь метод построения максимального по включению множества вершинно–непересекающихся M–чередующихся цепей и увеличения текущего паросочетания M с помощью построенного множества.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Разберем теперь метод построения максимального по включению множества вершинно–непересекающихся M–чередующихся цепей и увеличения текущего паросочетания M с помощью построенного множества.

- Выберем произвольную вершину $x \in Xfree$.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Разберем теперь метод построения максимального по включению множества вершинно–непересекающихся M–чередующихся цепей и увеличения текущего паросочетания M с помощью построенного множества.

- Выберем произвольную вершину $x \in Xfree$.
- Проведем поиск в глубину с корнем в x, помечая вершины по тем же правилам, которые использовались при построении графа $G(M)$, и продвигаясь из вершины $x \in X$ по светлым ребрам, а из вершин $y \in Y$ — по темным.

Разберем теперь метод построения максимального по включению множества вершинно–непересекающихся M–чередующихся цепей и увеличения текущего паросочетания M с помощью построенного множества.

- Выберем произвольную вершину $x \in Xfree$.
- Проведем поиск в глубину с корнем в x, помечая вершины по тем же правилам, которые использовались при построении графа $G(M)$, и продвигаясь из вершины $x \in X$ по светлым ребрам, а из вершин $y \in Y$ — по темным.
- Помеченные в ходе поиска вершины помещаются в стек S.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Поиск в глубину из вершины x завершается либо по достижению свободной вершины $y \in Yfree$ (в этом случае существует искомая цепь, начинающаяся в x и заканчивающаяся в y), либо тогда, когда $S = nil$. Пустота стека S означает, что в $G(M)$ не существует M–чередующейся цепи, начинающейся в вершине x.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Поиск в глубину из вершины x завершается либо по достижению свободной вершины $y \in Yfree$ (в этом случае существует искомая цепь, начинающаяся в x и заканчивающаяся в y), либо тогда, когда $S = nil$. Пустота стека S означает, что в $G(M)$ не существует M–чередующейся цепи, начинающейся в вершине x.
- Если искомая цепь существует, то после завершения поискав S первая и последняя вершины свободны относительно M, а все промежуточные вершины насыщены в M. Отметим, что вершины чередуются: сначала из DX, потом из DY, и т.д.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Увеличить паросочетание M с помощью найденной цепи очень просто.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Увеличить паросочетание M с помощью найденной цепи очень просто.

- Считываем попарно вершины их стека S (первой считывается $y \in Y$) и корректируем значения массивов $Xdouble$ и $Ydouble$. При этом у первой и последней из считанных вершин y и x значения $Ydouble[y]$ и $Xdouble[x]$ равные nil, получат значения соответствующих вершин соседних вершин из S, а у всех прочих произойдет смена значений массивов $Ydouble$ и $Xdouble$ с одних вершин на другие.
- Считывая вершины из S, мы одновременно удаляем их из графа $G(M)$. В результате при построении следующей M–чередующейся цепи вновь найденная цепь не пересекается с прежде построенными цепями и вершинами.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Процесс построения цепей ведется до полного исчерпания одного из списков *Xfree* или *Yfree*, чем и обеспечивается максимальность по включению построенного множества M–чередующ цепей.

Детали описанного процесса представлены в процедуре $Increase(M)$. В ней без формального описания используется функция $Choice(x)$, которая возвращает непомеченную в ходе поиска вершину $y \in DY$, смежную с x, если такая существует, и $Choice(x) = nil$, если все вершины из DY, смежные с x, уже помечены.

Переменная *indication* сигнализирует о том, достигнута ли свободная вершина $y \in DY$, т.е. $indicatio = 0$, если свободная вершина еще не достигнута, и $indication = 1$, если достигнута.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

```
procedure Increase(M);
begin
  while (Xfree ≠ ∅) and (Yfree ≠ ∅) do
    begin
      x ⇐ Xfree; Xfree := Xfree \ {x};
      S := nil; S ⇐ x; indication := 0;
      while (S ≠ nil) and (indication = 0) do
        begin
          x ⇐ S; y := Choice(x);
          if y ≠ nil then
            begin
              S ⇐ y; z := Ydouble[y];
              if z ≠ nil  then S ⇐ z;
            else
              begin
                indication := 1; Yfree := Yfree \ {y};
```

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

```
                end
            end
        else
            begin
                x ⇐ S; DX := DX \ {x};
                if S ≠ ∅ then
                    begin
                        y ⇐ S; DY := DY \ {y};
                    end
                end
            end;
            if indication = 1 then
                while S ≠ nil  do
                    begin
                        y ⇐ S; DY := DY \ {y};
                        x ⇐ S; DX := DX \ {x};
                        Xdouble[x] := y; Ydouble[y] := x;
                    end
        end
end;
```

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Структура этой процедуры следующая.

- Основной цикл 3–35 ведется до полного опустошения одного из списокв свободных в X или в Y вершин.

Структура этой процедуры следующая.

- Основной цикл 3–35 ведется до полного опустошения одного из списокв свободных в X или в Y вершин.
- Каждый проход этого цикла начинается с выбора произвольного элемента x из $Xfree$ и последнего поиска в глубину с корнем в x (строки 6–27).

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Структура этой процедуры следующая.

- Основной цикл 3–35 ведется до полного опустошения одного из списокв свободных в X или в Y вершин.
- Каждый проход этого цикла начинается с выбора произвольного элемента x из $Xfree$ и последнего поиска в глубину с корнем в x (строки 6–27).
  Важно отметить, что если текущая вершина x имеет непомеченную насыщенную смежную с ней вершину y, то в стек S помещается сразу две вершины: сначала y, затем $z = Ydouble[y]$ (строки 9–13). Можно сказать, что промежуточные вершины в ходе поиска в S помещаются парами.

Именно это обстоятельство объясняет действия в строках 20–25.

Именно это обстоятельство объясняет действия в строках 20–25.

- Если вершины помещали в S парами, то и удалять их оттуда нужно парами (кроме самой первой).

Именно это обстоятельство объясняет действия в строках 20–25.

- Если вершины помещали в S парами, то и удалять их оттуда нужно парами (кроме самой первой).
- В строках 20–25 сначала удаляется из S и графа вершина x, все соседи которой уже посещались, а затем, если x не первой попала попала в S, удаляется та вершина y, для которой $Ydouble[y] = x$ (также верно равенство $Xdouble[x] = y$).

Именно это обстоятельство объясняет действия в строках 20–25.

- Если вершины помещали в S парами, то и удалять их оттуда нужно парами (кроме самой первой).
- В строках 20–25 сначала удаляется из S и графа вершина x, все соседи которой уже посещались, а затем, если x не первой попала попала в S, удаляется та вершина y, для которой $Ydouble[y] = x$ (также верно равенство $Xdouble[x] = y$).
- В сроке 28 анализируется, каким именно исходом закончился поиск в глубину.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

- Если поиск удачный, т.е. *indication* $= 1$, то в цикле 29–34 увеличивается паросочетание M и удаляются из графа $G(M)$ все вершины найденной M–чередующейся цепи.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Соберем описанные процедуры в один алгоритм построения наибольшего паросочетания в двудольном графе.

Алгоритм Хопкрофта–Карпа

Вход: двудольный граф $G=(X,E,Y)$, заданный матрицей $A[1\dots p, 1\dots q]$, где $p=|X|$, $q=|Y|$.

Выход: наибольшее паросочетание в графе G, задаваемое массивами $Xdouble[1\dots p]$ и $Ydouble[1\dots q]$.

```
begin
   for x ∈ X  do  Xdouble[x] := nil;
   for y ∈ X  do  Ydouble[y] := nil;
   repeat
      Graph(M);
      if Yfree ≠ ∅ then Increase(M);
   until Yfree = ∅;
end.
```

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Дадим небольшой комментарий к этому алгоритму.

- В сроках 2–3 строится пустое паросочетание M.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Дадим небольшой комментарий к этому алгоритму.

- В сроках 2–3 строится пустое паросочетание M.
- После построения вспомогательного графа процедурой $Graph(M$ анализируется (условие в сроке 6), содержит ли граф $G(M)$ свободные в M вершины $y \in Y$.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Дадим небольшой комментарий к этому алгоритму.

- В сроках 2–3 строится пустое паросочетание M.
- После построения вспомогательного графа процедурой *Graph*(M анализируется (условие в сроке 6), содержит ли граф $G(M)$ свободные в M вершины $y \in Y$.
- Ксли таковые имеются, то процедура *Increase*(M) увеличивает текущее паросочетание и (условие в сроке 7) фаза повторяется.

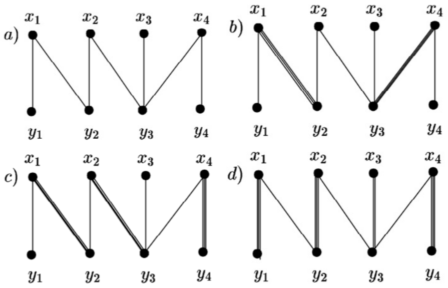
Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Дадим небольшой комментарий к этому алгоритму.

- В сроках 2–3 строится пустое паросочетание M.
- После построения вспомогательного графа процедурой $Graph(M$ анализируется (условие в сроке 6), содержит ли граф $G(M)$ свободные в M вершины $y \in Y$.
- Ксли таковые имеются, то процедура $Increase(M)$ увеличивает текущее паросочетание и (условие в сроке 7) фаза повторяется.
- Если же список достигнутых процедурой $Graph(M)$ свободных вершин $y \in Y$ пуст, то алгоритм завершает работу и по теореме Бержа текущее паросочетание является наибольшим.

Разберем работу алгоритма на простом примере

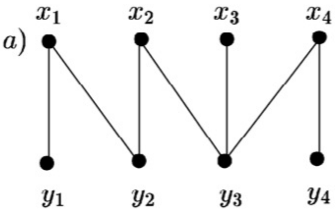


На рисунке a) содержится исходный граф, а на b), c) и d) — паросочетания, полученные после каждой из трех фаз. Фаза — построение вспомогательного графа и последующее увеличение текущего паросочетания (один проход цикла 4–7).

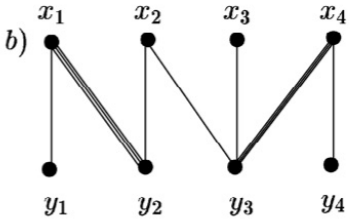
Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

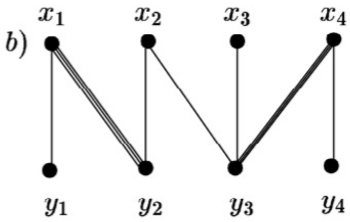
Ясно, что вспомогательный граф $G(M)$, построенный в первой фазе, совпадает с исходным, ибо все вершины $y \in Y$ свободны. и неизолированы в G. Следовательно, выполняются равенства $Xfree = X$, $Yfree = Y$.



Предположим, что поиск в глубину в процедуре *Increase*(M) начинается с вершины x_1. Теперь все зависит от процедуры *Choice*(x_1). Пусть *Choice*(x_1) $= y_2$ (случай *Choice*(x_1) $= y_1$ не так интересен). Тогда цепь x_1, x_1y_2, y_2 является M–чередующейся, и в M будет добавлено ребро x_1y_2.



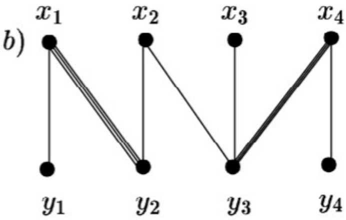
Пусть следующей вершиной, взятой из списка *Xfree*, была вершина x_4, и $Choice(x_4) = y_3$ (действуем максимально злоумышленно). Тогда в M добавится ребро $x_4 y_3$. Первая фаза на этом закончится.



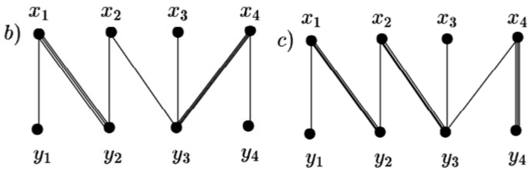
Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Интересно отметить, что вспомогательный граф $G(M)$, построенный на второй фазе, в точности совпадает с исходным. Действительно, поиск в ширину начнется из вершин x_2 и x_3. Полный просмотр первого фронта приведет к тому, что в $G(M)$ попадут вершины y_2 и y_3, а через них — вершины x_1 и x_4. Затем из вершин x_1 и x_4 будут достигнуты вершины y_1 и y_4. Конечно же во вспомогательный граф попадут и все ребра исходного графа.



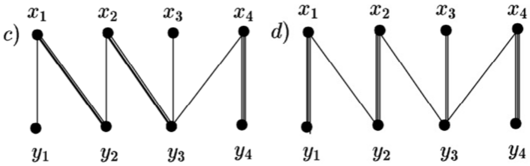
Предположим, что поиск в глубину начнется из вершины x_2 и $Choice(x_2) = y_3$. Тогда будет найдена M–чередующаяся цепь с последовательностью вершин x_2, y_3, x_4, y_4. Паросочетание M увеличится следующим образом: будет удалено ребро x_4y_3 и добавлены ребра x_2y_3 и x_4y_4. После удаления вершин x_2, y_3, x_4, y_4 ни одной M–чередующейся цепи в графе $G(M)$ уже не существует. Результат второй фазы изображен на рисунке c).



Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Наконец, в третьей фазе поиск в ширину начнется из единственной свободной вершины x_3. Вспомогательный граф представляет собой M–чередующуюся цепь, включающую вершины $x_3, y_3, x_2, y_2, x_1, y_1$. Увеличение паросочетания M вдоль этой цепи, приводит к паросочетанию, изображенному на рисунке d), которое является наибольшим паросочетанием в этом графе.



Оценка сложности алгоритма Хопкрофта–Карпа

Лемма 1

Пусть M и N — два паросочетания в двудольном графе $G = (X, Y, E)$ и

$$|M| = r < s = |N|.$$

Тогда симметрическая разность $M \oplus N$ содержит не менее $s - r$ непересекающихся по вершинам M–чередующихся цепей.

Доказательство. Рассмотрим граф $\tilde{G} = (X, Y, M \oplus N)$ и обозначим через $G_1, \ldots, G_k$ компоненты связности этого графа.

Доказательство. Рассмотрим граф $\tilde{G} = (X, Y, M \oplus N)$ и обозначим через $G_1, \ldots, G_k$ компоненты связности этого графа.

- Поскольку M и N — паросочетания, каждая вершина графа $\tilde{G}$ инцидентна не более чем одному ребру из $M \setminus N$ и не более чем одному ребру из $N \setminus M$.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Доказательство. Рассмотрим граф $\tilde{G} = (X, Y, M \oplus N)$ и обозначим через $G_1, \ldots, G_k$ компоненты связности этого графа.

- Поскольку M и N — паросочетания, каждая вершина графа $\tilde{G}$ инцидентна не более чем одному ребру из $M \setminus N$ и не более чем одному ребру из $N \setminus M$.
- Следовательно, каждая компонента связностей имеет один из следующих трех видов:
  1. изолированная вершина;
  2. цикл четной длины с ребрами попеременно из $M \setminus N$ и $N \setminus M$;
  3. цепь с ребрами попеременно из $M \setminus N$ и $N \setminus M$.

Обозначим через E_i множество ребер компоненты G_i. Пусть

$$d_i = |E_i \cap N| - |E_i \cap M|.$$

Обозначим через E_i множество ребер компоненты G_i. Пусть

$$d_i = |E_i \cap N| - |E_i \cap M|.$$

В тех случаях, когда компонента G_i имеет тип 1) или 2), получаем $d_i = 0$.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Обозначим через E_i множество ребер компоненты G_i. Пусть

$$d_i = |E_i \cap N| - |E_i \cap M|.$$

В тех случаях, когда компонента G_i имеет тип 1) или 2), получаем $d_i = 0$.

В случае 3) либо $d_i = -1$, либо $d_i = 1$, либо $d_i = 0$. Причем, случай $d_i = 1$ возможен только тогда, когда цепь начинается ребром из N и заканчивается ребром из N, т.е. является M–чередующейся.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Обозначим через E_i множество ребер компоненты G_i. Пусть

$$d_i = |E_i \cap N| - |E_i \cap M|.$$

В тех случаях, когда компонента G_i имеет тип 1) или 2), получаем $d_i = 0$.

В случае 3) либо $d_i = -1$, либо $d_i = 1$, либо $d_i = 0$. Причем, случай $d_i = 1$ возможен только тогда, когда цепь начинается ребром из N и заканчивается ребром из N, т.е. является M–чередующейся.

Остается показать, что $d_i = 1$ для не менее чем $s - r$ индексов.

$$\sum_{i=1}^{k} d_i =$$

$$= \sum_{i=1}^{k} (|E_i \cap N| - |E_i \cap M|) = |N \setminus M| - |M \setminus N| = |N| - |M| =$$

$$= s - r.$$

Из этого уравнения следует, что в графе имеется не менее $s - r$ различных M–чередующихся цепей. ■

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Лемма 2

Пусть M — паросочетание в двудольном графе G, и $M = r < s$, где s — мощность наибольшего паросочетния в G. Тогда существует M–чередующаяся цепь длины не превосходящей $\dfrac{2r}{s-r} + 1$.

Доказательство. Пусть N — наибольшее паросочетание в G. Тогда $|N| = s$, и по лемме 1 множество $M \oplus N$ содержит не менее $s - r$ непересекающихся по вершинам (а, следовательно, и по ребрам) M–чередующихся цепей.

Доказательство. Пусть N — наибольшее паросочетание в G. Тогда $|N| = s$, и по лемме 1 множество $M \oplus N$ содержит не менее $s - r$ непересекающихся по вершинам (а, следовательно, и по ребрам) M–чередующихся цепей.

Пусть кратчайшая из них имеет длину $2k + 1$ (длина M–чередующейся цепи нечетна). Тогда ровно k ребер этой цепи принадлежат M.

Доказательство. Пусть N — наибольшее паросочетание в G. Тогда $|N| = s$, и по лемме 1 множество $M \oplus N$ содержит не менее $s - r$ непересекающихся по вершинам (а, следовательно, и по ребрам) M–чередующихся цепей.

Пусть кратчайшая из них имеет длину $2k + 1$ (длина M–чередующейся цепи нечетна). Тогда ровно k ребер этой цепи принадлежат M.

Следовательно, $(s - r)k \leqslant r$, т.к. каждая M–чередующаяся цепь содержит не менее k ребер из M.

Доказательство. Пусть N — наибольшее паросочетание в G. Тогда $|N| = s$, и по лемме 1 множество $M \oplus N$ содержит не менее $s - r$ непересекающихся по вершинам (а, следовательно, и по ребрам) M–чередующихся цепей.

Пусть кратчайшая из них имеет длину $2k + 1$ (длина M–чередующейся цепи нечетна). Тогда ровно k ребер этой цепи принадлежат M.

Следовательно, $(s - r)k \leqslant r$, т.к. каждая M–чередующаяся цепь содержит не менее k ребер из M.

Из этого соотношения вытекает требуемое неравенство. ■

Следующий результат является ключевым.

Следующий результат является ключевым.

Лемма 3

Пусть P и $\tilde{P}$ — чередующиеся цепи, построенные в разных фазах алгоритма Хопкрофта–Карпа, причем цепь P построена раньше, чем цепь $\tilde{P}$. Тогда

$$|P| < |\tilde{P}|.$$

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Достаточно рассмотреть фазы, которые следуют друг за другом.

Достаточно рассмотреть фазы, которые следуют друг за другом.

Пусть M и N — паросочетания, которые были построены перед началом соответствующих фаз. Пусть $\{P_1, \ldots, P_k\}$ — максимальное по включению множество вершинно непересекающихся кратчайших M–чередующихся цепей, построенное алгоритмом. r — длина каждой из этих цепей.

Достаточно рассмотреть фазы, которые следуют друг за другом.

Пусть M и N — паросочетания, которые были построены перед началом соответствующих фаз. Пусть $\{P_1, \ldots, P_k\}$ — максимальное по включению множество вершинно непересекающихся кратчайших M–чередующихся цепей, построенное алгоритмом. r — длина каждой из этих цепей.

Тогда $P = P_i$ для некоторого i и $N = M \oplus P_1 \oplus \ldots \oplus P_k$.

Положим $L = N \oplus \tilde{P}$. Поскольку цепи P_i не пересекаются по вершинам (а поэтому и не имеют общих ребер), имеем:

$$
\begin{aligned}
M \oplus L &= \\
&= M \oplus M \oplus P_1 \oplus P_2 \oplus \ldots \oplus P_k \oplus \tilde{P} = \\
&= \tilde{P} \oplus (P_1 \cup P_2 \cup \ldots \cup P_k)\,.
\end{aligned}
$$

Положим $L = N \oplus \tilde{P}$. Поскольку цепи P_i не пересекаются по вершинам (а поэтому и не имеют общих ребер), имеем:

$$M \oplus L = \\ = M \oplus M \oplus P_1 \oplus P_2 \oplus \ldots \oplus P_k \oplus \tilde{P} = \\ = \tilde{P} \oplus (P_1 \cup P_2 \cup \ldots \cup P_k)\,.$$

Отсюда

$$|M \oplus L| = |\tilde{P}| + kr - 2|\tilde{P} \cap Q|,$$

где $Q = P_1 \cup P_2 \cup \ldots \cup P_k$.

С другой стороны, т.к. $|L| = |M| + k + 1$, множество $M \oplus L$ содержит не менее $k+1$ реберно непересекающихся M–чередующихся цепей (лемма 1). Следовательно,

$$|\tilde{P}| + kr - 2|\tilde{P} \cap Q| \geqslant (k+1)r.$$

С другой стороны, т.к. $|L| = |M| + k + 1$, множество $M \oplus L$ содержит не менее $k + 1$ реберно непересекающихся M–чередующихся цепей (лемма 1). Следовательно,

$$|\tilde{P}| + kr - 2|\tilde{P} \cap Q| \geqslant (k+1)r.$$

Отсюда

$$|\tilde{P}| \geqslant r + 2|\tilde{P} \cap Q|.$$

Т.к. в алгоритме Хопкрофта–Карпа выбиралось максимальное по включению множество вершинно непересекающихся цепей, цепь $\tilde{P}$ имеет общую вершину v хотя бы с одной из цепей $P_1, \ldots, P_k$.

Т.к. в алгоритме Хопкрофта–Карпа выбиралось максимальное по включению множество вершинно непересекающихся цепей, цепь $\tilde{P}$ имеет общую вершину v хотя бы с одной из цепей $P_1, \ldots, P_k$.

Поскольку после каждого увеличения относительно какой-либо чередующейся цепи все вершины цепи становятся насыщенными, эта общая вершина v является насыщенной, т.е. не первой и не последней в цепи $\tilde{P}$.

Т.к. в алгоритме Хопкрофта–Карпа выбиралось максимальное по включению множество вершинно непересекающихся цепей, цепь $\tilde{P}$ имеет общую вершину v хотя бы с одной из цепей $P_1, \ldots, P_k$.

Поскольку после каждого увеличения относительно какой-либо чередующейся цепи все вершины цепи становятся насыщенными, эта общая вершина v является насыщенной, т.е. не первой и не последней в цепи $\tilde{P}$.

Тогда темное относительно N ребро, инцидентное вершине v, входит в обе цепи, т.е. $\tilde{P} \cap Q \neq \varnothing$.

Т.к. в алгоритме Хопкрофта–Карпа выбиралось максимальное по включению множество вершинно непересекающихся цепей, цепь $\tilde{P}$ имеет общую вершину v хотя бы с одной из цепей $P_1, \ldots, P_k$.

Поскольку после каждого увеличения относительно какой-либо чередующейся цепи все вершины цепи становятся насыщенными, эта общая вершина v является насыщенной, т.е. не первой и не последней в цепи $\tilde{P}$.

Тогда темное относительно N ребро, инцидентное вершине v, входит в обе цепи, т.е. $\tilde{P} \cap Q \neq \varnothing$.

Отсюда $|\tilde{P}| > r$. ∎

Теорема 4

Число фаз алгоритма Хопкрофта–Карпа не превышает $2\lfloor s \rfloor + 1$, *где* s *— мощность наибольшего паросочетания в данном графе.*

Доказательство. Пусть $M_0 = \varnothing$, $M_1, \ldots, M_s$ — все паросочетания, $P_0, P_1, \ldots, P_{s-1}$ — все чередующиеся цепи, последовательно построенные алгоритмом, и

$$M_i = M_{i-1} \oplus P_{i-1}, \quad i = 1, 2, \ldots, s.$$

Каждая цепь P_i — чередующаяся для некоторого паросочетания $M_j (j \leqslant i)$, которое было построено перед началом очередной фазы.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Каждая цепь P_i — чередующаяся для некоторого паросочетания $M_j (j \leqslant i)$, которое было построено перед началом очередной фазы.

Поскольку внутри каждой фазы цепи не пересекаются по вершинам, цепь P_i является не только M_j–чередующейся, но также и M_k–чередующейся для всех k таких, что $j \leqslant k \leqslant i$.

Каждая цепь P_i — чередующаяся для некоторого паросочетания $M_j (j \leqslant i)$, которое было построено перед началом очередной фазы.

Поскольку внутри каждой фазы цепи не пересекаются по вершинам, цепь P_i является не только M_j–чередующейся, но также и M_k–чередующейся для всех k таких, что $j \leqslant k \leqslant i$.

Еще раз отметим, что цепи, построенные в одной и той же фазе, имеют одинаковую длину, а в разных — разную. Таким образом, число фаз алгоритма равно количеству различных чисел в последовательности

$$|P_0|, |P_1|, \ldots, |P_{s-1}|.$$

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Пусть $r = \lfloor s - \sqrt{s} \rfloor$. Тогда $|M_r| = r < s$.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Пусть $r = \lfloor s - \sqrt{s} \rfloor$. Тогда $|M_r| = r < s$.

Используя лемму 2 и несложные арифметические преобразования получаем цепочку неравенств:

$$|P_r| \leqslant \frac{2r}{s-r} + 1 = \frac{2s}{s-r} - 1 = \frac{2s}{s = \lfloor s - \sqrt{s} \rfloor} - 1 \leqslant$$

$$\leqslant \frac{2s}{\sqrt{s}} - 1 = s\sqrt{s} - 1 \leqslant 2\lfloor \sqrt{s} \rfloor + 1.$$

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Поскольку длина каждой цепи нечетна, последовательность чисел $|P_0|, \ldots, |P_r|$ содержит не более $\lfloor\sqrt{s}\rfloor + 1$ различных чисел. Последовательность $|P_{r+1}| \ldots, |P_{s-1}|$ может содержать не более $(s-1)-r$ других чисел.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Поскольку длина каждой цепи нечетна, последовательность чисел $|P_0|, \dots, |P_r|$ содержит не более $\lfloor\sqrt{s}\rfloor + 1$ различных чисел. Последовательность $|P_{r+1}| \dots, |P_{s-1}|$ может содержать не более $(s-1)-r$ других чисел.

Кроме того,

$$(s-1) - r = (s-1) - \lfloor s - \sqrt{s} \rfloor \leqslant (s-1) - \big((s - \sqrt{s}) - 1\big) = \sqrt{s}.$$

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Поскольку длина каждой цепи нечетна, последовательность чисел $|P_0|, \ldots, |P_r|$ содержит не более $\lfloor\sqrt{s}\rfloor + 1$ различных чисел. Последовательность $|P_{r+1}| \ldots, |P_{s-1}|$ может содержать не более $(s-1)-r$ других чисел.

Кроме того,

$$(s-1) - r = (s-1) - \lfloor s - \sqrt{s}\rfloor \leqslant (s-1) - \big((s-\sqrt{s}) - 1\big) = \sqrt{s}.$$

Окончательно получаем, что в последовательности чисел $|P_0|, \ldots, |P_{s-1}|$ имеется не более $2\lfloor\sqrt{s}\rfloor + 1$ различных нечетных чисел. ■

Теперь мы можем оценить вычислительную сложность алгоритма Хопкрофта–Карпа. Для двудольного графа $G = (X, Y, E)$ положим

$$n = \max\{|X|, |Y|\}.$$

Ясно, что мощность наибльшего паросочетания не превосходит n.

Паросочетания в двудольных графах

Задача о наибольшем паросочетании. Алгоритм Хопкрофта–Карпа

Теорема 5

Алгоритм Хопкрофта–Карпа имеет сложность $O(n^{\frac{5}{2}})$.

Доказательство. По теореме 4 число фаз имеет порядок $\sqrt{n}$. Остается оценить сложность выполнения каждой фазы.

Доказательство. По теореме 4 число фаз имеет порядок $\sqrt{n}$. Остается оценить сложность выполнения каждой фазы.

В процедуре $Graph(M)$ просматривается каждое ребро не более одного раза. Просмотр ребра означает просмотр соответствующего элемента матрицы смежности A, т.е. сложность этой процедуры $O(n^2)$.

Доказательство. По теореме 4 число фаз имеет порядок $\sqrt{n}$. Остается оценить сложность выполнения каждой фазы.

В процедуре $Graph(M)$ просматривается каждое ребро не более одного раза. Просмотр ребра означает просмотр соответствующего элемента матрицы смежности A, т.е. сложность этой процедуры $O(n^2)$.

Поиск в глубину, выполняемый в процедуре $Increase(M)$, имеет сложность $O(p+q)$, поскольку использованные ребра и вершины удаляются из графа. Следовательно, сложность этой процедуры равна $O(n^2)$.

Отсюда вытекает, что сложность всего алгоритма Хопкрофта–Карпа равна

$$O(\sqrt{n}) \times O(n^2) = O(n^{\frac{5}{2}}).$$

■