Алгоритмы на графах

Модуль 3. Потоки в сетях и паросочетания.

Лекция 14.

Паросочетания в графах общего вида.

Адигеев Михаил Георгиевич

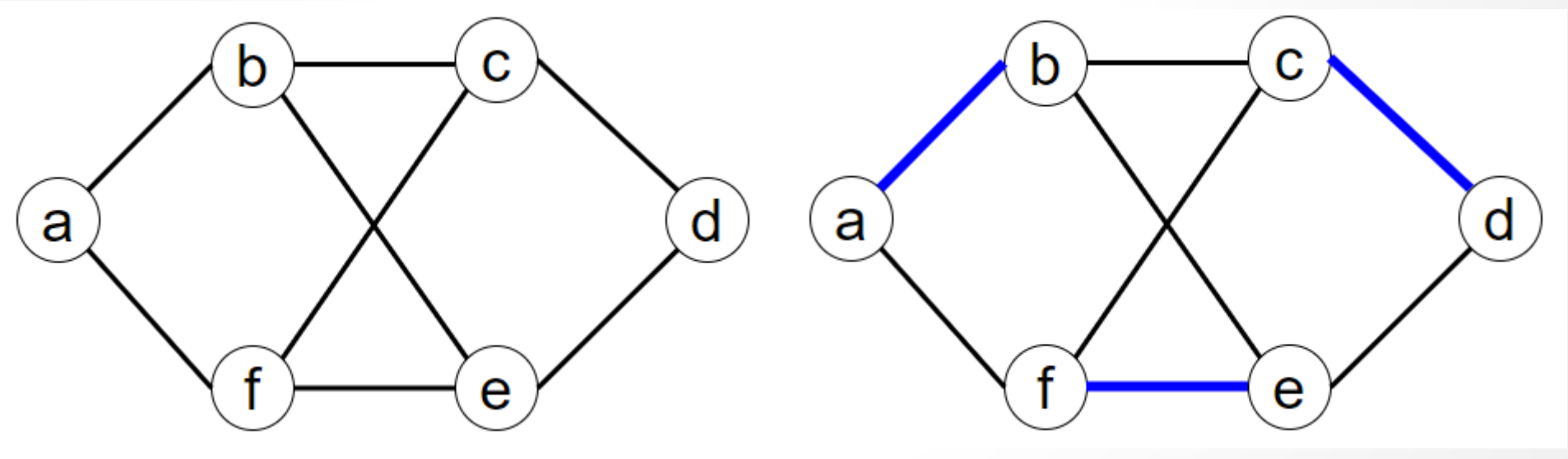
2022

План лекции

1. Паросочетания в графах общего вида.
   - ✓ Вспоминаем определение.
   - ✓ Чередующиеся цепи.
   - ✓ Увеличивающая цепь.
2. Алгоритм нахождения максимального паросочетания.
   - ✓ Поиск увеличивающей цепи.
   - ✓ Чередующееся дерево, увеличивающее дерево.
   - ✓ Цветки. Срезание цветка. Теорема о корректности операции срезания цветка.
   - ✓ Оценка сложности алгоритма.

Паросочетания

Определение. ***Паросочетанием*** на графе $G(V, E)$ называется подмножество рёбер $M \subseteq E$, в котором все рёбра попарно не смежны (не имеют общих вершин).

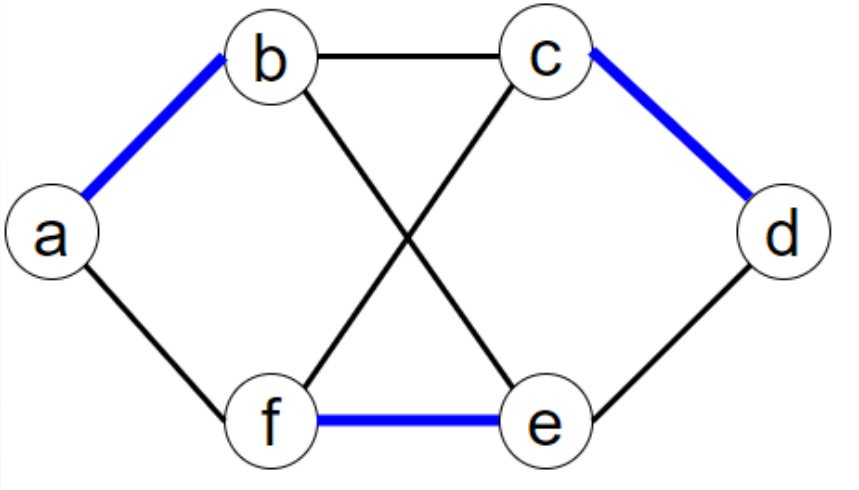


Паросочетания

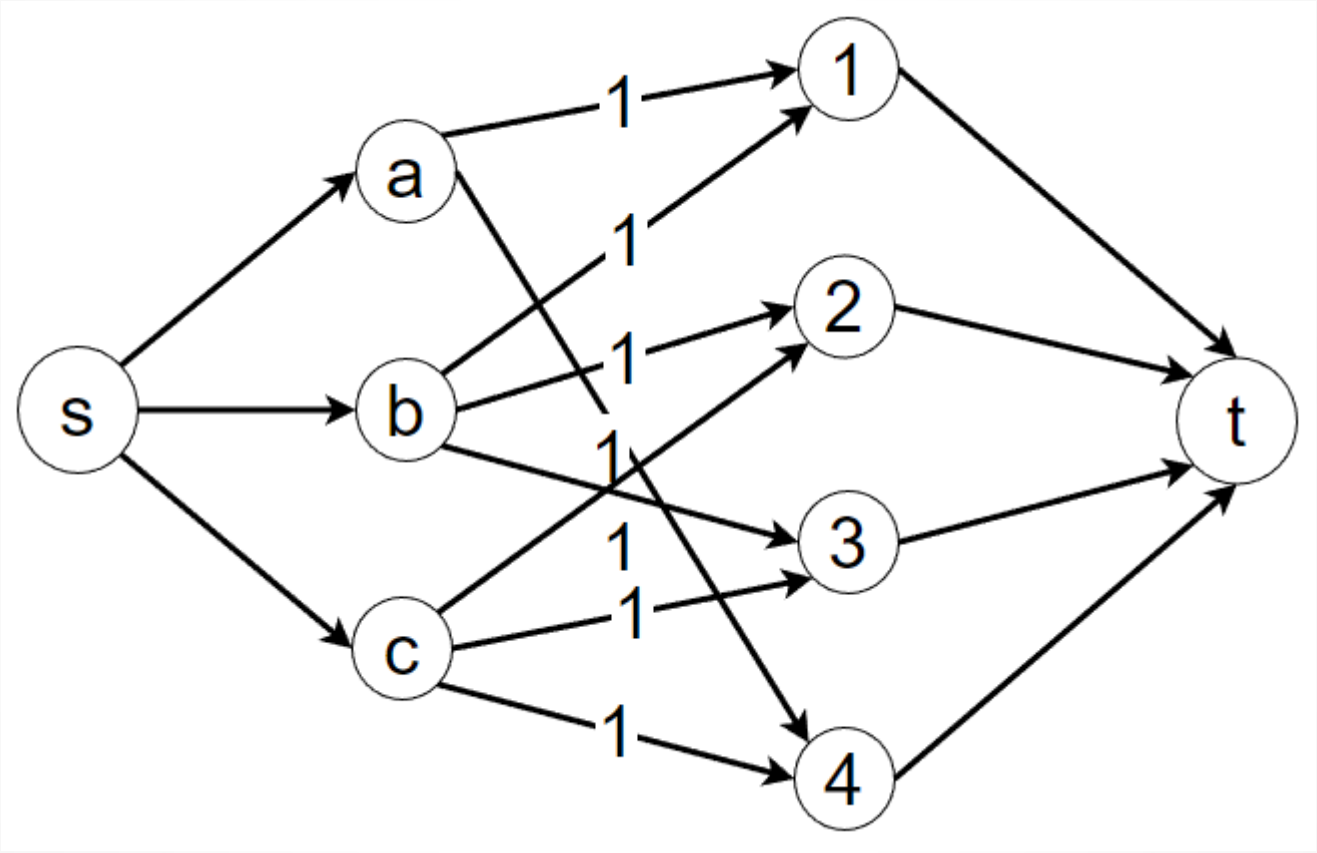
Определение. ***Совершенным паросочетанием*** называется паросочетание, которое покрывает все вершины графа, то есть каждая вершина инцидентна ребру паросочетания.

Определение. Паросочетание называется ***наибольшим***, если на данном графе не существует паросочетания с бОльшим количеством рёбер.

Очевидно, что совершенное паросочетание является наибольшим, но обратное, вообще говоря, не верно.

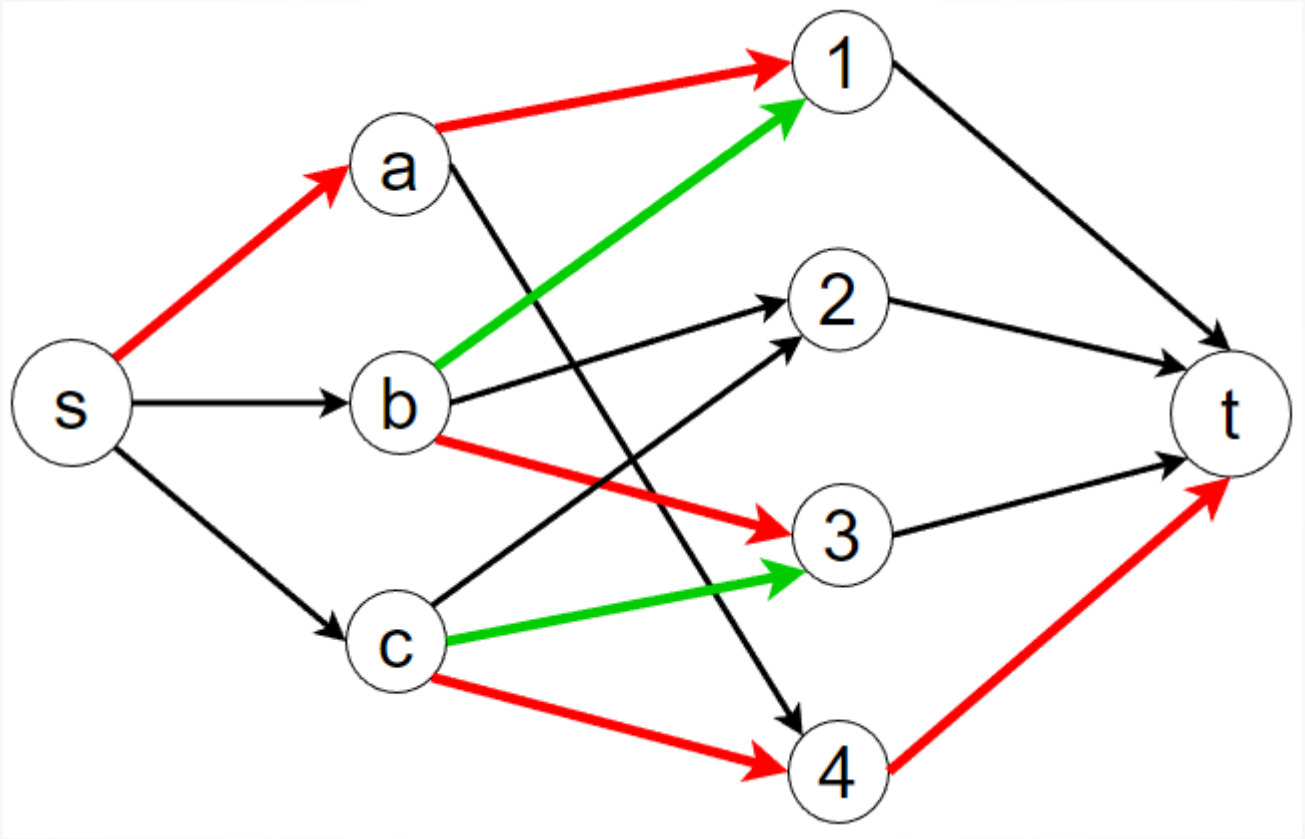


Паросочетания



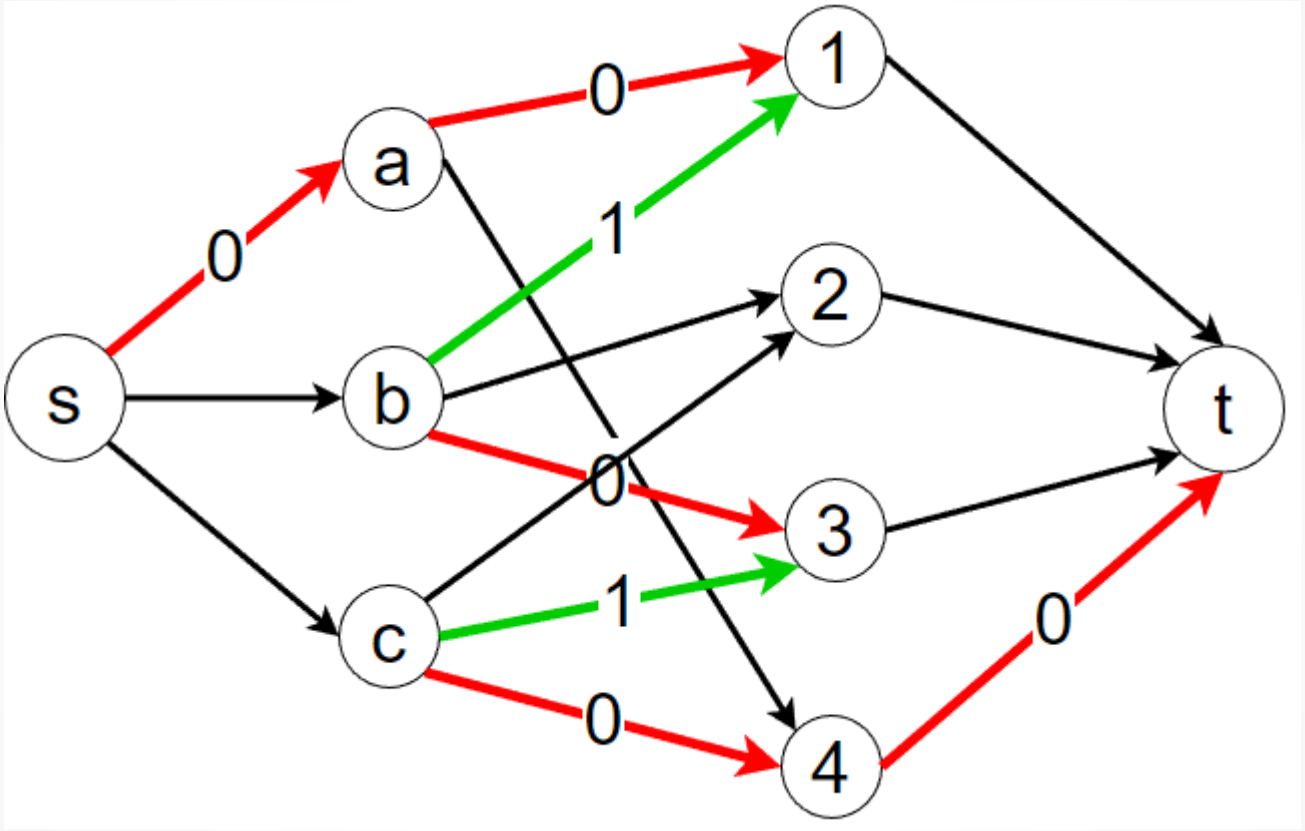
Наибольшее паросочетание на двудольном графе можно найти с помощью алгоритма нахождения максимального потока.

Паросочетания



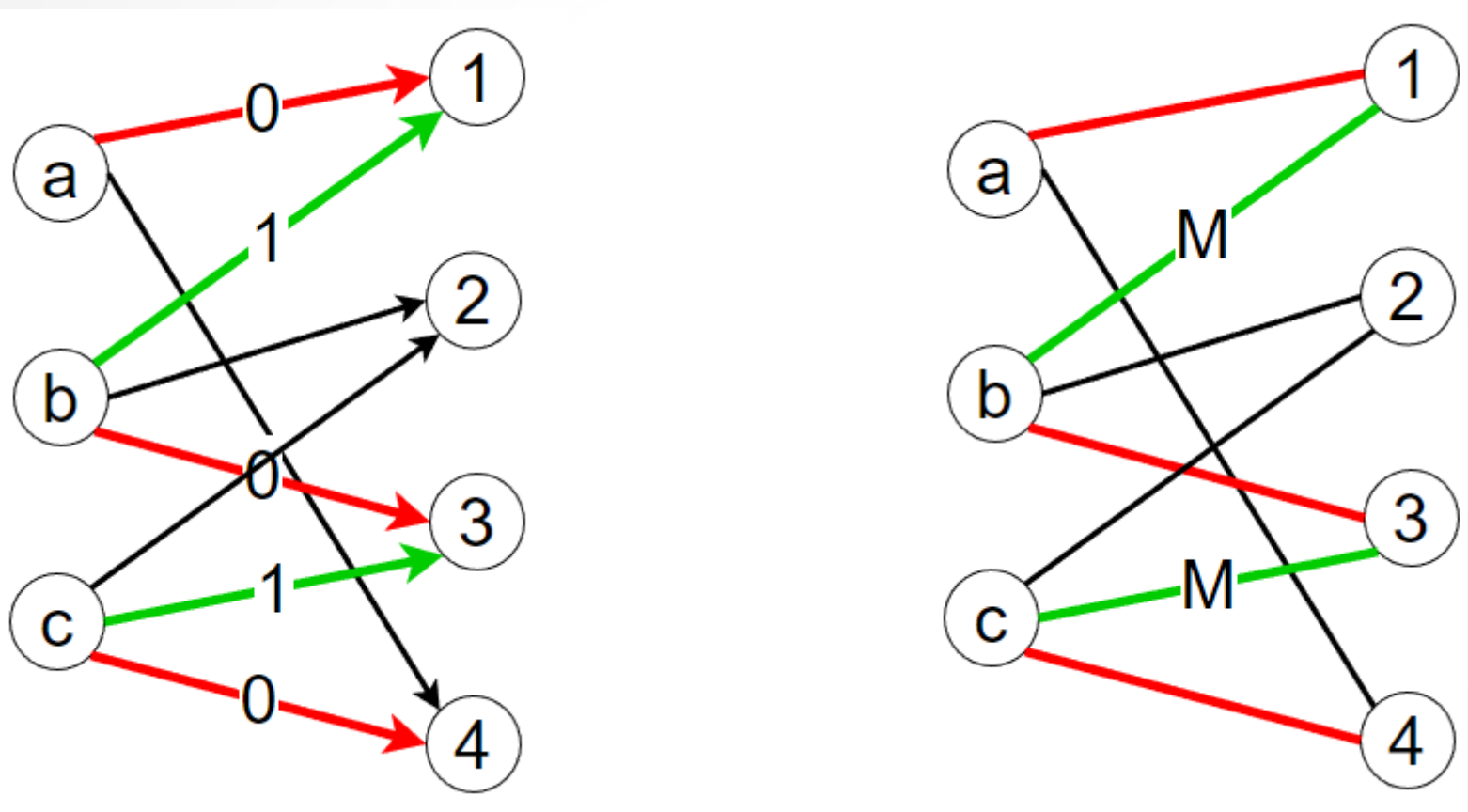
Для нахождения максимального потока мы находим увеличивающую цепь (например, как на рисунке).

Паросочетания



Для нахождения максимального потока мы находим увеличивающую цепь (например, как на рисунке).

Паросочетания



Увеличивающая цепь содержит попеременно рёбра, принадлежащие и не принадлежащие паросочетанию.

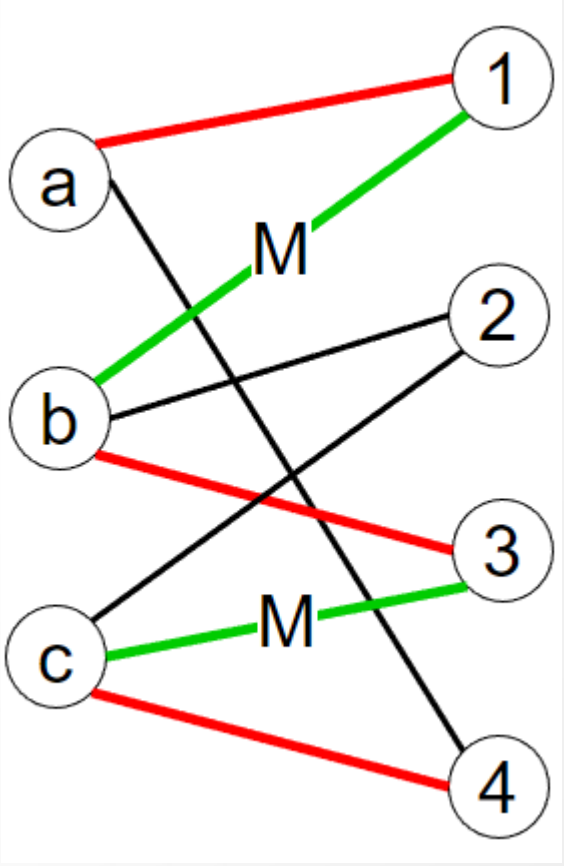
Паросочетания

Пусть M --- паросочетание на графе $G(V, E)$.

Определение. ***Чередующейся цепью*** для M называется цепь, в которой чередуются рёбра, принадлежащие и не принадлежащие M.

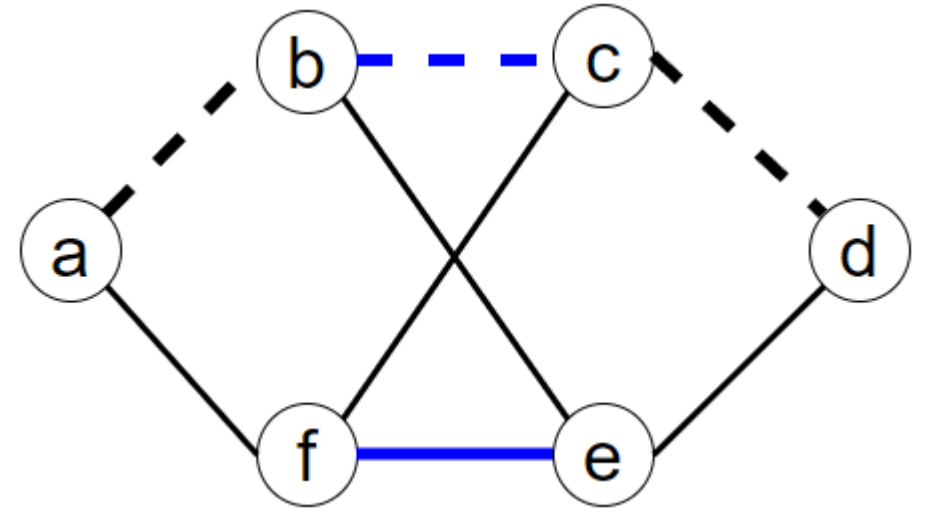
Определение. Вершину v будем называть ***покрытой***, если она инцидентна ребру из M.

Определение. Чередующаяся цепь называется ***увеличивающей*** цепью, если обе её концевые вершины не покрыты.



Паросочетания

Утверждение. Увеличивающая цепь содержит нечётное количество рёбер, причём рёбер, не принадлежащих паросочетанию, на 1 больше, чем принадлежащих.



Паросочетания

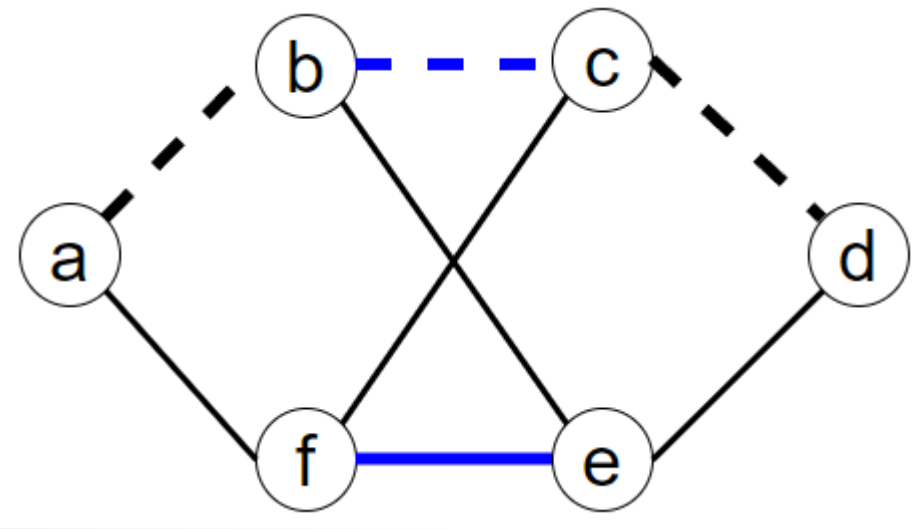
Пусть M --- паросочетание на графе $G(V, E)$.

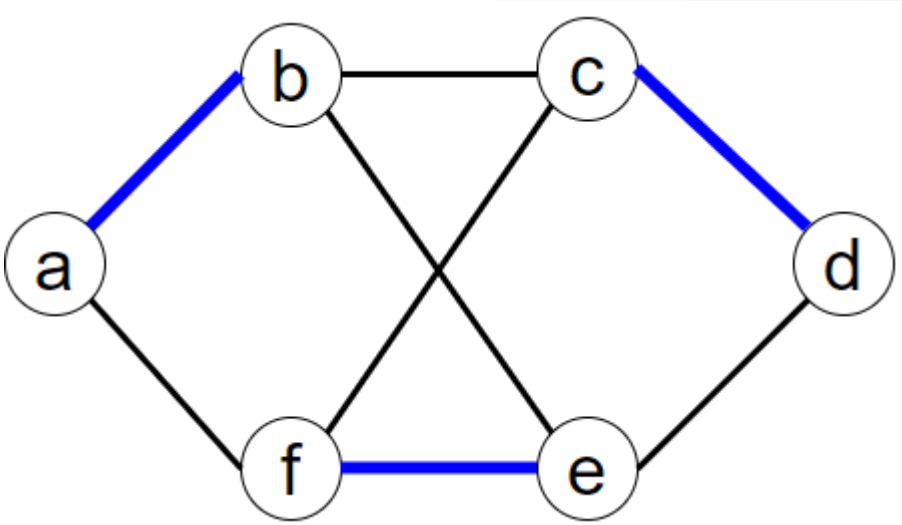
Теорема. Паросочетание M является наибольшим для $G \Leftrightarrow$ на G нет увеличивающей цепи для M.

Доказательство

$\Rightarrow$ Пусть M --- наибольшее паросочетание. И допустим, что на G есть увеличивающая цепь π. Построим новое паросочетание M': за пределами цепи π оно совпадает с M, а на рёбрах из π содержит дополнение к M.

Паросочетания





В силу предыдущего утверждения, $|M'| = |M| + 1$. Получили противоречие, M --- не наибольшее паросочетание.

Паросочетания

$\Leftarrow$Пусть на G нет увеличивающей цепи для M.

Пусть M^* --- наибольшее паросочетание.

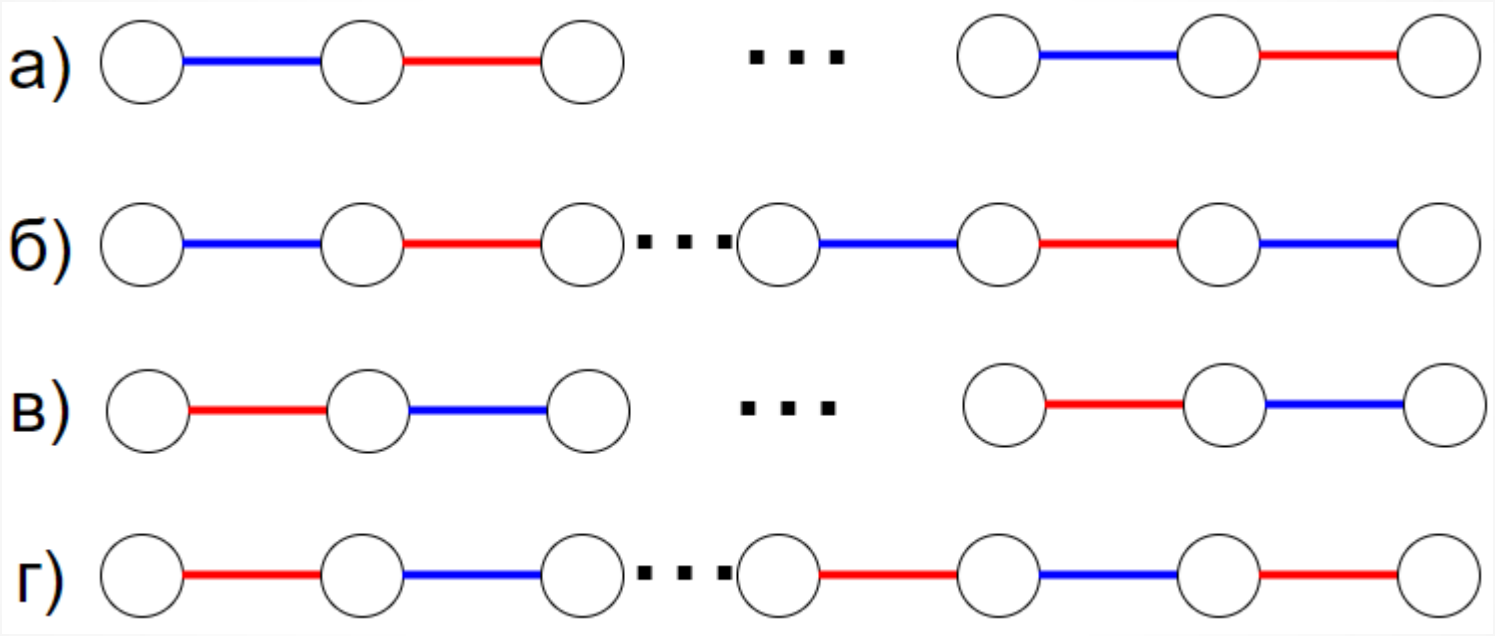
Рассмотрим G' --- частичный граф, порождённый множеством рёбер $M \Delta M^*$, где Δ означает симметрическую разность множеств ($M \Delta M^* = (M \cup M^*) \backslash (M \cap M^*)$).

На графе G' степень каждой вершины не превышает 2. Поэтому каждая связная компонента G' представляет собой либо изолированную вершину, либо цепь, либо цикл.

На G' не может быть цикла с нечётным количеством рёбер, т.к. иначе M или M^* не было бы паросочетанием.

Паросочетания

Рассмотрим, какие цепи могут быть на G'.



Цепь (б) является увеличивающей для M, цепь (г) – увеличивающая для M^*. Поэтому таких цепей не может быть.

Паросочетания

Итак, G' состоит из чередующихся циклов чётной длины и чередующихся цепей чётной длины. Поэтому $|M| = |M^*|$. Следовательно, M является наибольшим паросочетанием.

Теорема доказана.

В доказательстве теоремы в части «$\Rightarrow$» содержится описание процедуры, позволяющей на основании увеличивающей цепи построить паросочетание, имеющее бОльшую мощность, чем текущее паросочетание.

Паросочетания

Алгоритм Эдмондса

Идея алгоритма:

1) Выбрать начальное паросочетание M.
2) Итерационно увеличиваем размер паросочетания, пока это возможно:
    - Находим увеличивающую цепь.
    - Используем её для увеличения паросочетания.

Как искать увеличивающую цепь?

Паросочетания

Поиск увеличивающей цепи можно выполнить с помощью чередующихся/увеличивающих деревьев.

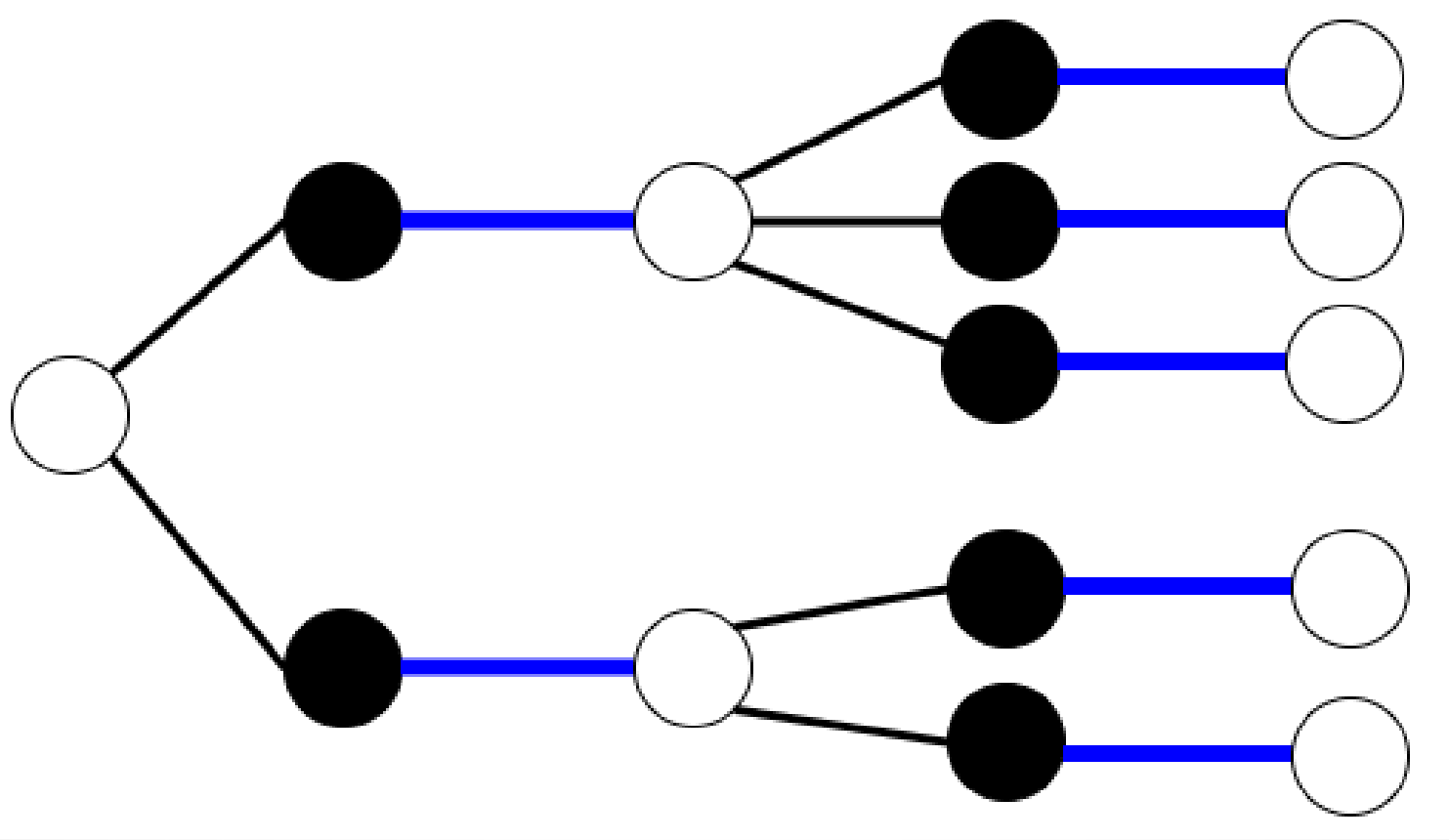
Построение чередующегося дерева:

1) Выбрать в качестве стартовой вершины произвольную ***непокрытую*** вершину. Сделать её корнем дерева и классифицировать как ***внешнюю*** вершину.

Паросочетания

2) Расширять дерево, добавляя к листьям новые рёбра по следующему правилу:
    a) Если лист – внешняя вершина, то добавим к дереву все инцидентные рёбра (они не принадлежат паросочетанию) и их другие вершины (*если они ещё не в дереве*). Добавленные вершины классифицируем как ***внутренние***.
    b) Если лист – внутренняя вершина, и она покрыта, то добавляем в дерево инцидентное ей ребро паросочетания и другой конец этого ребра (классифицируем как ***внешнюю*** вершину дерева).

Паросочетания



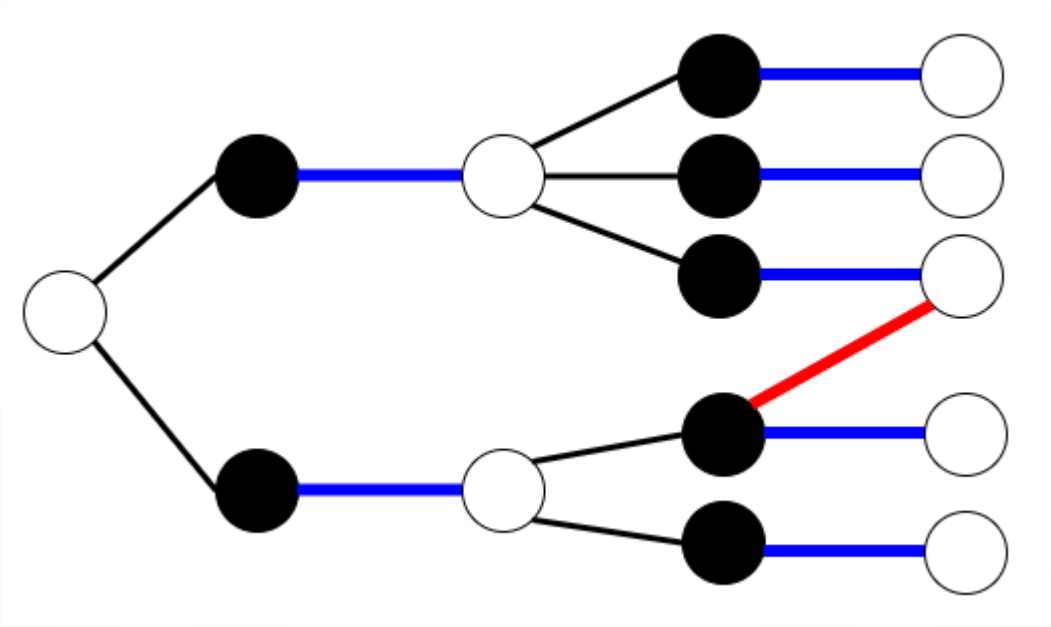
Паросочетания

В ходе этой процедуры возможны следующие ситуации:

1) В дерево будет добавлена *внутренняя* вершина, которая *не покрыта* текущим паросочетанием. В этом случае найдена увеличивающая цепь (начало – в корне дерева, конец – в этой вершине).

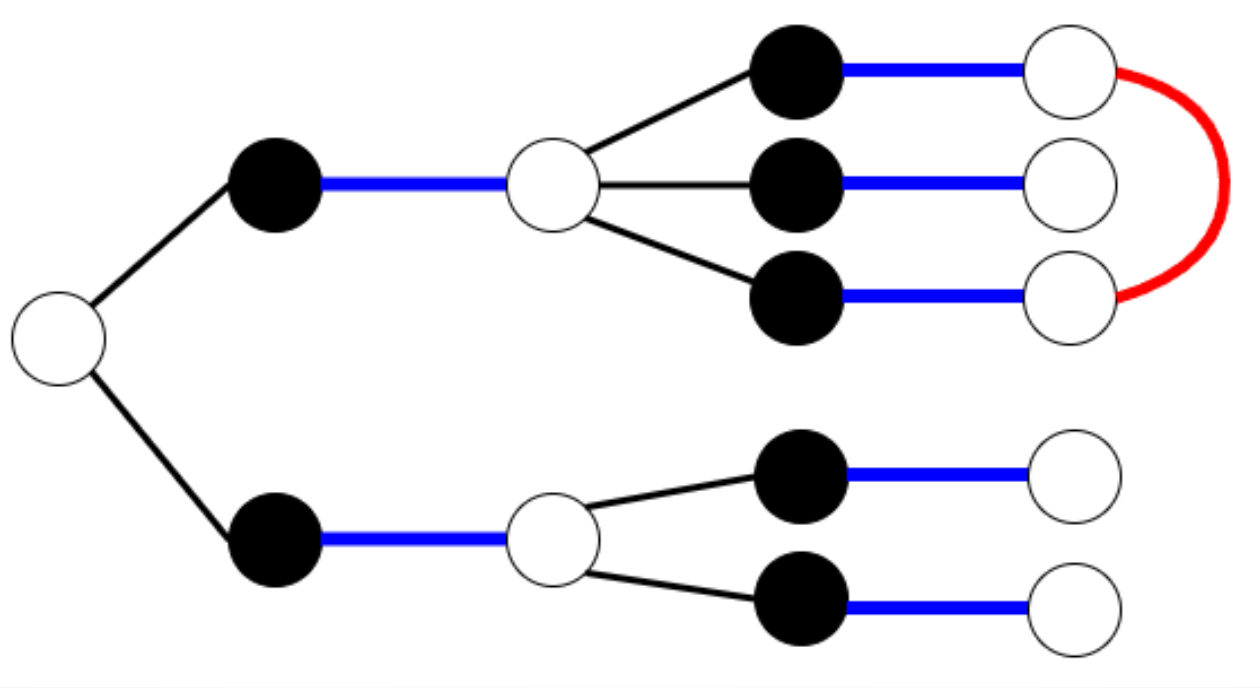
Паросочетания

2) На шаге (2а) нашли ребро, соединяющее текущую внешнюю вершину дерева с внутренней вершиной. Мы обнаружили на графе цикл чётной длины. Продолжаем выполнение алгоритма.



Паросочетания

3) На шаге (2а) нашли ребро, соединяющее две внешние вершины графа. Значит, нашли цикл нечётной длины. (Для двудольных графов такая ситуация не возникает!)



Паросочетания

Более точно: мы нашли ***цветок*** --- цикл нечётной длины $(2k + 1)$, в котором k внутренних вершин и $k + 1$ внешняя вершина, из которых одна соединена чередующейся цепью (***стеблем***) с корнем чередующегося дерева.

В этом случае надо ***срезать цветок***: стянуть все вершины цветка в одну новую ***квазивершину*** v'; все оставшиеся вершины графа, которые были смежны вершинам цветка, сделать смежными новой вершине v'. Получим новый граф G' и паросочетание M' на нём.

Теорема. На G есть увеличивающая цепь для $M \Leftrightarrow$ на G' есть увеличивающая цепь для M'.

(без доказательства).

Паросочетания

4) Мы не можем больше продолжать расширение увеличивающего дерева (все листья – внешние вершины, но они смежны только с внутренними вершинами этого же дерева). Такое дерево называется *венгерским*. Но при этом ещё не все вершины графа включены в дерево.

   В этом случае выбираем новую непокрытую вершину в качестве корня нового дерева и запускаем построение нового дерева.

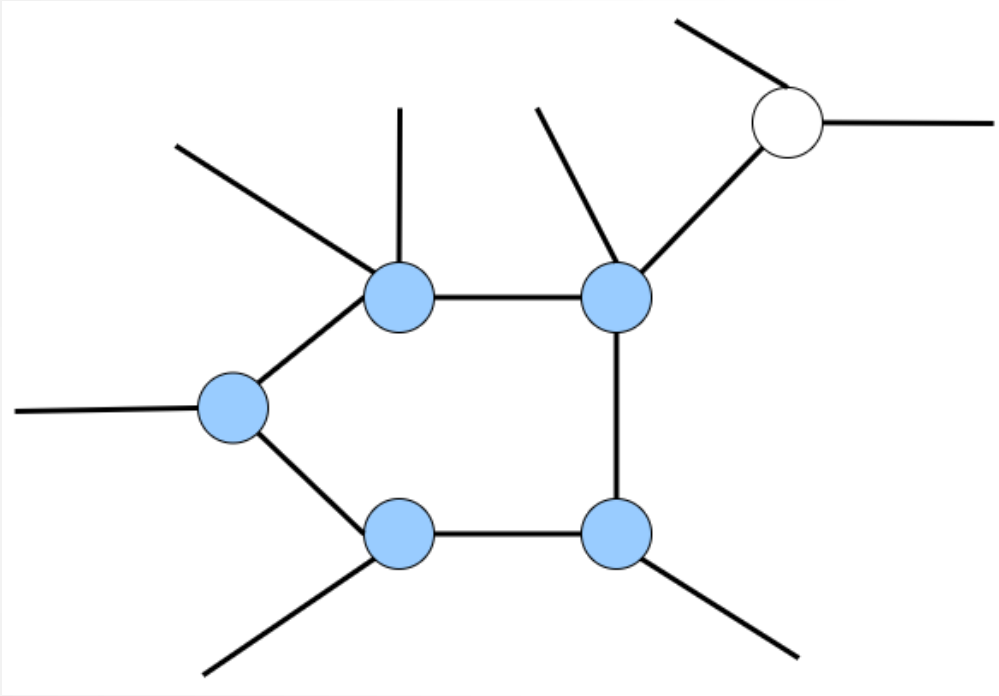
Паросочетания

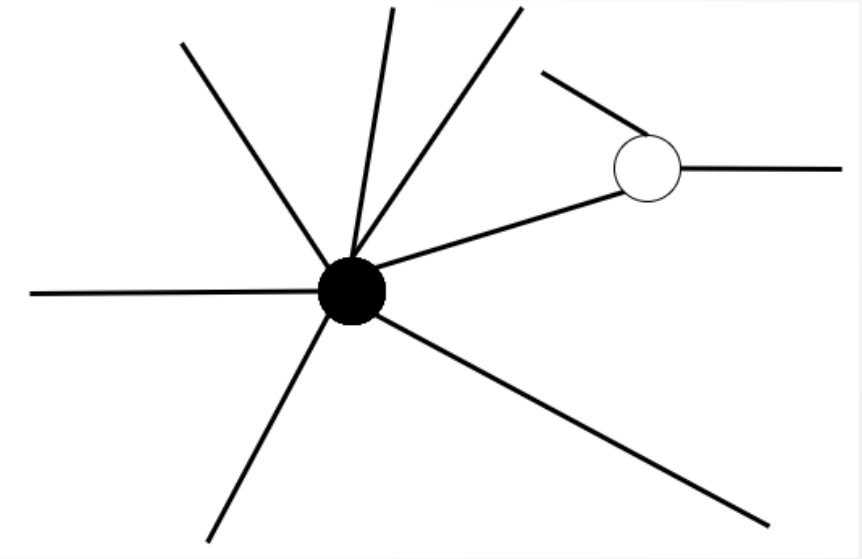
В итоге мы либо найдём увеличивающую цепь, либо разобьём граф на чередующиеся деревья (и рёбра между ними). В последнем случае текущее паросочетание является наибольшим.

Если в ходе алгоритма мы *срезали цветок*, то должны восстановить исходный граф, *распустив* каждый цветок.

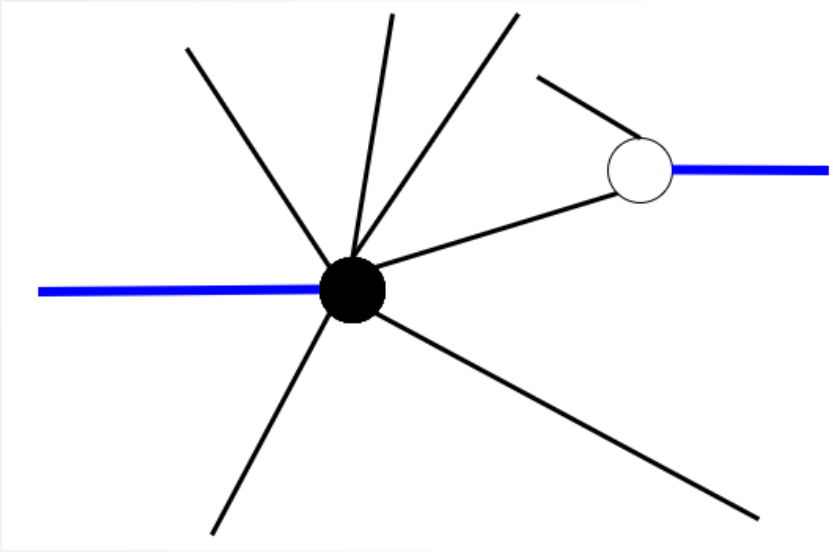
Распуская цветок, мы заменяем *квазивершину* на цикл. При этом, поскольку цикл имеет нечётную длину ($2k + 1$), мы всегда можем включить в итоговое паросочетание k рёбер из него таким образом, чтобы соблюсти требования к паросочетанию.

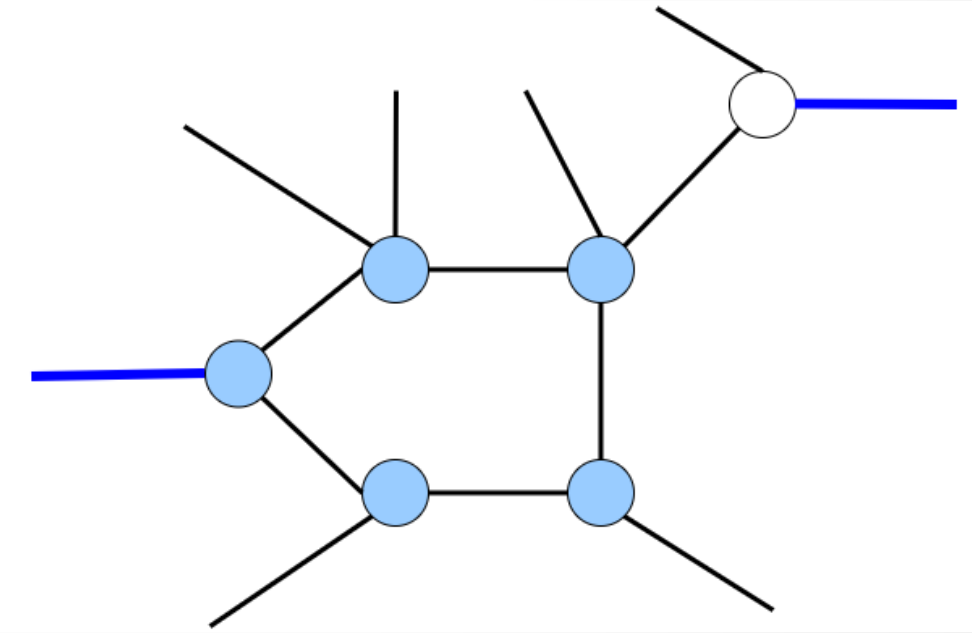
Распускание цветка (пример)



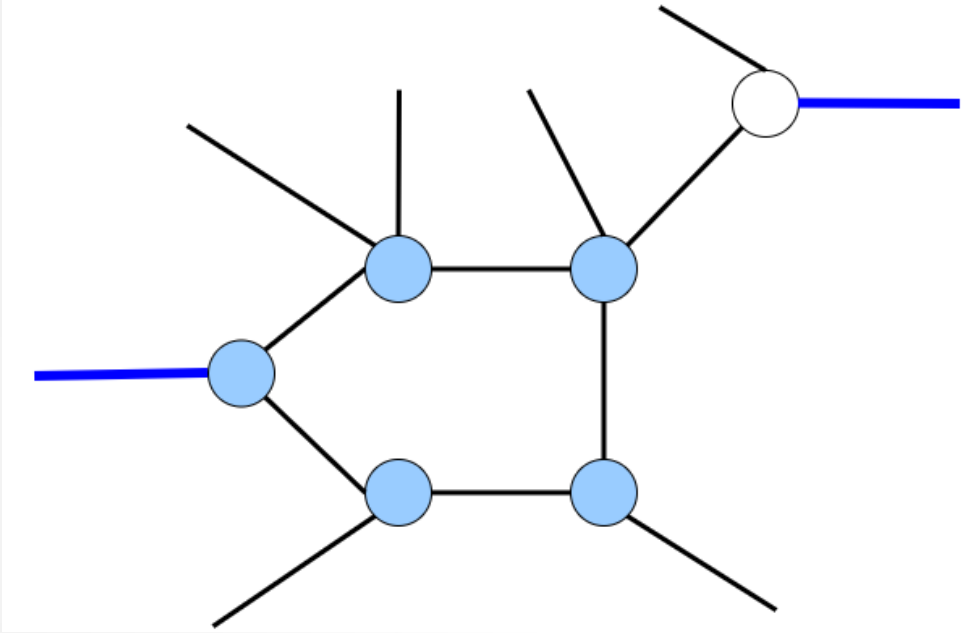


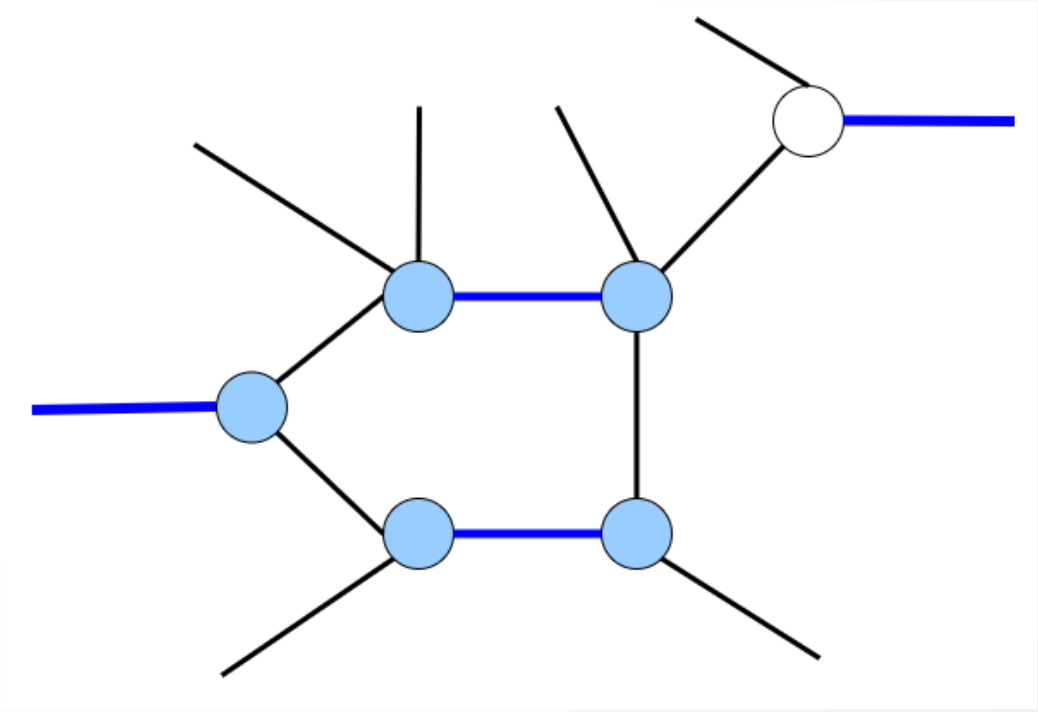
Распускание цветка (пример)





Распускание цветка (пример)





Паросочетания

Оценим временную сложность алгоритма.

- Поиск увеличивающей цепи на графе, без учёта срезания цветка, требует времени $O(m) = O(n^2)$.
- Срезание одного цветка требует времени $O(m) = O(n^2)$ при *наивной* реализации (при срезании цветка строим новый граф). Количество таких последовательных срезаний может достигать величины $O(n)$.
- Поэтому общая сложность нахождения одной увеличивающей цепи: $O(n^3)$.
- Количество итераций «найти увеличивающую цепь»+ «увеличить паросочетание»: $O(n)$.

Поэтому итоговая сложность алгоритма при «наивной» реализации: $O(n^4)$.

Паросочетания

При более продуманной реализации срезания цветка (хранить и обрабатывать только отличия G' от G) общая сложность алгоритма Эдмодса может быть снижена до $O(n^3)$.

Самый быстрый алгоритм (усовершенствованная версия алгоритма Эдмондса) имеет сложность $O(\sqrt{n} \cdot m)$.