Паросочетания в графах

Подготовил Сивухин Никита. По вопросам пишите на почту sivukhin.work+teach@gmail.com

Паросочетания в графах

Для неориентированного графа $G = (V, E)$ подмножество рёбер $M \subseteq E$ является *паросочетанием*, если в M нет соседних рёбер (иными словами, каждая вершина графа инцидентна не более чем одному ребру из M).

- *Размером паросочетания* $|M|$ называется количество рёбер в множестве M
- Рёбра из M будем называть *«тёмными»*, а рёбра не из паросочетания — *«светлыми»*
- Вершины, инцидентные рёбрам M, будем называть *«насыщенными»*, а остальные — *«свободными»*
- M является *максимальным по включению*, если оно не содержится ни в каком другом паросочетании как подмножество (иными словами его нельзя расширить новым ребром не нарушив свойство паросочетания)
- M является *наибольшим*, если размер любого другого паросочетания не превосходит $|M|$
- M является *совершенным* (perfect), если все вершины графа являются насыщенными

Замечание 1. *Добавление одной вершины к графу может увеличить размер наибольшего паросочетания не более чем на 1*

Замечание 2. *Для любой пары максимальных по включению паросочетаний M_1 и M_2 верно, что $|M_1| \leq 2|M_2|$*

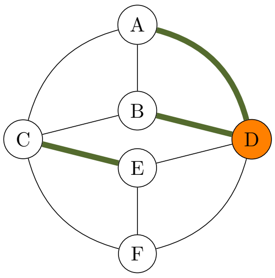

Рис. 1: Не паросочетание

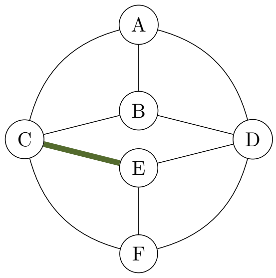

Рис. 2: Корректное паросочетание
F – *свободная* вершина
E – *насыщенная* вершина

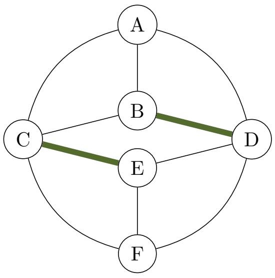

Рис. 3: Максимальное по включению паросочетание

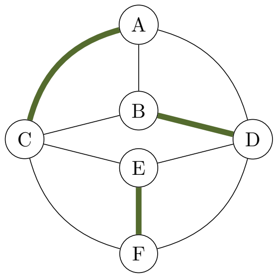

Рис. 4: *Совершенное* паросочетание

Свойство наибольшего паросочетания

Для графа $G = (V, E)$ и паросочетания $M \subseteq E$ определим понятия M*-цепи* и M*-чередующейся цепи*, необходимые для последующего изложения.

- M*-цепь* — это такой *простой путь* нечётной длины $(v_1, v_2, \dots v_{2k})$, где v_1 — свободная вершина, а рёбра пути чередуются по правилу светлое-тёмное (например, путь (F, C, E, D) на рис. 3).
- M*-чередующаяся цепь* — это M-цепь $(v_1, v_2, \dots v_{2k})$, где обе крайние вершины v_1 и v_{2k} являются свободными (например, путь (F, C, E, D, B, A) на рис. 3)

Теорема 1 (Берж, 1957)**.** *Паросочетание M в графе $G = (V, E)$ является наибольшим тогда и только тогда, когда в G не существует M-чередующихся цепей.*

Доказательство.

- **Необходимость условия**: если относительно паросочетания M в графе есть M-чередующаяся цепь $(v_1, v_2, \dots v_{2k})$, то паросочетание $M' = M \setminus \{\{v_{2i}, v_{2i+1}\}\}_{i=1}^{k-1} \cup \{\{v_{2i-1}, v_{2i}\}\}_{i=1}^{k}$ имеет размер $|M| + 1$, а значит M не являлось наибольшим.
- **Достаточность условия**: допустим, что в паросочетании M нет чередующихся цепей, но при этом есть M' большего размера.

  Рассмотрим граф $H = (V, M \oplus M')$, множеством ребёр которого является симметрическая разность рёбер из паросочетаний M и M'. Построенный граф обладает парой особых свойств:

  – Каждая вершина графа H имеет степень не больше двух
  – Рёбра вдоль любого пути в H чередуются между M и M'

  Из этих свойств следует, что рёбра графа можно представить в виде объединения непересекающихся компонент, являющихся цепочками и циклами, где в каждой компоненте рёбра чередуются между паросочетаниями. Из второго свойства также следует, что все циклы в графе H имеют чётную длину.

  Осталось заметить, что так как $|M'| > |M|$, а в любом цикле и цепочке чётной длины рёбер из каждого паросочетания поровну, то значит в H существует цепочка нечётной длины начинающаяся с ребра из M'. Такая цепочка является чередующейся в G относительно M, что противоречит условию теоремы.

$\square$

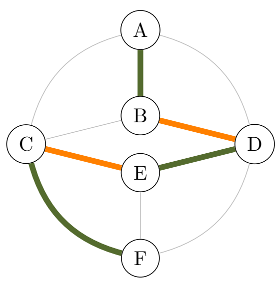

Рис. 5: Пример цепи нечётной длины в графе H

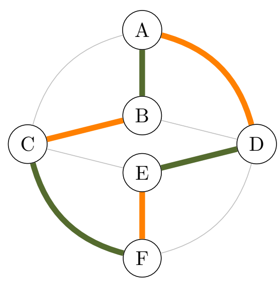

Рис. 6: Пример цикла в графе H

Заметим, что несмотря на свою простую формулировку, проверка наличия M-чередующейся цепи в произвольном графе является непростой задачей, из-за того что каждая вершина может быть посещена в разных «позициях» цепи (чётная/нечётная), что играет существенную роль при поиске M-чередующейся цепи.

Упражнение 1. *Постройте граф, на котором простой обход в глубину не сможет найти M-чередующуюся цепь*

Паросочетания в двудольных графах

Теорема 2 (Холл, 1935 (теорема о свадьбах))**.** *В двудольном графе $G = (X, Y, E)$ существует паросочетание насыщающее все вершины доли X тогда и только тогда, когда для любого подмножества $S \subseteq X$ верно, что $|N(S)| \geq |S|$, где $N(S)$ — множество всех соседей S.*

Доказательство.

- **Необходимость условия**: в графе с паросочетанием M насыщающим долю X, соседей $N(S)$ любого подмножества $S \subseteq X$ не меньше чем $|S|$, так как для каждой вершины S есть пара из M.
- **Достаточность условия**: докажем утверждение по индукции от количества вершин в графе ($|X| + |Y|$).

  База индукции. $|X| = 1$ — утверждение тривиально.
  Индуктивный переход. Рассмотрим два случая в графе:

  – Для любого собственного подмножества $S \subset X$ верно, что $|N(S)| > |S|$. В таком случае можно рассмотреть граф без произвольной вершины доли Y, для которого верно условие теоремы, а значит в графе существует совершенное паросочетание по предположению индукции
  – Существует собственное подмножество $S \subset X$, для которого $|N(S)| = |S|$. В таком случае рассмотрим два графа $G_I = G[S \cup N(S)]$ и $G_R = G[X \cup Y \setminus (S \cup N(S))]$. Утверждение теоремы очевидно выполняется для графа G_I, поэтому нужно лишь доказать выполнение свойства дла G_R. **Допустим** что это неверно и в G_R есть подмножество S' такое, что $|N_R(S')| < |S'|$. Заметим, что $N(S' \cup S) = N_R(S') \sqcup N(S)$, а значит $|N(S' \cup S)| = |N_R(S')| + |N(S)|$. Но так как для $S \cup S'$ верно свойство теоремы, то $|N(S \cup S')| \geq |S \cup S'| = |S| + |S'|$, а значит $|N_R(S')| \geq |S'|$, так как $|N(S)| = |S|$, что противоречит изначальному **предположению**.

□

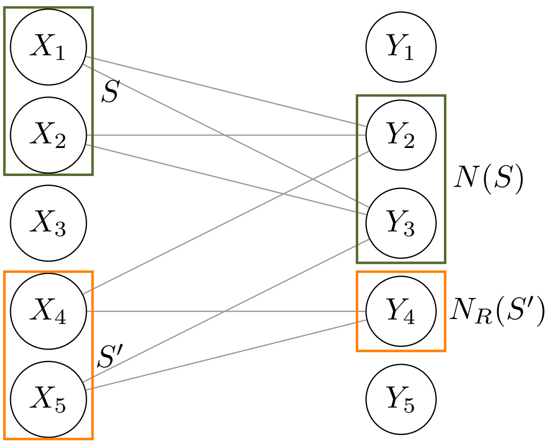

Рис. 7: Если $|N_R(S')| < |S'|$, то изначальный граф нарушал свойство теоремы

Замечание 3 (Ориентация рёбер наведённая паросочетанием)**.** *В двудольном графе $G = (X, Y, E)$ с паросочетанием M можно определить естественную ориентацию рёбер такую, что каждое «светлое» ребро будет ориентировано от доли X к Y, а каждое «тёмное» — от Y к X.*

Важно, что ориентация рёбер вдоль любой M-цепи буде одинаковая, так как цепь имеет нечётную длину, а значит крайние вершины принадлежат разным долям графа.

Будем обозначать граф, ориентированный относительно паросочетания M таким образом как G_M.

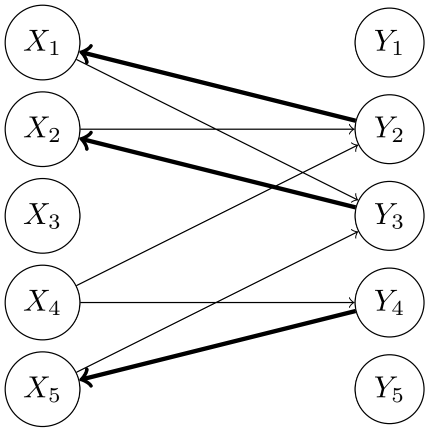

Рис. 8: Наведённая ориентация. Ориентация рёбер вдоль M-цепи (Y_3, X_1, Y_2, X_4) одинаковая.

Замечание 4 (Пути в графе G_M)**.** *Любой ориентированный путь между парой «свободных» вершин в графе G_M соответствует некоторой M-чередующейся цепи и наоборот.*

Алгоритм Куна

Комбинируя теорему Бержа с замечанием 4 получаем первый алгоритм нахождения наибольшего паросочетания в двудольном графе: с помощью поиска в глубину ищем любую M-чередующуяся цепь как путь между парой свободных вершин в ориентированном графе, пока они есть. Финальное паросочетание по теореме Бержа является максимальным.

Одна итерация поиска M-чередующейся цепи работает за время $O(n + m)$, а всего их будет не больше чем размер максимального паросочетания, т.е. $O(n)$. Итого, алгоритм работает за $O(n^2 + nm)$, что вырождается в $O(n^3)$ для случая плотных графов.

Алгоритм 1 (Алгоритм Куна)**.** *Для того, чтобы получить каноничный алгоритм, нужно обрабатывать вершины одной из долей по порядку, где на каждом шаге алгоритма будет построено наибольшее паросочетание относительно обработанных вершин одной из долей. Тогда, пользуясь замечанием 2, достаточно искать M-чередующуюся цепь только относительно новой рассматриваемой вершины.*

В таком виде алгоритм будет работать за время $O(n_1 m)$, где n_1 — размер выбранной доли. Заметим, что в зависимости от выбора доли время работы алгоритма может сильно отличаться и для графов с большой разницой в размерах долей итерирование по меньшей доле может принести заментный прирост в скорости работы.

Стоит также отметить несколько неасимптотических оптимизаций, которые на практике могут ускорить алгоритм в несколько раз:

- Жадная инициализация построит 2-приближение к оптимальному паросочетанию и уменьшает количество шагов основного алгоритма минимум в 2 раза

- Рандомизация порядка вершин перед инициализацией может увеличить размер жадного паросочетания из-за структуры входных данных

- Если не сбрасывать пометки с посещенных вершин, можно искать максимальное по включению множество непересекающихся M-чередующихся цепей за один обход графа, что существенно ускоряет алгоритм

  Заметим, что жадная инициализация в данном подходе не нужна, так как жадное паросочетание будет построено автоматически в рамках первой итерации этого подхода

```cpp
struct Graph {
  int n, m; // n = |X|, m = |Y|
  vector<vector<int>> edges; // edges from X part to Y part
  // match[y] = x - pair for vertex "y" from Y to vertex "x" from X
  bool try_kuhn(v, vector<int> &match, vector<int> &used) {
    if (used[v]) {
      return false;
    }
    used[v] = 1;
    for (int to : edges[v]) {
      if (match[to] == -1 || try_kuhn(match[to], match, used)) {
        match[to] = v;
        return true;
      }
    }
    return false;
  }
  // unoptimized version
  vector<int> match() {
    auto match = vector<int>(m, -1);
    auto used = vector<int>(n, 0);
    for (int v = 0; v < edges.size(); v++) {
      fill(used.begin(), used.end(), false);
      try_kuhn(v, match, used);
    }
    return match;
  }
};
```

Ссылки

- Быстрый Кун: https://codeforces.com/blog/entry/17023
- Паросочетания в DI-контейнере: https://github.com/skbkontur/GroboContainer
- Паросочетания в postgresql: https://github.com/postgres/postgres
- Теорема о свадьбах: https://ru.wikipedia.org/wiki
- Обзорная статья про паросочетания и их приложения: https://web.eecs.umich.edu