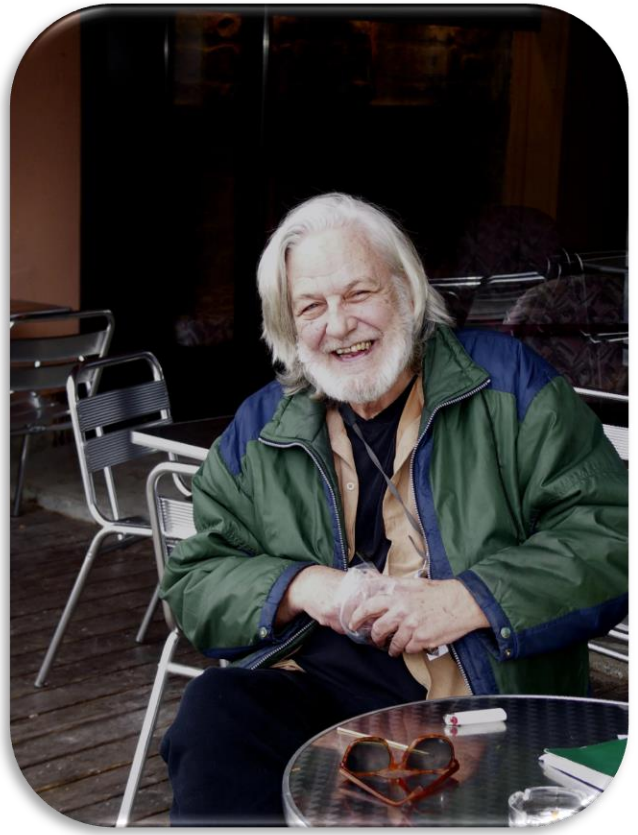
Алгоритм сжатия цветков

Студент Яцевич Ксения
Б9121-09.03.03 пикд

Историческая справка

Алгоритм разработал Джек Эдмондс в 1961 году и опубликовал в 1965 году.

Основной причиной, почему алгоритм сжатия цветков важен, является то, что он дал первое доказательство возможности нахождения наибольшего паросочетания за полиномиальное время. Другой причиной является то, что метод приводит к описанию многогранника линейного программирования для многогранника паросочетаний, что приводит к алгоритму паросочетания минимального веса.



Постановка задачи

Разрешение проблемы нахождения максимального паросочетания в графах с нечетными циклами

Описание алгоритма

Идея алгоритма

Алгоритм сжатия цветков (англ. Blossom algorithm) — это алгоритм в теории графов для построения наибольших паросочетаний на графах.

Описание алгоритма

Оценка сложности

Всего имеется n итераций, на каждой из которых выполняется обход в ширину за O(m) кроме того, могут происходить операции сжатия цветков — их может быть O(n).

Сжатие соцветий работает за O(n), то есть общая асимптотика алгоритма составит O(n(m+n^2))=O(n^3).

Описание алгоритма

Дополняющий(увеличивающий) путь

Мы сможем найти максимальное паросочитание путем инверсии дополняющего пути.

Дополняющий путь - чередующаяся цепь, которая начинается и кончается голыми вершинами.

Описание алгоритма

Сжатие цветка

Сжатие всего нечётного цикла в одну псевдо-вершину (соответственно, все рёбра, инцидентные вершинам этого цикла, становятся инцидентными псевдо-вершине).

Описание алгоритма

Теорема Эдмондса

Пусть граф $\overline{G}$ был получен из графа G сжатием одного цветка. Тогда в графе $\overline{G}$ существует увеличивающая цепь тогда и только тогда, когда существует увеличивающая цепь в G.

Описание алгоритма

Общая схема алгоритма

```
void edmonds() {
    for (int i=0; i<n; ++i)
        if (вершина i не в паросочетании) {
            int last_v = find_augment_path (i);
            if (last_v != -1)
                выполнить чередование вдоль пути из i в last_v;
        }
}

int find_augment_path (int root) {
    обход в ширину:
        int v = текущая вершина;
        перебрать все рёбра из v
            если обнаружили цикл нечётной длины, сжать цикл
            если пришли в свободную вершину, возвращаем дополняющий путь
            если пришли в несвободную вершину, то добавить в очередь смежную ей в паросочетании
    return -1;
}
```

Описание алгоритма

Пример работы алгоритма

Исходный граф | Сжатие первого цветка | Сжатие второго цветка | Сжатие третьего цветка | Инверсия дополняющего пути | **Восстановление графа**