Теория графов

Построение базы циклов. Рационализация

Константин Вернер

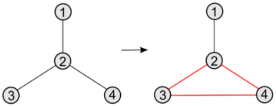
30.10.2020

База цикла

$(\Gamma_V, \oplus, \cdot)$ - векторное пространство графов на множестве вершин V
$C[G]$ - *пространство циклов* графа $G \in \Gamma_V$

Определение

Фундаментальный цикл графа G относительно остова T - простой цикл C, полученный путем добавления к остову T ребра $e \notin T$



База циклов

Утверждение

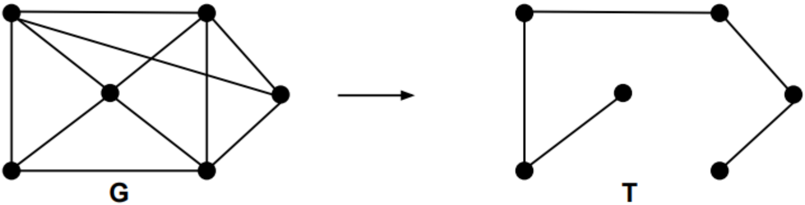
Множество всех фундаментальных циклов относительно любого каркаса T графа G образует базис пространства циклов этого графа.

На основе этого утверждения предлагается следующий способ построения базиса пространства циклов графа (базы цикла):

1) Найти какой-нибудь каркас T;
2) Для каждого ребра $e \notin T$ найти единственный цикл, который содержит это ребро e и ребра каркаса T, поместить его в базу циклов.

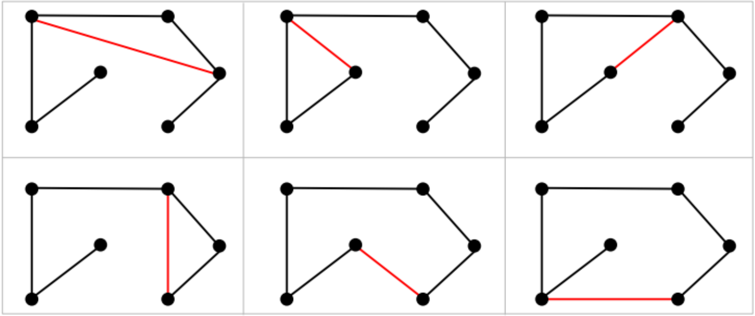
Пример

Дан граф G, найдем его каркас T



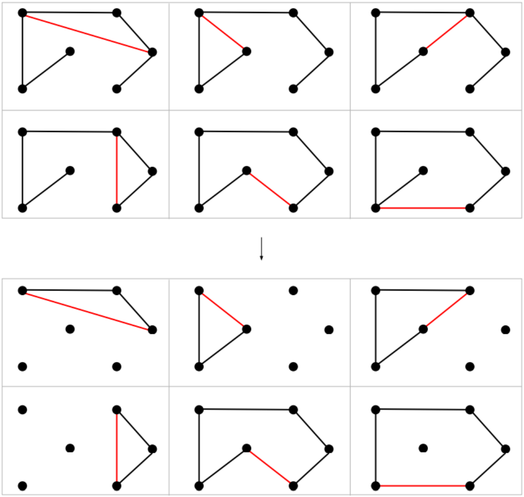
Пример

Ищем все графы $T \oplus \{e\}\ \forall e \notin T$



Пример

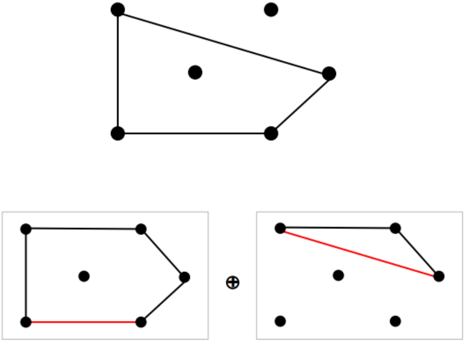
Ищем все графы $T \oplus \{e\} \; \forall e \notin T$



Изолируем циклы и помещаем их в базис.

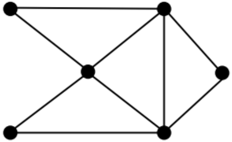
Пример

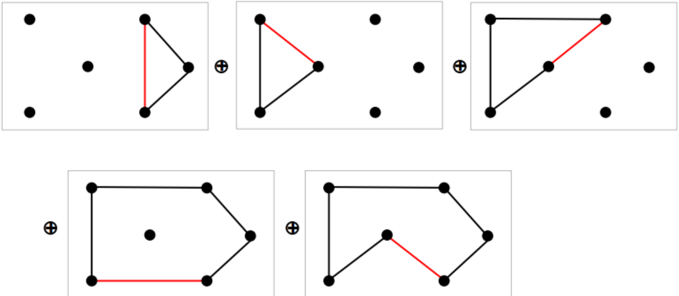
Любой эйлеров подграф G может быть представлен как линейная комбинация элементов базы циклов.



Пример

Любой эйлеров подграф G может быть представлен как линейная комбинация элементов базы циклов.





Построение базы циклов

Будем описывать алгоритм на основе алгоритма DFS

Построение Базы Циклов

Вход: $G = (V, E)$

помечаем все вершины как новые
помечаем вее ребра как новые
k := 1 - счетчик циклов
for v in V
 if v новая
  CycleBase(v)

Построение базы циклов

```
Procedure CycleBase(v)
 открыть v
 T(v) = v
 x := v
 while x открытая:
  if (x,y) неисследованно:
    пометить (x,y) как иследованное
    if y новая
       открыть y
       T(y) = x
        x := y
    else
       newCycle(x,y)
   else
     закрыть вершину x
     x = T(x)
```

Построение базы циклов

```
Procedure newCycle(x,y)
 k := k + 1
 C(k) = [x]
 z := x
 while z != y
    z := T(z)
    Добавить z в C(k)
```

База Циклов: Анализ Алгоритма

DFS выполняется за линейное от числа вершин и ребер время.

В поиске базы циклов также необходимо запоминать встречающиеся циклы. Подсчитаем суммарную длину этих циклов для полного графа с n вершинами. Относительно DFS-дерева будет $n-2$ цикла длины 3, $n-3$ цикла длины 4, ..., 1 цикл длины n. Сумма длин всех фундаментальных циклов равна:

$$\sum_{i=1}^{n-2} i(n+1-i) = \frac{n^3+3n^2-16n+12}{6}$$

Таким образом, сложность алгоритма $\mathcal{O}(n^3)$

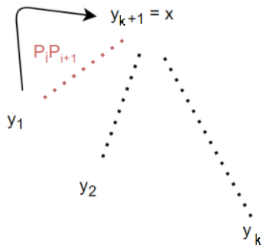
База Циклов: Рационализация

Покажем, что сложность приведенного алгоритма можно свести к $O(n^2)$.

Рассмотрим произвольную $x \in V$
$y_1, ..., y_n$ - предки x в DFS-дереве, соединенные с x обратными ребрами. И пусть $y_{k+1} = x$

$P_i, i \in (1, ..., n)$ - путь в DFS-дереве между $y_i = y_{i+1}$

Алгоритм дает циклы вида $C_i = xP_iP_{i+1}...P_kx, i \in (1, ..., n)$



База Циклов: Рационализация

Алгоритм дает циклы вида $C_i = aP_iP_{i+1}...P_ka, i \in (1, ..., k)$

Рассмотрим циклы $C'_i = aP_ia, i \in (1, ..., k)$

$C_i = C'_i \oplus C'_{i+1} \oplus ... \oplus C'_k$ - Совокупность таких циклов образует базу циклов графа.

Будем называть такую систему *сокращенной*

База Циклов: Рационализация

Изначальный алгоритм дает нам C_i, но мы хотим получать C_i'.

Для этого нужно после обнаружения обратного ребра от предка x к потомку y выписать вершины, содержащиеся в стеке, начиная с y и заканчивая следующей вершиной, смежной с x.

Анализ алгоритма

Оценим суммарную длину S циклов сокращенной системы для графа с n вершин и с m ребер.

$C_i' = xP_ix$.
P_i состоит из y_i и y_{i+1}; каждое из них имеет обратное ребро с x. P_{i+1} имеет общее ребро с P_i

Анализ алгоритма

$C_i' = xP_ix$.
P_i состоит из y_i и y_{i+1}; каждое из них имеет обратное ребро с x. P_{i+1} имеет общее ребро с P_i

$$\Rightarrow$$

Каждое обратное ребро принадлежит не более чем двум циклам сокращенной системы.

$$\Rightarrow$$

Суммарный вклад обратных ребер в S не более чем $2m$.

Анализ алгоритма

В каждом цикле системы назовем *верхушкой* этого цикла вершину x, при исследовании окрестности который был найден этот цикл.

Для каждого прямого ребра в сокращенной системе имеется не более одного цикла с данной верхушкой.

$$\Rightarrow$$

Число циклов, в которые входит данное прямое ребро, не превосходит числа вершин, лежащих в дереве выше этого ребра. Это число не превосходит числа всех вершин графа n.

Анализ алгоритма

Для каждого прямого ребра в сокращенной системе имеется не более одного цикла с данной верхушкой.

$$\Rightarrow$$

Число циклов, в которые входит данное прямое ребро, не превосходит числа вершин, лежащих в дереве выше этого ребра. Это число не превосходит числа всех вершин графа n.

Имеется не более чем $n - 1$ прямое ребро,

$$\Rightarrow$$

Для суммарного вклада всех прямых ребер в S получаем верхнюю оценку n^2.

Анализ алгоритма

Для каждого прямого ребра в сокращенной системе имеется не более одного цикла с данной верхушкой.

$$\Rightarrow$$

Число циклов, в которые входит данное прямое ребро, не превосходит числа вершин, лежащих в дереве выше этого ребра. Это число не превосходит числа всех вершин графа n.

Имеется не более чем $n-1$ прямое ребро,

$$\Rightarrow$$

Для суммарного вклада всех прямых ребер в S получаем верхнюю оценку n^2.

Таким образом,

$$S < 2m + n^2 = \mathcal{O}(n^2)$$