

2-приближение для взвешенной задачи вершинного покрытия

Проект по курсу
Сложности вычислений

Поминов Денис
группа Б05-824

Аннотация

В данной работе приведен анализ алгоритма 2-приближения для взвешенной задачи вершинного покрытия[1]. Описание алгоритма 1 находится в разделе 2. В разделе 3 доказана корректность. Также в разделе 4 приведен пример графа, на котором достигается худшее приближение и анализ точности приближения и времени работы алгоритма.

1 Введение

Задача поиска вершинного покрытия является классической задачей в теории алгоритмов. Эта задача так же интересна тем, что она является **NP**-полной[2]. Суть задачи состоит в отыскании минимального вершинного покрытия, т.е. содержащего наименьшее число вершин. Не менее интересной является задача отыскания минимального вершинного покрытия в графе со взвешенными вершинами[3].

В данной статье мы рассмотрим полиномиальный алгоритм1, который решает задачу 2-приближения минимального взвешенного вершинного покрытия. Далее мы проанализируем алгоритм и разберем пример графа, на котором достигается худшее приближение, а также рассмотрим зависимости некоторых метрик работы алгоритма от количества вершин.

В дальнейшем мы обсуждаем реализацию алгоритма. Для проверки его были сгенерированы наборы тестов. На практике алгоритм успешно решает задачу 2-приближения и в некоторых случаях находит оптимальное решение.

2 Описание алгоритма

В данной части приведены описание алгоритма и определения, необходимые для понимания алгоритма.

Определение. Дан граф $G = (V, E)$. Вершинным покрытием называется множество $S \subset V$, такое что каждое ребро из E имеет хотя бы один из концов в S .

Определение. Дан граф $G = (V, E)$ и $w : V \rightarrow \mathbb{R}^+$. Минимальным вершинным покрытием называется вершинное покрытие S , такое что $\sum_{v \in S} w(v) = \min$.

Для поиска минимального вершинного покрытия в графе со взвешенными вершинами воспользуемся методом оценивания ребер, а также свойством честности.

Определение. Дан граф $G = (V, E)$. Каждому ребру сопоставим некоторое число $p_e \geq 0$. Тогда свойство честности: $\forall i \in V : \sum_{e=(i,j)} p_e \leq w_i$.

Определение. Дан граф $G = (V, E)$. Назовем вершину i жадной, если $\sum_{e=(i,j)} p_e = w_i$.

3 Доказательство корректности

В этой части приводится доказательство корректности алгоритма, а так же доказательство вспомогательных утверждений.

Algorithm 1 Алгоритм 2-приближения для взвешенной задачи вершинного покрытия

```
1: procedure WEIGHTEDVERTEXCOVER
2:    $S \leftarrow \emptyset$ 
3:   for  $e \in E$  do
4:      $p_e \leftarrow 0$ 
5:   while  $\exists e = (i, j) : i \text{ и } j \text{ — не жадные}$  do
6:     увеличиваем  $p_e$  пока  $i$  или  $j$  не станет жадной
7:      $S \leftarrow S \cup$  жадные вершины, получившиеся на этой итерации
8:   return  $S$ 
```

3.1 Вспомогательные утверждения

Лемма 1. Пусть $G = (V, E)$ — граф со взвешенными вершинами. Тогда $\forall S$ — вершинное покрытие и $\forall p_e$ — честных оценок ребер: $\sum_{e \in E} p_e \leq w_S$.

Доказательство. Т.к. S — вершинное покрытие, $\sum_{e \in E} p_e \leq \sum_{i \in S} \sum_{e=(i,j)} p_e$. В силу условия честности оценок ребер имеем: $\sum_{i \in S} \sum_{e=(i,j)} p_e \leq \sum_{i \in S} w_i = w(S)$. \square

3.2 Корректность

Теорема 1 (о корректности алгоритма). Алгоритм1 находит 2-приближение решения задачи минимального вершинного покрытия

Доказательство. Алгоритм останавливается, т.к. после каждой итерации в множество S — всех жадных вершин, добавляется хотя бы одна вершина. Рассмотрим множество S после завершения алгоритма. Для каждого ребра $e = (i, j)$ имеем, что $i \in S$ или $j \in S$, следовательно S — вершинное покрытие.

В силу того, что $\forall i \in S : i$ — жадная, $S \subset V$ и $\forall e \in E : p_e \geq 0$, имеем:

$$w(S) = \sum_{i \in S} w_i = \sum_{i \in S} \sum_{e=(i,j)} p_e \leq \sum_{i \in V} \sum_{e=(i,j)} p_e = 2 \cdot \sum_{e \in E} p_e$$

Теперь рассмотрим S^* — минимальное вершинное покрытие. Из леммы имеем: $\sum_{e \in E} p_e \leq w(S^*)$. \square

4 Анализ программы

В данном разделе проводится анализ алгоритма на основе этой реализации.

4.1 Пример графа с наихудшим приближением

Заметим, что работа алгоритма зависит от порядка вершин и порядка ребер. Нетрудно увидеть, что примером графа, на котором достигается худшее приближение будет следующая конструкция.

Пусть имеем n вершин — $\{v_1, \dots, v_n\}$. Каждая из вершин имеет $w = 1$. Проведем ребра между v_i и v_{i+1} вершинами $\forall i \in \{1, \dots, n-1\}$. Очевидно, что вес минимального вершинного покрытия будет равен $\lceil \frac{n}{2} \rceil$.

Лемма 2. На этом графе в результате работы алгоритма достигается худшее приближение.

Доказательство. Докажем методом математической индукции.

База. Для $n = 2$ — после итерации $p_e = 1$, где $e = (v_1, v_2)$, а $S = \{v_1, v_2\}$.

Шаг. Пусть верно для $\forall n \leq k-1$. Докажем для $n = k$. Рассмотрим первый шаг алгоритма: $e = (v_1, v_2)$ и $p_e = 1$, $S = \{v_1, v_2\}$. На следующем шаге мы рассмотрим ребро $e_1 = (v_2, v_3)$ и ничего не сделаем. Дальнейший ход алгоритма эквивалентен запуску алгоритма на $n = k-2$ вершинах. Пусть s — ответ на предыдущем шаге.

$$\frac{s+2}{\lceil \frac{k}{2} \rceil} = x$$

$$\frac{s}{\lfloor \frac{k-2}{2} \rfloor} = 2$$

Решая систему получим, что $x = 2$.

□

4.2 Описание тестов

Наборы тестов можно посмотреть в репозитории с реализацией.

Здесь и далее $G(n, p)$ — случайный граф на n , где ребро проводится с вероятностью p .

- G — графы на n вершинах, построенные в прошлом подразделе, где $n \in \{2, \dots, 21\}$. Количество вершин достаточно маленькое, чтобы проверить степень приближения, сравнивая результаты алгоритма с полным перебором вершинных покрытий.
- $G(n, 0.5)$, где $n \in \{7, 10\}$ — 5000 тестов каждого вида
- $G(n, p)$, где $n \in \{3, 500\}$, а $p \in \{0.5, 0.7, 0.9\}$

Реализация алгоритма была проверена на корректность приведенными тестами и многократным запуском на случайно сгенерированных графах.

4.3 Выводы

- Отношение приближенного и точного весов стремится к некоторой константе на Рис. 1 и Рис. 3
- Отношение времен не стабилизируется, видно, что нахождение оптимального решения занимает крайне много времени по сравнению с 2-приближением. (Рис. 2 и Рис. 4)
- Нахождение покрытия *corner-case* графа занимает небольшое количество времени для рассматриваемого алгоритма, однако целесообразность применения будет зависеть от конкретной задачи. (Рис. 5 и Рис. 6)
- Из графика можем понять, что алгоритм 2-приближения является полиномиальным. Четкой зависимости от плотности графа нет, т.к. скорость и оптимальность работы зависит от порядка обхода ребер. (Рис. 7)

5 Заключение

Приведенный в данной статье алгоритм является хорошим приближенным решением задачи взвешенного вершинного покрытия. Проанализировав его работу, можно утверждать, что скорость алгоритма подходит для применения его на практике. Теоретические методы, которые используются в данном алгоритме так же могут иметь широкое применение. Хотя нельзя не заметить, что алгоритм может давать наихудшее приближение на довольно тривиальных примерах.

Список литературы

- [1] Josef Řimánek. *Linear programming: Vašek CHVÁTAL W.H. Freeman and Co., New York, 1983, xiii + 478 pages.* авг. 1985.
- [2] Ronald Book. *Review: Michael R. Garey and David S. Johnson, Computers and intractability: A guide to the theory of NP-completeness.* сент. 1980.
- [3] Jon Kleinberg и Éva Tardos. *Algorithm Design.* янв. 2006.

6 Приложение

График отношения приближенного и точного веса минимального вершинного покрытия для $G(7, 0.5)$

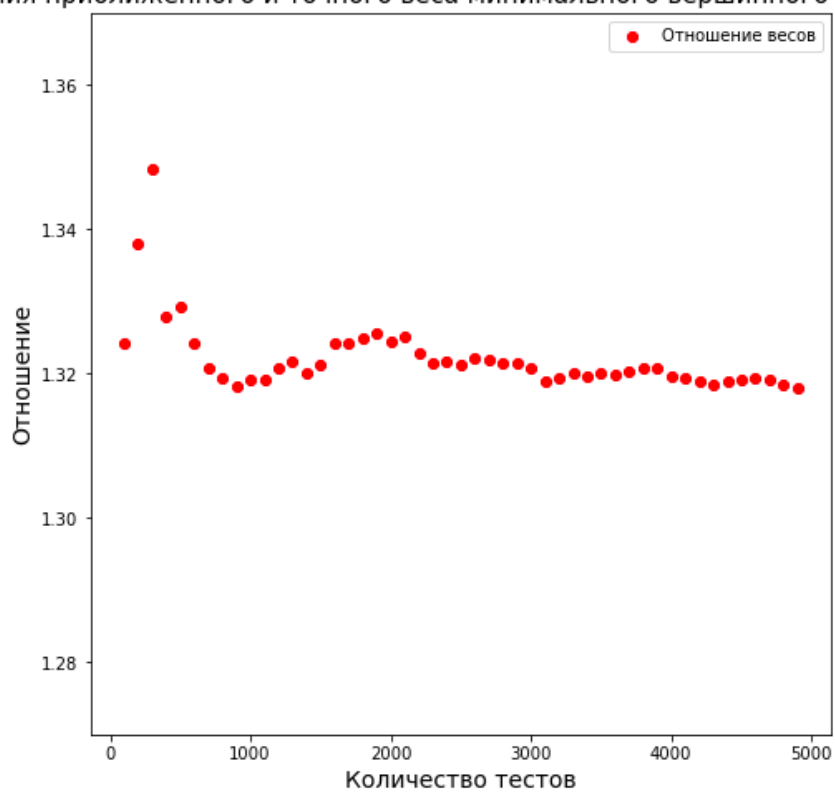


Рис. 1: График отношения приближенного и точного веса минимального вершинного покрытия для $G(7, 0.5)$

График отношения времени работы алгоритма приближенного и оптимального нахождения покрытия для $G(7, 0.5)$

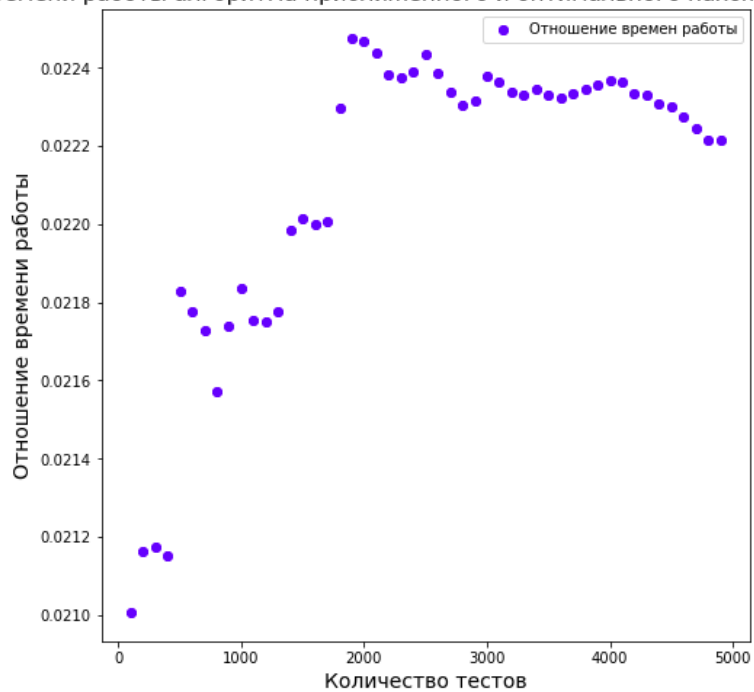


Рис. 2: График отношения времени работы алгоритма приближенного и оптимального нахождения покрытия для $G(7, 0.5)$

График отношения приближенного и точного веса минимального вершинного покрытия для $G(10, 0.5)$

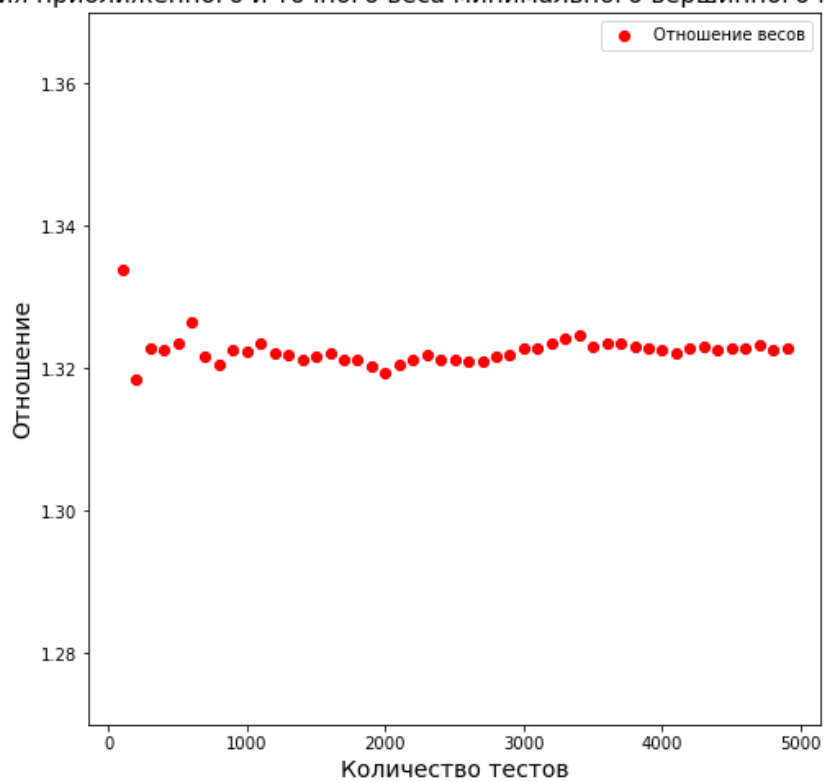


Рис. 3: График отношения приближенного и точного веса минимального вершинного покрытия для $G(10, 0.5)$

График отношения времени работы алгоритма приближенного и оптимального нахождения покрытия для $G(10, 0.5)$

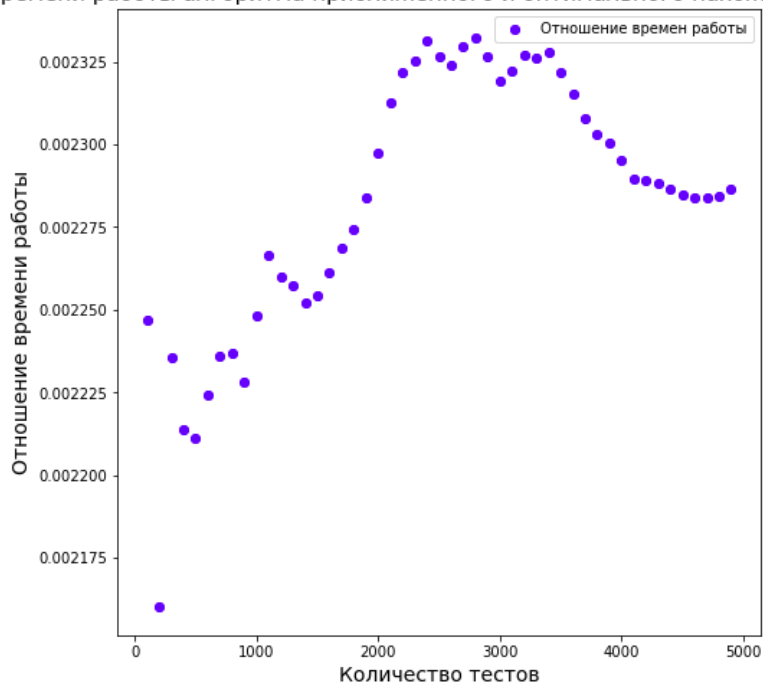


Рис. 4: График отношения времени работы алгоритма приближенного и оптимального нахождения покрытия для $G(10, 0.5)$

График отношения времени работы алгоритма приближенного и оптимального нахождения покрытия для corner-case графа

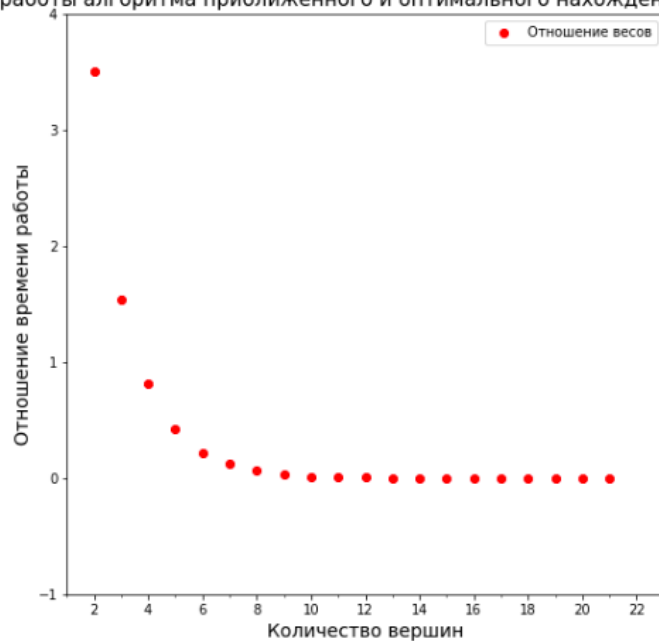


Рис. 5: График отношения времени работы алгоритма приближенного и оптимального нахождения покрытия для corner-case графа

График отношения времени работы алгоритма приближенного и оптимального нахождения покрытия для corner-case графа

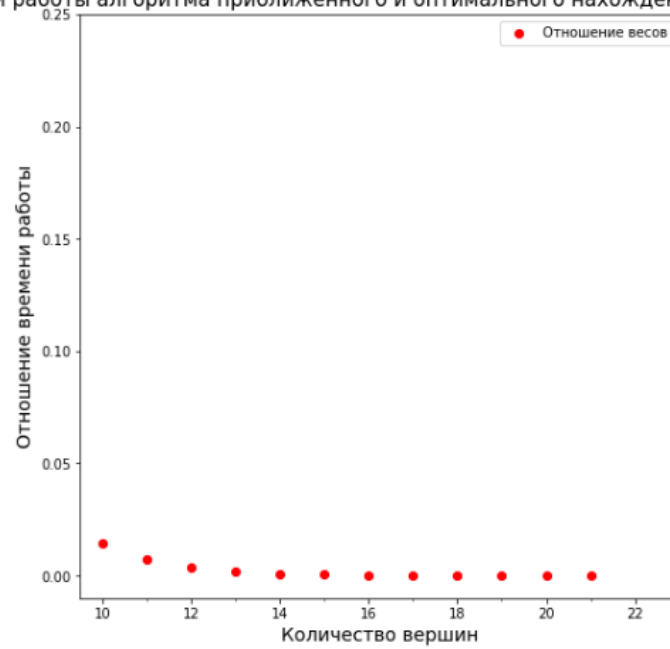


Рис. 6: График отношения времени работы алгоритма приближенного и оптимального нахождения покрытия для corner-case графа(другой масштаб)

График зависимости времени работы алгоритма от размера графа $G(n, p)$ для некоторых p

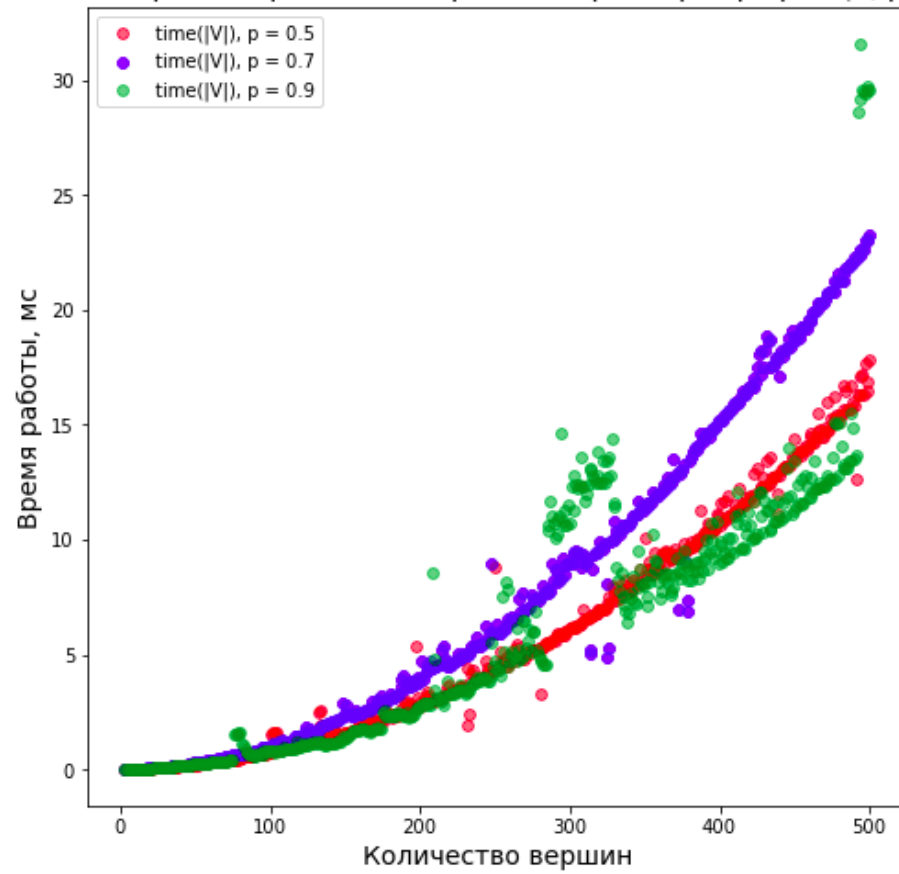


Рис. 7: График зависимости времени работы алгоритма от размера графа $G(n, p)$ для некоторых p