

Московский физико-технический институт  
Физтех-школа прикладной математики и информатики

# Д е р е в о   Ш т е й н е р а

Автор: [Евтушенко Иван \(2025\)](#)

*Проект в Overleaf*

*Проект на GitHub*

осень 2025

# Содержание

<b>1 Введение</b>	<b>2</b>
<b>2 Обзор существующей литературы</b>	<b>2</b>
<b>3 NP-полнота</b>	<b>2</b>
<b>4 Сведение к метрической версии</b>	<b>3</b>
4.1 Построение метрической версии . . . . .	3
4.2 Корректность метрической версии . . . . .	4
<b>5 Метрическая задача: 2-приближение</b>	<b>4</b>
5.1 Алгоритм . . . . .	4
5.2 Корректность и оценка приближения . . . . .	5
5.3 Реализация алгоритма . . . . .	5
5.4 Анализ алгоритма . . . . .	6
<b>6 Заключение</b>	<b>6</b>
<b>7 Источники</b>	<b>7</b>

## 1 Введение

Задача о нахождении дерева Штейнера является одной из базовых задач комбинаторной оптимизации. Она впервые была предложена Якобом Штайнером в контексте геометрии, в последствии была обобщена на графы. Задача заключается в поиске дерева минимального веса во взвешенном графе, соединяющего некоторое заранее известное подмножество вершин, быть может, через дополнительные вершины исходного графа.

Эта задача часто возникает при решении задач оптимизации или построения карт телекоммуникационных и транспортных сетей.

В настоящей работе было рассмотрено доказательство **NP**-полноты задачи и сведение её к метрической версии, представлено приближенное решение обоих. Проведён анализ на основе датасета I080.

## 2 Обзор существующей литературы

Задача поиска дерева Штейнера - есть одна из классических **NP**-полных задач, последнее было доказано Ричардом Карпом в 1972 году [1]. Задача не имеет точного алгоритмического решения за полиномиальное время, если только не  $P = NP$ , поэтому естественное желание найти приближенный алгоритм.

Первый такой был предложен Такахаси и Мацуямой в 1980 году [2]. После Бырковой и др. в 2010 году был перестроен и реализован алгоритм, использующий метод релаксации направленного покомпонентного разреза исходной задачи, что позволило получить аппроксимацию для restricted-варианта  $\ln(4)$ , что даёт  $\ln(4) + \varepsilon$  для исходной задачи.

Существует также жадный алгоритм, который был предложен Чи-Йен Чен в 2018 году [3], который гарантирует решение с константой не хуже, чем 1.4295.

Кроме того, в открытом доступе были обнаружены датасеты исходных графов для поиска деревьев Штейнера [4], на которых был протестирован простейший алгоритм, дающий 2-приближение.

## 3 NP-полнота

Докажем, что задача поиска дерева Штейнера (**STEINERTREE**) является **NP**-полнотой, используя сведение из задачи о нахождении вершинного покрытия.

**Определение 3.1.** **Задача STEINERTREE:** дан взвешенный граф  $G' = (V', E')$ , множество терминалов  $V'_0 \subset V'$  и число  $K$ . Также имеются веса на рёбрах  $w' : E' \rightarrow \mathbb{R}^+$ . Требуется определить, существует ли дерево, содержащее все вершины  $V'_0$ , веса не больше  $K$ .

**Замечание.**  $STEINERTREE \in NP$ : по предъявленному дереву можно за полиномиальное время проверить, что оно связно, содержит все терминалы и его суммарный вес  $\leq K$ .

**Определение 3.2.** **Задача VERTEXCOVER:** дан граф  $G = (V, E)$  и  $k$  - количество вершин. Необходимо определить, существует ли в  $G$   $k$  вершин, которые инцидентны всем рёбрам  $E$ .

**Решение 3.1.** Построим  $f : G \rightarrow G'$  явно: для любой вершины  $v$  из  $V$  создадим нетерминальную вершину  $v' \in G'$ ; для каждого ребра  $e \in G$  создадим терминальную вершину  $v'_e \in G'$  и соединим её рёбрами  $g'_{e1}$  и  $g'_{e2}$  веса 1 с образами инцидентных  $e$  вершин. Кроме

того, в  $G'$  создадим нетерминалы вершину  $n$  и соединим её ребрами  $g'_n$  веса 1 со всеми нетерминальными вершинами  $G''$ .

**Утверждение 3.1.** Для графа  $f(G)$  (где  $G$  имеет вершинное покрытие мощности  $k$ ) можно построить дерево Штейнера весом не больше, чем  $\omega'(E') + k$ .

*Доказательство.*

1. Построим подграф  $T'$  и покажем, что это дерево Штейнера: рассмотрим множество  $X$ , явл. верш. покрытием  $G$ , для каждой вершины  $x \in X$  покрасим в красный её образ. Для каждой терминальной вершины  $v'_e$  возьмём одно ребро, соед. её с красной вершиной, пусть оно приходит в вершину  $u$  (возможно в силу существования вершинного покрытия  $X$ ). Так же для каждой такой вершины  $u$  возьмём ребро, соединяющее её с вершиной  $n$ . Взяв корнем дерева вершину  $n$ , получим, что его глубина ровно 3 (вырожденный граф  $G$  с нулём ребер тривиален), т.к. терминальная вершина может быть соединена только с вершиной типа  $v'$ , а те, в свою очередь - только с вершиной  $n$ . Получили, что все терминалы доступны + нет циклов.
2. По построению  $T'$  очевидно, что  $\omega'(T') \leq \omega'(E') + k$

□

**Утверждение 3.2.** Если в графе  $f(G)$ , существует дерево Штейнера  $T'$ , т.ч.  $|T'| \leq |E| + k$ , то мы умеем явно предъявлять вершинное покрытие  $G$  на  $k$  вершинах

*Доказательство.*

1. Найдём вершинное покрытие исходного графа: рассмотрим множество вершин  $T'$ , соединённых с терминалами. По построению они соответствуют вершинам графа  $G$  и все терминалы покрыты → они получены из вершин, на которых можно выбрать вершинное покрытие, пусть  $X$
2. Мы взяли все терминалы, т.е.  $|T'| \geq |E|$ , но при этом  $|T'| \leq |E| + k$ , т.е. оставшиеся связи с  $n$  дополнили  $|T'|$  не строго меньше, чем на  $k \rightarrow |X| \leq k$

□

## 4 Сведение к метрической версии

### 4.1 Построение метрической версии

Исходно имеем взвешенный граф  $G = (V, E)$ , с весовой функцией  $\omega : E \rightarrow \mathbb{R}^+$ ,  $\{v \in V_0\}$  - терминальные вершины.

**Определение 4.1.** Получим из него метрический граф: пусть  $G' = (V', E')$  - полный граф на множестве вершин  $V$  ( $V' := V$ ), введём на нём следующую весовую функцию  $d : E' \rightarrow \mathbb{R}^+$ , т.ч.

$$d(u, v) = \min_{l: u \rightarrow v} \left( \sum_{e \in l} (\omega(e)) \right),$$

иначе говоря, вес каждого ребра полного графа приравнивается к минимальному весу пути, соединяющему вершины в исходном графе.

**Утверждение 4.1.**  $d$  – есть метрика.

*Доказательство.*  $d$  удовлетворяет неравенству треугольника, остальное очевидно.  $\square$

**Утверждение 4.2.** Построение взвешенного графа возможно за полином. время от кол-ва вершин.

*Доказательство.* Единственный сложный шаг построения - вычисление всех кратчайших путей. Это возможно сделать, запустив алг. дейкстры от каждой вершины.  $\square$

## 4.2 Корректность метрической версии

Покажем явно, что полученный граф эквивалентен предыдущему с точки зрения веса минимального дерева Штейнера:

*Доказательство.*

( $\Rightarrow$ ) Пусть  $T$  - д.Ш. (здесь и далее "дерево Штейнера") в  $G$ , рассмотрим его образ при построении: рассмотрим подграф  $T'$  в  $G'$ , получим его сохранив каждое ребро из  $T$  ( $d(u,v) \leq \omega(u,v)$ ), т.о. мин. вес д.Ш. в  $G'$  не больше  $\omega(T)$ , т.е. веса мин. д.Ш. в  $G$ .

( $\Leftarrow$ ) Рассмотрим мин. д.Ш в  $G'$ , пусть это  $T'$ , каждое его ребро "развернём" в исходном графе, получим  $T_r \subset G$ .  $T_r$  содержит все терминалы по построению, но в нём могли появиться циклы. Удалим их ( $T_r \rightarrow T$ ) и получим, что  $\omega(T) \leq \omega(T')$ .  $\square$

## 5 Метрическая задача: 2-приближение

### 5.1 Алгоритм

**Решение 5.1.**

**Входные данные:** полный метрически-взвешенный граф  $G = (V, E)$  (взвешенный граф  $G = (V, E)$  в общем случае), вес. функция  $\omega : E \rightarrow \mathbb{R}^+$ ,  $V_0 \subset V$  - подмножество терминальных вершин.

**Выходные данные:** Дерево, содержащее все терминалы  $V_0$ .

**Операции:**

0. Если граф не является полным метрически-взвешенным, привести его к соответствующему виду по алгоритму выше.
1. Рассмотреть подграф на вершинах  $V_0$ .
2. Вычислить по нему MST - минимальное оствовное дерево.
3. Вернуть MST.

## 5.2 Корректность и оценка приближения

**Утверждение 5.1.** Пусть  $T^*$  – минимальное дерево Штейнера в метрическом графе. Тогда дерево, возвращаемое алгоритмом, имеет вес не больше чем  $2\omega(T^*)$ .

*Доказательство.* Рассмотрим дерево  $T^*$  и выполним по нему обход, не проходящий по одному ребру больше двух раз, начиная с произвольного терминала (это возможно, т.к. если удвоить все рёбра, то можно выделить эйлеров путь). Пусть полученный путь равен  $l$ , тогда

$$\omega(l) \leq 2\omega(T^*).$$

Сославшись на индуцированную метрику допустимо перейти к рассмотрению  $l$  только как набора прямых ребер между терм. вершинами, пусть на них индуцировано д.Ш.  $T'$ .  $T'$  имеет вес равный  $\omega(\text{MST}(V_0))$ , что есть не более чем  $\omega(l)$ , т.е.  $\leq 2\omega(T^*)$ .  $\square$

## 5.3 Реализация алгоритма

Реализация 2-приближения для метрического графа, работающая за  $O(|V_0|^2 \log(|V_0|))$ :

```

1 import networkx as nx
2
3 def steiner_metric(G_m, V0):
4     return nx.minimum_spanning_tree(G_m.subgraph(list(V0)))

```

Дополнение к предыдущей реализации для решения задачи, поставленной на классическом графе, работающее за  $O(|V_0|^2 \log(|V_0|) + |V_0| \cdot |E| \log(|V|))$ :

```

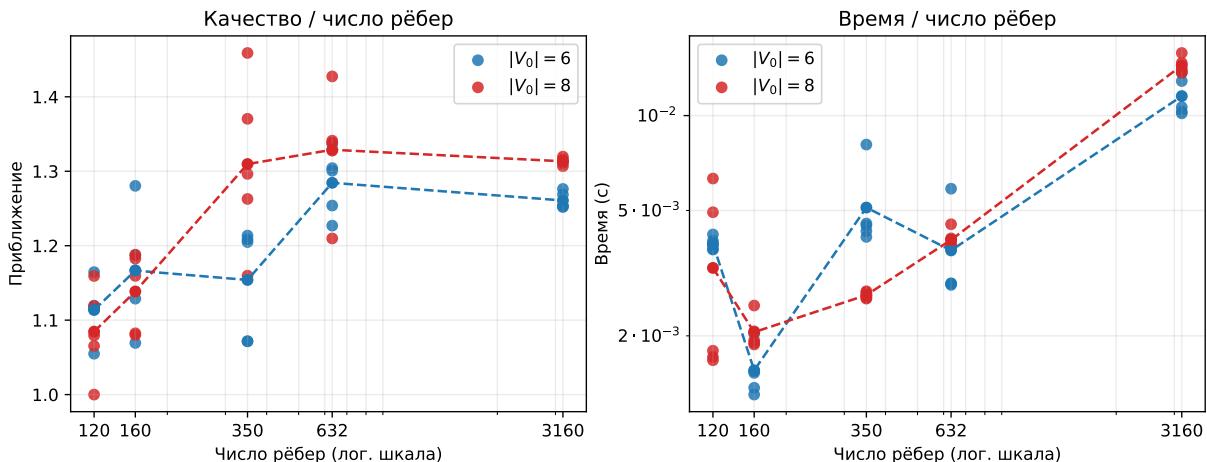
1 def metric_closure(G, V0):
2     R = list(V0)
3     K = nx.Graph()
4     K.add_nodes_from(R)
5     P = {}
6     for i, s in enumerate(R):
7         d, path = nx.single_source_dijkstra(G, s)
8         for t in R[i+1:]:
9             K.add_edge(s, t, weight=d[t])
10            P[s, t] = path[t]
11            P[t, s] = path[t][::-1]
12    return K, P
13
14 def steiner_common(G, V0):
15     K, P = metric_closure(G, V0)
16     Tt = steiner_metric(K, V0)
17     H = nx.Graph()
18     for u, v in Tt.edges():
19         p = P[u, v]
20         for a, b in zip(p, p[1:]):
21             H.add_edge(a, b, weight=G[a][b].get("weight", 1))
22     return nx.minimum_spanning_tree(H)

```

## 5.4 Анализ алгоритма

Для тестирования и анализа алгоритма был использован классический датасет, а точнее, его фрагмент I080 [4]. Наибольший интерес представляют зависимости качества приближения и время работы от количества вершин и рёбер в графе. Протестировать на большом количестве вершин возможности нет (задача NP-полна), поэтому ограничимся случаями для 6 и 8 терминальных вершин.

Результаты:



Выводы на основе результатов:

1. На малом количестве рёбер результатам алгоритма свойственна высокая дисперсия. С увеличением количества рёбер последняя резко падает и аппроксимационный результат стремится к 1.3.
2. С увеличением количества нетерминальных вершин точность алгоритма снижается вместе с увеличением дисперсии.
3. Время работы алгоритма зависит от количества рёбер, что соответствует асимптотике алгоритма Дейкстры.

## 6 Заключение

В рамках проделанной работы была исследована задача нахождения дерева Штейнера и реализован алгоритм, дающий 2-приближение, который вопреки существованию усиленных версий, претендует на важность по праву простоты и быстроты работы. Рассмотренный фрагмент датасета предлагает разряженные графы, на них алгоритм дал приближение, не уступающее результатам более сильных алгоритмов.

## 7 Источники

### Список литературы

- [1] Karp, R. M. Reducibility among combinatorial problems. In Complexity of computer computations, pp. 85–103. Springer, Boston, MA, 1972. [https://link.springer.com/chapter/10.1007/978-1-4684-2001-2\\_9](https://link.springer.com/chapter/10.1007/978-1-4684-2001-2_9)
- [2] Byrka, J., Grandoni, F., Rothvoß, T., Sanità, L. An improved LP- based approximation for Steiner tree. In Proceedings of the forty-second ACM symposium on Theory of computing, pp. 583–592, 2010. <https://dl.acm.org/doi/10.1145/1806689.1806769>
- [3] Chi-Yeh Chen. An Efficient Approximation Algorithm for the Steiner Tree Problem. National Cheng Kung University, Taiwan, ROC. 2018. <https://arxiv.org/pdf/1709.03867.pdf>
- [4] Thorsten Koch. SteinLib Testsets. 2017. <https://steinlib.zib.de/testset.php>