

Сложность вычислений
"Дерево Штейнера"

Иванов Вячеслав, группа 699

2 декабря 2018 г.

Оглавление

1	Постановка задачи	3
2	Практическая значимость	3
2.1	Построение филогенетических деревьев	3
2.2	Анализ сетей биологических взаимодействий	4
2.3	Конструирование интегральных схем	4
3	Доказательство NP-полноты задачи	4
4	Сведение к метрическому случаю	5
5	2-оптимальный алгоритм	6
6	Реализация	7
6.1	Асимптотика работы алгоритма и ограничения по его применению	7
6.2	Использованные технологии	8
6.3	Проверка корректности	8
7	Список литературы	8

1 Постановка задачи

$G = (V, E)$ — неориентированный граф, $V_0 \subset V$ — непустое множество *терминальных* вершин $w : E \rightarrow \mathbb{R}^+$ — весовая функция. Требуется решить задачу оптимизации:

$$\begin{aligned} \min_{T \subset G} \quad & \sum_{e \in E(T)} w(e) \\ \text{s.t.} \quad & T \text{ — дерево} \\ & V_0 \subset V(T) \end{aligned}$$

Т.е. найти дерево минимального веса, покрывающее все терминальные вершины.

В нетривиальных частных случаях задача имеет полиномиальный алгоритм решения:

1. $|V_0| = 2$: задача о кратчайшем пути между выделенными вершинами
2. $|V_0| = |V|$: задача о минимальном остовном дереве

Алгоритмы поиска минимального остовного дерева, как будет показано далее, составляют основу 2-оптимального алгоритма поиска дерева Штейнера в метрическом случае.

2 Практическая значимость

Деревья Штейнера возникают в таком количестве практических задач, что на эту тему пишут [целые книги](#). Автору особенно приглянулись следующие примеры:

2.1 Построение филогенетических деревьев

Определение 2.1. *Филогенетическое дерево* — корневое дерево, отражающее эволюционные связи и степень сходства между организмами. Построение таких деревьев — частая задача эволюционной биоинформатики.

Определение 2.2. *Расстояние Левенштейна* или *редакторское расстояние* $L(s_1, s_2)$ — минимальная стоимость получения строки s_2 из строки s_1 применением операций вставки, удаления и замены символов друг на друга, каждая из которых имеет некоторую стоимость.

Определение 2.3. *Расстояние Хэмминга* $H(s_1, s_2)$ определено для пар строк одинаковой длины и выражается как число позиций, в которых они различаются:

$$H(s_1, s_2) = \sum_{i=1}^{|s_1|} \mathbf{I}(s_1^{(i)} \neq s_2^{(i)})$$

Phylogenetic Tree of Life

★ = You are here

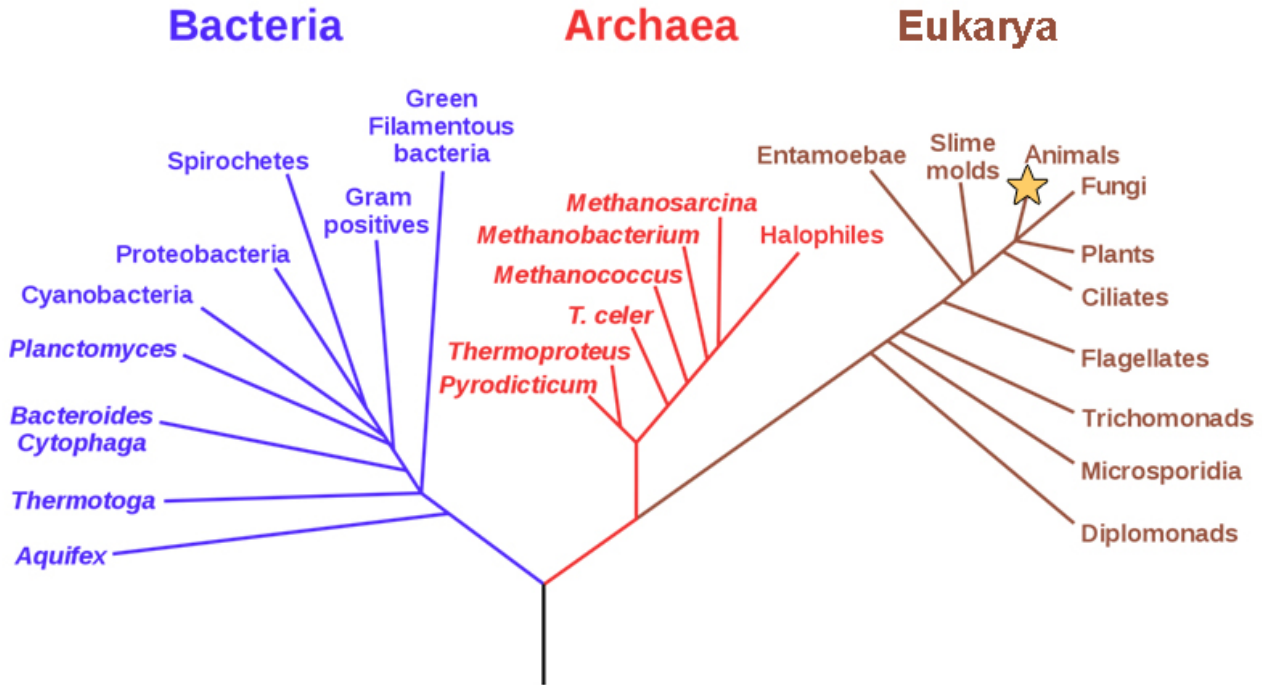


Рис. 1: Теоретически укоренённое филогенетическое дерево, показывающее общее происхождение организмов из всех трёх доменов: Бактерии, Археи, Эукариоты.

2.2 Анализ сетей биологических взаимодействий

2.3 Конструирование интегральных схем

3 Доказательство NP-полноты задачи

Теорема 3.1.

$\{(G, k) \mid \text{в неориентированном графе } G \text{ есть дерево Штейнера веса } \leq k \in \mathbb{Z}\} \in \text{NPC}$

Доказательство.

1. **STEINER-TREE** \in **NP**: Сертификат должен проверять, что поданный ему на вход подграф T является деревом, содержит все терминальные вершины и имеет вес $\leq k$, причём вторая и третья подзадачи тривиальны. Согласно одному из эквивалентных определений дерева, достаточно проверить связность T и то, что $|E(T)| = |V(T)| - 1$, для чего достаточно обхода в глубину. Т.е. полиномиальный сертификат существует и **STEINER-TREE** \in **NP**.
2. **VERTEX-COVER** \leq_p **STEINER-TREE**: Полиномиальное сведение устроено так:

(а) Дополним G до $K_{|V|}$, где каждое ребро поделим на 2 и назовём результат $G' = (V', E')$.

$$W := \{w_i \mid e_i = (u, v) \in E \implies (u, w_i), (w_i, v) \in E'\}$$

$$V' = V \cup W$$

$$E' = V^2 \cup \{(u, w) \in V \times W \mid \exists v \in V : (u, w), (w, v) \in E'\}$$

- (b) Если $\forall e' \in E' : w'(e') = 1$ и $V'_0 = W$, то в G' есть дерево Штейнера веса не более $|E| + k - 1 \iff$ в G есть вершинное покрытие мощности $\leq k$.

Доказательство.

- \implies : Пусть T — дерево Штейнера для V'_0 в G' , тогда $C := V(T) \setminus V'_0$ — вершинное покрытие в G : $V(T) \setminus V'_0 \subset V$ и покрывает каждое ребро $e \in E$ по построению. $|C| = |V(T)| - |V'_0| = \omega(T) + 1 - |V'_0| \leq (|E| + k - 1) + 1 - |V'_0| = k$, т.к. $|E| = |V'_0|$.
- \impliedby : Пусть $C \subset V$ — вершинное покрытие, $|C| \leq k$, T — дерево на вершинах C в G' .¹ Чтобы гарантировать $V'_0 \subset V(T)$, для каждой вершины $v'_0 \in V'_0 \setminus V(T)$ добавим в T ребро (v'_0, c) , где $c \in C$ — вершина покрытия, накрывающая ребро, подразделением которого получена v'_0 . Полученный граф содержит $\leq |E| + |C| - 1 \leq |E| + k - 1$ рёбер. Если в процессе расширения в T образовались циклы, их можно раскрыть, уменьшив суммарный вес. По завершении получим дерево Штейнера веса $\leq |E| + k - 1$ в G' .

□

Все шаги построения G' полиномиальны: добавляется $O(|V|^2)$ рёбер и вершин. Для восстановления вершинного покрытия по дереву Штейнера в G' нужно брать разность множеств — $O(|V|^2)$ операций, а в обратную сторону нужно перебрать V'_0 и исходящие из него рёбра (не более двух на каждую вершину) — тоже $O(|V|^2)$ операций. Следовательно, сведение полиномиально, а его корректность была доказана выше.

□

4 Сведение к метрическому случаю

Часто хочется потребовать, чтобы для весовой функции выполнялось правило треугольника:

$$\omega(x, y) \leq \omega(x, z) + \omega(z, y)$$

причём она должна быть определена на V^2 . Такая весовая функция играет роль метрики на множестве вершин и становится проще для восприятия. В таком случае говорят о поиске *метрического* дерева Штейнера, и именно такой вид задачи был исторически первым.

Утверждение 4.1.

1. Существует полиномиальное сведение задачи о дереве Штейнера к метрическому случаю.
2. Оптимальные ответы к обоим задачам совпадают.

Доказательство. Предложенная конструкция основана на понятии *метрического замыкания*:

Определение 4.1. Пусть $G = (V, E, \omega)$ — неориентированный взвешенный граф, $d : V^2 \rightarrow \mathbb{R}^+$ — функция расстояния, сопоставляющая паре вершин длину кратчайшего пути между ними. Тогда граф G' , построенный по следующим правилам, называется *метрическим замыканием* графа G :

$$G' = (V, E', d), \quad E' = \{(u, v, d(u, v)) \mid u, v \in V, u \neq v\}$$

Полученный граф является метрическим, т.к. для d выполнено правило треугольника: если $d(x, y) > d(x, z) + d(z, y)$, то путь $x \rightarrow y$ можно было бы прорелаксировать конкатенацией путей $x \rightarrow z$ и $z \rightarrow y$, что противоречит определению $d(x, y)$ как длины *кратчайшего* пути $x \rightarrow y$.

Построить метрическое замыкание можно за $O(|V|^3)$ алгоритмом Флойда-Уоршелла.

¹ Такое дерево всегда существует, т.к. $V^2 \subset V(G')$.

Пусть T, T' — деревья Штейнера для V_0 в G и G' соответственно. Докажем, что $\omega(T) = d(T')$. Очевидно, что $d(T') \leq \omega(T)$, т.к. переход к кратчайшим путям не ухудшает ответ. Более того, каждое ребро в T' можно "разжать" в тот кратчайший путь, из которого он был получен, после чего выбрать в полученном графе минимальное остовное дерево T'' . $\omega(T'') \leq d(T')$, т.к. теперь каждое ребро встречается ровно один раз, а ранее могло вносить свой вклад одновременно в несколько кратчайших путей. Но T'' по построению — дерево Штейнера для V_0 в G ! Следовательно, $\omega(T) = \omega(T'') \leq d(T') \leq \omega(T)$, т.е. $\omega(T) = d(T')$. \square

5 2-оптимальный алгоритм

Теорема 5.1.

Существует 2-оптимальный алгоритм для метрического случая задачи поиска дерева Штейнера:

1. Считать граф $G = (V, E, \omega)$ и множество терминальных вершин V_0 .
2. Построить метрическое замыкание $G' = (V, E', d)$ графа G .
3. Выделить H' — подграф в G' , индуцированный вершинами V_0 .
4. Построить минимальное остовное дерево T_{MST} в H' .
5. Вернуть полученный T_{MST} в качестве ответа.

Доказательство. Пусть T_S — дерево Штейнера для V_0 в G' . Вершины T_S в порядке обхода в глубину

$$u_0, u_1, \dots, u_m = u_0$$

задают Эйлеров цикл, потому:

$$\sum_{i=0}^{m-1} d(u_i, u_{i+1}) = 2 \cdot d(T_S)$$

Если исключить из рассмотрения все нетерминальные вершины и оставить только первое вхождение всех терминальных, то получим путь:

$$v_0, v_1, \dots, v_k$$

содержащий все терминальные вершины (т.к. изначально это был обход G'). По неравенству треугольника:

$$d(v_i, v_{i+1}) \leq \sum_{j=1}^t d(u_{i_j}, u_{i_{j+1}})$$

Здесь $v_{i_j}, \dots, v_{i_{t+1}}$ — последовательные вершины в обходе в глубину, причём $v_i = u_{i_1}, u_{i+1} = v_{i_{t+1}}$. Поскольку u_1, \dots, u_k — путь, это также дерево, причём неравенство треугольника гарантирует, что добавление любой вершины только увеличивает его вес. Более того, в силу того, как задана весовая функция, это ещё и минимальное остовное дерево в H' . Следовательно, это дерево Штейнера T_{MST} для V_0 в G' , причём:

$$\begin{aligned} d(T_{\text{MST}}) &\leq \sum_{i=1}^n d(u_i, u_{i+1}) = 2 \cdot d(T_S) \\ 1 &\leq \frac{d(T_{\text{MST}})}{d(T_S)} \leq 2 \end{aligned}$$

По утверждению 3.1., по нему восстанавливается дерево Штейнера в G . \square

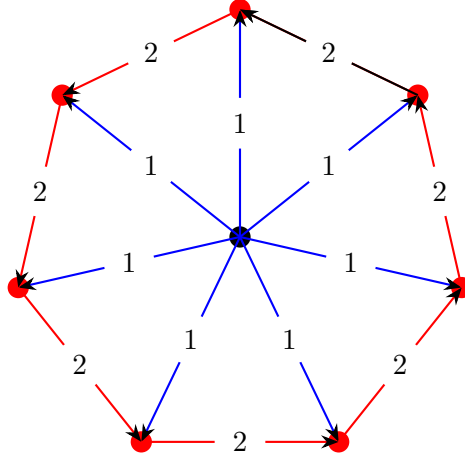


Рис. 2: Пример графа, на котором оценка достигается (в пределе). Терминальные вершины выделены красным. Реальное дерево Штейнера T_S выделено синим, а T_{MST} , построенное алгоритмом, — красным. Если внешний цикл имеет вид n -угольника, то верно, что $\frac{d(T_{\text{MST}})}{d(T_S)} = \frac{2n-2}{n} = 2 - \frac{2}{n} \rightarrow 2$.

6 Реализация

6.1 Асимптотика работы алгоритма и ограничения по его применению

1. Алгоритм Флойда-Уоршалла поиска кратчайших путей между всеми парами вершин: $\Theta(|V|^3)$ операций, $\Theta(|V|^2)$ памяти на хранение матрицы расстояний.
2. Алгоритм Краскала поиска минимального остовного дерева с помощью системы непересекающихся множеств: $O(|E| \log |V|)$ операций, $\Theta(|V|)$ памяти.

Т.к. метрическое замыкание делает граф полным, итогом имеем $\Theta(|V|^3)$ операций и $\Theta(|V|^2)$ памяти. Следовательно, алгоритм оправданно применять при $|V| \leq 1000$, что является существенным недостатком предложенного подхода.

Также стоит отметить, что алгоритмы Флойда-Уоршалла и Краскала по своей природе последовательные, в связи с чем воспользоваться преимуществами вычислительного кластера МФТИ, к которому автор имеет авторизованный доступ, не представляется возможным при таком подходе.

Более того, алгоритм Флойда-Уоршалла хорош тем, что не совершает больших прыжков по памяти при хранении матрицы кратчайших расстояний в виде последовательно расположенной в памяти двумерной таблицы, что уменьшает число промахов по кэшу процессора и оказывает положительный эффект на скорость работы алгоритма по сравнению с параллельными запусками алгоритмов поиска кратчайших путей из данной вершины во все остальные (в духе алгоритма Дейкстры) при малых $|V|$.

Тем не менее, при $|V| \geq 10^3$ не лишено смысла хранить исходную матрицу расстояний в базе данных (на жёстком диске или распределённо) и действительно выполнять параллельные запуски, скажем, алгоритма Дейкстры на доступных процессорах для параллельного подсчёта матрицы кратчайших расстояний. Эта неасимптотическая оптимизация делает возможной обработку существенно больших графов, чем было заявлено изначально, т.к. это типичная map-reduce задача, что с точки зрения автора является довольно дешёвой в реализации, но полезной идеей, пусть и выходящей за рамки требований поставленной учебной задачи.

6.2 Используемые технологии

Для работы с графами в Python была использована библиотека [graph-tool](#). По опыту автора, альтернативы в лице [networkx](#) и [igraph](#) имеют слишком много нетривиальных недостатков: [networkx](#) написана на Pure Python, из-за чего производительность становится неприемлемой уже на маленьких графах, а разработка Python-API для написанной на C библиотеки [igraph](#) уже больше года как приостановлена, а по состоянию на данный момент пользоваться ей очень уж неудобно по причинам, расписывать которые слишком долго для того, чтобы приводить их здесь. R-API гораздо лучше, но Python для меня предпочтительнее.

6.3 Проверка корректности

Тесты разделены на несколько групп:

1. Маленькие ($|V| \leq 20$) случайные графы в модели Эрдеша-Рейни, реальное дерево Штейнера в которых ищется полным перебором (возможно, с некоторыми эвристиками).
2. В качестве примера реальной задачи рассматривается поиск дерева Штейнера на географической карте города, где некоторые строения помечены и играют роль терминальных вершин. Данные взяты из [Принстонской статьи](#), но вынужденно ограничены до 1000 точек на карте в силу кубической асимптотики алгоритма.

7 Список литературы