Метрическая задача коммивояжёра

Максимова Анна Михайловна

13 января 2023 г.

Аннотация

Задача коммивояжёра (TSP) — одна из самых известных задач комбинаторной оптимизации, заключающаяся в поиске самого выгодного маршрута, проходящего через указанные города хотя бы по одному разу с последующим возвратом в исходный город. В данной работе будет рассмотрен частный случай TSP, называемый метрическим TSP. Метрический TSP - это подкласс TSP, в котором выполняется неравенство треугольника на ребрах (ETSP).

В ходе нашей работы будет доказана NP-полнота стандартной TSP и приведены 2 алгоритма аппроксимации. Первый - простой алгоритм с фактором аппроксимации 2, объяснение которого изложено в лекциях Tufts University, Spring 2002. И второй алгоритм, который дает 1.5 -приближение для ETSP. Этот алгоритм, являющийся главной частью нашей работы, также известен как алгоритм Кристофидеса и назван именем Никоса Кристофидеса и Анатолия Ивановича Сердюкова, которые независимо друг от друга нашли его в 1976, также он обладает лучшим аппроксимационным коэффициентом на данный момент. Во второй части проекта будет приложена реализация Алгоритма Кристофидеса.

1 План

1. Введение
2. Необходимые определения
3. Для стандартной задачи коммивояжёра не существует константных алгоритмов приближения, если $\mathbf{P} \neq \mathbf{NP}$
4. Аппроксимационные алгоритмы для ETSP
   (a) 2-аппроксимационный алгоритм
   (b) Алгоритм Кристофидеса
5. Имплементация Алгоритма Кристофидеса
6. Литература

2 Введение

В задаче коммивояжера рассматривается городов и матрица попарных расстояний между ними. Требуется найти такой порядок посещения городов, чтобы суммарное пройденное расстояние было минимальным, каждый город посещался ровно один раз и коммивояжер вернулся в тот город, с которого начал свой маршрут. Другими словами, во взвешенном полном графе требуется найти гамильтонов цикл минимального веса.
Вместе с простотой определения и сравнительной простотой нахождения хороших решений задача коммивояжёра отличается тем, что нахождение действительно оптимального пути является достаточно сложной задачей. Учитывая эти свойства, начиная со второй половины XX века исследование задачи коммивояжёра имеет не столько практический смысл, сколько теоретический в качестве модели для разработки новых алгоритмов оптимизации. Многие современные распространенные методы дискретной оптимизации, такие как метод отсечений, ветвей и границ и различные варианты эвристических алгоритмов, были разработаны на примере задачи коммивояжёра.
Существует очень много приложений данной задачи в реальной жизни. Например, чтобы оптимальным образом строить маршруты для курьеров, которым нужно развести груз по точкам и вернуться обратно.

Формальная постановка задачи:
В общем случае задача коммивояжёра ставится так: дан граф $G = (V, E)$ с функцией весов на рёбрах $w : E \longrightarrow \mathbb{R}_+$. Требуется найти гамильтонов цикл минимального веса.
В метрическом случае граф полный, а функция весов метрическая (то есть для любых трёх вершин x, y и z выполнено $w(x, z) \leq w(x, y) + w(y, z)$).

3 Необходимые определения

Определение 1. *Детерминированной многоленточной машиной Търинга называется кортеж объектов* $\langle \Sigma, \mathit{\Gamma}, Q, q_{start}, q_{accept}, q_{reject}, k, \delta \ \rangle, \Sigma, \mathit{\Gamma}, Q$ *- непустые конечные множества,* $\Sigma \subset \mathit{\Gamma}$*,* k *- целое положительное число,* $q_{start}, q_{accept}, q_{reject}$ *- три различных элемента* Q*,* $\delta : (Q \backslash \{q_{accept}, q_{reject}\}) \times \mathit{\Gamma}^k \longrightarrow Q \times \mathit{\Gamma}^k \{L, N, R\}^k$*. В таком случае* Σ *называется входным алфавитом,* $\mathit{\Gamma}$ *- ленточным алфавитом,* Q *- множеством состояний,* q_{start} *- начальным состоянием,* q_{accept}, q_{reject} *- принимающим и отвергающим состоянием соответственно,* k *- числом лент,* δ *- функцией перехода*

Определение 2. *Язык* $L \subset \Sigma^*$ *распознается машиной Тьюринга* M*, если для каждого* $x \in \Sigma^*$ *вычисление* $M(x)$ *останавливается и* $M(x) = L(x)$*, где* $L(x)$ *- характеристическая функция языка* L*.*

Определение 3. *Язык* $L \subset \Sigma^*$ *распознается машиной Тьюринга* M *за время* $O(T(n))$ *(или просто* $T(n)$*), если* M *распознает язык* L*, а также существует глобальная константа* C*, такая что для каждого* $x \in \Sigma^*$ *вычисление* $M(x)$ *останавливается не более чем за* $C \cdot T(|x|)$ *шагов, то есть за* $O(T(n))$ *шагов, где* $n = |x|$*.*

Определение 4. *Если* $T(n) : \mathbb{N} \longrightarrow \mathbb{N}$ *- функция, то класс* $\mathbf{DTIME}(T(n))$ *состоит в точности из тех языков, которые распознаются хоть какой нибудь машиной Тьюринга за время* $T(n)$

Определение 5. $\mathbf{P} = \cup_{c=1}^{\infty} \mathbf{DTIME}(n^c)$

Определение 6. *Класс* $\mathbf{NP}$ *состоит в точности из тех языков* A*, для которых существует некая машина* V *двух аргументов, работающая за полином от длины первого из них, т.ч.* $\forall x \in \Sigma^*$ *выполнено* $x \in A \Leftrightarrow \exists s : V(x, s) = 1$*. В таком случае* V *называют верификатором, а подходящее* s *— сертификатом для данного* x

Определение 7. *Язык* B *называется* $\mathbf{NP}$*-трудным, если* $\forall A \in \mathbf{NP} : A \leq_p B$

Определение 8. *Язык* B *называется* $\mathbf{NP}$*-полным, если* $B \in \mathbf{NP}$ *и* B *является* $\mathbf{NP}$*-трудным.*

Определение 9. *Если граф имеет простой цикл, содержащий все вершины графа по одному разу, то такой цикл называется гамильтоновым циклом, а граф называется гамильтоновым графом.*

Определение 10. *Считается, что алгоритм – полиномиальный или имеет полиномиальную временную сложность, если существует полином* $p(x)$ *такой, что на любом входном слове длины* n *алгоритм завершает вычисления после выполнения не более чем* $p(n)$ *операций.*

Определение 11. *Полный граф — простой неориентированный граф, в котором каждая пара различных вершин смежна.*

4 Для стандартной задачи коммивояжёра не существует константных алгоритмов приближения, если $\mathbf{P} \neq \mathbf{NP}$.

Известно, что задача о гамильтоновости графа является **NP**-полной. В стандартной задаче коммивояжера требуется найти гамильтонов цикл минимальной длины. Это не задача распознавания. Она не лежит в классе NP, но она не проще проверки гамильтоновости графа. Действительно, если существует точный полиномиальный алгоритм для задачи коммивояжера, то можно легко построить точный полиномиальный алгоритм и для проверки гамильтоновости графа. Для этого достаточно по графу $G = (V, E)$, чью гамильтоновость мы исследуем, построить матрицу расстояний (c_{ij}) задачи коммивояжера по следующему правилу:

$$c_{ij} = \begin{cases} 1, \text{ если}(i,j) \in E \\ 2, \text{ если}(i,j) \notin E \end{cases}$$

Если решение задачи коммивояжера имеет ответ n, то граф G является гамильтоновым. Обратное тоже верно.
Фактически мы получили доказательство даже более сильного утверждения.

Теорема 1. *Задача коммивояжера является* **NP***-трудной даже, ко гда матрица (c_{ij}) является симметричной и состоит из 1 и 2.*

Полученное утверждение говорит о том, что точный полиномиальный алгоритм для задачи коммивояжера построить, скорее всего, не удастся. По-видимому, таких просто не существует. Следовательно, надо либо выйти за рамки полиномиальных алгоритмов, либо искать приближенные решения задачи. Следующая теорема говорит о том, что второй подход может оказаться не проще точного решения задачи. Пусть для примера I задачи коммивояжера величина $Opt(I)$ задает длину кратчайшего гамильтонова цикла, а некоторый алгоритм A на этом примере дает ответ $A(I)$

Теорема 2. *Если существует приближенный полиномиальный алгоритм A и такая положительная константа r ,что для любого примера I задачи коммивояжера справедливо неравенство*

$$A(I) \leq r \cdot Opt(I)$$

то классы **P** *и* **NP** *совпадают.*

Доказательство. Рассмотрим снова задачу о гамильтоновом цикле и построим матрицу (c_{ij}) по следующему правилу:

$$c_{ij} = \begin{cases} 1, \text{ если} (i,j) \in E \\ \text{nr}, \text{ если} (i,j) \notin E \end{cases}$$

Где $n = |V|$
Применим приближенный алгоритм A. Если $A(I) = n$, то граф содержит гамильтонов цикл.
Предположим, что $A(I) > n$ Тогда $A(I) \geq nr + (n-1)$, так как путь коммивояжера проходит как минимум через одно «тяжелое» ребро. Но в этом случае граф не может иметь гамильтонов цикл, так как алгоритм ошибается не более чем в r раз. Следовательно, полиномиальный алгоритм A дает правильный ответ для **NP**-полной задачи и $\mathbf{P} = \mathbf{NP}$. □

Из доказательства теоремы следует, что константу r можно заменить на любую, например, экспоненциально растущую функцию от n Повторяя рассуждения, получаем, что относительная точность полиномиальных алгоритмов в худшем случае не может быть ограничена, в частности, величиной $2^{p(n)}$, где $p(n)$ — произвольный полином от размерности задачи, если $\mathbf{P} \neq \mathbf{NP}$.
Заметим, что в отличие от доказательства теоремы 1, новая конструкция уже не гарантирует выполнение неравенства треугольника. Это наводит на мысль, что при дополнительных ограничениях на матрицу (c_{ij}) можно получить полиномиальные алгоритмы с гарантированной точностью. В следующих главах мы рассмотрим такие алгоритмы.

5 Аппроксимационные алгоритмы для ETSP.

2-аппроксимационный алгоритм

Приведем описание алгоритма, удовлетворяющего $A(I) \leq 2 \cdot Opt(I)$.
Для этого рассмотрим граф $G = (V, E)$ с функцией весов на рёбрах $w : E \longrightarrow \mathbb{R}_+$, G полный, а функция весов метрическая (то есть для любых трёх вершин x, y и z выполнено $w(x, z) \leq w(x, y) + w(y, z)$), изображенный на рисунке:

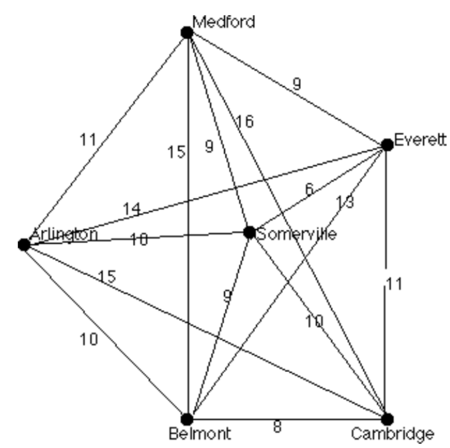

Рис. 1: Исходный граф ETSP

Построим минимальное остовное дерево (MST) на основе G. Остовным деревом графа называется дерево, которое можно получить из него путём удаления некоторых рёбер. Это можно сделать за полиномиальное время, используя, например, алгоритм Краскала или Прима.

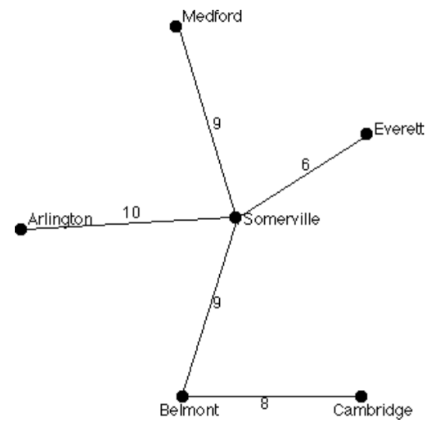

Рис. 2: Минимальное остовное дерево графа G

Лемма 1. *Вес* $MST \leq OPT$

Доказательство. От противного: Предположим по графу G построили MST так, что $OPT < MST$. Удалим одно ребро из гамильтонового графа OPT, тем самым сделав его ацикличным. Тогда получили связное дерево T с весами $T < OPT \Rightarrow T < OPT < MST$. Тогда MST - не минимальное остовное дерево. Противоречие. □

Следующим шагом запустим DFS (поиск в грубину, за полиномиальное время) по MST, проходя через каждое ребро дважды. Тогда получим некоторый «псевдопуть» PT. Стоимость $PT = 2 \cdot MST \leq 2 \cdot OPT$. Выпишем все вершины в порядке обхода.

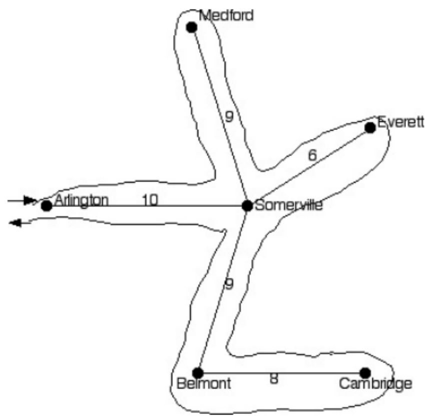

Рис. 3: DFS по графу MST. Получаем путь: ASMSESBCBSA

Далее перейдем от PT к P^*: Перепишем список вершин, полученный на предыдущем шаге, записывая каждую вершину только по одному разу. В нашем примере цикл в $PT =$ ASMSESBCBSA $\Rightarrow$ цикл в $P^* =$ ASMEBC

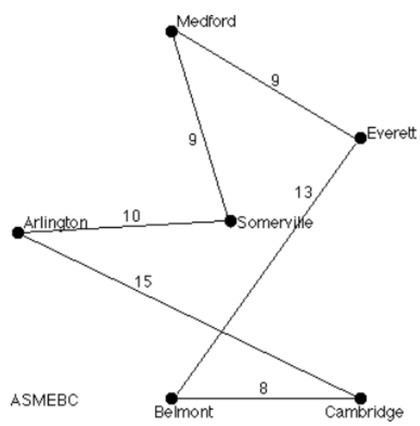

Рис. 4: граф P^*

Лемма 2. *Вес* $P^* \leq PT \Rightarrow P^* \leq 2 \cdot OPT$

Доказательство. Доказательство следует из того, что на ребрах выполняется неравенство треугольника. Значит путь из A в B должен быть короче, чем путь из A в C и затем в B $\square$

1.5-аппроксимационный алгоритм

Перед темкак перейдем к рассмотрению алгоритма, нужно указать два факта:
1) Паросочетание минимального веса в полном взвешенном графе может быть найдено за полиномиальное время (Венгерский алгоритм)
2) Граф имеет Эйлеров цикл тогда и только тогда, когда степени всех вершин четные (Эйлеров цикл посещает каждое ребро только по одному разу и содержит в себе все ребра графа)

Алгоритм Кристофидеса:

Задан граф $G = (V, E)$ с функцией весов на рёбрах $w : E \longrightarrow \mathbb{R}_+$, G полный, а функция весов метрическая.

1. Как и в прошлом алгоритме, построим за полиномиальное время минимальное остовное дерево (MST).

   Лемма 3. *В любом графе количество вершин нечетной степени должно быть четным.*

   Доказательство. Сумма степеней всех вершин в графе равна удвоенному числу ребер, следовательно, это четное число.
   $$\sum_{v \in V} deg(v) = 2 \cdot E$$
   Поскольку четное + нечетное число дают нечетное число, из этого следует, что сумма степеней вершин нечетной степени + сумма степеней вершин четной степени также должна быть четной.
   $$\sum_{v \in V} deg(v) = \sum_{v \in V_{odd}} deg(v) + \sum_{v \in V_{even}} deg(v)$$
   А так как сумма степеней вершин четной степени - четное число, значит количество вершин нечетной степени должно быть четным □

2. Найдем паросочетание минимального веса M^* в графе G между вершинами которые имеют нечетную степень в MST

Лемма 4. $M^* \leq 0.5 \cdot OPT$

Доказательство. Любой путь в графе можно разбить на два паросочетания M_1 и M_2, путем чередования ребер (то есть по очереди выбираем сначала в M_1, а потом в M_2)

$$OPT = M_1 + M_2 \geq M^* + M^*$$

Рассмотрим подграф $MST + M^*$: Каждая вершина теперь в этом подграфе имеет четную степень (потому что к нечетным вершинам в MST добавили еще по одному ребру). Поэтому в этом подграфе существует Эйлеров путь E. Его стоимость равна в точности стоимости всех ребер в графе $MST + M^*$, потому что в эйлеровом пути используется каждое ребро ровно один раз. Имеем:

$$MST + M^* \leq 0.5 \cdot OPT + OPT \leq 1.5 \cdot OPT$$

Запишем каждую вершину, которая встречается впервые в эйлеровом пути E и получим путь E^* со стоимостью $\leq 1.5 \cdot OPT$, который и будет является ответом. □

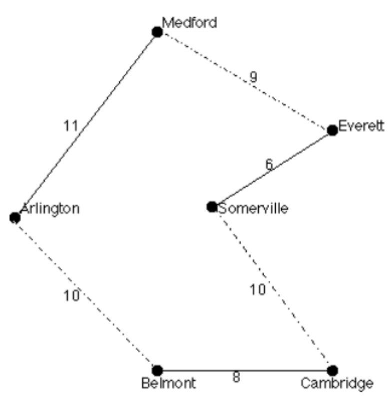

Рис. 5: Путь E^*

6 Имплементация Алгоритма Кристофидеса

https://github.com/anyatamax/Euclid_TSP_Problem

7 Литература

Список литературы

[1] Advanced Algorithms. Professor Lenore Cowen. Tufts University, Spring 2002. Lecture 3

[2] Глава 4. Задача коммивояжера. URL: http://old.math.nsc.ru/LBRT/k5/OR-MMF/TSPr.pdf

[3] Minimum-Cost-Perfect-Matching. Author github: dilsonpereira. URL: https://github.com/dilsonpereira/Minimum-Cost-Perfect-Matching