Алгоритмы для задачи коммивояжёра

Виктор Васильевич Лепин

Формулировка задачи

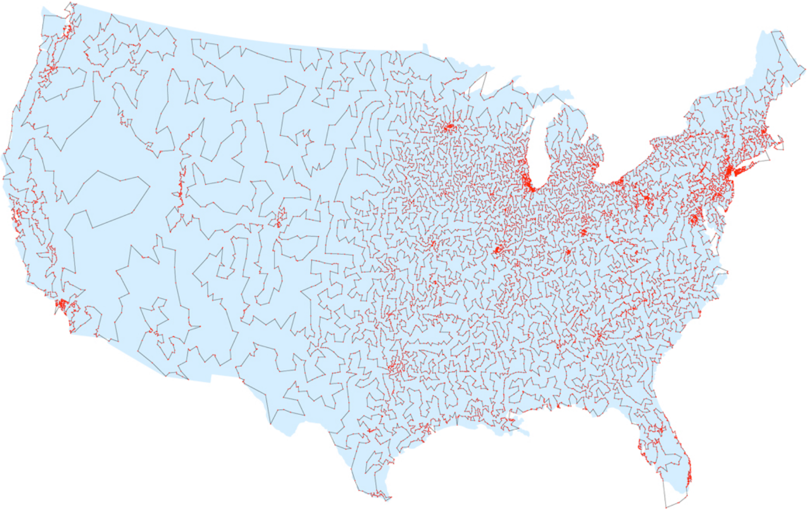
- Задача о гамильтоновом цикле: проверить, есть ли в графе цикл, проходящий по каждой вершине ровно один раз.
- Задача коммивояжёра: найти в данном полном взвешенном графе гамильтонов цикл минимального веса.
- Периодически мы будем искать не цикл, а путь.
- Применения: проектирование схем, планирование, сборка генома.
- Сложность полного перебора: $O(n!)$.

Цикл по 15 городам Германии

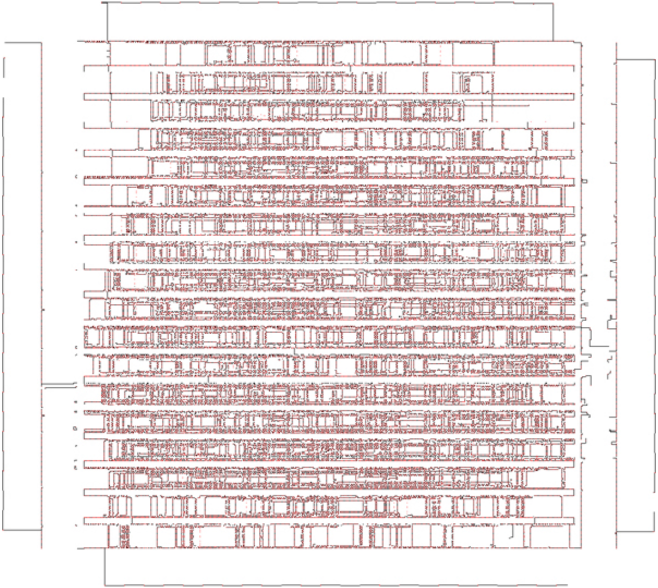
Оптимальный маршрут коммивояжёра через 15 крупнейших городов Германии. Указанный маршрут является самым коротким из всех возможных 43 589 145 600.

http://en.wikipedia.org/wiki/Travelling_salesman_problem

Цикл по 13 509 городам США



Оптимальный путь лазера 85 900 «городов»



Метод ветвей и границ

Начать с некоторой задачи P_0
$S = \{P_0\} \leftarrow$ множество активных подзадач
лучшийрезультат $= \infty$
while S не пусто
  do <u>выбрать</u> подзадачу (частичное решение) $P \in S$
    и удалить её из S
   <u>разбить</u> P на меньшие подзадачи $P_1, P_2, \cdots, P_k$
   for каждой P_i
    do if P_i является полным решением
      then обновить лучшийрезультат
     elseif <u>нижняяграница</u>$(P_i) <$ лучшийрезультат
      then добавить P_i в S
return лучшийрезультат

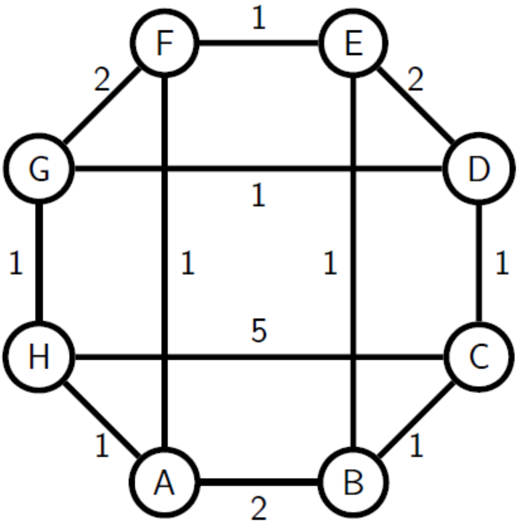
Подзадачи и нижняя граница

- подзадача: $[a, S, b]$ — достроение простого пути из a в b, проходящего по всем вершинам из $S \ni a, b$ (то есть кратчайший путь из b в a, проходящий по $V \setminus S$)
- начальная задача: $[a, a, a]$
- нижняя граница — сумма из
  - самого лёгкого ребра из a в $V \setminus S$,
  - самого лёгкого ребра из b в $V \setminus S$ и
  - минимального покрывающего дерева графа на вершинах $V \setminus S$.

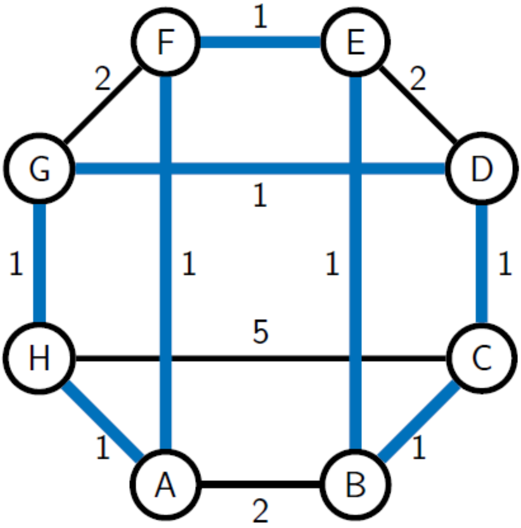
Кстати, о минимальных покрывающих деревьях

- Задача о минимальном покрывающем дереве — оставить в графе $(n-1)$ ребро, так чтобы граф остался связным и чтобы суммарный вес был минимальным. Решается почти за линейное время.
- Задача о минимальном пути коммивояжёра — то же самое, но запрещаем вершины степени больше двух. До сих пор не умеем решать быстрее 2^n.

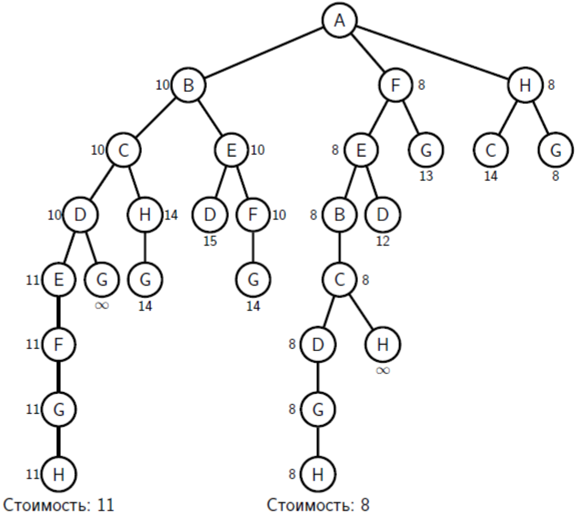
Пример графа



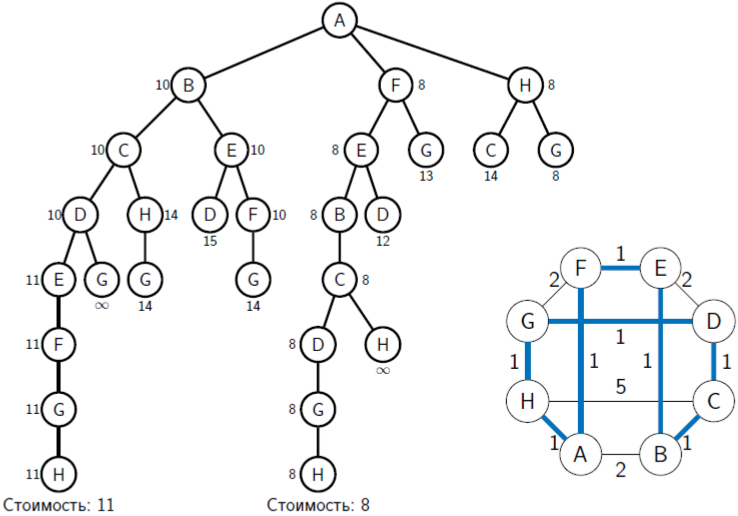
Пример графа



Дерево поиска

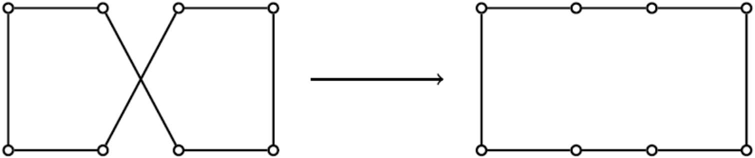


Дерево поиска

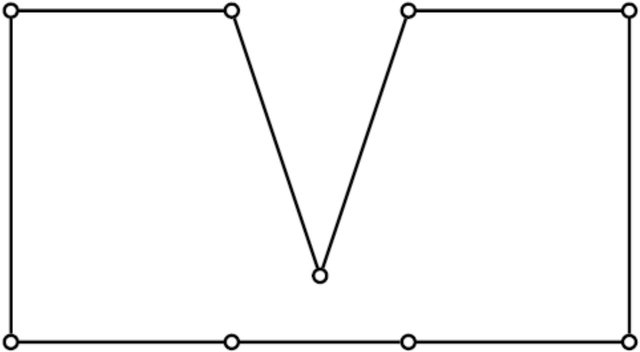


1. $s \leftarrow$ какое-нибудь начальное решение
2. **while** в окрестности s есть решение s' большей стоимости
3. **do** заменить s на s'
4. **return** s

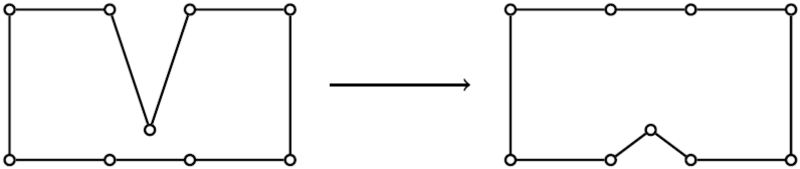
2-окружение



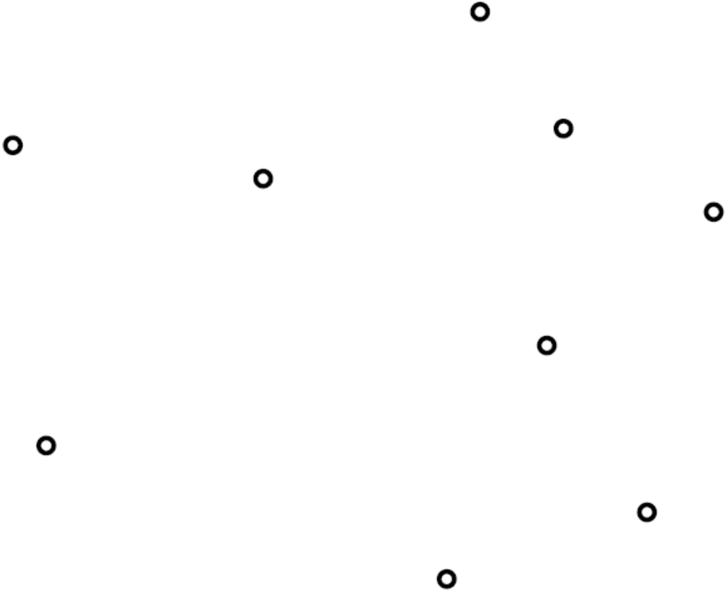
Узкое место



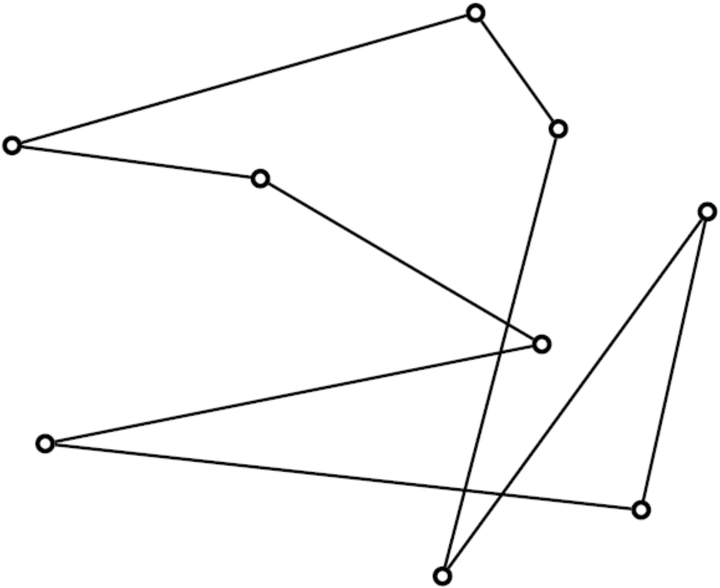
3-окружение



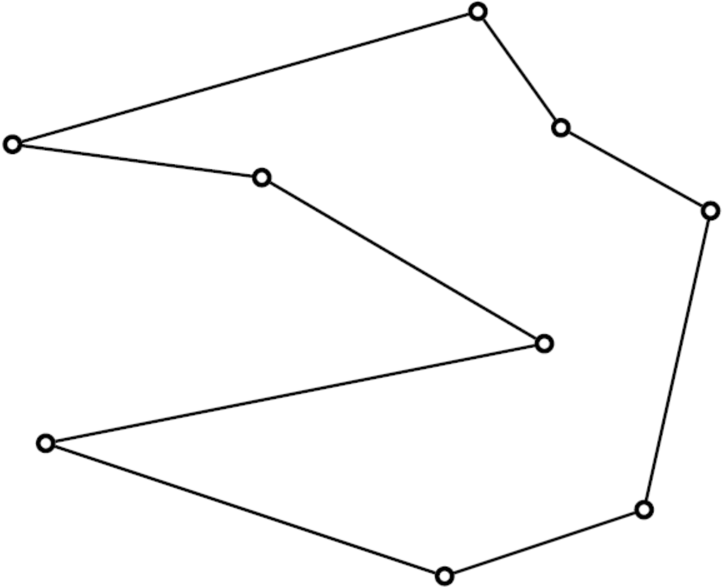
Пример локального поиска (с 3-окружением)



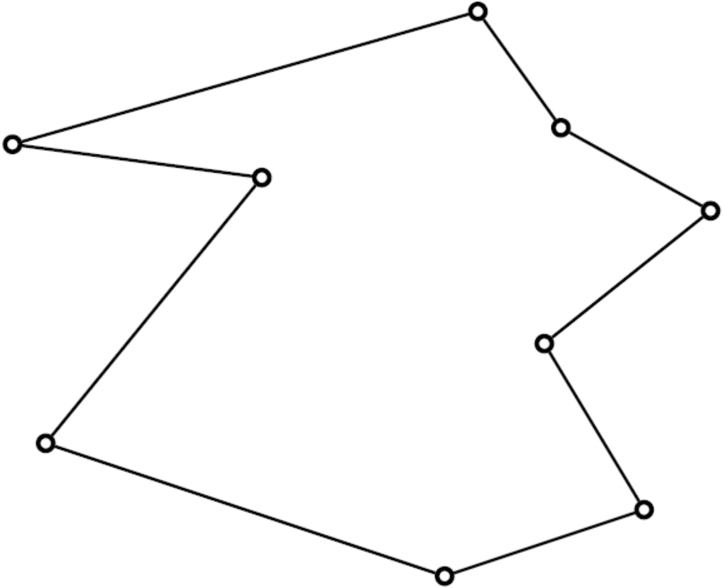
Пример локального поиска (с 3-окружением)



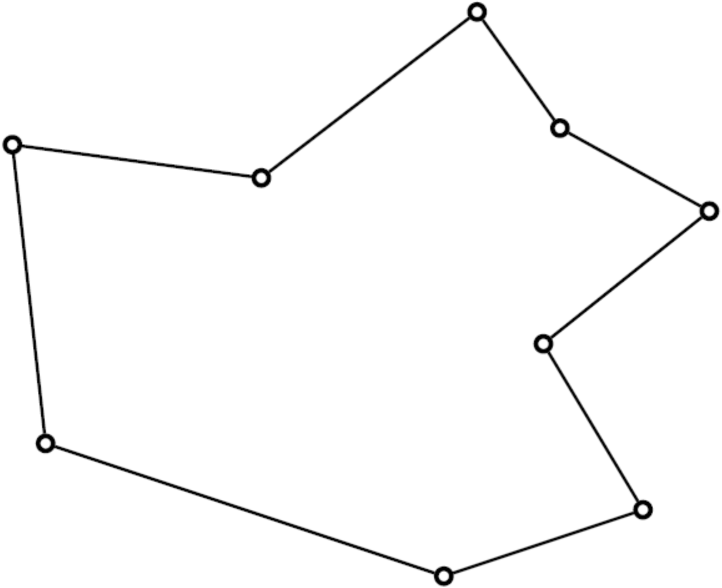
Пример локального поиска (с 3-окружением)



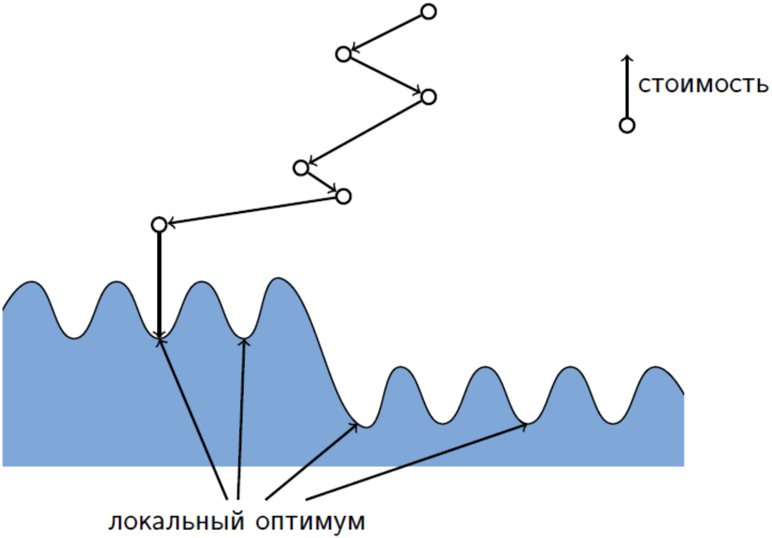
Пример локального поиска (с 3-окружением)



Пример локального поиска (с 3-окружением)



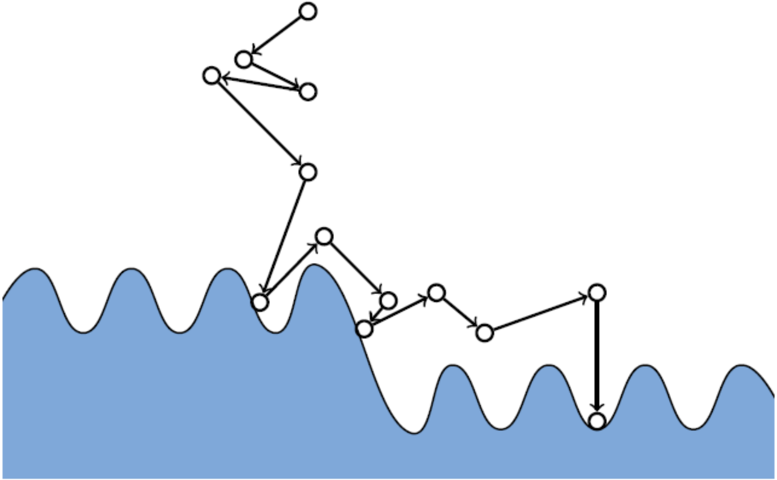
Локальный поиск абстрактно



Метод имитации отжига

$s \leftarrow$ какое-нибудь начальное решение
repeat
    выбрать случайное решение s' из окружения s
    $\Delta \leftarrow \mathrm{cost}(s') - cost(s)$
    if $\Delta < 0$
        then заменить s на s'
        else заменить s на s' с вероятностью $e^{-\Delta/T}$

Метод имитации отжига абстрактно



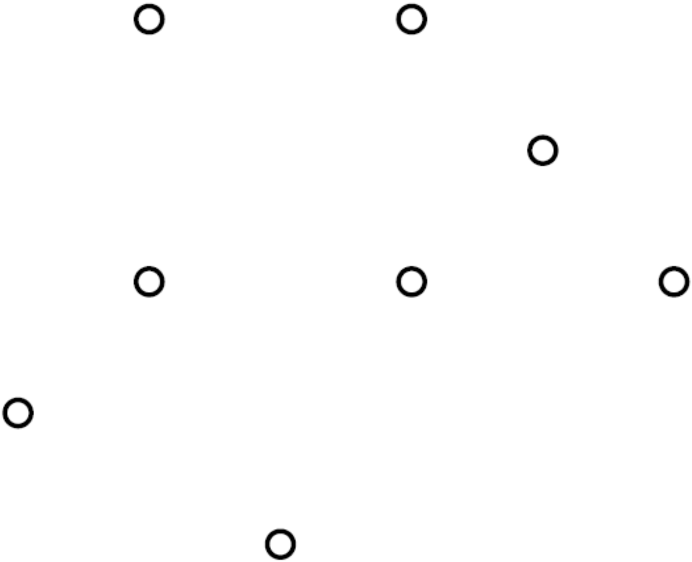
Задача коммивояжёра в метрическом пространстве

Задача коммивояжёра в метрическом пространстве (Metric TSP): частный случай для графов, веса рёбер которых удовлетворяют неравенству треугольника ($w(i, j) \leq w(i, k) + w(k, j)$).

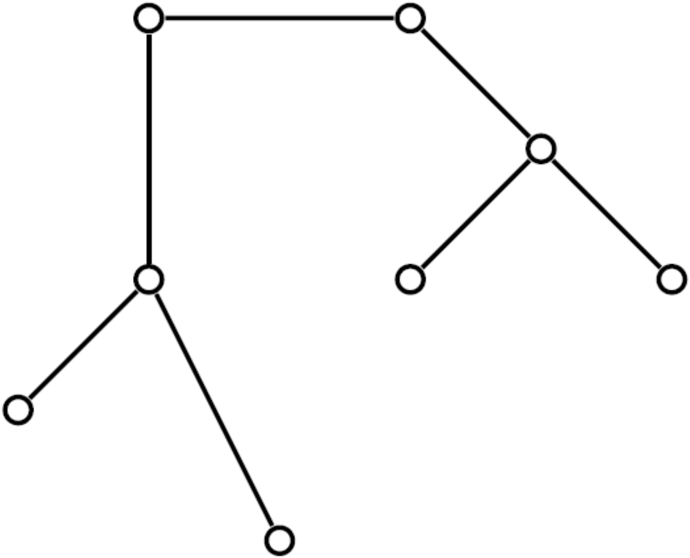
2-приближённый алгоритм

1. построить минимальное покрывающее дерево T;
2. продублировать каждое ребро дерева T и в полученном графе найти эйлеров цикл;
3. выкинуть из этого цикла все повторения вершин и вернуть полученный цикл.

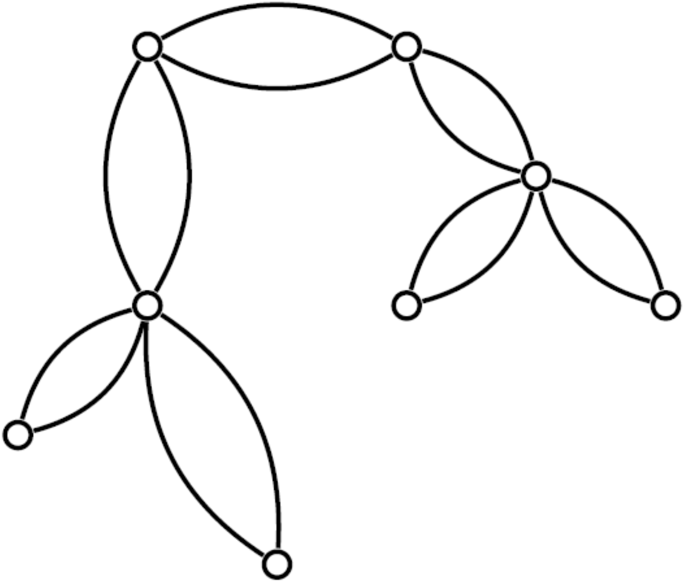
Пример



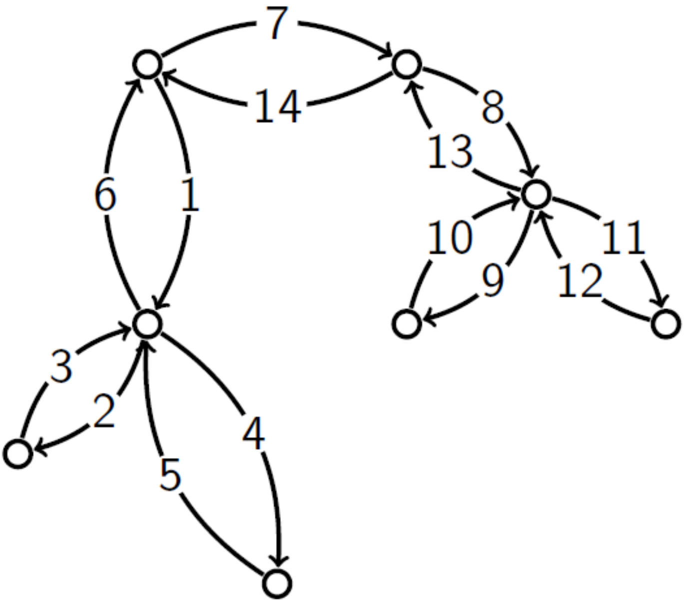
Пример



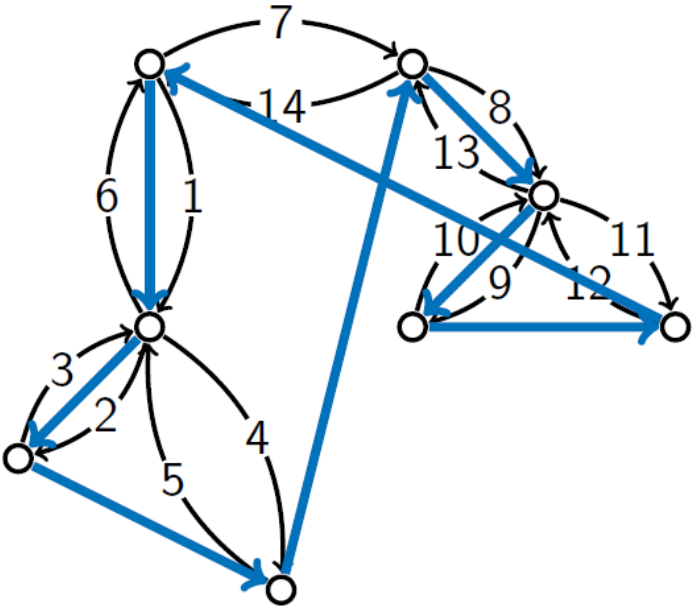
Пример



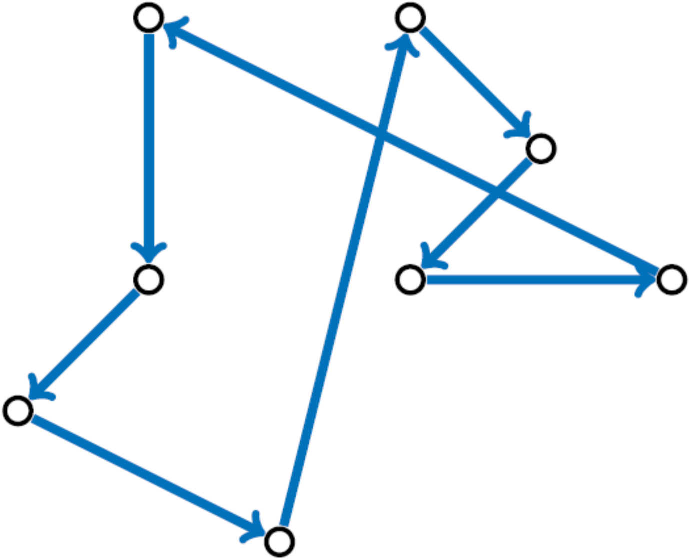
Пример



Пример



Пример



Доказательство

- пусть WT — вес минимального остовного дерева, а W_{opt} — вес оптимального гамильтонова цикла;
- $WT \leq W_{opt}$, поскольку при выкидывании ребра из гамильтонва цикла получается остовное дерево;
- каждое ребро построенного гамильтонова цикла заменяет какой-то путь эйлерова цикла, длина которого по неравенству треугольника не менее длины этого ребра;
- значит, длина найденного пути не превосходит $2WT$, а следовательно, и $2W_{opt}$.

1.5-приближённый алгоритм

1. построить минимальное покрывающее дерево T;
2. найти минимальное полное паросочетание всех вершин дерева T нечетной степени;
3. добавить найденные рёбра в дерево T и найти в полученном графе эйлеров цикл;
4. выкинуть из этого цикла все повторения вершин и вернуть полученный цикл.

Доказательство

- как и в предыдущем доказательстве, вес построенного цикла не превосходит $WT + WP$, где WP — вес минимального паросочетания вершин нечетной степени дерева T;
- нужно показать, что $WP \leq W_{opt}/2$;
- обозначим через A множество всех вершин нечётной степени дерева T;
- рассмотрим такой гамильтонов цикл на вершинах множества A : вершины множества A в нём будут встречаться в такой последовательности, в какой они идут в оптимальном гамильтоновом цикле графа G

Доказательство (продолжение)

- важно отметить, что нам не нужно строить такой цикл; нам важен лишь факт его существования
- нужно показать, что $WP \leq W_{opt}/2$;
- разбив вершины только что построенного цикла на чётные и нечётные, мы получим два паросочетания
- вес хотя бы одного из них будет не более $W_{opt}/2$
- значит, и вес минимального паросочетания не превосходит $W_{opt}/2$

Неприближаемость общего случая

Неприближаемость

- Предположим, что существует α-приближённый алгоритм для задачи коммивояжёра.
- Возьмём тогда произвольный (невзвешенный и необязательно полный) граф и присвоим всем его рёбрам вес 1.
- Между любыми двумя не соединёнными ребром вершинами добавим ребро веса $\alpha n + 1$.
- Заметим теперь, что если в исходном графе существует гамильтонов цикл, то в новом графе существует гамильтонов цикл веса n.

Неприближаемость (продолжение)

- Если же такого цикла в исходном графе нет, то самый лёгкий цикл в новом графе имеет вес хотя бы $(\alpha n + 1) + (n - 1) > \alpha n$.
- Таким образом, с помощью α-приближенного алгоритма для задачи о коммивояжёре мы можем понять, стоимость оптимального цикла в построенном графе превосходит n или нет.
- А это позволит нам понять (за полиномиальное время!), есть в исходном графе гамильтонов цикл или нет.
- Но тогда P = NP.

Точные алгоритмы

- Подзадачи: для подмножества городов $S \subseteq \{1, 2, \ldots, n\}$, включающего 1, и $j \in S$, обозначим через $C[S, j]$ длину кратчайшего пути, начинающегося в 1 и заканчивающегося в j, проходящего через каждый город из множества S ровно один раз.
- Пересчёт: $C[S, j] = \min_{i \in S, i \neq j}\{C[S \setminus \{j\}, i] + d_{ij}\}$.

Псевдокод

$C[\{1\},1] \leftarrow 0$

for $s \leftarrow 2$ **to** n

  do for всех $S \subseteq \{1,2,\ldots,n\}$ размера s, содержащих 1

    do $C[S,1] \leftarrow \infty$

      for всех $j \in S, j \neq 1$

        do $C[S,j] \leftarrow \min\limits_{i \in S, i \neq j} \{C[S \setminus \{j\}, i] + d_{ij}\}$

return $\min\limits_{j} C[\{1,\ldots,n\},j] + d_{j1}$

Сложность алгоритма

- Время работы данного алгоритма есть $O(n^2 2^n) = O^*(2^n)$.
- Более того, памяти ему требуется тоже $O^*(2^n)$, что делает его совсем непрактичным.

Пусть A — некоторое множество, $f, g : 2^A \to \mathbb{R}$, т.ч. $f(X) = \sum_{Y \subseteq X} g(Y)$. Тогда

$$g(X) = \sum_{Y \subseteq X} (-1)^{|X-Y|} f(Y).$$

Доказательство

$$\begin{aligned}
\sum_{Y \subseteq X} (-1)^{|X-Y|} f(Y) &= \sum_{Y \subseteq X} \sum_{Z \subseteq Y} (-1)^{|X-Y|} g(Z) \\
&= \sum_{Z \subseteq X} g(Z) \sum_{Z \subseteq Y \subseteq X} (-1)^{|X-Y|} f(Y) = g(X)
\end{aligned}$$

(последняя сумма равна 1, если $Z = X$, и нулю иначе).

Задача о гамильтоновом пути

- Формулировка задачи: необходимо проверить, есть ли в данном графе простой путь, проходящий через все вершины, начинающийся в заданной вершине s и заканчивающийся в заданной вершине t.
- Для $\{s, t\} \subseteq X \subseteq V$ обозначим через $g(X)$ количество путей (не обязательно простых! путь может проходить по некоторым вершинам несколько раз, а по некоторым вообще не проходить) длины $n - 1$ из s в t, проходящих только по вершинам множества X.
- Нетрудно видеть, что значение $g(X)$ содержится в строке s и столбце t матрицы A^{n-1}, где A — матрица смежности графа $G[X]$.

Задача о гамильтоновом пути (продолжение)

- Пусть теперь $f(X)$ есть количество путей длины $n-1$ из s в t, проходящих по всем вершинам множества X. В частности, $f(V)$ есть количество гамильтоновых путей из s в t.
- Тогда
$$f(V) = \sum_{Y \subseteq V} (-1)^{|V-Y|} g(Y).$$
- Таким образом, количество гамильтоновых путей в графе может быть найдено за время $O^*(2^n)$ и полиномиальную память.
- Интересно отметить, что данный алгоритм переизобретался три раза.

Задача о гамильтоновом пути (продолжение)