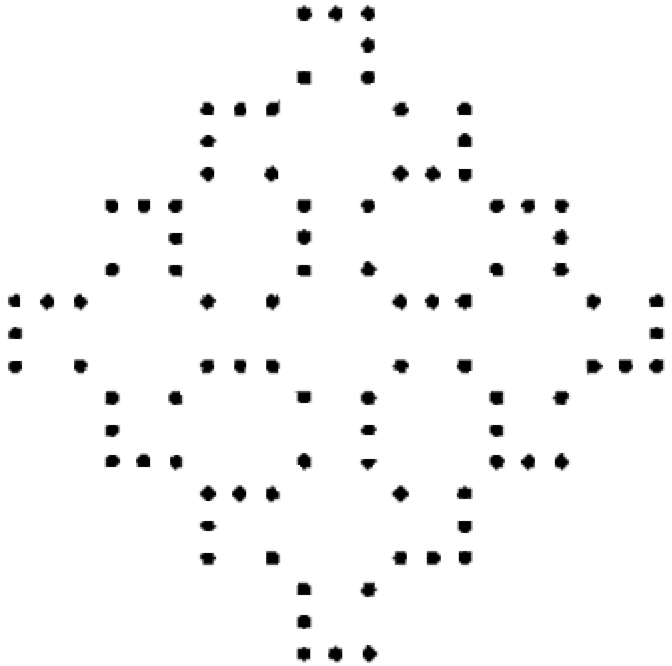
Задача коммивояжера

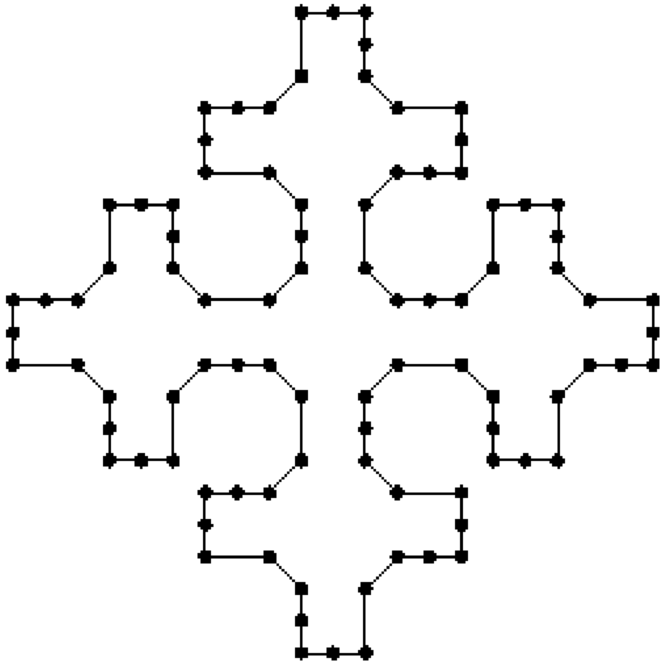
постановка, верхние и нижние оценки

Лекция 7

Задача коммивояжера

- Дана матрица (c_{ij}) попарных расстояний между городами, $1 \leq i, j \leq n$.
- Найти цикл, проходящий через каждую вершину ровно один раз и имеющий минимальный вес.





Близкие задачи

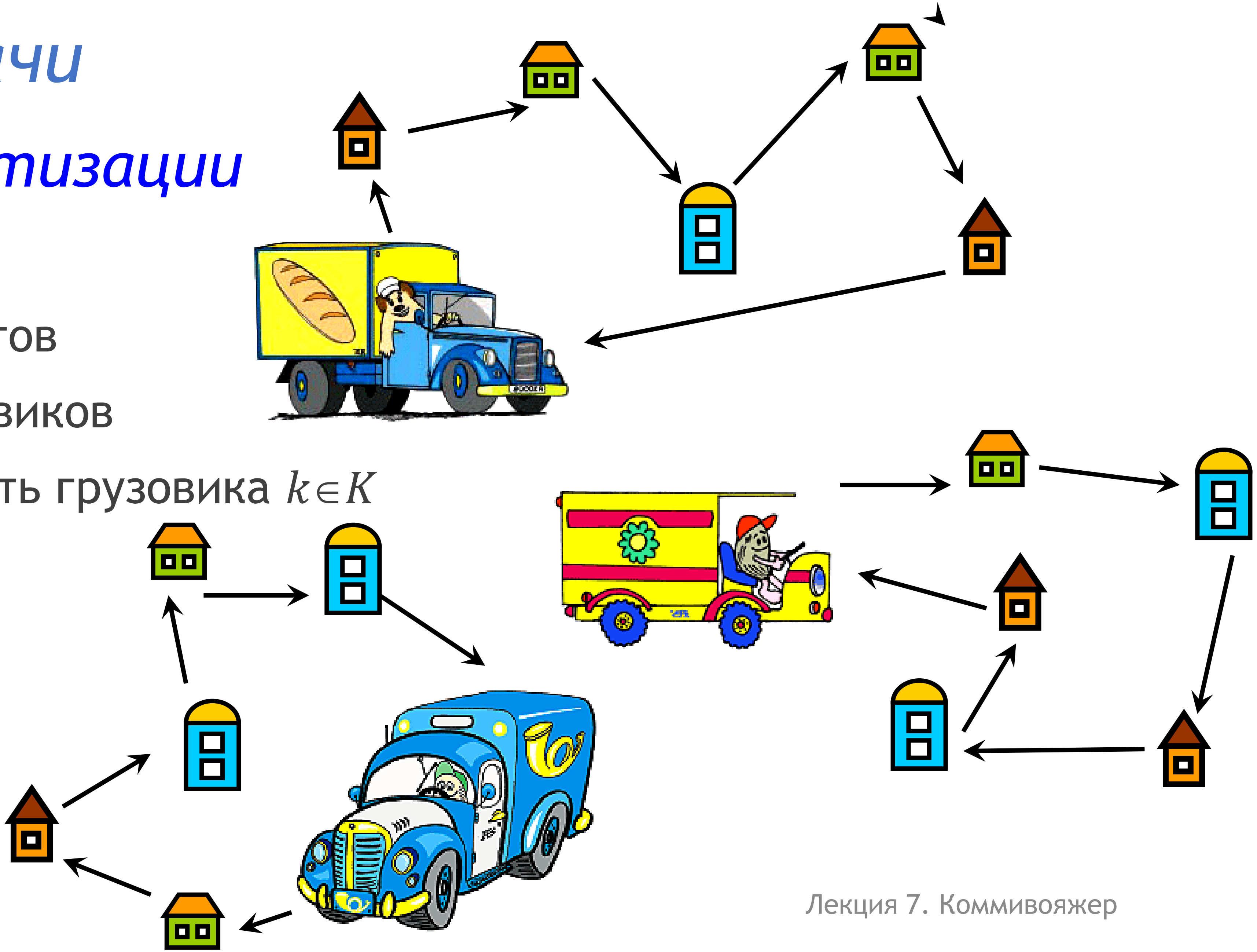
Задачи маршрутизации

Дано

J — множество клиентов

K — множество грузовиков

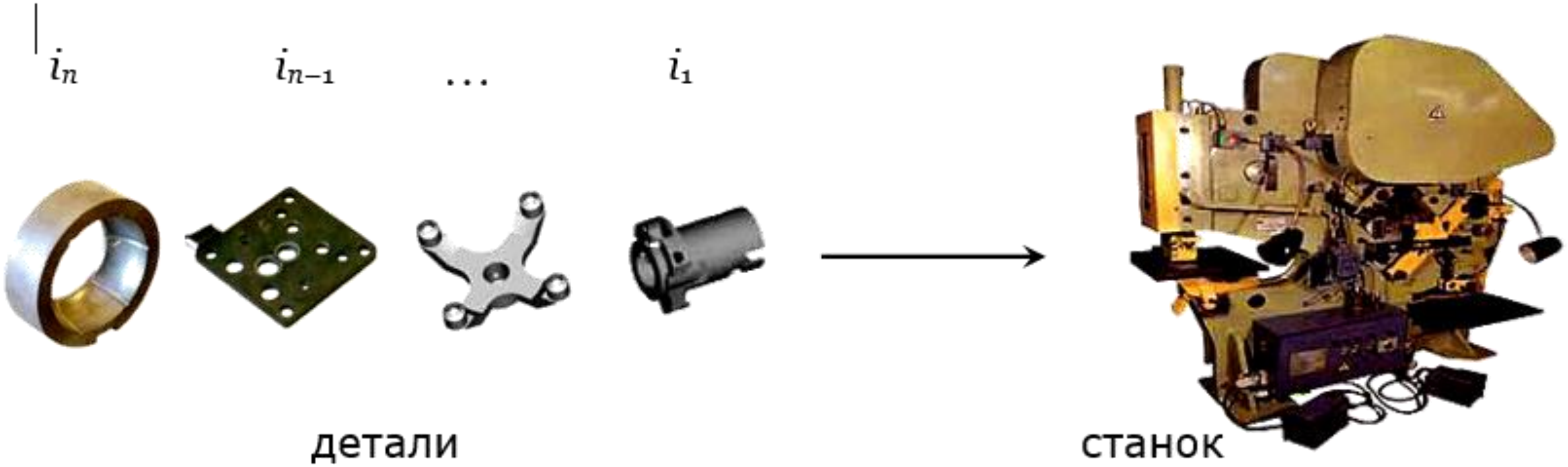
Q_k — грузоподъемность грузовика $k \in K$



Близкие задачи
Составление расписаний на одном станке

Дано: n деталей и один станок, c_{ij} — длительность переналадки станка для обработки j-й детали после i-й детали, p_j —длительность обработки j-й детали.

Найти последовательность обработки деталей, имеющую минимальную суммарную длительность.



Алгоритмическая сложность

Теорема 1. Задача коммивояжера является NP-трудной даже в случае, когда (c_{ij}) — евклидовы расстояния на плоскости, то есть матрица симметрична и удовлетворяет неравенству треугольника

$$c_{ij} \le c_{ik} + c_{kj}, 1 \le i, j, k \le n.$$

Теорема 2. Если существует приближенный полиномиальный алгоритм $\mathcal{A}$ и константа $r, 1 \le r < \infty$ такие, что для любого примера I задачи коммивояжера верно $\mathcal{A}(I) \le r\, OPT(I)$, то $P = NP$.

Доказательство Теоремы 1

Заметим, что задача коммивояжера не является задачей распознавания.

Рассмотрим NP–полную задачу о гамильтоновом цикле: дан граф $G = (V, E)$, правда ли, что он содержит гамильтонов цикл? С помощью решения задачи коммивояжера можно определить существует ли решение задачи о гамильтоновом цикле. Как?
По заданному графу $G = (V, E)$ построим пример задачи коммивояжера, положив

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

Если в оптимальном решении задачи коммивояжера получится цикл длины n, это будет означать, что граф содержит гамильтонов цикл, если нет, значит цикла нет. Задача коммивояжера не проще NP-полной задачи о гамильтоновом цикле, задача коммивояжера является NP-трудной.

Доказательство Теоремы 2

Рассмотрим NP-полную задачу о гамильтоновом цикле: дан граф $G = (V, E)$, правда ли, что он содержит гамильтонов цикл? Если условия теоремы верны и такие $\mathcal{A}$ и r существуют, то мы получим точный полиномиальный алгоритм решения задачи о гамильтоновом цикле. Как?
По заданному графу $G = (V, E)$ построим пример задачи коммивояжера, положив

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

Применим алгоритм $\mathcal{A}$ и посмотрим на ответ. Если получили цикл длины n, то граф G, очевидно, содержит гамильтонов цикл.

Если длина цикла больше n, то она не меньше чем $n \cdot r + (n - 1)$, так как включает вес хотя бы одного из «тяжелых» ребер. Но в этом случае граф G не может иметь гамильтонов цикл, так как алгоритм $\mathcal{A}$ ошибается не более чем в r раз и ответ в задаче коммивояжера не должен превосходить $n \cdot r$, если гамильтонов цикл есть. Итак, алгоритм $\mathcal{A}$ всегда дает правильный ответ для NP-полной задачи и имеет полиномиальную трудоемкость, то есть P = NP.

Нижние оценки в задаче коммивояжера

Примитивная оценка

Плата за выезд $a_i = \min_{i \neq j} c_{ij}, i = 1, \dots, n.$

Плата за въезд $b_j = \min_{i \neq j}(c_{ij} - a_i), j = 1, \dots, n$

Теорема 3. $OPT(c_{ij}) \geq \sum_{i=1}^{n} a_i + \sum_{j=1}^{n} b_j$.

Доказательство. Положим $c'_{ij} = c_{ij} - a_i, 1 \leq i, j \leq n$. Тогда $OPT(c_{ij}) = OPT(c'_{ij}) + \sum_{i=1}^{n} a_i$

Аналогично, $c''_{ij} = c'_{ij} - b_j, 1 \leq i, j \leq n$ и получаем

$$OPT(c_{ij}) = OPT(c''_{ij}) + \sum_{i=1}^{n} a_i + \sum_{j=1}^{n} b_j \geq \sum_{i=1}^{n} a_i + \sum_{j=1}^{n} b_j .$$

Нижние оценки в задаче коммивояжера

Оценка линейного программирования

Введем переменные $x_{ij} = \begin{cases} 1, \text{если из города } i \text{ едем в город } j, \\ 0, \quad \text{в противном случае.} \end{cases}$

Математическая модель

$$min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad j \in J,$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i \in J,$$

$$\sum_{i \in S} \sum_{j \in J \setminus S} x_{ij} \geq 1, \qquad \forall S \subset J, S \neq \emptyset$$ (Исключение подциклов)

$$x_{ij} \in \{0,1\}, \qquad i, j \in J.$$

Заменяя $x_{ij} \in \{0,1\}$ на $0 \leq x_{ij} \leq 1$, получаем задачу линейного программирования, которая дает нижнюю оценку для оптимума, не хуже предыдущей.

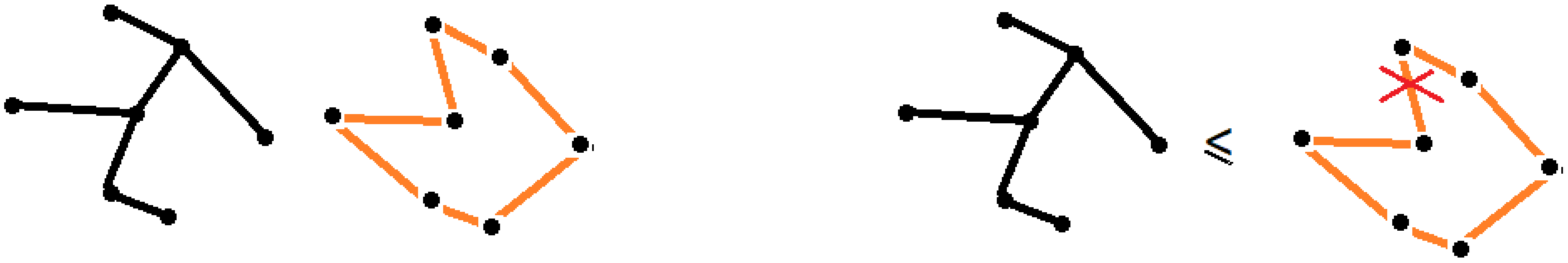
Нижние оценки в задаче коммивояжера

Минимальное остовное дерево для симметричных матриц

Пусть задан пример I задачи коммивояжера.
Рассмотрим вес минимального остовного дерева $T_{min}(I)$ и оптимальный цикл коммивояжера $OPT(I)$.

Почему $T_{min}(I) \leq OPT(I)$?

Если из цикла убрать любое ребро, то получается некоторое остовное дерево веса $T(I)$ не меньше веса минимального, получаем: $T_{min}(I) \leq T(I) < OPT(I)$



Разница между $T(I)$ и $OPT(I)$ это удаленное ребро, как его учесть в оценке?

Самый простой способ добавить к $T_{min}(I)$ минимальное из оставшихся ребер.

Нижние оценки в задаче коммивояжера

1-дерево для симметричных матриц

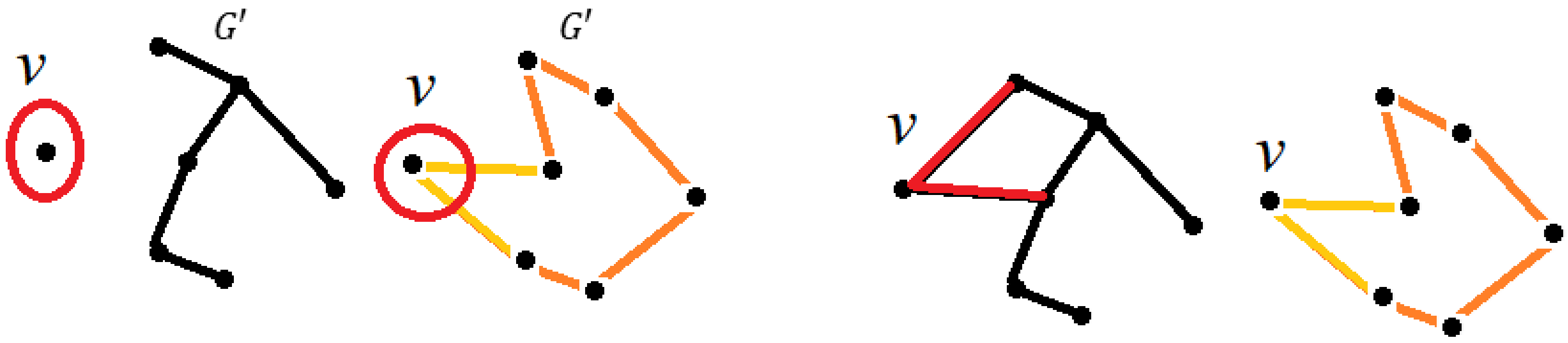
Пусть задан пример I задачи коммивояжера на графе G.

Удалим одну из вершин, а для оставшегося графа G' построим минимальное остовное дерево веса $T_{min}(G')$.

Заметим, что вес $T_{min}(G')$ не превосходит веса оптимального цикла без ребер смежных с вершиной v (т.к. это некоторое дерево на множестве вершин G' оранжевого цвета).

Вернем вершину v и два самых коротких смежных с ней ребра красного цвета. Получается 1-дерево.

Вес данного 1-дерева не превосходит $OPT(I)$, поскольку красные ребра не длиннее желтых.



Построение допустимого решения задачи коммивояжера

Алгоритм $A_б$ «Иди в ближайший»

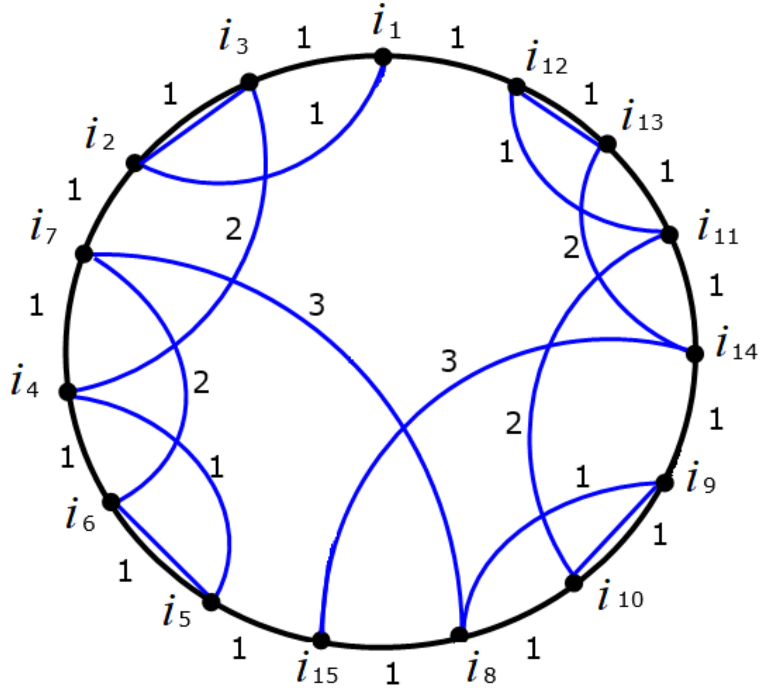
1. Выбираем произвольный город i_1
2. Находим ближайший город к i_1, обозначаем его i_2 и помечаем город i_1:
$$c_{i_1 i_2} = \min_{j \neq i_1} c_{i_1 j}.$$
3. На k-м шаге находим ближайший город к i_k, обозначаем его i_{k+1} и помечаем город i_k
$$c_{i_k i_{k+1}} = \min_{j \neq i_1, \dots, i_k} c_{i_k j}.$$
4. На шаге n возвращаемся в город i_1.

Построение допустимого решения задачи коммивояжера

Теорема 4

Для любого $r > 1$ найдется пример I задачи коммивояжера такой, что $A_Б(I) \geq r\, OPT(I)$ даже при условии, что $c_{ij} \leq c_{ik} + c_{kj}$, для всех $1 \leq i, j, k \leq n$.

На рисунке представлен пример для $1 < r < \frac{29}{15}$.
Используется 15 городов. Расстояния между городами заданы на рисунке, остальные получаются как длина минимального пути от вершины до вершины, неравенство треугольника будет выполняться. Начиная с города i_1 можно двигаться в ближайший по маршруту $i_1, i_2, \ldots, i_{15}, i_1$. Все промежуточные расстояния заданы, а $c_{15,1} = 5$, получаем $A_Б(I) = 29$, в то время как $OPT(I) = 15$, просто при движении по окружности. Для больших r нужно масштабировать пример, увеличивая количество городов.

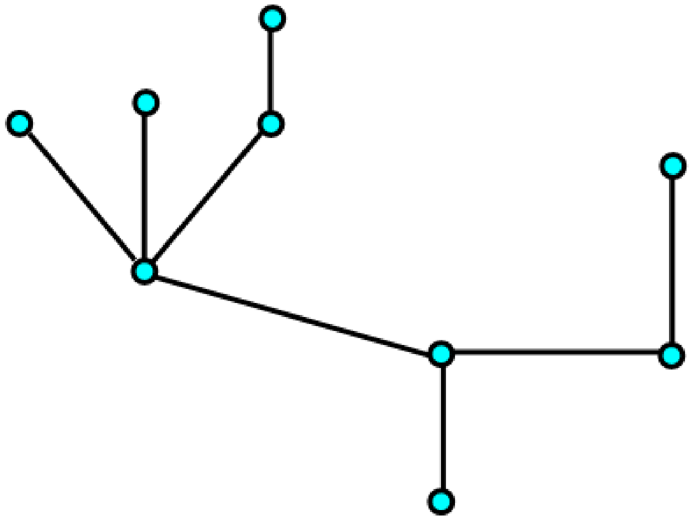


Алгоритм с гарантированной точностью для Евклидовой задачи
(матрица симметрична и выполняется неравенство треугольника)

1. Построить остовное дерево минимального веса с помощью Алгоритма Краскала.

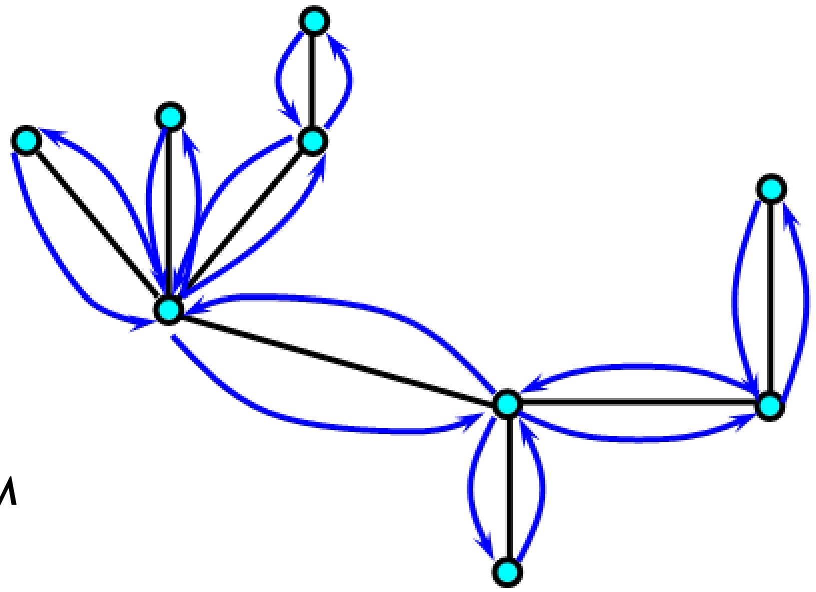
Алгоритм Ак дает нижнюю оценку для задачи коммивояжера

$$A_k(I) \leq OPT(I).$$



Алгоритм с гарантированной точностью для Евклидовой задачи
(матрица симметрична и выполняется неравенство треугольника)

2. Выполнить обход остовного дерева.



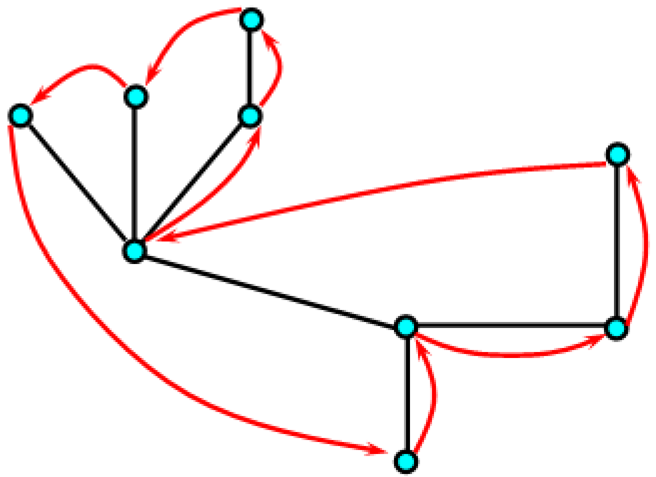
Обходим остовное дерево по правилу алгоритма «Поиск в глубину». Получаем маршрут, проходящий через все вершины. Листья посещаются один раз, но внутренние вершины посещаются несколько раз. Заметим, что каждое ребро дерева будет пройдено ровно 2 раза. Таким образом длина такого обхода $2A_k(I)$

$$2A_k(I) \leq 2OPT(I).$$

Алгоритм с гарантированной точностью для Евклидовой задачи

(матрица симметрична и выполняется неравенство треугольника)

3. Перестроить обход остовного дерева, методом срезания углов.



Движемся вдоль стрелок и помечаем вершины. Если очередная вершина уже помечена, то пропускаем ее и двигаемся дальше, пока не найдем непомеченную вершину или не вернемся в первую вершину.
Цепочку дуг для помеченных вершин заменяем прямой дугой в непомеченную или первую вершину. Заметим, что из неравенства треугольника длина такой дуги меньше, чем длина цепочки дуг, а значит длина нового обхода $A_{ST}(I)$ не длиннее $2A_k(I)$.

$$A_{ST}(I) \leq 2A_k(I) \leq 2OPT(I).$$

Алгоритм с гарантированной точностью для Евклидовой задачи
(матрица симметрична и выполняется неравенство треугольника)

Теорема 4. Если матрица (c_{ij}) удовлетворяет неравенству треугольника, то алгоритм перестройки двойного обхода остовного дерева A_{ST} получает Гамильтонов цикл не более чем в 2 раза хуже оптимального для любого примера I задачи коммивояжера, то есть

$$A_{ST}\,(I) \leq 2\,OPT(I).$$

Доказательство.
Для длины двойного обхода имеем

$$2\,A_K\,(I) \leq 2\,OPT(I).$$

Пусть новое ребро e, не содержащееся в двойном обходе, заменяет цепочку ребер $\{e_1, e_2, \dots, e_k\}$. Из неравенства треугольника следует, что $w_e \leq \sum_{i=1}^{k} w_{e_i}$.
А значит

$$A_{ST}(I) \leq 2A_k(I) \leq 2OPT(I).$$