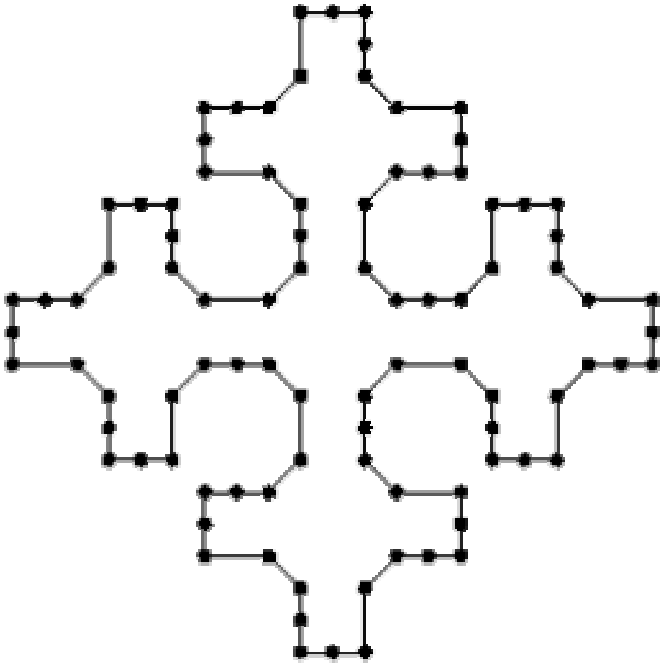
Задача коммивояжера

Дана матрица (c_{ij}) попарных расстояний между городами, $1 \le i, j \le n$.

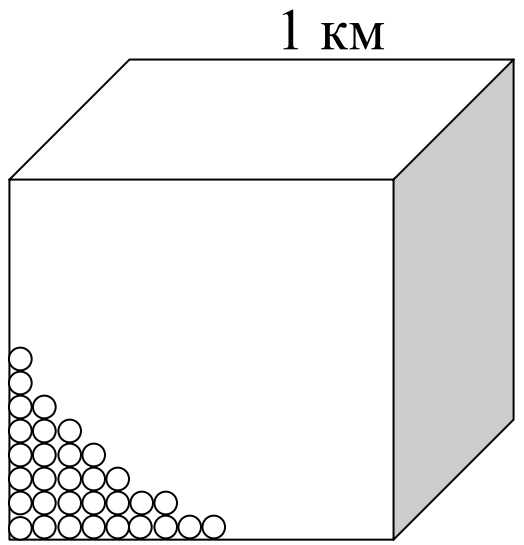
Найти контур минимальной длины, то есть цикл, проходящий через каждую вершину ровно один раз и имеющий минимальный вес.



http://www.ing.unlp.edu.ar/cetad/mos/TSPBIB_home.html

Трудоемкость полного перебора

Фантастический компьютер:



Куб со стороной 1 км.

Оперативная память – количество ячеек размером в атом водорода.

Все операции обмена проходят со скоростью света.

Производительность – 10^{52} операций в секунду.

Всего вариантов $(n-1)!$ Трудоемкость $n!$.

Для $n = 100$ получаем 10^{152} операций.

Время счета 10^{100} секунд.

Упражнение. Придумать алгоритм динамического программирования для решения задачи коммивояжера с трудоемкостью $O(n^2 2^n)$.

Задачи маршрутизации

J — множество клиентов

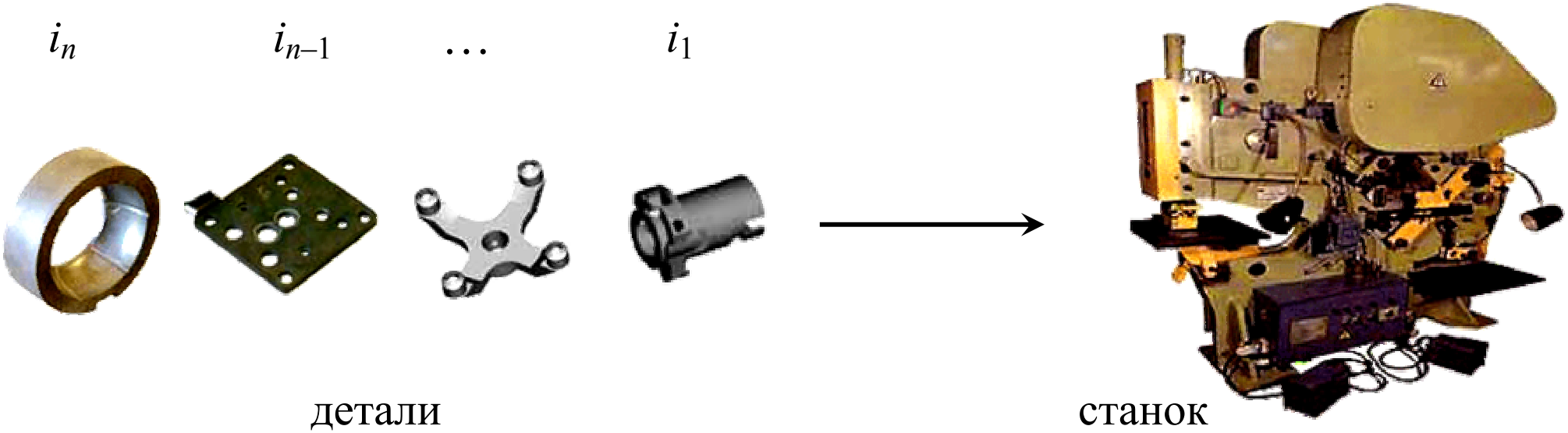
K — множество грузовиков

Q_k — грузоподъемность грузовика $k \in K$

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.



Раскрой рулонного материала с рисунком

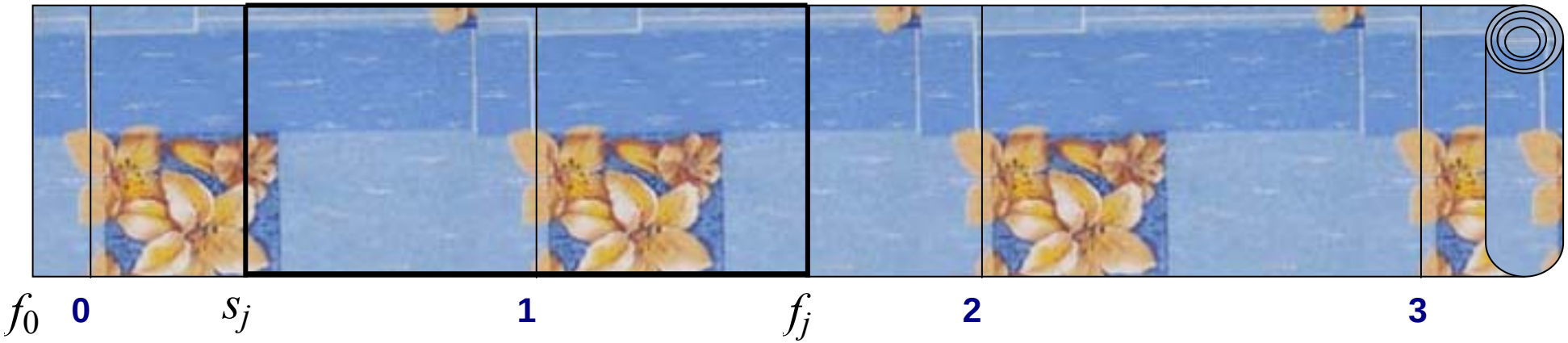
Дано бесконечный рулон с рисунком в 1м и размеры n кусков

$0 \le s_j \le 1$ — начало куска j, $0 \le f_j \le 1$ — конец куска j. Начало рулона и конец раскроя — f_0.

Найти последовательность отрезания кусков с минимальной длиной использованного материала

$$c_{ij} = \begin{cases} s_j - f_i, & \text{если } s_j \ge f_i, \\ 1 + s_j - f_i, & \text{если } s_j < f_i \end{cases}$$

Задача коммивояжера для $(n + 1)$-го города.



Алгоритмическая сложность

Задачи распознавания — задачи с ответом *«да»* или *«нет»*. Пример: есть ли в графе гамильтонов цикл?

Класс NP — класс задач распознавания, в которых можно проверить ответ «да» за полиномиальное время.

NP-полные задачи — самые трудные задачи в NP, то есть если существует точный полиномиальный алгоритм для решения одной из них, то существует точный полиномиальный алгоритм для решения всех задач из класса NP.

NP-трудные задачи — задачи которые могут не лежать в NP, но которые не проще NP-полных.

Класс P — класс задач распознавания, которые можно решить полиномиальным алгоритмом.

Проблема P = NP? стоит $1 млн. http://www.claymath.org/millennium/

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 A и константа r, $1 \le r < \infty$ такие, что для любого примера I задачи коммивояжера верно $A(I) \le r\, OPT(I)$, то P = NP.

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

$$c_{ij} = \begin{cases} 1, & \text{если } (ij) \in E, \\ nr, & \text{если } (ij) \notin E, \end{cases} \qquad n = |V|.$$

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

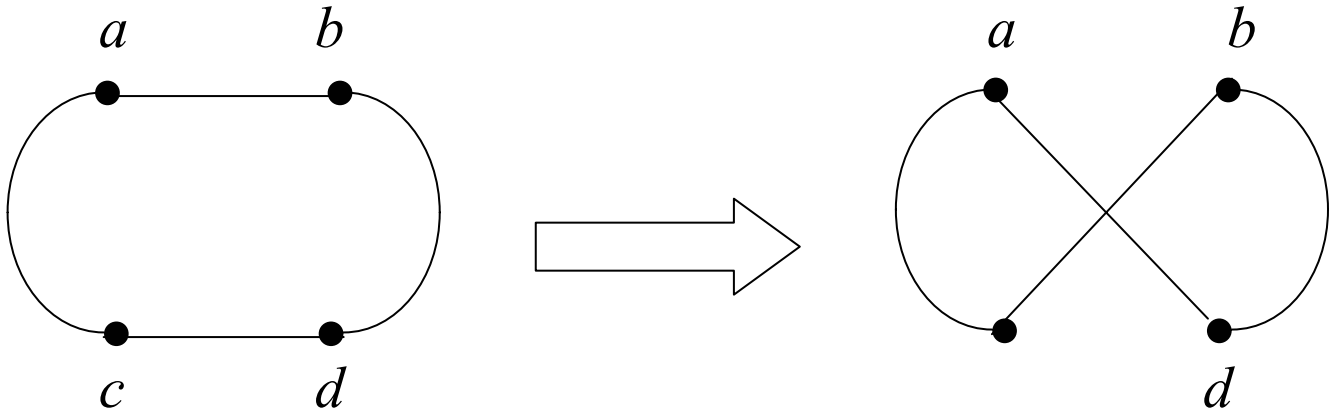
Если длина цикла больше n, то она не меньше чем $n \cdot r + (n - 1)$, так как включает вес хотя бы одного из «тяжелых» ребер. Но в этом случае граф G не может иметь гамильтонов цикл, так как алгоритм A ошибается не более чем в r раз и ответ в задаче коммивояжера не должен превосходить $n \cdot r$, если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

Алгоритмы локального спуска

Пусть C — гамильтонов цикл. $N(C)$ — окрестность полиномиальной мощности, т. е. число элементов окрестности полиномиально ограничено.

Примеры окрестностей:

Окрестность 2–замена



Выбираем в C два несмежных ребра и заменяем их другими так, чтобы снова получился гамильтонов цикл. Всего таких соседей $n(n-3)/2$.

Аналогично получаем окрестности 3– замена, 4–замена и т.д.

Стандартный алгоритм локального спуска

1. Выбрать C_0 — начальный цикл и вычислить его длину $F(C_0)$, $i := 0$.
2. Найти наилучшего соседа C_{i+1} для цикла C_i :

$$F(C_{i+1}) = \min \{F(C) \mid C \in N(C_i)\}.$$

3. Если $F(C_{i+1}) < F(C_i)$, то положить $i := i + 1$ и вернуться на 2,

   иначе STOP, C_i — локальный минимум.

Локальный спуск может потребовать экспоненциального числа шагов.

Погрешность локальных оптимумов

Теорема 3. Пусть A — алгоритм локального спуска и окрестность $N(C)$ имеет полиномиальную мощность. Если существует константа r, $1 \le r < \infty$ такая, что для любого примера I задачи коммивояжера справедливо неравенство

$A(I) \le r \cdot OPT(I)$, то P = NP.

Доказательство. (Аналогично предыдущему) Рассмотрим задачу в гамильтоновом цикле и получим точный полиномиальный алгоритм ее решения. Снова по графу $G = (V, E)$ строим матрицу

$$c_{ij} = \begin{cases} 1, & \text{если } (ij) \in E, \\ nr, & \text{если } (ij) \notin E, \end{cases} \qquad n = |V|.$$

Выбираем произвольный цикл C_0. Его длина в худшем случае равна $F(C_0)=(nr)n$.

На каждом шаге локального спуска часть «тяжелых ребер» заменяется на легкие. При самом длинном спуске одно «тяжелое» ребро заменяется на одно легкое. Значит число шагов (возвращений на п.2) не превосходит n. Каждый шаг имеет полиномиальную трудоемкость, так как окрестность $N(C)$ имеет полиномиальную мощность. Следовательно, алгоритм A является полиномиальным для данного класса примеров и, как следует из доказательства предыдущей теоремы, A — точный алгоритм. Значит P = NP. ■

Эвристические алгоритмы

Алгоритм $A_Б$ «Иди в ближайший из непройденных городов»

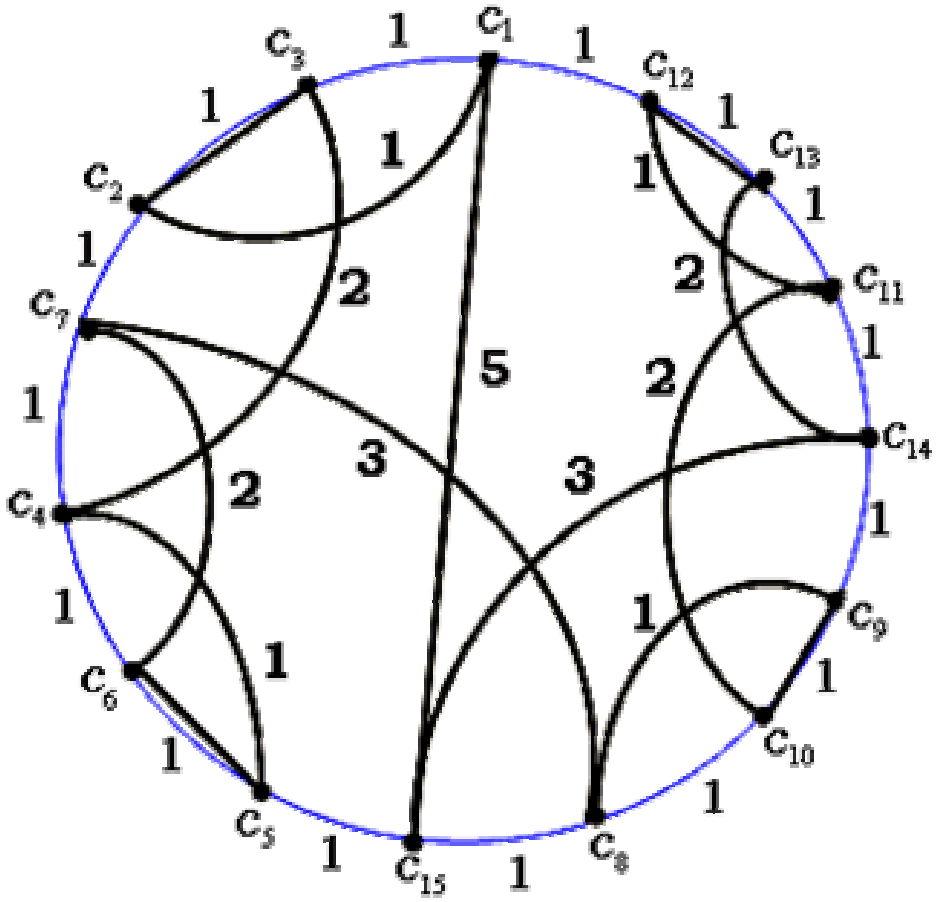
1. Выбираем произвольный город i_1.
2. Находим ближайший город к i_1, обозначаем его i_2 и помечаем город i_1:
$$c_{i_1 i_2} = \min_{j \neq i_1} c_{i_1 j}.$$
3. На k-м шаге находим ближайший город к i_k, обозначаем его i_{k+1} и помечаем город i_k :
$$c_{i_k i_{k+1}} = \min_{j \neq i_1, \ldots, i_k}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

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 $1 \leq i, j, k \leq n$.

Трудный пример для алгоритма $A_Б$

Имеется 15 городов, расположенных на окружности длиной 15 единиц, $OPT(I) = 15$.

Алгоритм $A_Б$ получает 27 единиц:



Вероятностный аналог алгоритма $A_Б$

Алгоритм $A_Б(\alpha)$, $\alpha \geq 1$.

На k-м шаге формируем список кандидатов

$$I_k(\alpha) = \left\{ j \neq i_1, \ldots, i_k \mid c_{i_k j} \leq \alpha \min_{j \neq i_1, \ldots, i_k} c_{i_k j} \right\}$$

Следующий город i_{k+1} выбирается из списка кандидатов случайным образом.

При $\alpha = 1$ получаем алгоритм «Иди в ближайший из непройденных городов».

Итерационный алгоритм

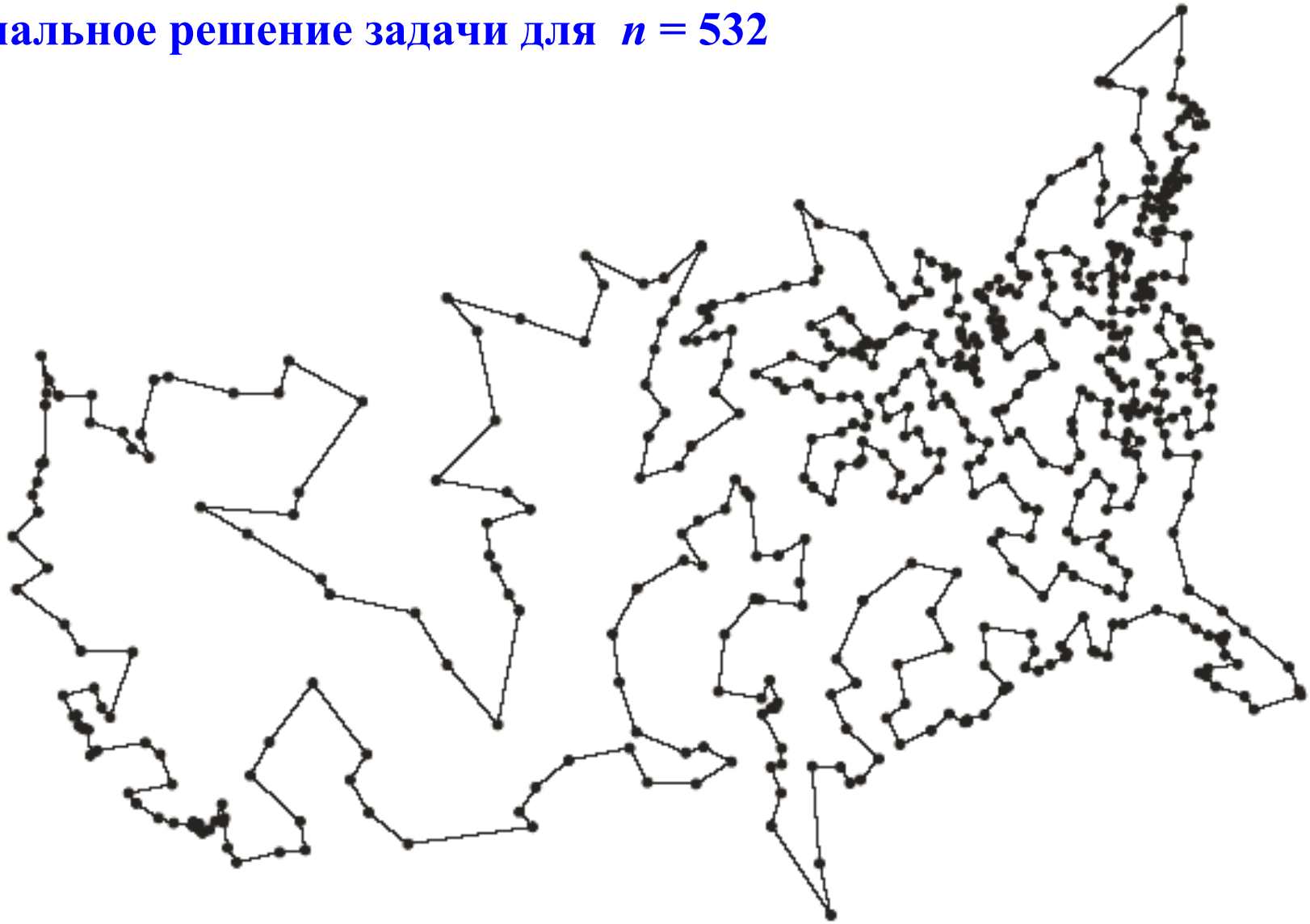
Алгоритм $A_Б(\alpha)$ применяется несколько раз и для каждого решения используется стандартная процедура локального спуска. Лучший из найденных локальных оптимумов предъявляется в качестве ответа.

1. Полагаем $F^* := \infty$
2. For $t := 1,\ldots, T$ do
   - 2.1. Применяем алгоритм $A_Б(\alpha)$ и получаем цикл C_t.
   - 2.2. Применяем алгоритм локального спуска с начальным решением C_t.
   - 2.3. Если полученный локальный минимум $\widetilde{C}_t$ меньше рекорда, т.е.

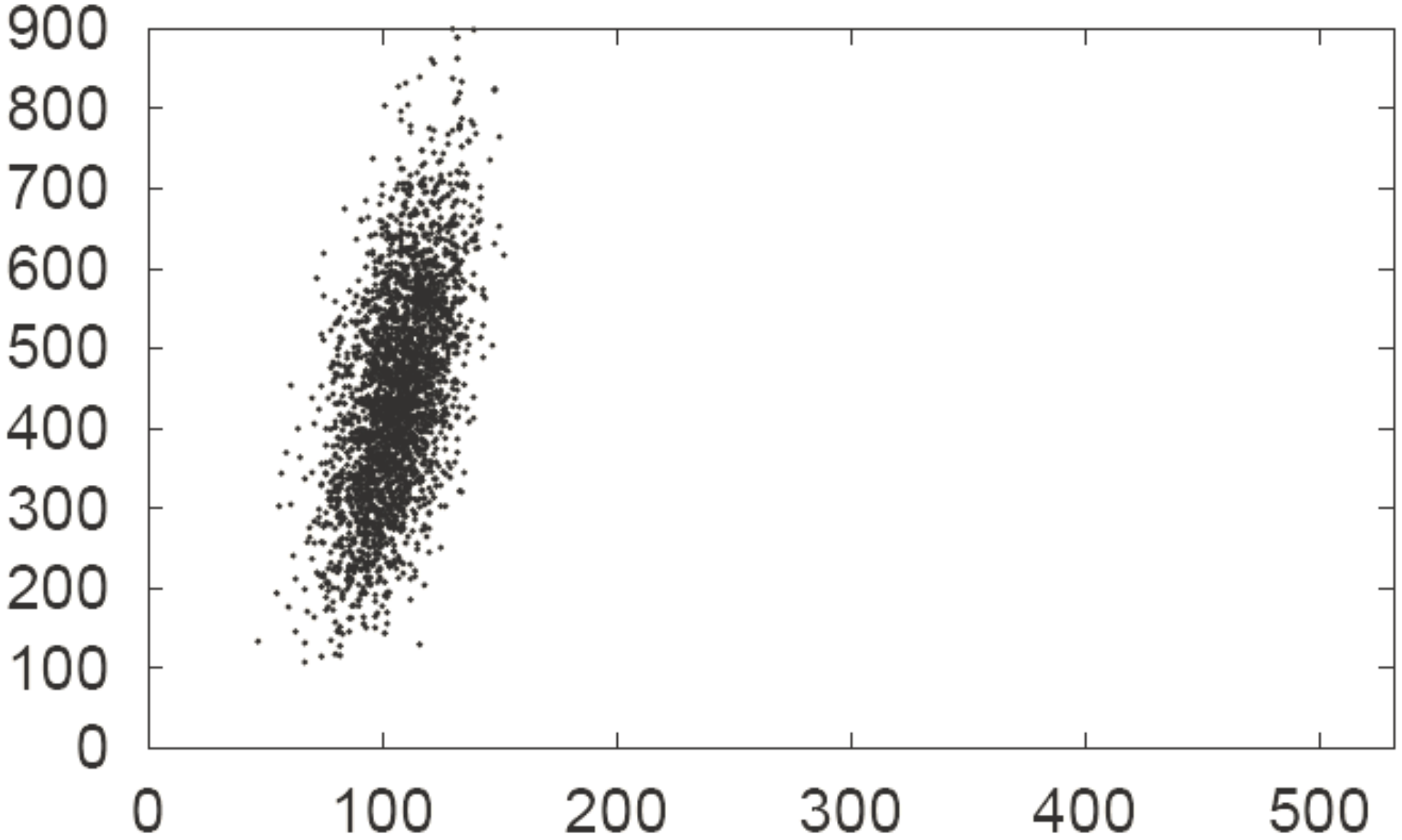
     $F(\widetilde{C}_t) < F^*$, то меняем рекорд $F^* := F(\widetilde{C}_t)$.

Увеличивая T и α, будем получать все более и более точные решения.

Оптимальное решение задачи для $n = 532$



Расстояние от локальных оптимумов до глобального



Поиск с чередующимися окрестностями (VNS)

1. Выбрать начальное решение C, положить $F^* = F(C)$, определить систему окрестностей $N_1, \ldots, N_K$.
2. Пока не выполнен критерий остановки, делать следующее:
   2.1. Положить $k := 1$.
   2.2. Повторять пока $k \le K$:
      2.2.1. Выбрать решение C' в окрестности $N_k(C)$ случайным образом;
      2.2.2. Применить локальный спуск с окрестностью N_1, взяв C' в качестве стартовой точки, и получить локальный минимум. Обозначим его C'';
      2.2.3. Если $F(C'') < F(C)$, то положить C := C'' и вернуться на 2.1, иначе положить $k := k + 1$.

Результаты численных экспериментов

n	Лучшее найденное решение		Время счета	
	2-opt	VNS	2-opt	VNS
100	825,69	811,95	0,25	0,17
200	1156,98	1132,63	3,88	2,82
300	1409,24	1376,76	12,12	9,35
400	1623,60	1577,42	46,13	34,37
500	1812,08	1756,26	110,64	91,00
600	1991,56	1925,51	204,60	173,07
700	2134,86	2089,33	347,77	259,06
800	2279,18	2190,83	539,94	462,23
900	2547,43	2342,01	699,33	624,74
1000	2918,10	2483,95	891,61	792,88
Среднее	1887,87	1768,67	285,63	244,97

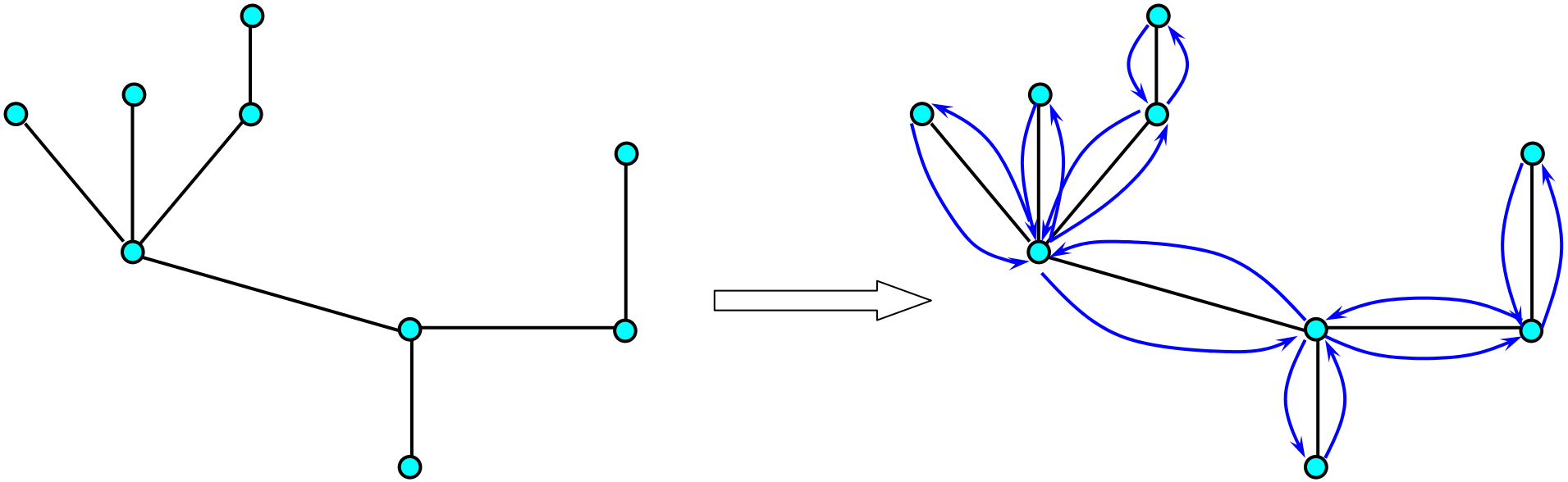
Алгоритмы с гарантированной точностью

Остовное дерево. Дан связный граф $G = (V, E)$, каждому ребру $e \in E$ приписан вес $w_e \geq 0$. Найти в G остовное дерево с минимальным суммарным весом ребер.

Алгоритм Крускала A_K дает точное решение задачи и нижнюю оценку для задачи коммивояжера:

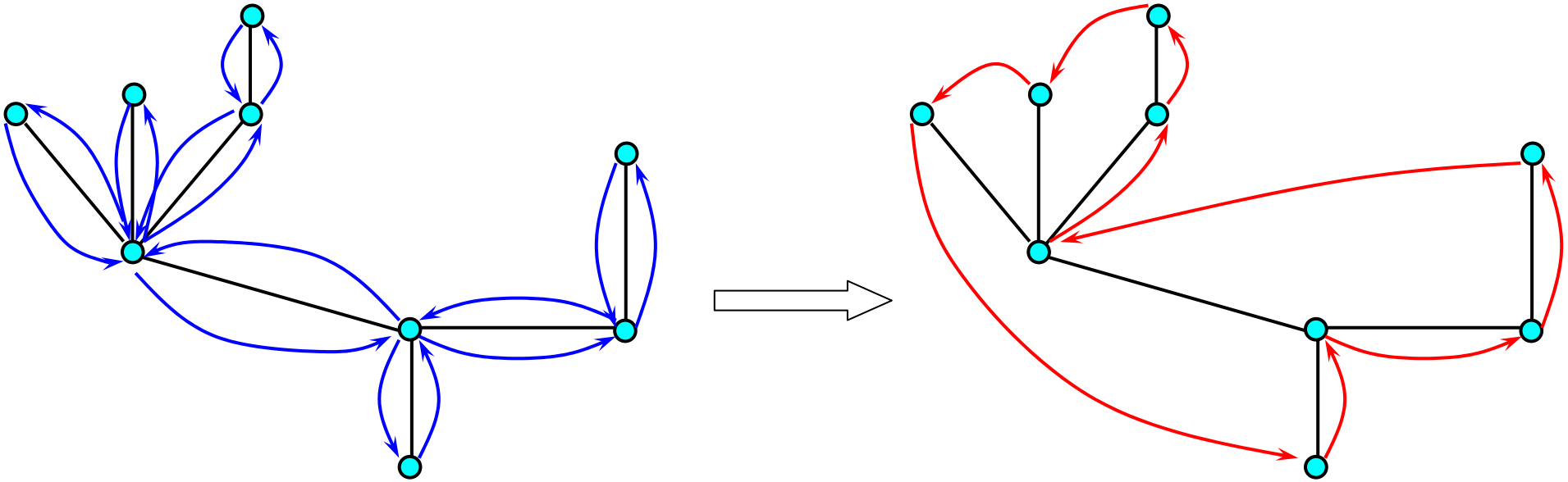
$$A_K(I) \leq OPT(I).$$

Двойной 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.

Перестройка остовного дерева



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.

Оценка точности алгоритма

Теорема 5. 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

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

Доказательство. Для длины двойного обхода имеем

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

Пусть новое ребро e, не содержащееся в двойном обходе, заменяет цепочку ребер $\{e_1, e_2, \ldots, e_k\}$. Из неравенства треугольника следует, что

$w_e \le \sum_{i=1}^{k} w_{e_i}$, то есть $A_{ST}(I) \le 2\, A_K(I) \le 2\, OPT(I)$. ■

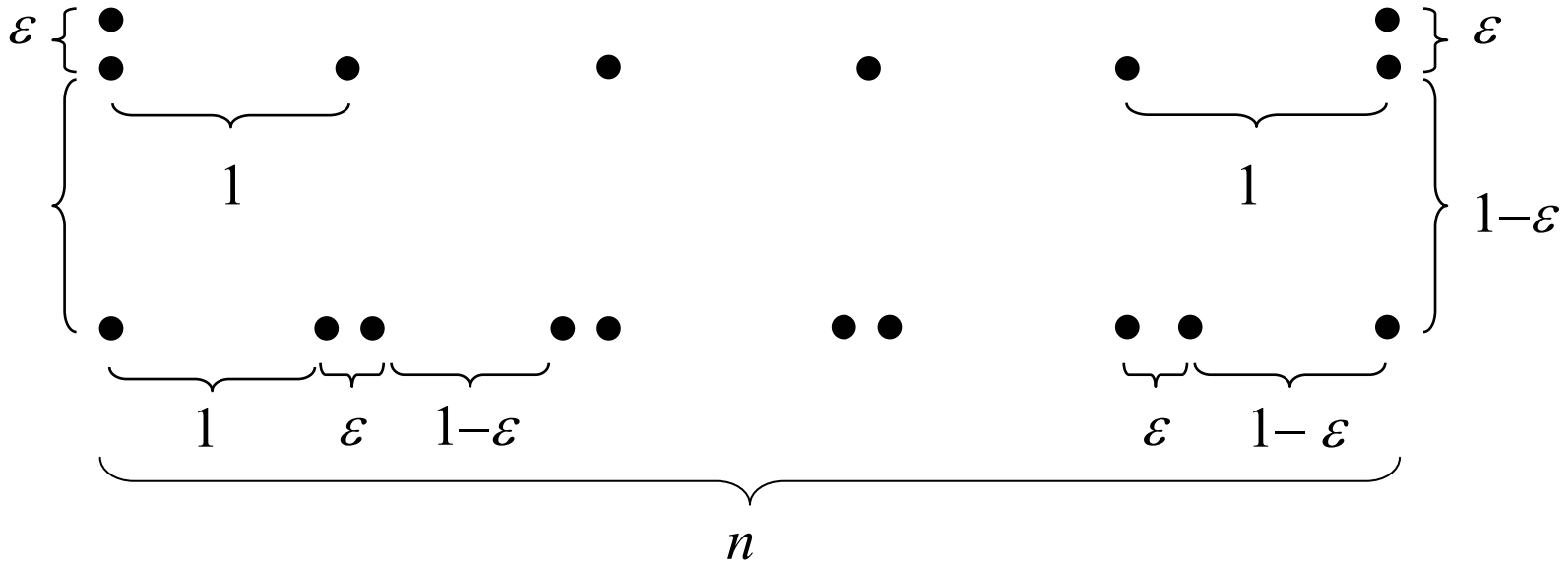
Теорема 6. Полученная оценка точности

$$A_{ST}(I) \le 2\ OPT(I)$$

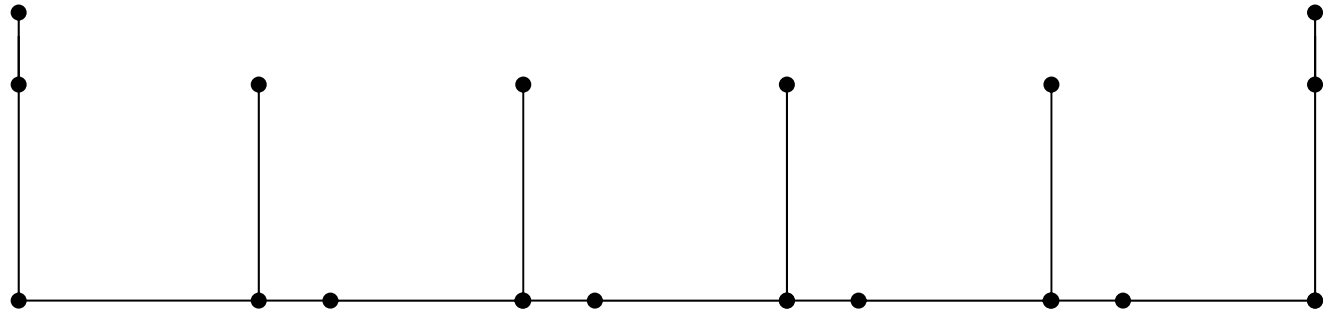
является неулучшаемой.

Доказательство. Приведем пример семейства исходных данных задачи коммивояжера, на котором оценка 2 достигается асимптотически.

Рассмотрим следующий пример на плоскости с $3(n+1)$ вершинами:

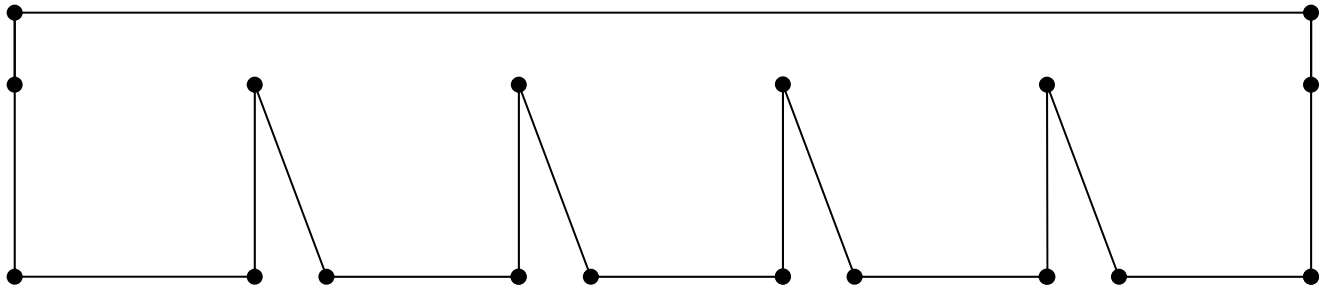


Для этого примера минимальное остовное дерево имеет вид:



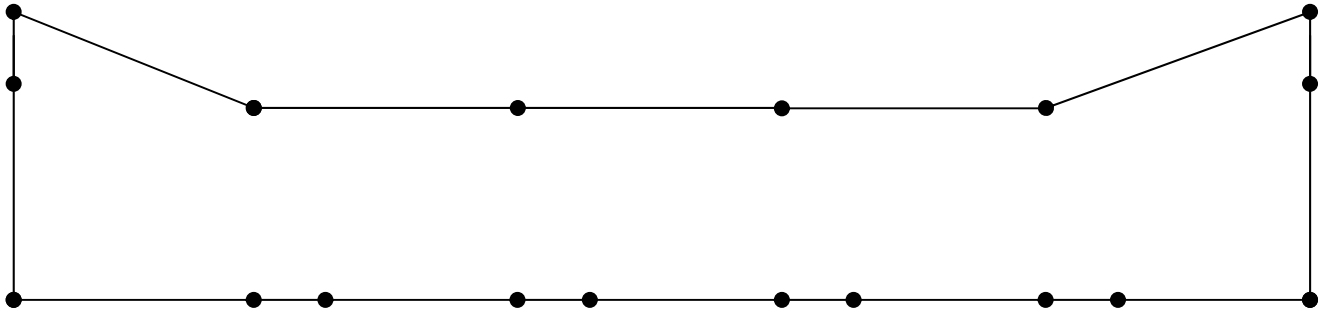
Длина остовного дерева $A_K = n + (n + 1)(1 - \varepsilon) + 2\varepsilon$

Алгоритм перестройки двойного обхода получит решение



Длина этого Гамильтонова цикла $A_{ST} \approx 2n + 2n\,(1 - \varepsilon)$

Оптимальное решение задачи $OPT(I) \approx 2n + 2$.



Таким образом, при $n \to \infty$ и $\varepsilon \to 0$ получаем $A_{ST}(I) / OPT(I) \to 2$ ■

Упражнение. Если матрица (c_{ij}) удовлетворяет неравенству треугольника, то существует алгоритм перестройки двойного обхода остовного дерева, который получает Гамильтонов цикл не более чем в 1,5 раза хуже оптимального для любого примера I задачи коммивояжера, то есть

$$A_{ST}(I) \le 1{,}5\ OPT(I).$$