Лекция 4. Задача коммивояжера

Екатерина Алексеева

Новосибирский Государственный Университет
Механико-математический факультет
http://math.nsc.ru/∼alekseeva/

30 сентября, 2012

Содержание лекции

Задача коммивояжера (ЗК)

Дано

$I = \{1, \ldots, n\}$ – *множество городов, матрица* (c_{ij}) – *попарные расстояния между городами,* $1 \leq i, j \leq n$.

Найти

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

http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

Математическая модель (ЗК)

Переменные задачи:

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

$$\min \sum_{i \in I} \sum_{j \in I} c_{ij} x_{ij} \tag{1}$$

при ограничениях

$$\sum_{i \in I} x_{ij} = 1, \forall j \in I, \tag{2}$$

$$\sum_{j \in I} x_{ij} = 1, \forall i \in I, \tag{3}$$

$$\sum_{i \in S} \sum_{j \in I \setminus S} x_{ij} \geq 1, \forall S \subset I, S \neq \emptyset,$$

или

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \forall S \subset I, S \neq \emptyset,$$

Математическая модель (ЗК)

Переменные задачи:
$u_i \geq 0$ – номер шага, на котором посетили город i
Исключение подциклов полиномиальным числом ограничений

$$u_i - u_j + nx_{ij} \leq n - 1, i, j \in I \setminus i_1, i \neq j. \qquad (4)$$

Упражнение Проверить (1)–(4), что математическая модель ЗК

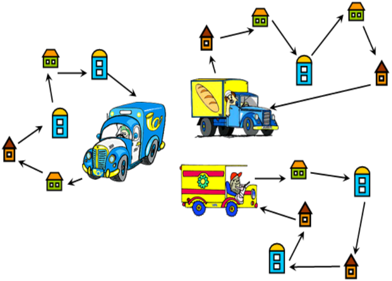
Задачи маршрутизации

Дано

J – множество клиентов,
K – множество грузовиков,
Q_k - грузоподъемность грузовика
$k \in K$

Найти

порядок посещения клиентов, так чтобы всем доставить груз с минимальными затратами



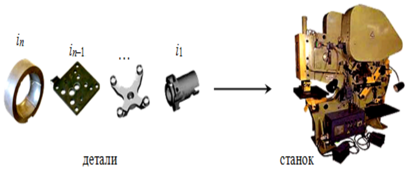
Составление расписаний на одном станке

Дано

*n деталей,
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

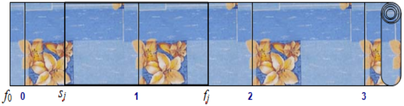


Раскрой рулонного материала с рисунком

Дано

бесконечный рулон с рисунком в 1м,
размеры n кусков,
$0 \leq s_j \leq 1$ – *начало куска j,*
$0 \leq f_j \leq 1$ – *конец куска j,*
начало рулона и конец раскроя f_0

$$c_{ij} = \begin{cases} s_j - f_i, & \text{если } s_j \geq f_i, \\ 1 + s_j - f_i, & \text{если } s_j < f_i, \end{cases} \tag{5}$$



Найти

последовательность отрезания кусков с минимальной длиной использованного материала

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

Класс P

класс задач распознавания, которые можно решить полиномиальным алгоритмом

NP-трудные задачи

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

Алгоритмическая сложность

Проблема P = NP?

стоит $1 млн. http://www.claymath.org/millennium/

Алгоритмическая сложность ЗК

Теорема 1

ЗК является NP-трудной даже в случае, когда (c_{ij}) – евклидовы расстояния на плоскости, то есть матрица симметрична и удовлетворяет неравенству треугольника

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

Теорема 2

Если существует приближенный полиномиальный алгоритм A и константа r, $1 \leq r < \infty$ такие, что для любого примера I ЗК верно $A(I) \leq r \cdot OPT(I)$, 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 ЗК, положив

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

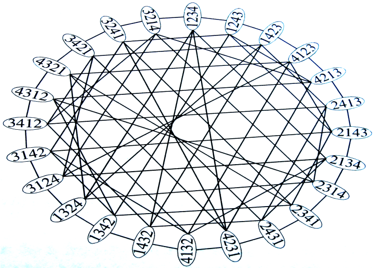
Применим алгоритм A и посмотрим на ответ.
Если получили цикл длины n, то граф G содержит гамильтонов цикл.
Если длина цикла больше n, то она не меньше чем $n \cdot r + (n-1)$, так как включает вес хотя бы одного из "тяжелых" ребер. Но в этом случае граф G не может иметь гамильтонов цикл, т.к. как алгоритм A ошибается не более чем в r раз и ответ в ЗК 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 $P = NP$.

Понятие окрестности

$s \in S$ — допустимое решение
$\mathcal{N} \subseteq S \times S$ — окрестность на множестве S
$\mathcal{N}(s) = \{s' \in S | \mathcal{N}(s, s')\} \subseteq S$ — окрестность решения s

окрестность k-замена

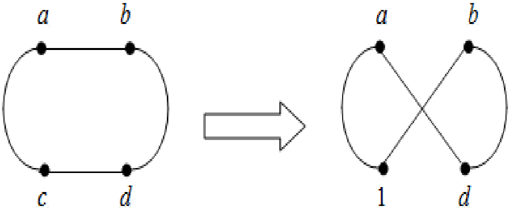
решения отличаются в k компонентах $\mathcal{N}_k(s) = \{s' \in S | d(s, s') = k\}$,
$1 \leq k \leq n$ *где d — расстояние Хэмминга, n — число компонент в решении.*



Множество всех решений заданных перестановкой и их соседи по окрестности 2-замена,
$|\mathcal{N}_2(s)| = \frac{n(n-1)}{2}$

Окрестность 2 - замена

Выбираем в два несмежных ребра и заменяем их другими так, чтобы снова получился гамильтонов цикл.



Общая схема стандартного алгоритма локального спуска

Шаг 1. Выбрать начальное решение $s \in S$, вычислить $f(s)$;
Шаг 2. Выбрать наилучшее соседнее решение s' для s:

$$f(s') = \min\{f(s'')|s'' \in \mathcal{N}(s)\};$$

Шаг 3. Если $f(s') < f(s)$, то $s := s'$ и вернуться на шаг 2
иначе STOP, s – локальный минимум.

Сколько шагов может потребоваться алгоритму локального спуска?

Общая схема стандартного алгоритма локального спуска

Шаг 1. Выбрать начальное решение $s \in S$, вычислить $f(s)$;
Шаг 2. Выбрать наилучшее соседнее решение s' для s:

$$f(s') = \min\{f(s'')|s'' \in \mathcal{N}(s)\};$$

Шаг 3. Если $f(s') < f(s)$, то $s := s'$ и вернуться на шаг 2
иначе STOP, s – локальный минимум.

Сколько шагов может потребоваться алгоритму локального спуска?

- экспоненциальное число шагов

Погрешность локальных оптимумов

Теорема 3

Пусть A *– алгоритм локального спуска и окрестность* $\mathcal{N}(s)$ *имеет полиномиальную мощность. Если существует константа* r, $1 \leq r < \infty$ *такая, что для любого примера* I *ЗК справедливо неравенство* $A(I) \leq r \cdot OPT(I)$, *то* $P = NP$.

Доказательство. (Аналогично предыдущему)

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

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

Выбираем произвольный цикл s_0. Его длина в худшем случае равна $f(s_0) = (n \cdot r) \cdot n$. 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, т.к. окрестность $\mathcal{N}(s)$ имеет полиномиальную мощность. Следовательно, алгоритм A является полиномиальным для данного класса примеров и, как следует из доказательства предыдущей теоремы, A – точный алгоритм. Значит $P = NP$.

Алгоритм "Иди в ближайший из непройденных городов"($A_Б$)

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

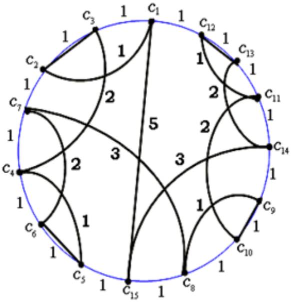
Для любого $r > 1$ найдется пример I ЗК такой, что $A_Б(I) \geq r \cdot OPT(I)$ даже при условии, что $c_{ij} \leq c_{ik} + c_{kj}$, для всех $1 \leq i, j, k \leq n$.

Пример работы алгоритма $A_Б$

15 городов, расположены на окружности длиной 15 единиц.

$OPT(I) = 15$

Алгоритм $A_Б$ получает 27 единиц.



Вероятностный аналог алгоритма $A_{\text{Б}}$

Алгоритм $A_{\text{Б}}(\alpha)$, $\alpha \geq 1$
На k-м шаге формируем список кандидатов

$$I_k(\alpha) = \{j \neq i_1, \dots, i_k | c_{i_k j} \leq \alpha \cdot \min_{j \neq i_1, \dots, i_k} c_{i_k j}\}$$

Следующий город i_{k+1} выбирается из списка кандидатов случайным образом.
При $\alpha = 1$ получаем алгоритм "Иди в ближайший из непройденных городов".

Итерационный вероятностный алгоритм $A_{\text{Б}}(\alpha)$

1. Полагаем рекорд $f^\star := \infty$.
2. Цикл по $t := 1, \dots, T$:

   Применяем алгоритм $A_{\text{Б}}(\alpha)$ и получаем s_t.

   Применяем алгоритм локального спуска с начальным решением s_t.

   Если полученный локальный минимум $\tilde{s}_t$ меньше рекорда, то поменять рекорд: $f(\tilde{s}_t) < f^\star$, то $f^\star := f(\tilde{s}_t)$.

Поиск с чередующимися окрестностями (VNS: N. Mladenović, P. Hansen, 1997)

Задана система разнотипных окрестностей $\mathcal{N}_1, \ldots, \mathcal{N}_k$.

Выбрать начальное решение $s \in S$, вычислить $f(s)$
Пока не выполнен критерий остановки повторять:
  $i := 1;$
   Пока $i \leq k$ выполнить:
    выбрать решение $s' \in \mathcal{N}_i(s)$ случайным образом;
    применить алгоритм локального спуска с окрестностью $\mathcal{N}$ к s',
    s''—полученное решение
    если $f(s'') < f(s)$, то $s := s''$
    иначе $i := i + 1;$
Выдать в качестве ответа s — локальный оптимум.

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

Задача об остовном дереве минимального веса

Дано

связный граф $G = (V, E)$, *каждому ребру* $e \in E$ *приписан вес* $w_e \geq 0$.

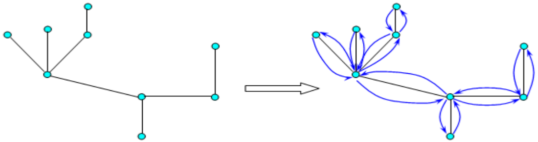
Найти

остовное дерево в G *с минимальным суммарным весом ребер*

Алгоритм Крускала A_K дает точное решение задачи и нижнюю оценку для ЗК: $A_K(I) \leq OPT(I)$.

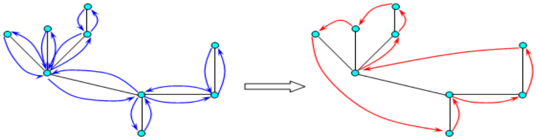
Двойной обход остовного дерева

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 ЗК, т.е. $A_{ST}(I) \leq 2OPT(I)$.

Доказательство

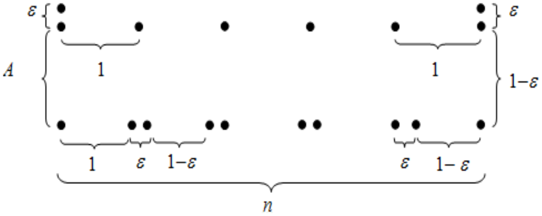
Для длины двойного обхода имеем $2A_K(I) \leq 2OPT(I)$. Пусть новое ребро e, не содержащееся в двойном обходе, заменяет цепочку ребер $e_1, e_2, \ldots, e_k$. Из неравенства треугольника следует, что $w_e \leq \sum_{i=1}^{k} w_{e_i}$, т.е. $A_{ST}(I) \leq 2A_K(I) \leq 2OPT(I)$.

Теорема 6

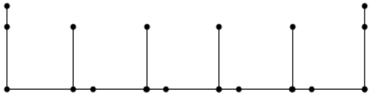
Полученная гарантированная оценка точности $A_{ST}(I) \leq 2OPT(I)$ *является неулучшаемой.*

Доказательство

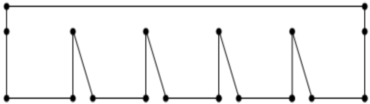
Приведем пример семейства исходных данных ЗК, на котором оценка 2 достигается асимптотически.
Рассмотрим следующий пример на плоскости с $3(n+1)$ *вершинами:*



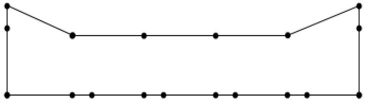
Для этого примера минимальное остовное дерево имеет вид:
Длина остовного дерева $A_K = n + (n+1)(1-\varepsilon) + 2\varepsilon$
Алгоритм перестройки двойного обхода получит решение



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

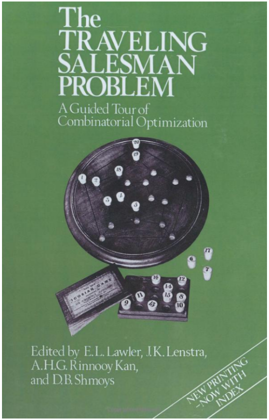


Оптимальное решение задачи $OPT(I) \approx 2n + 2$.



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

Рекомендованная литература



E. L. Lawler, Jan Karel Lenstra, A. H. G. Rinnooy Kan, D. B. Shmoys The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization. Wiley; 1 edition, September 1985