计算机问题求解 – 论题4-10

- 旅行推销商问题（TSP）
- 课程研讨
  - JH第4章第3节第5小节

Traveling Salesperson Problem (TSP)

Input: A weighted complete graph (G, c), where $G = (V, E)$ and $c : E \to \mathbb{N}$. Let $V = \{v_1, \ldots, v_n\}$ for some $n \in \mathbb{N} - \{0\}$.

Constraints: For every input instance (G, c), $\mathcal{M}(G, c) = \{v_{i_1}, v_{i_2}, \ldots, v_{i_n}, v_{i_1} \mid (i_1, i_2, \ldots, i_n)$ is a permutation of $(1, 2, \ldots, n)\}$, i.e., the set of all Hamiltonian cycles of G.

Costs: For every Hamiltonian cycle $H = v_{i_1} v_{i_2} \ldots v_{i_n} v_{i_1} \in \mathcal{M}(G, c)$, $cost((v_{i_1}, v_{i_2}, \ldots v_{i_n}, v_{i_1}), (G, c)) = \sum_{j=1}^{n} c(\{v_{i_j}, v_{i_{(j \bmod n)+1}}\})$, i.e., the cost of every Hamiltonian cycle H is the sum of the weights of all edges of H.

Goal: *minimum*.

原问题很难，也许子问题没有那么难

The **metric traveling salesperson problem, Δ-TSP**, is a subproblem of TSP such that every problem instance (G, c) of Δ-TSP satisfies the triangle inequality

$$c(\{u, v\}) \leq c(\{u, w\}) + c(\{w, v\})$$

for all vertices u, w, v of G.

我们先考虑简单一点的: Δ-TSP

Algorithm 4.3.5.1.

Input: A complete graph $G = (V, E)$, and a cost function $c : E \to \mathbb{N}^+$ satisfying the triangle inequality

$$c(\{u, v\}) \leq c(\{u, w\}) + c(\{w, v\})$$

Δ-TSP

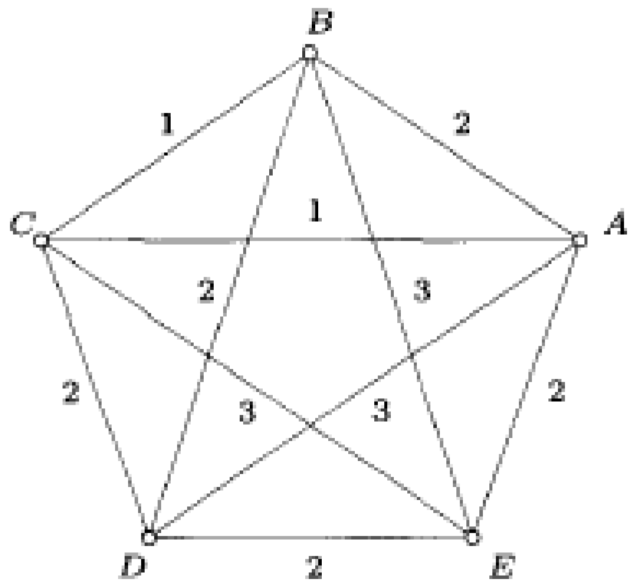
for all three different $u, v, w \in V$ {i.e., $(G, c) \in L_\Delta$}.

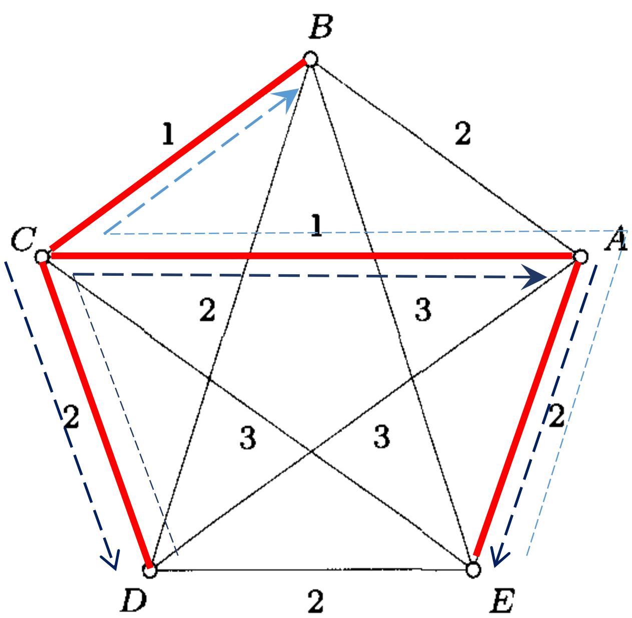
Step 1: Construct a minimal spanning tree T of G according to c.

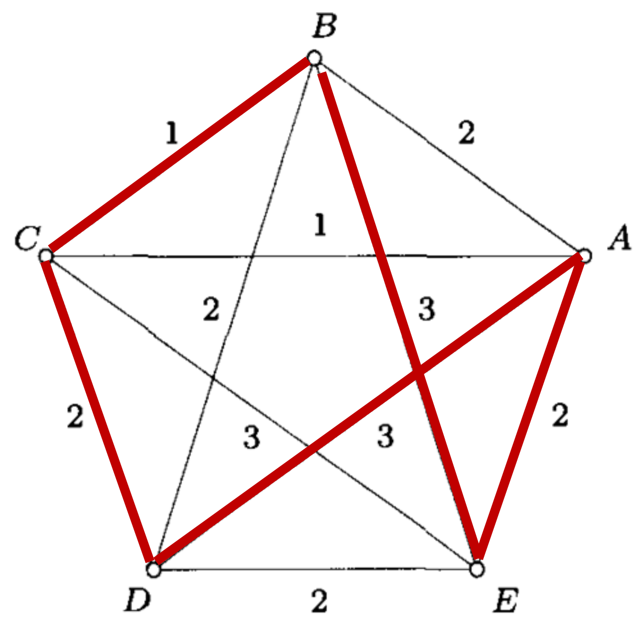
Step 2: Choose an arbitrary vertex $v \in V$. Perform depth-first-search of T from v, and order the vertices in the order that they are visited. Let H be the resulting sequence.

Output: The Hamiltonian tour $\overline{H} = H, v$.

问题：算法的基本思路

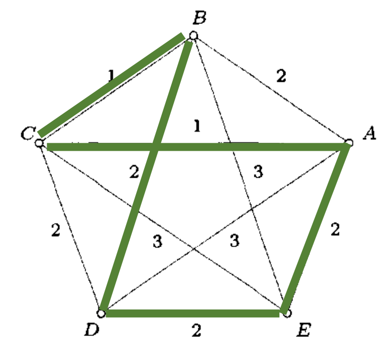






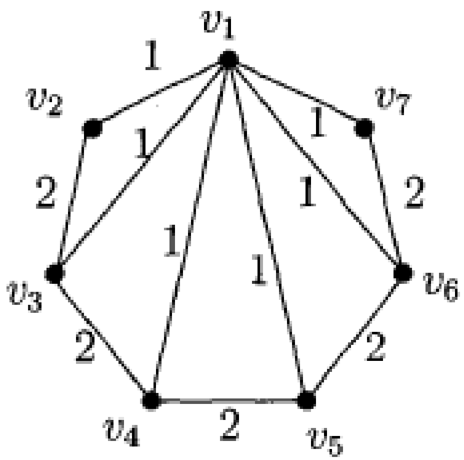
cost(T)=2+3+2+3+1=11

$cost_{opt}$=1+1+2+2+2=8

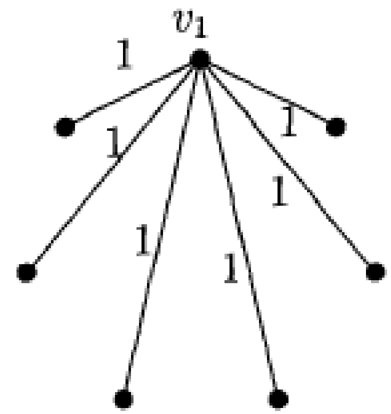


问题：

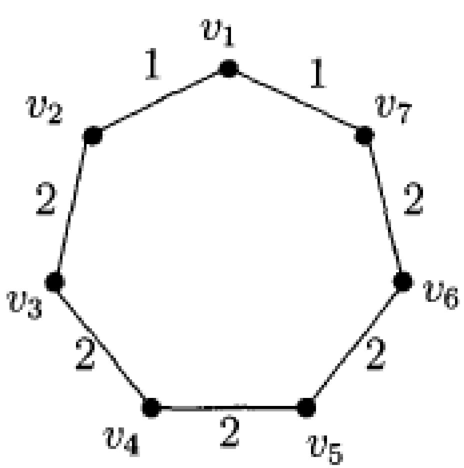
- 三角不等式在这里有什么意义？
- 算法近似比？
- 相对误差最坏的例子？


v_1
v_2
v_3
v_4
v_5
v_6
v_7
1
1
1
1
1
1
2
2
2
2
2
2

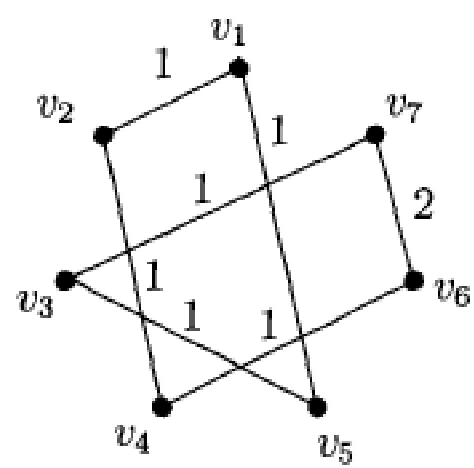
(a)


v_1
1
1
1
1
1
1

(b)


v_1
v_2
v_3
v_4
v_5
v_6
v_7
1
1
2
2
2
2
2

(c)


v_1
v_2
v_3
v_4
v_5
v_6
v_7
1
1
1
2
1
1
1

(d)

Algorithm 4.3.5.4. CHRISTOFIDES ALGORITHM

Input: A complete graph $G = (V, E)$, and a cost function $c : E \to \mathbb{N}^+$ satisfying the triangle inequality.

Step 1: Construct a minimal spanning tree T of G according to c.

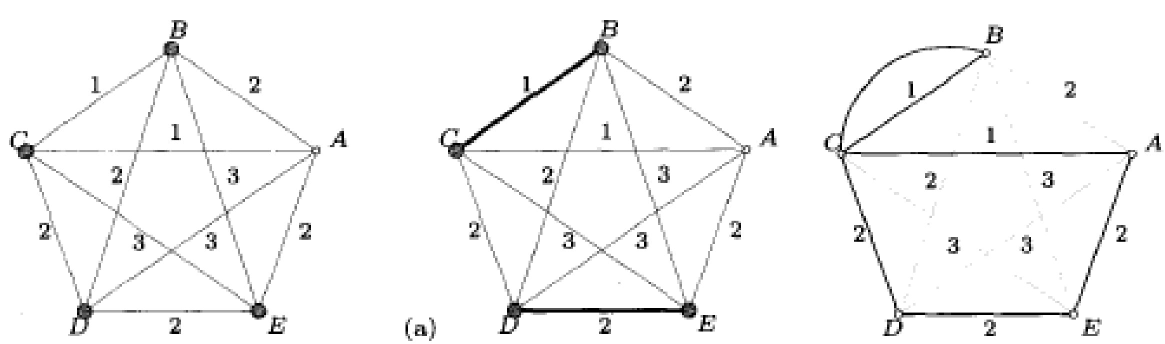
Step 2: $S := \{v \in V \mid deg_T(v) \text{ is odd}\}$.

Step 3: Compute a minimum-weight[21] perfect[22] matching M on S in G.

Step 4: Create the multigraph $G' = (V, E(T) \cup M)$ and construct an Eulerian tour ω in G'.

Step 5: Construct a Hamiltonian tour H of G by shortening ω (i.e., by removing all repetitions of the occurrences of every vertex in ω in one run via ω from the left to the right).

Output: H.



(a) (b)

问题:

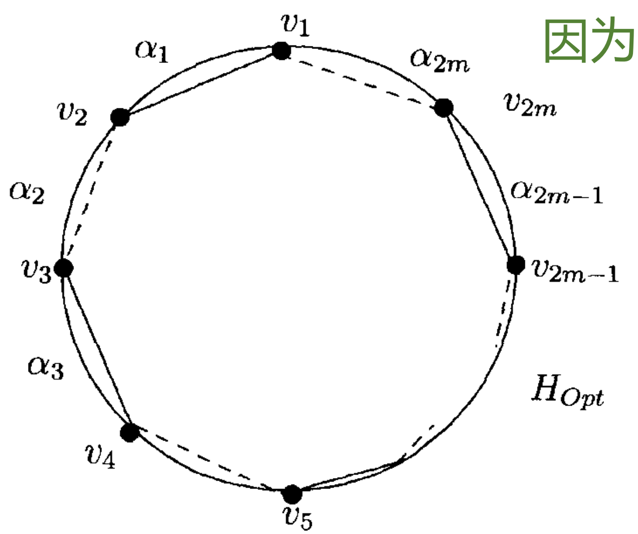
- 算法思路?
- 时间复杂度?
- 算法近似比?
- 最坏例子?

cost(M)的上限

假设共有$2m$个奇次顶点: $\{v_1, \ldots, v_{2m}\}$ 并以此次序出现在最优解中。

$$M_1 := \{\{v_1, v_2\}, \{v_3, v_4\}, \ldots, \{v_{2m-1}, v_{2m}\}\}, \text{ and}$$
$$M_2 := \{\{v_2, v_3\}, \{v_4, v_5\}, \ldots, \{v_{2m-2}, v_{2m-1}\}, \{v_{2m}, v_1\}\}$$

均是S的完美匹配，所以不会小于最小完美匹配M。



因为三角不等式，最优解的权不小于上述两个匹配的权之和:

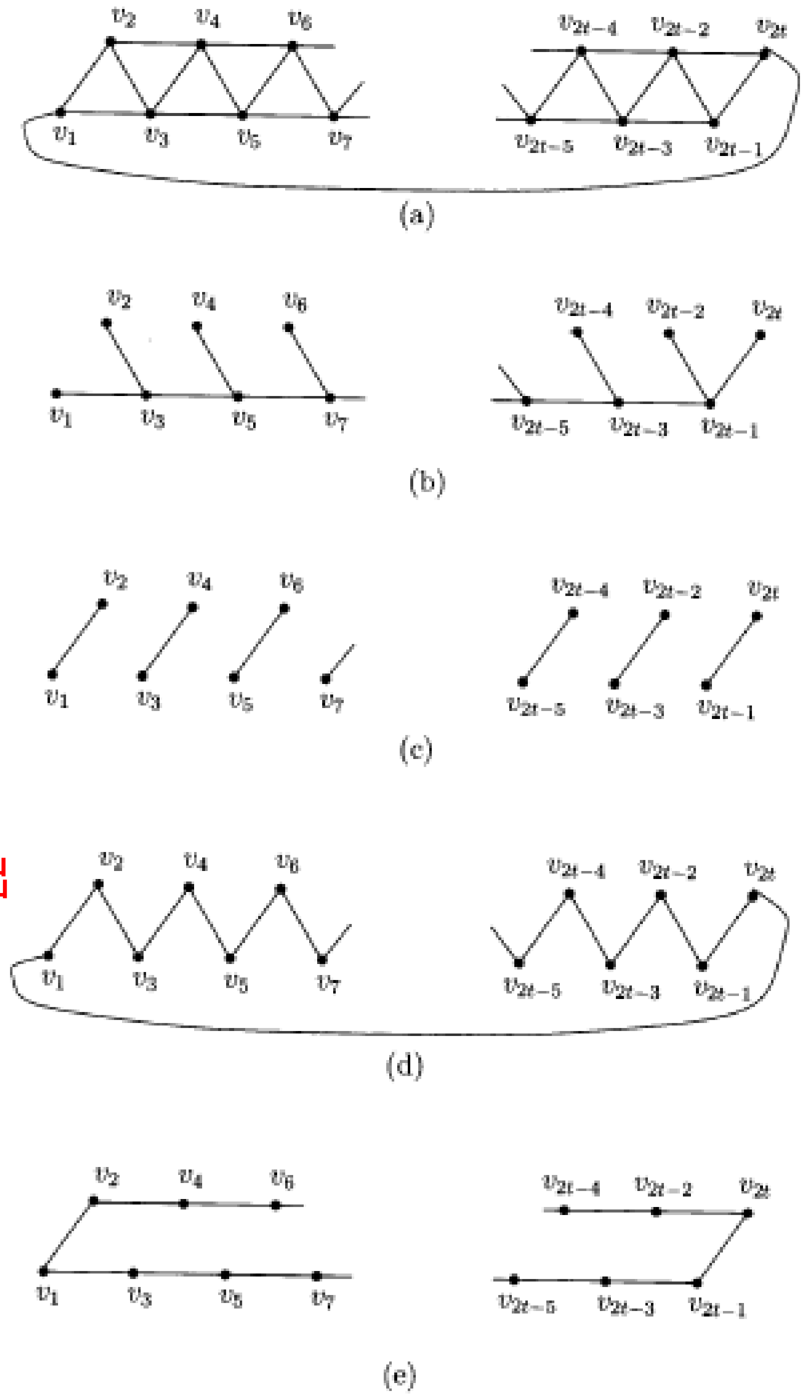
$$cost(H_{Opt}) \geq \sum_{i=1}^{2m} c(\{v_i, v_{i+1}\}) = cost(M_1) + cost(M_2).$$

$$cost(M_1) \geq cost(M) \text{ and } cost(M_2) \geq cost(M).$$

$$cost(M) \leq \min\{cost(M_1), cost(M_2)\} \leq \frac{1}{2} cost(H_{Opt}).$$

$$cost(H) \underset{(4.37)}{\leq} cost(T) + cost(M) \leq cost(H_{Opt}) + \frac{1}{2} cost(H_{Opt}) = \frac{3}{2} cost(H_{Opt}).$$

最优解去除一条边可得到一个生成树，
所以其权不可能小于最小生成树的权。



算法的输出

最优解

$$\frac{cost(H)}{cost(H_{Opt})} = \frac{3t-1}{2t},$$

Δ-TSP的不可近似性

- Papadimitriou and Vempala (2006): 220/219
- Lampis (2014): 185/184
- Karpinski, Lampis, and Schmied (2015): 123/122

(gap)

- Christofides (1976): 3/2

Δ-TSP→TSP

- There are many real inputs that are not in L_{Δ}, but that do not break the triangle inequality too much.

- 基本思路：衡量与Δ-TSP的距离

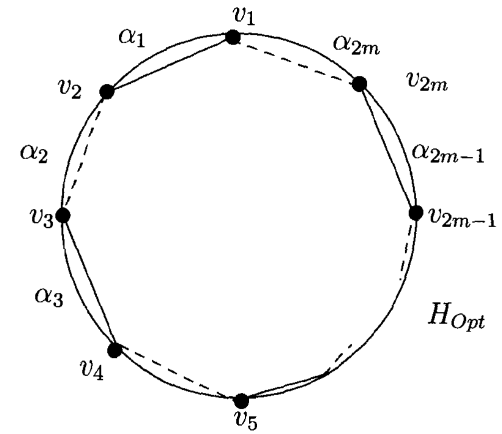
如果 $dist(G, c) \le r$, 则图中任意一条边的权不大于连接其两个端点的长度为2的通路的总权的(1+r)倍；

如果 $distance(G, c) \le r$, 则图中任意一条边的权不大于连接其两个端点的**任意**一条通路的总权的(1+r)倍。

Christofides算法解TSP问题

Lemma 4.3.5.9. CHRISTOFIDES ALGORITHM *is stable according to distance.*

问题：证明的基本思路？



基本逻辑与证明Christofides算法的近似度完全一样，差别只在于：$c(v_i v_{i+1}) \leq (1+r)\, c(v_i \alpha_i v_{i+1})$

因此：

$$cost(M_I) \leq \frac{1}{2}(1+r) \cdot cost(H_{Opt})$$

于是：

$$cost(\omega_I) \underset{(4.44)}{=} cost(T_I) + cost(M_I) \underset{\substack{(4.38)\\(4.46)}}{\leq} cost(H_{Opt}) + (1+r) \cdot cost(H_{Opt})$$

$$= (2+r) \cdot cost(H_{Opt}) \qquad (4.47)$$

Finally,

$$cost(H_I) \underset{(4.43)}{\leq} (1+r) \cdot cost(\omega_I) \underset{(4.47)}{\leq} (1+r) \cdot (2+r) \cdot cost(H_{Opt}).$$

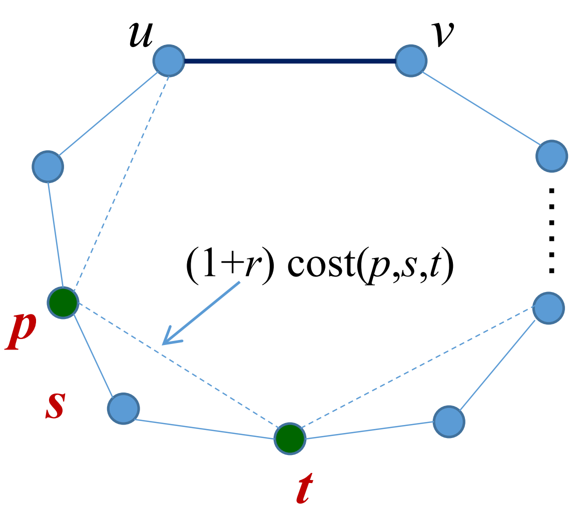
从$distance$到 $dist$

Let, for every positive real number r,

$$\triangle\text{-TSP}_r = (\Sigma_I, \Sigma_O, L, Ball_{r,dist}(L_\triangle), \mathcal{M}, cost, minimum).$$

Lemma 4.3.5.12. *For every positive real number r,* CHRISTOFIDES ALGORITHM *is $(r, O(n^{\log_2((1+r)^2)}))$-quasistable for dist.*

理解证明的关键：



that reducing the length m of a path to the length $\lceil m/2 \rceil$ increases the cost of the connection between u and v by at most $(1+r)$ times. After at most $\lceil \log_2 m \rceil$ such reduction steps one reduces the path v, P, u of length m to the path v, u, and

$$cost(u,v) = c(\{v,u\}) \leq (1+r)^{\lceil \log_2 m \rceil} \cdot cost(v,P,u). \quad (4.48)$$

问题：近似率恶化较快的原因?

Algorithm 4.3.5.18. SEKANINA'S ALGORITHM

Input:	A complete graph $G = (V, E)$, and a cost function $c : E \to \mathbb{N}^+$.
Step 1:	Construct a minimal spanning tree T of G according to c.
Step 2:	Construct T^3.
Step 3:	Find a Hamiltonian tour H in T^3 such that $P_T(H)$ contains every edge of T exactly twice.
Output:	H.

问题：
- 算法的基本思路
- 算法近似比证明的基本思路
- 算法的意义（尽管近似比并不很好）

Theorem 4.3.5.19. SEKANINA'S ALGORITHM *is a polynomial-time* 2*-approximation algorithm for* $\triangle$-TSP.

Proof. Obviously, Step 1 and 2 of SEKANINA'S ALGORITHM can be performed in time $O(n^2)$. Using Lemma 4.3.5.17 one can implement Step 3 in time $O(n)$. Thus, the time complexity of SEKANINA'S ALGORITHM is in $O(n^2)$.

Let H_{Opt} be an optimal solution for an input instance (G, c) of $\triangle$-TSP. Following the inequality (4.32) we have $cost(T) \leq cost(H_{Opt})$. The output H of SEKANINA'S ALGORITHM can be viewed as shortening the path $P_T(H)$ by removing repetitions of vertices in $P_T(H)$. Since $P_T(H)$ contains every edge of T exactly twice,

$$cost(P_T(H)) = 2 \cdot cost(T) \underset{(4.32)}{\leq} 2 \cdot cost(H_{Opt}). \tag{4.51}$$

Since H is obtained from $P_T(H)$ by exchanging simple subpaths by an edge, and c satisfies the triangle inequality,

$$cost(H) \leq cost(P_T(H)). \tag{4.52}$$

Combining (4.51) and (4.52) we obtain $cost(H) \leq 2 \cdot cost(H_{Opt})$. □

得到TSP问题的更好解

- 有什么进一步的思路吗？
- 如何对TSP问题的所有实例进行划分？
  - *dist*
  - *p*-strengthen triangle inequality
  - *EUC*-TSP
- 换一个比Δ-TSP 更好的Kernel