

近似演算法與旅行推銷員問題

完整教材 (依據 5CCS2FC2 Foundations of Computing II 第七週內容編寫)

最佳化問題、近似比、2OPT 演算法與不可近似性

整理自 Dr. Christopher Hampson(King's College London) 之投影片
並參考補充文獻擴充編寫

2026 年 6 月

本週學習目標

1. 分辨判定問題與最佳化問題, 並理解最佳化問題的解空間、成本函數與全域/區域最佳。
2. 認識旅行推銷員問題 (TSP), 理解其判定版本是 NP-complete, 以及最佳化版與判定版的等價性。
3. 掌握近似比的定義、 R -可近似與不可近似的概念。
4. 理解 2OPT 近似演算法 (MST + 前序走訪 + swap), 並證明它對度量 TSP 的近似比為 2。
5. 認識 Christofides 的 $3/2$ 近似, 以及一般 TSP 在 $P \neq NP$ 下不可近似的證明。

Contents

1 最佳化問題與近似演算法	3
1.1 判定問題 vs. 最佳化問題	3
1.2 最佳化問題的框架	3
2 旅行推銷員問題 (TSP)	3
2.1 最佳化版與判定版等價	4
3 近似比 (Approximation Ratio)	4
4 2OPT 近似演算法	5
4.1 2OPT swap move	5
4.2 完整的 2OPT 演算法	5
5 證明近似比 $R = 2$	6
6 一般 TSP 不可近似	7
7 本週重點整理	9
8 練習題 (附解答)	9

1 最佳化問題與近似演算法

1.1 判定問題 vs. 最佳化問題

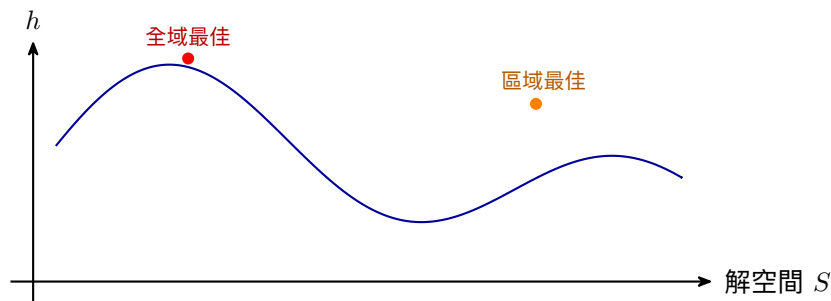
判定問題 (Decision)	最佳化問題 (Optimisation)
判斷解是否存在 (回答 Yes/No)	找出最佳的解
布林可滿足性 SAT	團問題 CLIQUE(最大團)
停機問題	頂點覆蓋 VERTEX-COVER(最小)
漢米頓迴圈問題	旅行推銷員問題 TSP
圖同構問題等	等

複雜度理論 (P、NP) 是以判定問題為主角; 但現實世界更常面對最佳化問題。兩者關係密切: 一個最佳化問題通常都有對應的判定版本 (「是否存在成本 $\leq D$ 的解?」)。

1.2 最佳化問題的框架

定義 1.1 (最佳化問題). 一個最佳化問題由以下構成:

- 解空間 $S = \{\text{所有可能的解}\}$;
- 成本函數 $h(\text{解}) = \text{該解的價值}$;
- 全域最佳 $s_{\text{global}}: h(s_{\text{global}}) \geq h(s)$ 對所有 $s \in S$ (以最大化為例);
- 區域最佳 $s_{\text{local}}: h(s_{\text{local}}) \geq h(s)$ 只對鄰近的 s 成立。



備註 1.2 (區域最佳的陷阱). 爬山式 (貪婪/局部搜尋) 演算法只能保證走到區域最佳, 未必是全域最佳——這正是近似演算法的核心難題, 也是上週 GSAT 卡住的原因。

2 旅行推銷員問題 (TSP)

定義 2.1 (TSP —最佳化版). 旅行推銷員問題 TSP:

輸入 一個完全加權圖 $G = (V, d)$, 其中 $d: V \times V \rightarrow \mathbb{Q}$ 是距離函數。

輸出 走遍所有頂點的最短漢米頓迴圈。

解空間 $S = \{\text{所有路線 } H\}$, 成本 $h(H) = \text{len}(H) = \sum_{(x,y) \in H} d(x,y)$ 。

定義 2.2 (TSP —判定版). TSP_{dec}:

輸入 完全加權圖 $G = (V, d)$, 以及門檻值 $D \in \mathbb{Q}$ 。

輸出 **True** 若且唯若 G 含一條漢米頓迴圈 H 使得 $\text{len}(H) \leq D$ 。

定理 2.3. TSP_{dec} 是 *NP-complete*。

備註 2.4. NP 性質顯然 (給一條路線可在多項式時間驗證長度); NP-hard 可由漢米頓迴圈問題 $HAM \leq_p TSP_{\text{dec}}$ 得到——把原圖的邊設距離 1、非邊設一個大數即可 (詳見第 6 節的歸約)。

2.1 最佳化版與判定版等價

定理 2.5. TSP 可在多項式時間求解 $\iff TSP_{\text{dec}}$ 可在多項式時間求解。

Proof. ($\Rightarrow$) 若能解最佳化版, 則用最佳路線 H 的長度與 D 比較即可回答判定版:

Algorithm 1 $TSP\text{-}DEC(G, D)$ ——由最佳化版實作

- 1: 用最佳化演算法求最佳路線 H
 - 2: **if** $\text{len}(H) \leq D$ **then return** 路線 H
 - 3: **else return** **False**
-

($\Leftarrow$) 反之, 若能解判定版, 則用二分搜尋逼近最佳長度。下界 $D_- = 0$, 上界 $D_+ =$ 所有邊總權重; 反覆對中點 mid 詢問判定版, 直到區間小於最短邊長 (此時區間內至多一個可行整數長度, 即最佳值)。

Algorithm 2 $TSP(G)$ ——由判定版二分搜尋實作

- 1: $D_- \leftarrow 0$; $D_+ \leftarrow G$ 所有邊的總權重
 - 2: **while** $(D_+ - D_-) \geq G$ 的最短邊長 **do**
 - 3: $\text{mid} \leftarrow (D_+ + D_-)/2$
 - 4: **if** $TSP_{\text{dec}}(G, \text{mid}) = H$ **then** $D_+ \leftarrow \text{mid}$
 - 5: **else** $D_- \leftarrow \text{mid}$
 - 6: **return** 長度為 mid 的路線 H
-

二分搜尋只需 $O(\log(\text{總權重}))$ 次詢問, 故為多項式。 □

備註 2.6. 此定理說明: 對 TSP , 「找最佳解」與「判斷是否存在夠短的解」難度相同。既然 TSP_{dec} 是 *NP-complete*, 最佳化版 TSP 也無多項式精確演算法 (除非 $P = NP$)。於是我們退而求其次, 尋找近似演算法。

3 近似比 (Approximation Ratio)

定義 3.1 (近似比). 近似演算法 M 在輸入 w 上的近似比是一個 ≥ 1 的實數:

$$R_M(w) = \max\left(\frac{C_{\text{approx}}}{C_{\text{global}}}, \frac{C_{\text{global}}}{C_{\text{approx}}}\right) = \begin{cases} C_{\text{approx}}/C_{\text{global}} & \text{若 } C_{\text{global}} \leq C_{\text{approx}} \text{ (最小化問題)} \\ C_{\text{global}}/C_{\text{approx}} & \text{若 } C_{\text{global}} > C_{\text{approx}} \text{ (最大化問題)} \end{cases}$$

其中 C_{approx} 是近似解的成本, C_{global} 是最佳解的成本。取 $\max$ 是為了讓比值無論最大化或最小化都 ≥ 1 , $R = 1$ 代表完全最佳。

定義 3.2 (R -可近似). 一個問題 L 稱為 R -可近似的, 若存在多項式時間近似演算法 M , 使得

$$R_M(w) \leq R \quad \text{對所有 } w \in \Sigma^*, \quad \text{即 } C_{\text{approx}} \leq R \times C_{\text{global}}.$$

定義 3.3 (不可近似). 一個問題 L 稱為不可近似 (unapproximable) 的, 若對任何固定 $R > 0, L$ 都不是 R -可近似的。換言之, 任何多項式時間近似演算法, 對某些輸入都會給出任意差的解。

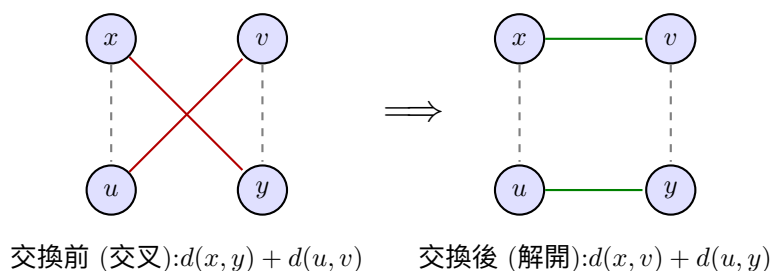
4 2OPT 近似演算法

2OPT 是針對 TSP 的經典局部搜尋: 反覆執行「2OPT swap move」——把兩條互相交叉 (或可改善) 的邊「解開」。

4.1 2OPT swap move

2OPT swap move 的四個步驟

1. 給定路線 H , 選兩條不相鄰的邊, 設為 (x, y) 與 (u, v) 。
2. 比較交換前後的權重: $d(x, y) + d(u, v)$ 對 $d(x, v) + d(u, y)$ 。
3. 只要交換能降低總權重就替換這兩條邊。
4. 確認交換後不會把迴圈拆成兩段 (仍是單一迴圈)!



當 $d(x, v) + d(u, y) < d(x, y) + d(u, v)$ 時, 交換可縮短路線。幾何上, 這正是把一個「交叉」解開——在平面歐氏距離下, 交叉的路線一定比解開後長。

Algorithm 3 SWAP-MOVE(H)

- 1: for H 中每一對不相鄰的邊 (x, y) 與 (u, v) do
 - 2: if $d(x, v) + d(u, y) < d(x, y) + d(u, v)$ then
 - 3: 令 H' 為把 $(x, y), (u, v)$ 替換成 $(x, v), (u, y)$ 的結果
 - 4: return H'
 - 5: return 未改變的 H (找不到改善)
-

4.2 完整的 2OPT 演算法

2OPT 需要一個初始路線。投影片採用「最小生成樹的前序走訪」作為起點 (這也是稍後證明近似比的關鍵):

Algorithm 4 2OPT(G)

- 1: 建構 G 的最小生成樹 T_0
 - 2: 令 H_0 為 T_0 的前序走訪 (preorder traversal) 所得的迴圈
 - 3: $i \leftarrow 0$
 - 4: **while** H_i 尚非 (區域) 最佳 **do**
 - 5: $H_{i+1} \leftarrow \text{SWAP-MOVE}(H_i); \quad i \leftarrow i + 1$
 - 6: **return** H_i
-

範例 (投影片 8 頂點)

從 MST 的前序走訪得到初始路線

$$H_0 : A, B, C, H, D, E, F, G, A, \quad \text{len}(H_0) \approx 19.06.$$

連續施行 swap move, 每次解開一個交叉、縮短長度:

$$H_1 : A, B, C, H, G, F, E, D, A (\approx 16.08) \rightarrow H_2 : A, B, C, H, F, G, E, D, A (\approx 14.71).$$

H_2 無法再改善, 為 (區域) 最佳解, 長度 ≈ 14.71 。

5 證明近似比 $R = 2$

定理 5.1. 對滿足三角不等式的 (度量)TSP, 2OPT 演算法的近似比為 $R = 2$, 即

$$\text{len}(H_{\text{approx}}) \leq 2 \times \text{len}(H_{\text{global}}).$$

Proof. 設 H_{approx} 為 2OPT 找到的區域最佳、 H_{global} 為全域最佳漢米頓迴圈。

步驟 1(目標). 要證 $\text{len}(H_{\text{approx}}) \leq 2 \text{len}(H_{\text{global}})$ 。

步驟 2(拿掉一條邊). 從 H_{global} 移除任一條邊, 得到一條路徑 T (它是生成樹)。因為去掉了非負權的邊,

$$\text{len}(T) \leq \text{len}(H_{\text{global}}).$$

步驟 3(MST 更短). 設 T_0 為 2OPT 第一步算出的最小生成樹。由於 T 也是一棵生成樹、而 T_0 是最小者,

$$\text{len}(T_0) \leq \text{len}(T).$$

步驟 4(前序走訪 $\leq$ 兩倍 MST). 前序走訪相當於把 MST 的每條邊「各走兩遍」(一去一回的歐拉走訪), 再用三角不等式把重複造訪的頂點「抄捷徑」略過。故

$$\text{len}(H_0) \leq 2 \times \text{len}(T_0).$$

步驟 5(swap 只會變短). 每次 2OPT swap 都使長度不增:

$$\text{len}(H_{\text{approx}}) \leq \cdots \leq \text{len}(H_1) \leq \text{len}(H_0).$$

步驟 6(串起來). 合併以上不等式:

$$\text{len}(H_{\text{approx}}) \leq \text{len}(H_0) \leq 2 \text{len}(T_0) \leq 2 \text{len}(T) \leq 2 \text{len}(H_{\text{global}}).$$

□



推論 5.2. 度量 TSP(距離滿足三角不等式) 是2-可近似的。

定理 5.3 (Christofides, 1976). 度量 TSP 是 $\frac{3}{2}$ -可近似的。

Christofides 演算法概要 (把 2 改進到 $\frac{3}{2}$)

2OPT 的初始路線把 MST 「每邊加倍」以得到歐拉圖 (故是 2 倍)。Christofides 用更省的方式造歐拉圖:

1. 建最小生成樹 T ;
2. 取 T 中奇度數頂點集合 O (由握手引理, $|O|$ 為偶數);
3. 在 O 上求最小權完美匹配 M , 令 $R = T \cup M$ ——此時所有頂點皆為偶度, R 是歐拉圖;
4. 求歐拉迴圈, 再抄捷徑 (三角不等式) 得漢米頓迴圈。

關鍵界限: $\text{len}(T) \leq \text{OPT}$, 且 $\text{len}(M) \leq \frac{1}{2}\text{OPT}$ (把最佳迴圈在 O 上抄捷徑後, 可拆成兩個完美匹配, 較便宜者 $\leq \frac{1}{2}\text{OPT}$)。故總長 $\leq \frac{3}{2}\text{OPT}$ 。Christofides 是度量 TSP 數十年來最佳的近似比 (直到 2020 年才有些微突破)。

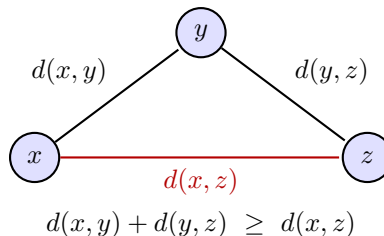
6 一般 TSP 不可近似

第 5 節的好消息建立在三角不等式上。若距離不滿足三角不等式, 情況急轉直下。

定義 6.1 (三角不等式). 距離函數 d 滿足三角不等式, 若對所有 x, y, z :

$$d(x, y) + d(y, z) \geq d(x, z).$$

即「繞經 y 」絕不會比「直接從 x 到 z 」更短——直線最短。



定理 6.2. 若 d 不必滿足三角不等式, 則一般的旅行推銷員問題 TSP 是不可近似的 (除非 $P = NP$)。

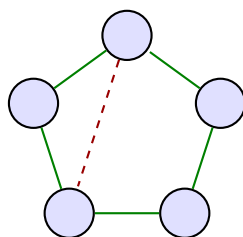
證明 (由漢米頓迴圈歸約). **步驟 1(反證假設)**. 假設存在多項式時間、近似比為 R 的近似演算法, 即

$$\text{len}(H_{\text{approx}}) \leq R \times \text{len}(H_{\text{global}}).$$

步驟 2(目標). 取任一漢米頓迴圈問題實例 $G = (V, E), n = |V|$. 我們要造一個 TSP 實例, 使其有「夠短」的近似解 $\iff G$ 有漢米頓迴圈。

步驟 3(造圖). 在相同頂點上造完全加權圖 $G' = (V, d)$:

$$d(x, y) = \begin{cases} 1 & \text{若 } (x, y) \in E \text{ (「短邊」)} \\ nR + 1 & \text{若 } (x, y) \notin E \text{ (「長邊」)} \end{cases}$$



綠實線: E 中的邊 (權 1); 紅虛線: 非邊 (權 $nR + 1$)

步驟 4(關鍵等價). 設 H_{approx} 為近似演算法回傳的迴圈。則

$$\text{len}(H_{\text{approx}}) \leq nR \iff G \text{ 有漢米頓迴圈}.$$

($\Leftarrow$) 若 G 有漢米頓迴圈, 則存在只用「短邊」的 TSP 迴圈 H_{global} , 其長度

$$\text{len}(H_{\text{global}}) = \underbrace{1 + 1 + \cdots + 1}_{n \text{ 條邊}} = n.$$

由近似保證, $\text{len}(H_{\text{approx}}) \leq R \times n = nR$ 。

($\Rightarrow$) 反之, 若 $\text{len}(H_{\text{approx}}) \leq nR$, 則此迴圈不可能用到任何「長邊」(任一條長邊就貢獻 $nR + 1 > nR$)。故 H_{approx} 只用短邊 $\Rightarrow$ 它是 G 的漢米頓迴圈。

步驟 5(導出快速演算法). 於是可用這個假設的近似演算法, 在多項式時間內精確判定漢米頓迴圈:

Algorithm 5 FAST-HAM(G)

- 1: $n \leftarrow |V|$; 依步驟 3 造完全加權圖 G'
 - 2: $H_{\text{approx}} \leftarrow$ 對 G' 跑 R -近似演算法
 - 3: **if** $\text{len}(H_{\text{approx}}) \leq nR$ **then return True**
 - 4: **else return False**
-

步驟 6(矛盾). 漢米頓迴圈問題是 NP-complete。若上述演算法存在, 則 $\text{HAM} \in \text{P}$, 推得 $\text{P} = \text{NP}$ 。故在 $\text{P} \neq \text{NP}$ 的假設下, 一般 TSP 不可近似。 $\square$

距離 vs. 時間: 一個現實的對照

- 以「距離」最小化的路線規劃是可近似的: 實體距離滿足三角不等式 (繞路不會更近), 故可用 2OPT/Christofides 求得好的近似。
- 以「時間」最小化卻不可近似!: 因為塞車、單行道、轉乘等, 「繞路」有時反而更快, 時間不滿足三角不等式——此時 TSP 退化成一般情形, 不可近似。

歸約 $\text{HAM} \leq_p \text{TSP}_{\text{dec}}$ 同時也證明了 TSP_{dec} 的 NP-hardness。

「旅行推銷員問題不是一個問題, 而是一種癮。」——Christos Papadimitriou

7 本週重點整理

主題	重點
判定 vs. 最佳化	判定問判斷「是否存在」; 最佳化問「最佳解」; 兩者常等價 (二分搜尋)
TSP	完全加權圖上的最短漢米頓迴圈; TSP_{dec} 是 NP-complete
近似比	$R_M(w) = \max(C_{\text{approx}}/C_{\text{global}}, C_{\text{global}}/C_{\text{approx}}) \geq 1$; R -可近似 $\Leftrightarrow C_{\text{approx}} \leq R C_{\text{global}}$
2OPT	MST $\rightarrow$ 前序走訪 $\rightarrow$ swap 解交叉; 區域最佳
近似比證明	$\text{len}(H_{\text{approx}}) \leq \text{len}(H_0) \leq 2\text{len}(T_0) \leq 2\text{len}(T) \leq 2\text{len}(H_{\text{global}})$
度量 TSP	三角不等式下: 2OPT 給 2-近似; Christofides 給 3/2-近似
一般 TSP	無三角不等式時不可近似 (除非 $\text{P} = \text{NP}$); 由 HAM 以「gap」歸約

8 練習題 (附解答)

練習 1. 某最小化問題的最佳成本 $C_{\text{global}} = 100$, 演算法給出 $C_{\text{approx}} = 130$ 。近似比 $R_M(w)$ 為何? 該演算法是否為 1.5-近似?

解答

$R_M = 130/100 = 1.3$ 。因 $1.3 \leq 1.5$, 在此輸入上滿足 1.5-近似 (但要稱「1.5-可近似」需對所有輸入都 ≤ 1.5)。

練習 2. 在 2OPT 的近似比證明中, 「前序走訪 $\leq$ 兩倍 MST」這一步用到了哪個性質? 為什麼需要它?

解答

用到三角不等式。前序走訪相當於把 MST 每邊走兩遍 (長度 $= 2\text{len}(T_0)$), 再把重複造訪的頂點「抄捷徑」略過; 三角不等式保證抄捷徑不會增加長度, 故結果 $\leq 2\text{len}(T_0)$ 。沒有三角不等式, 抄捷徑可能變長, 證明失效。

練習 3. 在不可近似性證明中, 若把非邊距離設為 $nR + 1$, 當 G 沒有漢米頓迴圈時, 任何迴圈的長度下界是多少? 為何 $> nR$?

解答

任何迴圈至少用一條非邊 (否則它就是 G 的漢米頓迴圈)。故長度 $\geq (n - 1) \cdot 1 + (nR + 1) = n + nR > nR$ 。因此「長度 $\leq nR$ 」可精確區分有無漢米頓迴圈。

練習 4. 為什麼「以距離為成本」的路線規劃可近似, 而「以時間為成本」的卻不可近似?

解答

實體距離滿足三角不等式 (直線最短、繞路不會更近), 屬度量 TSP, 可用 2OPT/Christofides 近似。但行車「時間」會因塞車、單行道而使繞路更快, 違反三角不等式, 退化為一般 TSP, 不可近似。

練習 5 (思考題). Christofides 為何能用「最小權完美匹配」而非「MST 邊加倍」? 其節省的關鍵不等式是什麼?

解答

歐拉圖只要求「所有頂點偶度」。MST 中只有奇度頂點需要修正, 且其個數為偶 (握手引理), 故只需在這些頂點上加一個完美匹配, 而非把整棵樹加倍。關鍵不等式為 $\text{len}(M) \leq \frac{1}{2}\text{OPT}$: 把最佳迴圈在奇度頂點上抄捷徑後可拆成兩個完美匹配, 較便宜者不超過 $\frac{1}{2}\text{OPT}$ 。於是總長 $\leq \text{len}(T) + \text{len}(M) \leq \text{OPT} + \frac{1}{2}\text{OPT} = \frac{3}{2}\text{OPT}$ 。

參考資料

1. C. Hampson, *5CCS2FC2 Foundations of Computing II, Week 7* 投影片 (approximation / 2opt / proving_2opt / unapproximable), King's College London.
2. N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem," Technical Report, CMU, 1976.
3. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, 3rd/4th ed., MIT Press. (近似演算法、TSP 章節)
4. D. P. Williamson, D. B. Shmoys, *The Design of Approximation Algorithms*, Cambridge Univ. Press, 2011.
5. V. V. Vazirani, *Approximation Algorithms*, Springer, 2003.
6. Wikipedia: *Travelling salesman problem; Christofides algorithm; 2-opt; Hardness of approximation*.