Traveling Salesperson Problem (TSP)

$$\text{Minimize } Z = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}\, x_{ij}$$

st

$$\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1, 2, \ldots, n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1, 2, \ldots, n$$

$$x_{ij} = (0, 1)$$

$$\sum_{i,j \in S} x_{ij} \le |S| - 1$$

distance matrix

	1	2	3	4	5
1		2	3	4	5
2	2		6	7	8
3	3	6		9	10
4	4	7	9		11
5	5	8	10	11	

$$x_{ij} = \begin{cases} 1, & \text{if city } j \text{ is reached from city } i. \\ 0, & \text{otherwise} \end{cases}$$

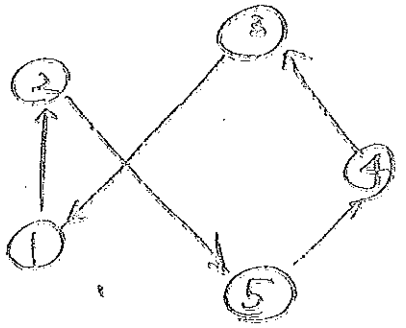
$$\text{Minimize } Z = 2x_{12} + 3x_{13} + 4x_{14} + 5x_{15}$$
$$+ 2x_{21} + 6x_{23} + 7x_{24} + 8x_{25}$$
$$+ 3x_{31} + 6x_{32} + 9x_{34} + 10x_{35}$$
$$+ 4x_{41} + 7x_{42} + 9x_{43} + 11x_{45}$$
$$+ 5x_{51} + 8x_{52} + 10x_{53} + 11x_{54}$$

subject to

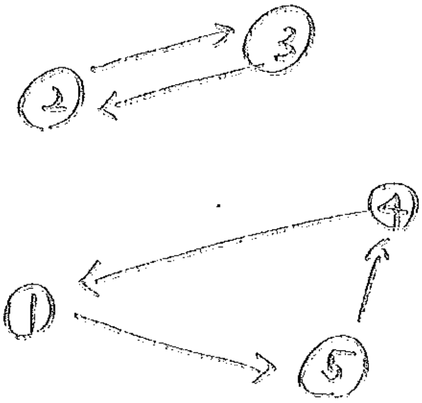
$$x_{12} + x_{13} + x_{14} + x_{15} = 1$$
$$x_{21} + x_{23} + x_{24} + x_{25} = 1$$
$$x_{31} + x_{32} + x_{34} + x_{35} = 1$$
$$x_{41} + x_{42} + x_{43} + x_{45} = 1$$
$$x_{51} + x_{52} + x_{53} + x_{54} = 1$$

$$x_{21} + x_{31} + x_{41} + x_{51} = 1$$
$$x_{12} + x_{32} + x_{42} + x_{52} = 1$$
$$x_{13} + x_{23} + x_{43} + x_{53} = 1$$
$$x_{14} + x_{24} + x_{34} + x_{54} = 1$$
$$x_{15} + x_{25} + x_{35} + x_{45} = 1$$

$$\forall\, x_{ij} \in (0, 1).$$



Tour solution



Subtour solution

$X_{12} = X_{25} = X_{54} = X_{43} = X_{31} = 1$

$X_{23} = X_{32} = X_{15} = X_{54} = X_{41} = 1$.

subtour breaking

$$X_{23} + X_{32} \le 1$$

$$X_{15} + X_{51} + X_{45} + X_{54} + X_{41} + X_{14} \le 2.$$

The Optional Stop TSP

$$y_j = \begin{cases} 1 & \text{if city } j \text{ is visited} \\ 0, & \text{otherwise} \end{cases}$$

V_j : the value of visiting city j

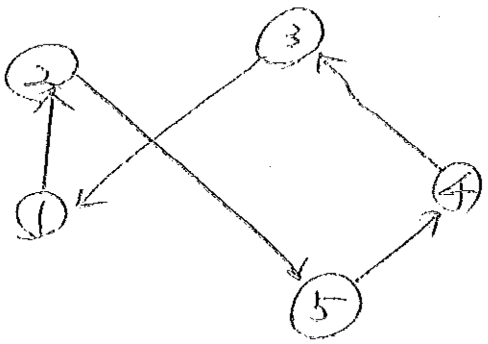
$$\text{Min} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} - \sum_{j=1}^{n} V_j y_j$$

$$\sum_{i \neq j} x_{ij} = y_j$$

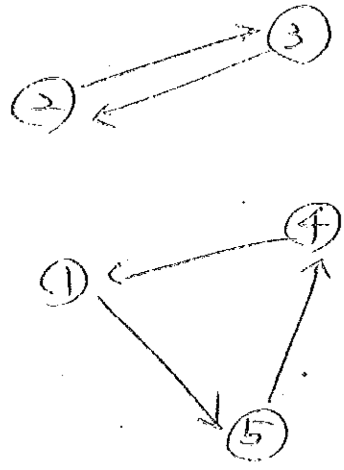
$$\sum_{k \neq j} x_{jk} = y_j$$

$$\sum_{i,j \in S} x_{ij} \le |S| - 1$$

$$X_{12} = X_{25} = X_{54} = X_{43} = X_{31} = 1$$



Tour Solution



Subtour solution

$$X_{23} = X_{32} = X_{15} = X_{54} = X_{41} = 1$$

Subtour breaking

$$X_{23} + X_{32} \leq 1$$

$$X_{15} + X_{51} + X_{45} + X_{54} + X_{41} + X_{14} \leq 2$$

☆ 會隨問題加大變得越困難，產生新的變化，求解之速度快慢會因人而異(功力)。所以當問題變大時，i.e. 50個以上，一般用以啟發式演算法求解。

	1	2	3	4	5
1	0	132	217	164	58
2	132	0	290	201	79
3	217	290	0	113	303
4	164	201	113	0	196
5	58	79	303	196	0

the nearest-neighbor heuristic (NNH) for TSP

可從任一節點開始，在未被拜訪的城市中，選擇城市為目前城市距離最近的。

1－5－2－4－3－1 optimal（距離 668）

3－4－1－5－2－3 距離 704 非最佳

一常見受歡迎的 NNH 為計算所有節點開始之距離，再從中找出最佳解。

the cheapest insertion (CIH) for TSP

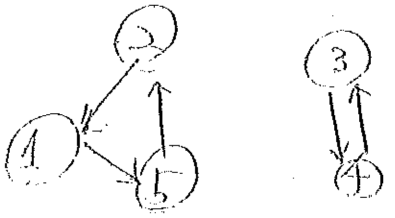
從任一節點開始，擇一與目前節點最近的節點。(i, j) 將 (i, j) 替換成 (i, k) 與 (k, j) 其中 k 為不在 (i, j) 中，k 使增加之距離最短（計算方式 $C_{ik} + C_{kj} - C_{ij}$）。一開始 (1, 5) 和 (5, 1)

k = 2 或 3 或 4

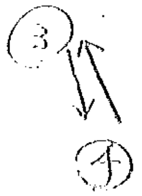
(1, 5)	(1, 2) - (2, 5)	$C_{12} + C_{25} - C_{15} = 153$ * 任選一
(1, 5)	(1, 3) - (3, 5)	$C_{13} + C_{35} - C_{15} = 462$
(1, 5)	(1, 4) - (4, 5)	$C_{14} + C_{45} - C_{15} = 302$
(5, 1)	(5, 2) - (2, 1)	$C_{52} + C_{21} - C_{51} = 153$ *
(5, 1)	(5, 3) - (3, 1)	$C_{53} + C_{31} - C_{51} = 462$
(5, 1)	(5, 4) - (4, 1)	$C_{54} + C_{41} - C_{51} = 302$

任何 X_{ij} 包含 subtour 將會違反上述限制式

例如 $X_{15}=X_{52}=X_{21}=X_{43}=X_{34}=1$ 包含兩個 subtour



選擇不包含城市 1 之 subtour



$$\left.\begin{aligned} u_3 - u_4 + 5X_{34} &\le 4 \\ u_4 - u_3 + 5X_{43} &\le 4 \end{aligned}\right\} \xrightarrow{\text{加總兩式產生}} 5X_{34} + 5X_{43} \le 8$$

$\Rightarrow$ 不可能產生 $X_{34}=X_{43}=1$

1. Any set of X_{ij}'s containing a subtour will be infeasible (that is, they violate the constraint).

2. Any set of X_{ij}'s that forms a tour will be feasible (there will exist a set of u_j's that satisfy the constraint)

令 t_i：為城市 i 為路徑中的位置(拜訪順序)

令 $u_i = t_i$ 將會滿足限制式

例如 1 - 3 - 4 - 5 - 2 - 1

$\underset{\text{城市}}{u_1} = \underset{\text{位置}}{1}$ $u_3=2$ $u_4=3$ $u_5=4$ $u_2=5$

例如考慮 $x_{ij}=1$，例如 x_{52}

對應 $u_5 - u_2 + 5x_{52} \le 4$

城市2緊接在城市5之後，則 $u_5 - u_2 = -1$

$-1 + 5x_{52} \le 4$　　$5x_{52} \le 5$ 成立。

考慮 $x_{ij} = 0$ 例如 $x_{32} = 0$、

$$u_3 - u_2 + 5x_{32} \le 4 \quad \because x_{32} = 0$$

$$u_3 - u_2 \le 4 \quad \because u_3 \le 5\text{、且 } u_2 \ge 1$$

$5 - 2 = 3$，∴ $u_3 - u_2$ 不可能超過3。

亦滿足上述限制式

$$u_i - u_j + p x_{ij} \le p-1 \quad \text{for } 2 \le i \ne j \le n \quad O(n^2)$$

subtour elimination constraints (subtour prevention constraints)

初期 (1, 2)(2, 5)-(5, 1) $k=3$ 式 十

(1, 2)	(1, 3)-(3, 2)	$C_{13}+C_{32}-C_{12} = 375$
(1, 2)	(1, 4)-(4, 2)	$C_{14}+C_{42}-C_{12} = 233$ *
(2, 5)	(2, 3)-(3, 5)	$C_{23}+C_{35}-C_{25} = 514$
(2, 5)	(2, 4)-(4, 5)	$C_{24}+C_{45}-C_{25} = 318$
(5, 1)	(5, 3)-(3, 1)	$C_{53}+C_{31}-C_{51} = 462$
(5, 1)	(5, 4)-(4, 1)	$C_{54}+C_{41}-C_{51} = 302$

(1, 4)-(4, 2)-(2, 5), (5, 1) $k=3$

(1, 4)	(1, 3)-(3, 4)	$C_{13}+C_{34}-C_{14} = 166$ *
(4, 2)	(4, 3)-(3, 2)	$C_{43}+C_{32}-C_{42} = 202$
(2, 5)	(2, 3)-(3, 5)	$C_{23}+C_{35}-C_{25} = 514$
(5, 1)	(5, 3)-(3, 1)	$C_{53}+C_{31}-C_{51} = 462$

(1, 3)(3, 4)-(4, 2)-(2, 5)-(5, 1)

optimal

The Traveling Salesman Problem — QA 164 T697 清大

A Guided Tour of Combinatorial Optimization

Edit by

E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan.

D. B. Shmoys. John Wiley & Sons Ltd 1985

Evaluation of Heuristics

The following three methods have been suggested for evaluating heuristics:

1. Performance guarantees
2. Probabilitistic analysis
3. Emprical analysis

1. gives a worst-case bound on how far away from optimality a tour constructed by the heuristic can be. (complexity). NNH 很差.

2. a heuristic is evaluated by assuming that the location of cities follows some known probability distribution.

$$\frac{\text{Expected length of the path found by the heuristic}}{\text{Expected length of an optimal tour}}$$

越接近 1, 越好

例如 城市是独立分布, uniformly distributed on a cube of unit length, width and height.

3. Empirical analysis: heuristics are compared to the optimal solution for a number of problems for which the optimal tour is known.

A Comparison of the growth of some polynomial functions to that of certain exponential functions

	Function	Approximate values		
polynomial function	n	10	100	1,000
	$n \log n$	33	664	9966
	n^3	1000	1,000,000	10^9
	$10^6 n^8$	10^4	10^{22}	10^{33}
	2^n	1024	1.27×10^{30}	1.05×10^{301}
	$n^{\log n}$	2099	1.93×10^{13}	2.89×10^{29}
	$n!$	3628800	10^{158}	4×10^{2567}

call an algorithm "good" when it is sufficiently efficient to be useable in practice. (if its worst-case complexity is bounded by a polynomial function of n.

O (big-O notation) to express the running time function

$O(n \log n)$ $O(n^3)$, 等.