中華民國第四十六屆中小學科學展覽會
作品說明書

高中組　數學科

040420

群最大圓半徑量測之 TSP 演算法

學校名稱： 國立馬祖高級中學

作者：	指導老師：
高二 劉瑞軒	胡裕仁
高二 林芃薇	洪儒
高二 林孟俞	
高二 曹景瑄	

關 鍵 詞：維度遞減矩陣運算、推銷員路徑

群最大圓半徑量測之 TSP 演算法

壹、摘要

在此次的科展中，利用了高中所學的知識，來進行有關推銷員路徑問題之探討。透過所學的基本原理[2][3][4]，來找出解決推銷員路徑問題，並提出一套可供電腦演算的路徑規劃演算法，並結合合計概程式設計概念，實際開發一套程式作驗證。

研究結果：

一. 理論分析：

引用高中數學所學的原理，並利用高二學的數學概念，來量測最短相對路徑，再配合定義出的維度遞減矩陣運算方式，及數學上常用的歐幾里德距離量測方法，來推算並分析路徑問題，最後找出其關係式以進行電腦模擬演算。

二. 演算法內容：參考內文第 13 頁

三. 經模擬後我們發現所提的方法可以順利解決推銷員問題（TSP），且透過電腦運算可以更快速地算出「路徑組合」，免去辛苦的人力規劃配對運算。

貳、研究動機

在中華民國全國中小學科展第四十三屆及第四十五屆高中數學組中有二件作品，分別爲「神奇推銷員」[6]及「迴圈迷宮探索-一筆劃問題」[7]，提到如果一個圖形有推銷員路徑，也就是說該圖形可以一筆劃連接完成所有點的路徑規劃問題，其路線必可用記憶輪來表示，亦即有記憶輪一定可以表示有推銷員路線的圖形，而含有推銷員路徑的圖形，至少會有一條可以標示成記憶輪的路徑，可惜該作品所提出的方法只能以人工計算的方式完成而且計算費時費力[1][6]。至於「迴圈迷宮探索-一筆劃問題」中提到任意兩個（以上）圖形，只要接在同一個點上的話均可應用組合數來解題，並且歸納出一些鏈狀及環狀兩種路徑的部分通解。而且該篇文章亦不能以非人工的方式

來求出一筆劃的推銷員路徑，而且也沒證明是最佳路徑，因此我們試著想以不同的方法來解決一筆劃的推銷員路徑規劃問題，希望可以快速地完成推銷員的路徑規劃，以求出最佳組合的解。

進行本次科學研究之前，我們看了很多參考資料及歷屆全國科展的得獎作品，使我們在思索推銷員問題及一筆劃問題之後，發現很多文章作品皆未能有效地以系統方式解決推銷員路徑規劃的問題，而是以費力的人工計算來產生路徑規劃的組合方式，進行最佳路徑編碼規劃，因此我們重新思考，希望能找出以非人工的方式來解決最佳路徑規劃。

因此經過半年多的睡眠不足，及組員與老師無以計數的午餐約會，我們思考出一套新的演算法，來解決由電腦自動規劃出的推銷員路徑問題。即利用高二所學的數學方法[3][4]，再加上計概課老師所提到的一些排序方式，我們設計了一個「維度遞減矩陣運算方式」，並配合「群最大圓半徑量測方法」來完成推銷員問題的路徑組合。

參、研究目的

本文研究目的，在於結合高中數學方法及計概課程中的一些程式設計概念[5]，來完成找出一組有效的推銷員路徑方法(即漢米爾頓環路)使其距離為最短，並利用 Visual basic 程式的數學函數功能模擬產生了三組分別為 x=20、y=20 的單位方格隨機圖、x=25、y=25 的單位方格隨機圖，及 x=30、y=30 單位方格的隨機圖，並且讓此三組圖分別對應生成 20、25、30 三組數目不同的城市點，然後再利用歐幾里德方法求解距離方式，來計算平面上任二個座標點的相對距離，並初始產生 C_2^n 個組合數的相對距離，來開始我們的維度遞減矩陣運算，最後利用本文新提出的「群最大圓半徑量測 TSP 演算法」的演算法則，來求出自動化的推銷員路徑規劃的組合結果。

肆、研究設備及器材

一、電腦壹台。

二、可以編輯數學方程式符號的 MathType、Word 軟體各壹套。

三、繪圖軟體的 Visio、GSP 各壹套。

四、一台彩色印表機。

五、程式設計 Visual basic 軟體壹套

六、一堆不要用的計算紙及四粒清楚的人腦。

伍、研究方法

在了解推銷員路徑問題時，我們先學習一些概念，即漢米爾頓環路，是由漢米爾頓所提出。他造了一個正十二面體，在其中每一個頂點上標示一個城市，並探討是否有一種捷徑存在，使其恰好可經過每一頂點一次的路徑。若滿足此一圖形中的每一頂點、皆可經過每一次的路徑稱爲漢米爾頓路徑，如下圖。

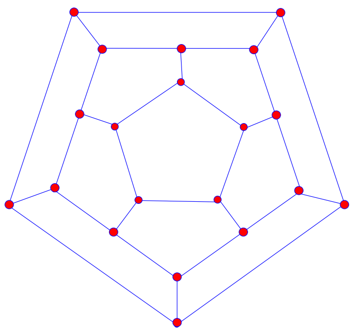


圖 4-1

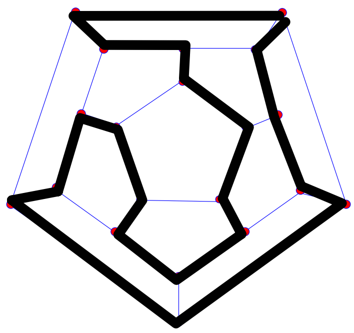


圖 4-2

由於漢米爾頓路徑到目前還沒有找到一個簡單又可快速判斷的充要條件；因此，我們只能利用這類路徑或迴路的基本特性來做判斷。首先，可以從下列之必要條件來判斷一個圖是否有漢米爾頓迴路：

1. 漢米爾頓迴路通過圖中除起點外之任一頂點最多一次。
2. 若一圖有漢米爾頓迴路，則該必具連通性。因爲在非連通性的圖中，不可能有通過每一頂點之迴路。
3. 若一圖有漢米爾頓迴路，則圖中任一頂點之進出次數至少是 2。因爲若有一頂點，其進出次數少於 2，即至多有一個邊連接該頂點，則要通過該頂點的任一迴路務必沿著同一邊回去才能通過其他頂點，如此一來必有一頂點重複通過，此爲矛盾也。

4. 若一圖存在漢米爾頓迴路，則任一條漢米爾頓迴路必經由那些次數為 2 的頂點之兩個邊。
5. 若一頂點，其次數大於 2，在建造一條漢米爾頓迴路時，一旦路過該頂點，則其他與該頂點相連接而尚未使用過的邊都不能再考慮使用。
6. 在一個具有n個頂點的圖中，任一漢米爾頓路徑必恰好經過n-1 個邊;而任一漢米爾頓迴路必恰好經過n個邊。若一圖有漢米爾頓迴路存在，則只要去掉該迴路上的任一個邊，即可得到此圖的漢米爾頓路徑[1][5] 。

一個連通的圖，必然只有一個做為起點的頂點和一個作為終點的頂點，其餘的頂點只能是“過路”的頂點，在過路頂點處必然是“到達”與“離開”成對存在的。
由於規定不用重複的路線，因而每次“到達”與“離開”都用不同的邊代表。有向圖表示是從起點u到終點v，即$\overrightarrow{uv}$這就構成有向圖，對於無向圖表示從起點u到終點v以$\overline{uv}$表示，即點u到點v沒有方向性[5]。

在我們的研究中最難的問題，是如何在n個點的無向圖中，判斷出是否可以一筆劃完成所有的點的連接，透過了本文所提之群最大圓半徑量測之 TSP 演算法及維度遞減矩陣。

我們找出了一套方法來將所有的樣本點予以標記，以確保一定有推銷員路徑，而標記完成之後，再以歐氏方法來求算圖上任兩點的距離，使其產生C_2^n組相對距離再以此設計出一套相對距離矩陣，再來將每一組距離排序，求出在所有資料點中的最大距離，當作群最大圓的直徑，並設群最大圓的直徑中點為地圖上的虛擬圓心O，再以歐幾里德距離方法找出圓心O與所有城市點的相對距離，進行排序以找出距圓心O最短距離的城市點，當作 TSP 路徑規劃的實際起始點。

陸、研究過程

本組主要是利用高二上學期所學的圓的關係概念，來量測出最短路徑，進而提出一套演算法來完成 TSP 路徑規劃問題，其優勢在於可避免複雜的人工計算及時間的浪

費。

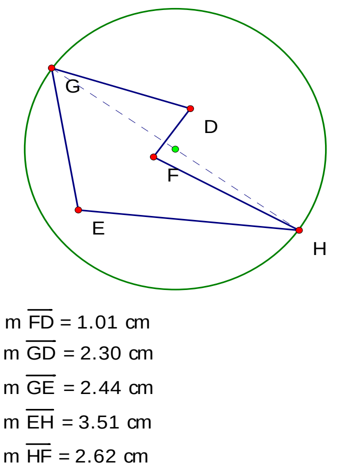


圖 5-1

我們所提的概念是利用在一群隨機散佈的資料點中，如果所有資料點可能存在不同的位置上，如圖 5-1 中的 D-E-F-G-H 五個座標點。首先我們計算出所有資料點的相對距離 $d_{i,j}$ ，$0 < i, j \le n$ ，$i \ne j$ ，n 爲樣本點個數。因此有 $C_2^n = \dfrac{n(n-1)}{2}$ 筆相對距離的 $d_{i,j}$ 存在，之後取出最大 $d_{i,j}$ 即圖 5-1 中的 G 和 H 兩點，找出 $\overline{GH}$ 的中點，並以 $\overline{GH}$ 中點爲圓心， $\dfrac{1}{2}\overline{GH}$ 長爲半徑畫出一圓，設圓心爲 O 點，在圓 O 中找出距圓心最近的點即 F 點，令 F 爲起始點，重新計算 F 與各點的距離，並進行排序找出最小距離並予以連接 F-D；再計算 D 和 E、G、H 三點的距離，並連接最小距離 D-G；仿照此方式即可完成本例的 TSP 路徑規劃組合問題，即 F-D-G-E-H-F。

但是在模擬產生 TSP 的都市點時，可能會有以下的特殊情形產生：

一、城市點發生在正三角形、正方形暨各式正多邊形以及最大圓上的頂點時，因爲此時圓心 O 至圓上各點等距，如圖 5-2a 及圖 5-2b，故可任取一城市點如圖 5-2a 中之 B 點，再依上列法則進行各點連線。

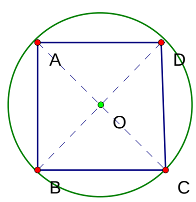


圖 5-2a

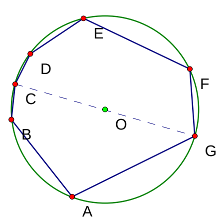


圖 5-2b

二、在下圖 5-3 中 O 為最大圓之圓心且 $\overline{OA}=\overline{OB}$，因此分別以 A 及 B 兩點開始依所提法則進行運算，最後再比較兩者之間的距離和，若以 B 為始點的 TSP 路徑最小，則出現以 B 開始的所有點的順序。

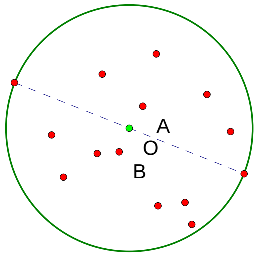


圖 5-3

三、數學架構：

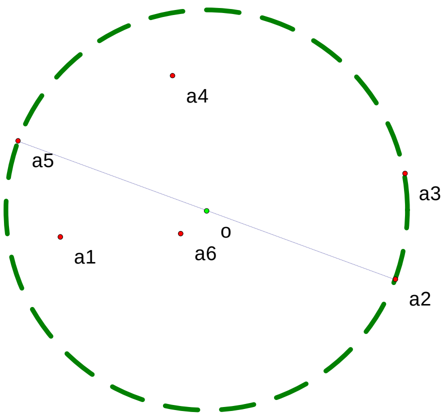


圖 5-4a

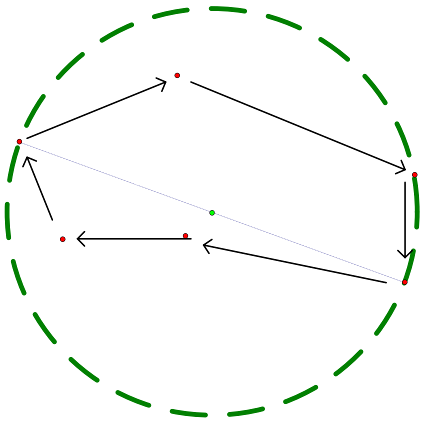


圖 5-4b

在圖 5-4a 中設各點座標如下：$a_1(x_1,y_1)$、$a_2(x_2,y_2)$、$a_3(x_3,y_3)$、$a_4(x_4,y_4)$、$a_5(x_5,y_5)$、$a_6(x_6,y_6)$，最大距離 $\overline{a_2a_5}$，因此圓心座標為 $O(\frac{x_2+x_5}{2},\frac{y_2+y_5}{2})$，故最近點為 a_6。因圖 5-4b 可得路徑順序為 a_6 - a_1 - a_5 - a_4 - a_3 - a_2 - a_6。

$$\begin{array}{c} \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{array}\begin{array}{c} \begin{array}{cccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{array} \\ \begin{bmatrix} 0 & d_{2,1} & d_{3,1} & d_{4,1} & d_{5,1} & d_{6,1} \\ d_{1,2} & 0 & d_{3,2} & d_{4,2} & d_{5,2} & d_{6,2} \\ d_{1,3} & d_{2,3} & 0 & d_{4,3} & d_{5,3} & d_{6,3} \\ d_{1,4} & d_{2,4} & d_{3,4} & 0 & d_{5,4} & d_{6,4} \\ d_{1,5} & d_{2,5} & d_{3,5} & d_{4,5} & 0 & d_{6,5} \\ d_{1,6} & d_{2,6} & d_{3,6} & d_{4,6} & d_{5,6} & 0 \end{bmatrix} \end{array}$$

、$d_{2,5}=\sqrt{(x_2-x_5)^2+(y_2-y_5)^2}=R$，為本群資

料的最大圓直徑，因此取$\overline{a_2a_5}$的中點$O(\frac{x_2+x_5}{2},\frac{y_2+y_5}{2})$。由於 TSP 路徑是無向圖的問題，因此我們只考慮以 0 爲對角線的矩陣內容下半或上半三角形即可，所以只列出方陣的下三角形半邊。在圖 5-4a 中重新計算O點和a_1、a_2、a_3、a_4、a_5、a_6因此可得下式：$d_{O,i}=\min\{d_{O,1},d_{O,2},d_{O,3},d_{O,4},d_{O,5},d_{O,6}\}=d_{O,6}$來產生起始點$a_6$，經維度遞減矩陣運算可得下式：

$$
\begin{array}{c|c}
 & \begin{array}{cccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{array} \\
\begin{array}{c} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{array} &
\begin{bmatrix} 0 & & & & & \\ d_{1,2} & 0 & & & & \\ d_{1,3} & d_{2,3} & 0 & & & \\ d_{1,4} & d_{2,4} & d_{3,4} & 0 & & \\ d_{1,5} & d_{2,5} & d_{3,5} & d_{4,5} & 0 & \\ d_{1,6} & d_{2,6} & d_{3,6} & d_{4,6} & d_{5,6} & 0 \end{bmatrix}
\end{array}
\Rightarrow
\begin{array}{c|c}
 & \begin{array}{cccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{array} \\
a_6 & \begin{bmatrix} d_{1,6} & d_{2,6} & d_{3,6} & d_{4,6} & d_{5,6} & 0 \end{bmatrix}
\end{array}
$$

得出

$d_{1,6}=\min\{d_{1,6},d_{2,6},d_{3,6},d_{4,6},d_{5,6},d_{6,6}=0\}$，連接$\overline{a_6a_1}$；由於$a_6$接$a_1$所以將$a_6$取消

$$
\begin{array}{c|c}
 & \begin{array}{ccccc} a_1 & a_2 & a_3 & a_4 & a_5 \end{array} \\
\begin{array}{c} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{array} &
\begin{bmatrix} 0 & & & & \\ d_{1,2} & 0 & & & \\ d_{1,3} & d_{2,3} & 0 & & \\ d_{1,4} & d_{2,4} & d_{3,4} & 0 & \\ d_{1,5} & d_{2,5} & d_{3,5} & d_{4,5} & 0 \end{bmatrix}
\end{array}
$$

，重新計算a_1與其它各點的相對距離

$$
\begin{array}{c|c}
 & a_1 \\
\begin{array}{c} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{array} &
\begin{bmatrix} 0 \\ d_{1,2} \\ d_{1,3} \\ d_{1,4} \\ d_{1,5} \end{bmatrix}
\end{array}
$$

得出

$d_{1,5}=\min\{d_{1,1}=0,d_{1,2},d_{1,3},d_{1,4},d_{1,5}\}$，連接$\overline{a_1a_5}$；同理將$a_1$取消，重新計算$a_5$與其它各點的相對距離

$$
\begin{array}{c|c}
 & \begin{array}{cccc} a_2 & a_3 & a_4 & a_5 \end{array} \\
\begin{array}{c} a_2 \\ a_3 \\ a_4 \\ a_5 \end{array} &
\begin{bmatrix} 0 & & & \\ d_{2,3} & 0 & & \\ d_{2,4} & d_{3,4} & 0 & \\ d_{2,5} & d_{3,5} & d_{4,5} & 0 \end{bmatrix}
\end{array}
\Rightarrow
\begin{array}{c|c}
 & \begin{array}{cccc} a_2 & a_3 & a_4 & a_5 \end{array} \\
a_5 & \begin{bmatrix} d_{2,5} & d_{3,5} & d_{4,5} & 0 \end{bmatrix}
\end{array}
$$

得出

$d_{4,5}=\min\{d_{2,5},d_{3,5},d_{4,5},d_{5,5}=0\}$，連接$\overline{a_5a_4}$；同理將$a_5$取消，重新計算$a_4$與其它各點的相對距離

$$
\begin{array}{c|c}
 & \begin{array}{ccc} a_2 & a_3 & a_4 \end{array} \\
\begin{array}{c} a_2 \\ a_3 \\ a_4 \end{array} &
\begin{bmatrix} 0 & & \\ d_{2,3} & 0 & \\ d_{2,4} & d_{3,4} & 0 \end{bmatrix}
\end{array}
\Rightarrow
\begin{array}{c|c}
 & \begin{array}{ccc} a_2 & a_3 & a_4 \end{array} \\
a_4 & \begin{bmatrix} d_{2,4} & d_{3,4} & 0 \end{bmatrix}
\end{array}
$$

得出

$d_{3,4}=\min\{d_{2,4},d_{3,4},d_{4,4}=0\}$，連接$\overline{a_4a_3}$；最後將$a_4$取消，重新計算$a_3$與其它各點的

相對距離 $\begin{array}{c|cc} & a_2 & a_3 \\ \hline a_2 & 0 & \\ a_3 & d_{2,3} & 0 \end{array} \Rightarrow \begin{array}{c|cc} & a_2 & a_3 \\ \hline a_3 & d_{2,3} & 0 \end{array}$ 得出$d_{2,3}=\min\{d_{2,3},d_{3,3}=0\}$，連接$\overline{a_3a_2}$；當

方陣算至2×2的維度時，表示沒有其他城市點故直接連接起始點$a_6 \Rightarrow \overline{a_2a_6}$，得路

徑順序為a_6-a_1-a_5-a_4-a_3-a_2-a_6。

四、演算法證明：

定理 1：在一群非共圓的點資料中，最大圓的圓心是唯一且存在。

證明：

存在性證明：

設平面上散佈有n個點（$n>2$），且此n個點可以任意位置擺放，則由歐幾里德距離量測平面上任兩點，必可找出一組位置點（a_i,a_j）$i\neq j$使得i至j的最大歐幾里德距離滿足$d_{i,j}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}$，若將$d_{ij}$視為點資料中最大圓的直徑，設圓心為$O$，則$O$的座標為$(\frac{x_i+x_j}{2},\frac{y_i+y_j}{2})$。

唯一性證明：

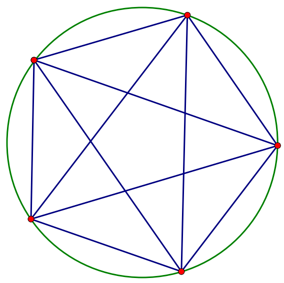
設平面上散佈有n個點（$n>2$），且此n個點可以任意位置擺放，若有二段以上的d_{ij}長度相等且為最大，則該二線段必為相交於一點或平行二種關係之一，或者是散佈點群恰構成為一正多邊形的頂點。若是前者可將最大線段頂點連線構成一多邊形，則多邊形中可由其最大對角線的兩端，來找出唯一的一組最大距離。後者因為有**共圓現象**，為唯一性的例外，若為後者則任取二對角線端點為直徑，依本演算法即可找出最佳路徑。

定理 2：在一群共圓的點資料中，最大圓的圓心是非唯一存在。

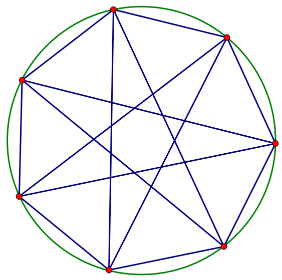
證明：

非唯一性證明：

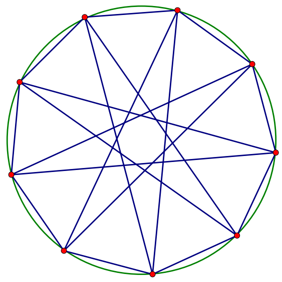
若城市點恰在正奇數 n 邊形的頂點上**共圓現象**，則在正奇數 n 邊形（ $n \geq 5$ ）時，由列舉法可證出有不只一個的最大圓的圓心存在（即頂點與頂點所正對的邊的兩端點皆可構成最大圓的直徑）。



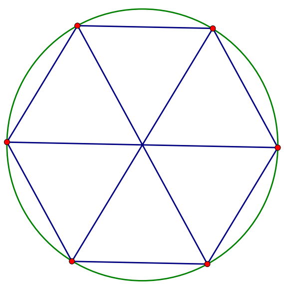
正 5 邊形



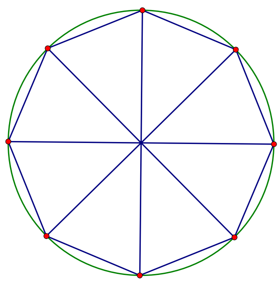
正 7 邊形



正 9 邊形



正 6 邊形



正 8 邊形

…

…

演算法收斂證明：

由第 9 頁的第三點數學架構所列之式，若將其式子引申至 n 維(即有 n 個城市點）本演算法必經 $1+(n-1)$ 次的維度遞減矩陣運算，而後得出一 2×2 的矩陣，因此故必返回起點城市、故得證，即本方法必收斂。

柒、研究結果

我們歸納並提出的演算法如下：

步驟 1：

(1) 初始化所有參數（計算城市點的個數 n ）。

(2) 計算所有城市點 $a_i(x_i, y_i)$ 、 $1 \le i \le n$ 的相對距離 $d_{i,j}^{t=0} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ 、t 表示第幾層的計算，若 $i = j$ 則 $d_{i,j}^{0} = 0$ 。

(3) 找出最大 d_{ij} ，取其中點為點群最大圓圓心 $O(\frac{x_i + x_j}{2}, \frac{y_i + y_j}{2})$ 。

步驟 2：

(1) 計算 o 和全部城市點的最小距離 $\min(d_{O,i})$ 。

(2) 重新設定實際起始城市點 a_i

$$\begin{array}{cc} & \begin{array}{ccccccc} a_1 & a_2 & L & a_i & a_{i+1} & L & a_n \end{array} \\ a_i & \begin{bmatrix} d_{1,i}^{t=1} & d_{2,i}^{1} & L & 0 & d_{i+1,i}^{1} & L & d_{n,i}^{1} \end{bmatrix} \end{array} 。$$

(3) 找出 $\min(d_{i,j}^{t=1})$，$1 \le j \le n$ 且 $i \ne j$ ，連接 $\overline{a_i a_j}$ ，消掉 a_i 重新設定實際起始城市點 a_j ，

$$\begin{array}{cc} & \begin{array}{ccccccc} a_1 & a_2 & L & a_j & a_{j+1} & L & a_n \end{array} \\ a_j & \begin{bmatrix} d_{1,i} & d_{2,i} & L & 0 & d_{j+1,i} & L & d_{n,i} \end{bmatrix} \end{array} ,$$

再找出 $\min(d_{j,k}^{t=2})$ $1 \le k \le n$ 且 $j \ne k$ ，依據步驟 2-(2)及步驟 2-(3)算至 $t = n-1$ 時結束並畫出所有路徑。

步驟 3：

若同時遇到 $\min(d_{i,j}^{t}) = \min(d_{i,k}^{t})$，$1 \le k \le n$ 且 $i \ne j \ne k$ 時，分成二組或更多組重新依步驟 2-(2)及步驟 2-(3)計算，並比較其路徑總和。

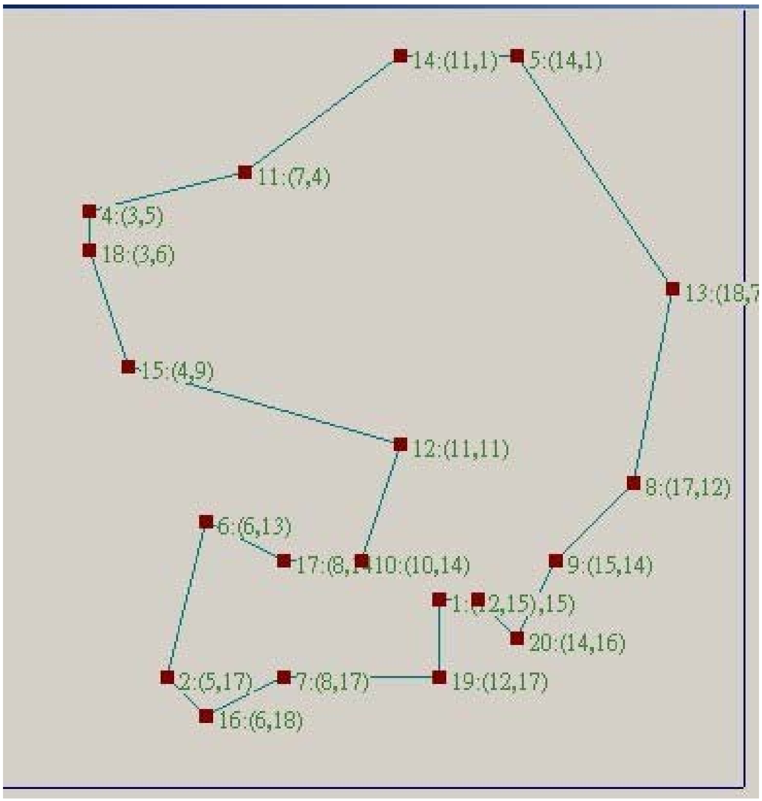
實驗探討一：在 20 乘 20 的方格中，我們以亂數產生 20 個點座標，使其代表地圖上的城市位置，經 $1+(n-1)$ 個演算次數之後得到如下的相對距離及 TSP 路線圖。

順序	城市點	X座標	Y座標
1	12	11	11
2	10	10	14
3	17	8	14
4	6	6	13
5	2	5	17
6	16	6	18
7	7	8	17
8	19	12	17
9	1	12	15
10	3	13	15
11	20	14	16
12	9	15	14
13	8	17	12
14	13	18	7
15	5	14	1
16	14	11	1
17	11	7	4
18	4	3	5
19	18	3	6
20	15	4	9

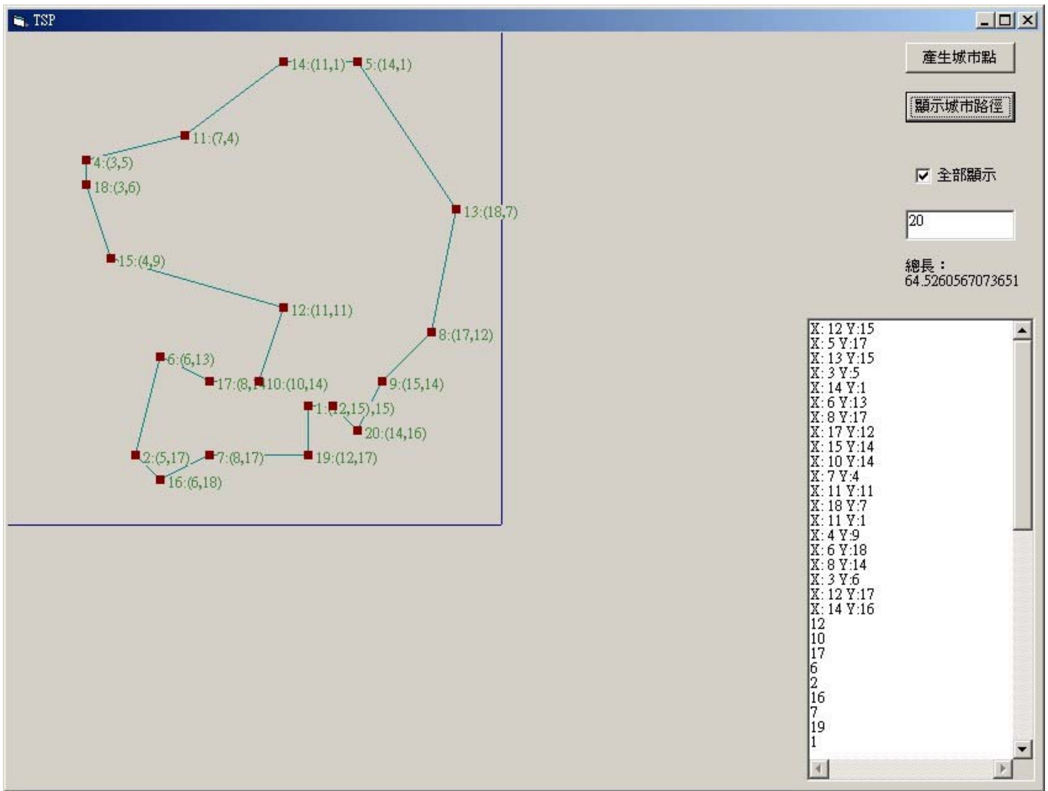
運算次數	相對位置點	相對距離
第1次	12-10	3.16227766016838
第2次	10-17	2
第3次	17-6	2.23606797749979
第4次	6-2	4.12310562561766
第5次	2-16	1.41421356237310
第6次	16-7	2.23606797749979
第7次	7-19	4
第8次	19-1	2
第9次	1-3	1
第10次	3-20	1.41421356237310
第11次	20-9	2.23606797749979
第12次	9-8	2.82842712474619
第13次	8-13	5.09901951359278
第14次	13-5	7.21110255092798
第15次	5-14	3
第16次	14-11	5
第17次	11-4	4.12310562561766
第18次	4-18	1
第19次	18-15	3.16227766016838
第20次	15-12	7.28010988928052

20 個點座標位置表

在上表 20 個點的「相對位置點」，可以很清楚地知道經演算法運算後地圖上的連線，由「城市點順序」爲 9 的位置開始一路完成其規劃順序爲【9-14-8-15-16-5-7-17-12-13-18-1-3-10-6-2-19-20-11-4-9】其圖形結果，如後所示。



TSP-20 路線圖之 1



TSP-20 路線圖之 2

由上圖 TSP-20 路線圖之 1 之 2，我們可以快速地算出推銷員如何以有效的拜訪方式完

成他所要拜訪的客戶，而返回起點。

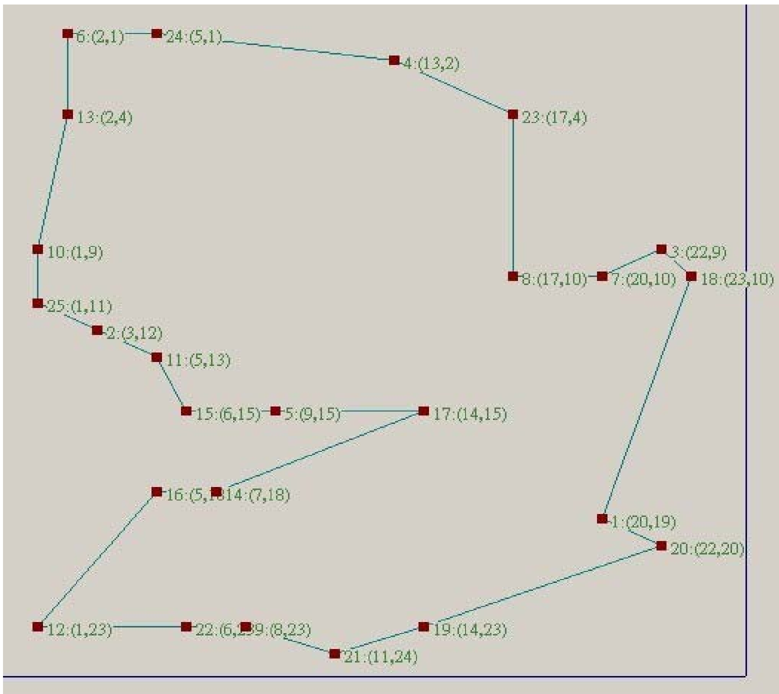
實驗探討二：在 25 乘 25 的方格中，我們以亂數產生 25 個點座標，使其代表地圖上的城市位置，經$1+(n-1)$個演算次數之後得到如下的相對距離及 TSP 路線圖。

順序	城市點	X座標	Y座標
1	17	14	15
2	5	9	15
3	15	6	15
4	11	5	13
5	2	3	12
6	25	1	11
7	10	1	9
8	13	2	4
9	6	2	1
10	24	5	1
11	4	13	2
12	23	17	4
13	8	17	10
14	7	20	10
15	3	22	9
16	18	23	10
17	1	20	19
18	20	22	20
19	19	14	23
20	21	11	24
21	9	8	23
22	22	6	23
23	12	1	23
24	16	5	18
25	14	7	18

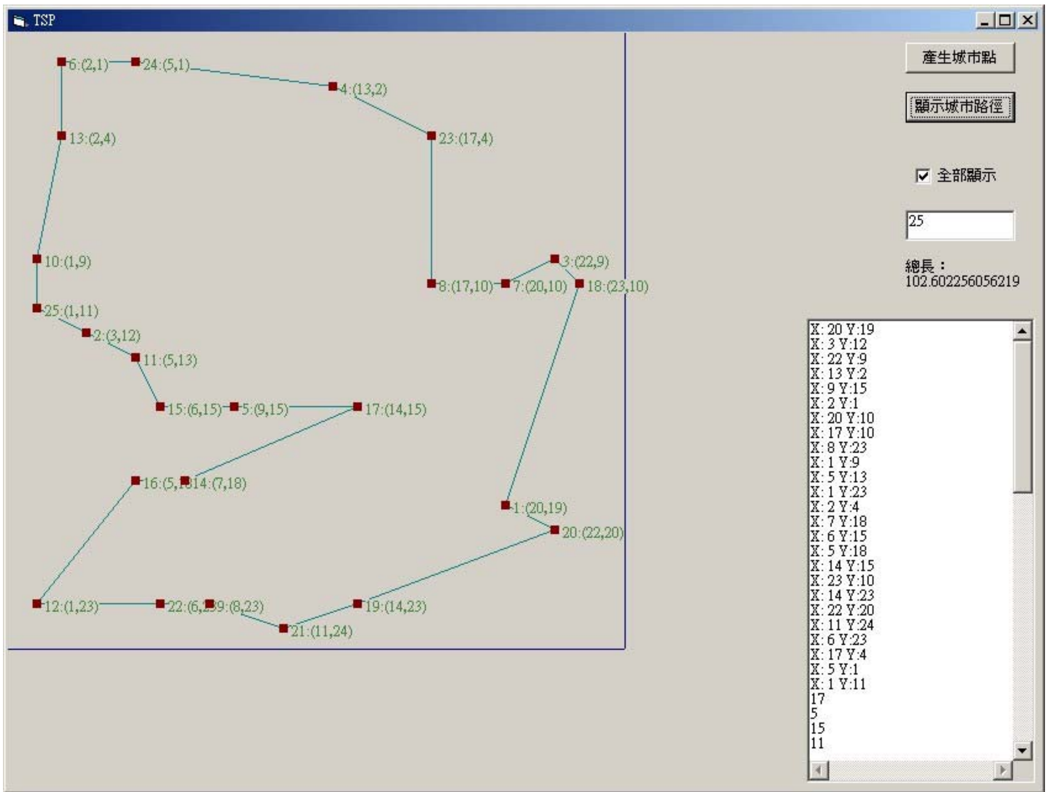
運算次數	相對位置點	相對距離
第1次	17-5	5
第2次	5-15	3
第3次	15-11	2.23606797749979
第4次	11-2	2.23606797749979
第5次	2-25	2.23606797749979
第6次	25-10	2
第7次	10-13	5.09901951359278
第8次	13-6	3
第9次	6-24	3
第10次	24-4	8.06225774829855
第11次	4-23	4.47213595499958
第12次	23-8	6
第13次	8-7	3
第14次	7-3	2.23606797749979
第15次	3-18	1.41421356237310
第16次	18-1	9.48683298050514
第17次	1-20	2.23606797749979
第18次	20-19	8.54400374531753
第19次	19-21	3.16227766016838
第20次	21-9	3.16227766016838
第21次	9-22	2
第22次	22-12	5
第23次	12-16	6.40312423743285
第24次	16-14	2
第25次	14-17	7.61577310586391

25 個點座標位置表

在上表 25 個點的「相對距離」，可以很清楚地知道經演算法運算之後地圖上的連線，由「城市點順序」爲 4 的位置開始一路完成其規劃順序爲【4-13-23-17-8-16-21-2-24-9-1-3-5-20-12-25-11-14-19-7-10-22-15-6-18-4】其圖形結果，如後所示。



TSP-25 路線圖之 1



TSP-25 路線圖之 2

實驗探討三：在 30 乘 30 的方格中，我們以亂數產生 30 個點座標，使其代表地圖上的

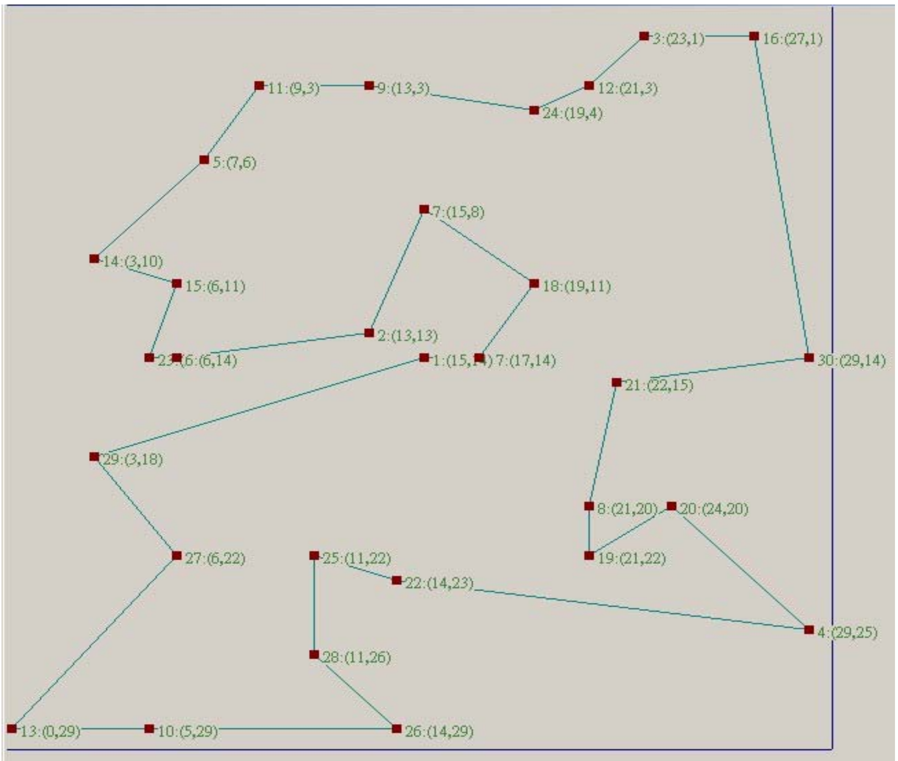
城市位置，經$1+(n-1)$個演算次數之後得到如下的相對距離及 TSP 路線圖。

順序	城市點	X座標	Y座標
1	1	15	14
2	17	17	14
3	18	19	11
4	7	15	8
5	2	13	13
6	6	6	14
7	23	5	14
8	15	6	11
9	14	3	10
10	5	7	6
11	11	9	3
12	9	13	3
13	24	19	4
14	12	21	3
15	3	23	1
16	16	27	1
17	30	29	14
18	21	22	15
19	8	21	20
20	19	21	22
21	20	24	20
22	4	29	25
23	22	14	23
24	25	11	22
25	28	11	26
26	26	14	29
27	10	5	29
28	13	0	29
29	27	6	22
30	29	3	18

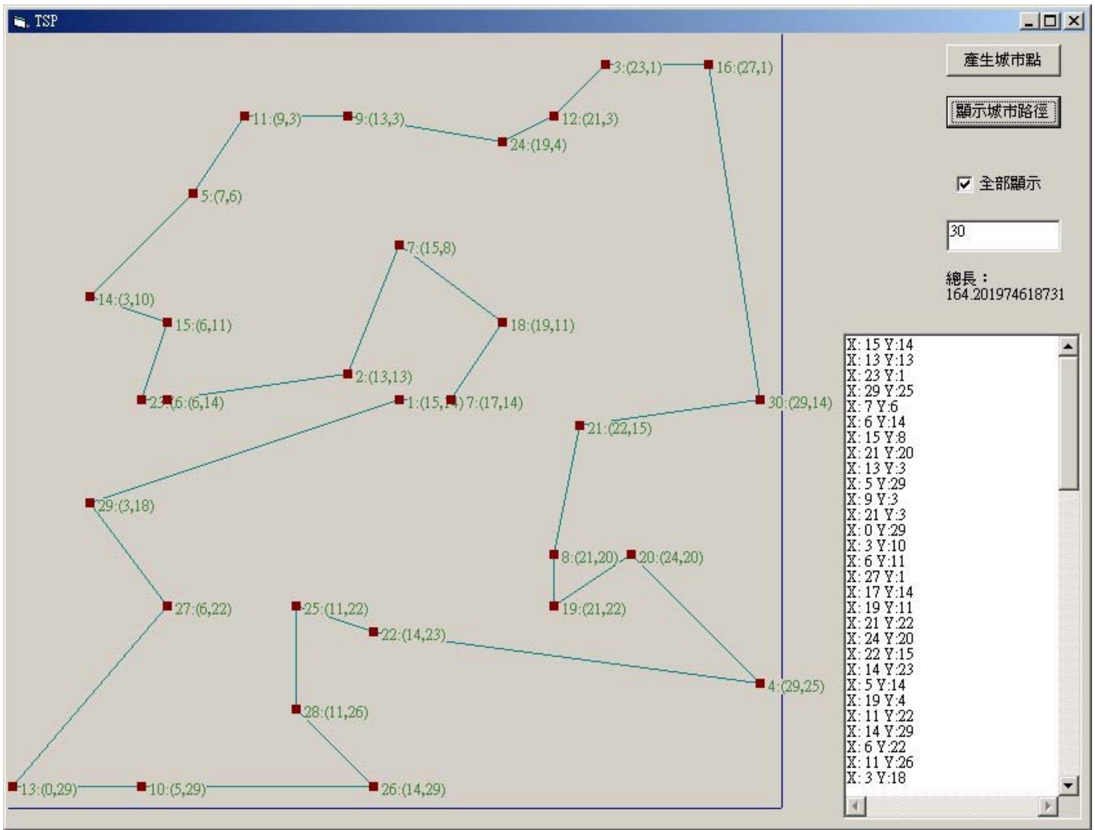
運算次數	相對位置點	相對距離
第1次	1-17	2
第2次	17-18	3.60555127546399
第3次	18-7	5
第4次	7-2	5.38516480713450
第5次	2-6	7.07106781186548
第6次	6-23	1
第7次	23-15	3.16227766016838
第8次	15-14	3.16227766016838
第9次	14-5	5.65685424949238
第10次	5-11	3.60555127546399
第11次	11-9	4
第12次	9-24	6.08276253029822
第13次	24-12	2.23606797749979
第14次	12-3	2.82842712474619
第15次	3-16	4
第16次	16-30	13.15294643796590
第17次	30-21	7.07106781186548
第18次	21-8	5.09901951359278
第19次	8-19	2
第20次	19-20	3.60555127546399
第21次	20-4	7.07106781186548
第22次	4-22	15.13274595042160
第23次	22-25	3.16227766016838
第24次	25-28	4
第25次	28-26	4.24264068711928
第26次	26-10	9
第27次	10-13	5
第28次	13-27	9.21954445729289
第29次	27-29	5
第30次	29-1	12.64911064067350

30 個點座標位置表

在上表 30 個點的「相對距離」，可以很清楚地知道經演算法運算之後地圖上的連線，由「城市點順序」為 16 的位置開始一路完成其規劃順序為【16-18-2-13-26-24-10-12-15-11-14-7-9-5-1-21-17-27-29-30-28-25-23-22-6-8-4-19-20-16】其圖形結果，如後所示。



TSP-30 路線圖之 1



TSP-30 路線圖之 2

捌、實驗結論與未來展望

本文所提的演算法經數據模擬之後，可以順利完成解決旅行推銷員 TSP 的路徑規劃問題，且可以快速地列出推銷員的行進路線組合，加速推銷員本身對於實際路線的決定。

雖然一筆劃問題看似簡單，可是點數愈多，其複雜度也愈高，因此本組希望可以設計一套可自動完成路線規劃的組合方法以順利解決 TSP 問題，經實際數值模擬探討之後，我們順利的完成這個結果，也完成了所有城市的路線組合。而且利用歐氏距離量測方法，也可將我們的應用，由平面拓展至三維以上之更高維度的空間，來進行立體路徑的 TSP 路徑規劃應用。

但是對於路線組合是否爲最佳解，目前仍然沒有想出一個比較合適的好結果，而且相關的證明也只證明出演算法爲可行，卻未證明出演算法爲一定是最佳，因此對於上述問題本組仍然有待改善的地方。

雖然這種組合最佳化的問題好像沒有最佳解，但是由於我們的圖形結果看來似乎結果不錯，因此希望能以此方法來作爲本次研究的結果，以提供研究旅行推銷員問題 TSP 的另一種思考方式。

玖、參考資料及其他

[1] 葉偉文譯，數學是啥玩意 I，天下文化出版社，2002。

[2] 李虎雄、陳昭地、黃登源、李政貴、林礽堂、儲啓政，康熙圖書高中數學第一冊第一章，康熙圖書網路股份有限公司，2006。

[3] 李虎雄、陳昭地、黃登源、李政貴、林礽堂、儲啓政，康熙圖書高中數學第三冊第四章，康熙圖書網路股份有限公司，2006。

[4] 李虎雄、陳昭地、黃登源、李政貴、林礽堂、儲啓政，康熙圖書高中數學第四冊第三章，康熙圖書網路股份有限公司，2006。

[5] 林福來，離散數學初步，九章出版社，1998。

[6] 曹儷馨、陳以叡、陳韋婷，神奇推銷員，中華民國第四十三屆全國中小學科展高

中數學組，2003。

[7] 林振倫、林暐勛、許詠晟、戴瑋辰，迴圈迷宮探索-一筆劃問題，中華民國第四十五屆全國中小學科展高中數學組，2005。

程式碼

```
Dim x, y As Integer
Dim z(1 To 30, 1 To 30) As Double
Dim Zx(1 To 30) As Integer
Dim Zy(1 To 30) As Integer
Dim iCounter As Integer
Dim a As Integer
Dim pic(1 To 30) As PictureBox
Dim txt(1 To 30) As Label
Dim isShowLine As Boolean
Dim isShowCity As Boolean
Dim iVisitCity(0 To 31) As Integer
Dim iCityNumber As Integer
Dim isGetPath    As Boolean

Dim isInit As Boolean

Private Sub Command3_Click()
    If isInit = False Then
        Call produce_Click
    End If

    Dim iIndexOfMaxDistance1 As Integer
    Dim iIndexOfMaxDistance2 As Integer
    Dim dDistance As Double
    Dim dMaxDis As Double
    Dim NewZx(0 To 30) As Double
    Dim NewZy(0 To 30) As Double
    Dim dMinDis As Double
    Dim iNextCity As Integer
    Dim dArrayDis(1 To 30) As Double
    Dim dSum As Double
```

```
If isGetPath = False Then
    '    找最長的距離
    dDistance = 0
    For i = 1 To iCounter
        For j = i + 1 To iCounter
            dDistance = Sqr((Zx(i) - Zx(j)) ^ 2 + (Zy(i) - Zy(j)) ^ 2)
            If dDistance > dMaxDis Then
                dMaxDis = dDistance
                iIndexOfMaxDistance1 = i
                iIndexOfMaxDistance2 = j
            End If
        Next
    Next i
    '新原點
    NewZx(0) = (Zx(iIndexOfMaxDistance1) + Zx(iIndexOfMaxDistance2)) / 2
    NewZy(0) = (Zy(iIndexOfMaxDistance1) + Zy(iIndexOfMaxDistance2)) / 2
    For i = 1 To 30
        NewZx(i) = Zx(i)
        NewZy(i) = Zy(i)
    Next i
    iVisitCity(0) = 0
    '    找第 i 個點
    For k = 1 To iCounter
        dMinDis = 10000
        For i = 1 To iCounter
            If (i <> iVisitCity(k - 1)) Then
                dDistance = Sqr((NewZx(i) - NewZx(iVisitCity(k - 1))) ^ 2 + (NewZy(i) - NewZy(iVisitCity(k - 1))) ^ 2)
                If dDistance < dMinDis Then
                    iVisitCity(k) = i
                    dMinDis = dDistance
                End If
            End If
        Next i
        NewZx(iVisitCity(k - 1)) = 100
        NewZy(iVisitCity(k - 1)) = 100
```

```
                txtShow.Text = txtShow.Text & iVisitCity(k) & vbCrLf
            Next
            iVisitCity(iCounter + 1) = iVisitCity(1)
            isGetPath = True

            dSum = 0
            For i = 1 To iCounter
                dArrayDis(i) = Sqr((Zx(iVisitCity(i + 1)) - Zx(iVisitCity(i))) ^ 2 + (Zy(iVisitCity(i +
                1)) - Zy(iVisitCity(i))) ^ 2)
                txtShow.Text = txtShow.Text & iVisitCity(i) & "-" & iVisitCity(i + 1) & "=" &
                dArrayDis(i) & vbCrLf
                dSum = dSum + dArrayDis(i)
            Next
            lblDistance.Caption = "總長：" & vbCrLf & dSum
        End If

        '    畫線
        If chkAll.Value = 1 Then
            iCityNumber = iCounter ' iCounter
            txtNext.Text = iCounter
        Else
            iNextCity = CInt(txtNext.Text) + 1
            If iNextCity <= iCounter Then
                iCityNumber = iNextCity ' iCounter
                txtNext.Text = iNextCity
            End If
        End If

        isShowCity = True
        Form1.Refresh
    End Sub

    Private Sub Form_Load()
        x = Val(InputBox("請輸入 x 座標的上限" + Chr(10) + Chr(13) + "x 在 30 之內", "x 座標",
        30))
        y = Val(InputBox("請輸入 x 座標的上限" + Chr(10) + Chr(13) + "y 在 30 之內", "y 座標",
        30))

        ' 格式不合，預設為 30
        If x <= 0 Or x > 30 Then
```

```
            x = 30
        End If
        If y <= 0 Or y > 30 Then
            y = 30
        End If

        isShowLine = False
        isShowCity = False

        For i = 1 To 30
            Set pic(i) = Controls.Add("VB.PictureBox", "DynBtn" & i, Me)
            With pic(i)
                .Height = 100
                .Width = 100
                .BackColor = &HFF&
                .ForeColor = &HFF&
                .Appearance = 0
                .AutoSize = False
                .BorderStyle = 0
                .FillStyle = 1
                .Visible = False
            End With

            Set txt(i) = Controls.Add("VB.Label", "DynTxt" & i, Me)
            With txt(i)
                .BackColor = &H8000000F
                .ForeColor = &H80000012
                .Appearance = 0
                .AutoSize = True
                .BackStyle = 1
                .BorderStyle = 0
                .Visible = False
            End With
        Next i
End Sub

Private Sub Form_Activate()
        Randomize
End Sub
```

```
Private Sub Form_Paint()
        ScaleMode = 3
        Line (x * 20, 0)-(x * 20, y * 20), QBColor(1) ' 右邊界
        Line (0, y * 20)-(x * 20, y * 20), QBColor(1) ' 下邊界

        If isShowCity Then
                For i = 1 To iCityNumber
                '       Point i To Point j
                        Line (Zx(iVisitCity(i)) * 20 + 3, Zy(iVisitCity(i)) * 20 + 3)-(Zx(iVisitCity(i + 1)) *
                        20 + 3, Zy(iVisitCity(i + 1)) * 20 + 3), QBColor(3)
                Next i
        End If
    'Form1.Refresh
End Sub

Private Sub produce_Click()
'iCounter
        iCounter = (x + y) / 2
        ' 產生隨機點
        i = 1
        While i <= iCounter
                Zx(i) = Int(Rnd * x)
                Zy(i) = Int(Rnd * y)
                If (i = 1) Then
                        i = i + 1
                Else
                        ' 檢查是否和之前的點重覆，若重覆就放棄此點，再產生一次
                        Dim isNear As Boolean
                        isNear = False
                        For j = 1 To (i - 1)
                                If (Abs(Zx(i) - Zx(j)) < 3) And Zy(i) = Zy(j) Then
                                        isNear = True
                                        Exit For
                                End If
                        Next

                        If Not isNear Then
                                Debug.Print "(" & Zx(i) & "," & Zy(i) & ")"
                                i = i + 1
                        End If
```

```
        End If
    Wend

    For k = 1 To iCounter
        pic(k).Left = Zx(k) * 20
        pic(k).Top = Zy(k) * 20
        pic(k).BackColor = QBColor(4)
        pic(k).Visible = True
        ' 顯示點座標
        txt(k).Left = Zx(k) * 20 + 10
        txt(k).Top = Zy(k) * 20
        txt(k).BackColor = &H8000000F
        txt(k).ForeColor = RGB(69, 134, 62)
        txt(k).Caption = k & ":(" & Zx(k) & "," & Zy(k) & ")"
        txt(k).Visible = True
    Next k

    txtShow.Text = ""
    For i = 1 To iCounter
       txtShow.Text = txtShow.Text & "X: " & Zx(i) & " Y:" & Zy(i) & vbCrLf
    Next i

    txtNext = 0
    chkAll.Value = 0
    isShowCity = False
    isGetPath = False
    lblDistance.Caption = "總長："
    Form1.Refresh
    isInit = True
End Sub
```

評　　語

040420　群最大圓半徑量測之 TSP 演算法

1. 最短路徑問題 Traveling Salesman Problem 的完整解答十分的困難，一般人都依照 Polya 原則「尚未解決一難題以前，先尋找一個簡單一點的問題來解決」來處理。本作品亦不例外。當然，科展的教育價值在於引發學生對於研究的熱誠及興趣。我們由作者的表現可以看出本作品符合此教育理念。

2. 何謂「最大圓」?何謂「非唯一存在」?請注意數學用語之恰當性及嚴謹性。

3. 建議能比較本方法與其他既有演算法則之優劣。

4. 起始點之挑選頗具創意。應說明該挑選之優點何在。