雙環電腦網路裡的L-形

黃光明

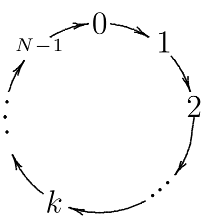
摘要: 本文借雙環網路中的L-形指出電腦網路中有很多有趣的數學問題, 像在 L-形的研究裡就用到了圖論、幾何、代數和數論, 可見數學是的確有用的。

1. 雙環網路

不久之前, 一個電腦中只有一個處理器, 它工作的速度已經比人快上萬倍。不久之後, 我們即嫌它的速度太慢, 因此在一個電腦內放入多個處理器讓他們同時進行一個工作。當然, 他們彼此間需要交換訊息, 這一個處理器間 (也可以是儲存器間) 的網路就是一種電腦網路。同時公司的辦公大樓裡也不僅已有一台電腦爲滿足, 而是每個桌上有一台。這時又需要把這些電腦連結起來, 連結的方式也是一種電腦網路。本文討論的是電腦網路的邏輯結構, 而非它的工程問題。

電腦網路有多種結構可選擇。選擇的標準當然隨應用而異, 但總有一些原則可循。第一是要有好的性質, 通常是指電腦間通訊時間短、可靠性強、含有多種類型的子結構 (這是爲了有的演算法需要特殊結構) 即使零件出差錯時仍能工作等。第二是結構要簡單, 以導致建造、裝置和使用都簡單。

本文介紹一型符合上面兩原則的網路, 叫做 (有向) 雙環網路。它的祖先是 (有向) 單環網路, 即把 N 具電腦 $0,1,\ldots,N-1$ 作環狀連接 (見下圖)。也可說對任何 i, $0\le i\le N-1$,i 有一條 (有向) 線接到 $i+1$ (N 視作0), 表示 i 可以送訊息到 $i+1$。

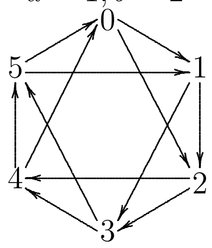


單環網路的確夠簡單, 也滿足了任一點都可 (沿環) 送訊息到另一點的基本任務。但從 i 點送到 $i-1$ 點要走完整個環共 $N-1$ 步。如有任一點或一線壞掉, 每一點就都不能送訊息到在它和壞 (線) 點間的點。它的子結構 (部分點間的連接) 也很貧乏, 唯一能連接所有子集合中的點的結構就是一條線。

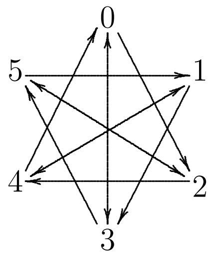
爲此有人建議了雙環網路, 即除了$i\to i+1$ 線組成的一個環外, 再加上 $i\to i+a$, $a\neq 1$, 的一個環。後來雙環網路的定義又被推廣成含 $i\to i+a$ 和 $i\to i+b$ 的兩個環。下圖給了一些 $N=6$ 的例子 (如 $i+a$ 或

$b) > N$, 則以 $i+a$ 或 b 除 N 的餘數代替):

$a=1, b=2$



$a=2, b=3$



($x \leftrightarrow y$ 代表 $x \to y$ 和 $y \to x$ 的兩條線), 一個網路必須滿足的一個性質就是每個點都有路線通到其他任一點, 稱爲連通性。雙環網路 $DL(N;a,b)$ 具有連通性的充要條件是 N,a, b 三個數的公因數爲1。

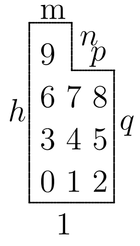
不像單環網路只有一種, 雙環網路隨著 a 和 b 的選擇有許多種, 因此存在一個選擇的問題。在較好的選擇下, 兩點間最遠的距離 (稱爲網路直徑) 從單環的 N 降到 $\sqrt{N}$, 有相當好的可靠性, 少數點和線壞掉時, 網路仍能以高效率運作, 也具有一些重要的子結構, 並能在最短時間內讓所有點同步傳輸訊息。本文限於篇幅, 主要只討論和快速通訊有關的的直徑和同步傳輸。讀者如想涉窺雙環網路的其它面貌, 請參閱[6]。

三環網路和一般的多環網路也有人研討過, 因爲他們含有雙環網路, 所以自然也具有雙環網路的性質, 甚至更好一些 (但數學分析很難, 具體結果不多), 這時要考慮的是新加環的成本, 及每個點要具有接受多條入-線, 及在多條出-線中選擇路線的控制能力的成本及控制所需的時間, 而決定是否值得。一般的意見是雙環網路已經夠好了。

2. L-型

用0-點爲出發點, 我們把到達所有其他點的最短路線畫出來。一個相當不錯的表示法是把0-點放在左下角, a-步 (即 $i \to i+a$) 往右, b-步往上。爲了保證路線爲最短, 王和 Coppersmith [8]提出一個逐步畫法, 即先標出一步能走到的點, 再標出兩步、三步$\cdots$能走到的點。如果一個點在 x 步已走到, 而又在 $y > x$ 步走到時, 我們即不再把它標示出。下列給出 $DL(10;1,3)$ 的標示法:

3	6	9	(2)
0 1	3 4	6 7	9 (0)
一步	0 1 2	3 4 5	6 7 8
	兩步	0 1 2 (3)	3 4 5 (6)
		三步	0 1 2 (3) (4)
			四步



L-型

標出的 N 點用一個多邊形畫出, 則恆得一 L-型。它有六條邊, 其長度分別標示爲 h, l, m, n, p, q, 但只有四個長度是獨立的, 且可證 $l \geq n$, $h \geq p$, 而其中至少有一個是不等式。有時 $n = 0$ 或 $p = 0$ 則 L-型退化爲一長方形。因爲這一退化型的分析較複雜, 本文只考慮純 L-型。

從上例所示的演算法, 可以看出 L-型可在 $O(N)$ 的時間中造出。如果我們要把 L-型中每一格中的點標出, 則因爲共有 N 點, 也必須要用到 $O(N)$ 時間。但如果我們只要 L-型的邊長, 則只有四個參數, 就不一定要 $O(N)$ 時間。最理想的當然是從 N, a, b 可以直接解出 h, l, p, n 四值, 但目前能作到的是程和黃 [4]利用歐幾里得輾轉相除法得到的一個 $O(\log N)$ 時間的演算法。我們用 $DL(50; 1, 19)$ 來舉例說明。

甲. 定義 $\rho_{-1} = N$, ρ_0, $0 \leq \rho_0 < N$, 是滿足 $a\rho_0 + b \equiv 0 (\bmod N)$ 的非負整數, ρ_i, q_i, $1 \leq i \leq m+1$, 則從下式輾轉得到:

$$\rho_i = \rho_{i-2} - q_i \rho_{i-1}, \quad 1 \leq \rho_i \leq \rho_{i-1}$$

m 是首次達到 $\rho_{m+1} \equiv 0 (\bmod N)$ 的數, 在例中, 我們得到

$$\begin{aligned} \rho_{-1} = 50,\ 因爲\ ,1 \cdot 31 + 19 \\ \equiv 0 (\bmod 50),\ 故\ \rho_0 = 31 \end{aligned}$$

$$\rho_1 = 50 - 1 \cdot 31 = 19,\ 故\ q_1 = 1$$

$$\rho_2 = 31 - 1 \cdot 19 = 12,\ 故\ q_2 = 1$$

$$\rho_3 = 19 - 1 \cdot 12 = 7,\ 故\ q_3 = 1$$

$$\rho_4 = 12 - 1 \cdot 7 = 5,\ 故\ q_4 = 1$$

$$\rho_5 = 7 - 1 \cdot 5 = 2,\ 故\ q_5 = 1$$

$$\rho_6 = 5 - 2 \cdot 2 = 1,\ 故\ q_6 = 2$$

$$\rho_1 = 2 - 2 \cdot 1 = 0,\ 故\ q_7 = 2, m = 2$$

乙. 定義 $U_{-1} = 0$, $U_0 = 1$, $U_{i+1} = q_{i+1} U_i + U_{i-1}$, $0 \leq i < m$, 在例中得

$$U_1 = 1 \cdot 1 + 0 = 0,\ U_2 = 1 \cdot 1 + 1 = 2,$$

$$U_3 = 1 \cdot 2 + 1 = 3,\ U_4 = 1 \cdot 3 + 2 = 5,$$

$$U_5 = 1 \cdot 5 + 3 = 8,\ U_6 = 2 \cdot 8 + 5 = 21,$$

$$U_7 = 2 \cdot 21 + 8 = 50$$

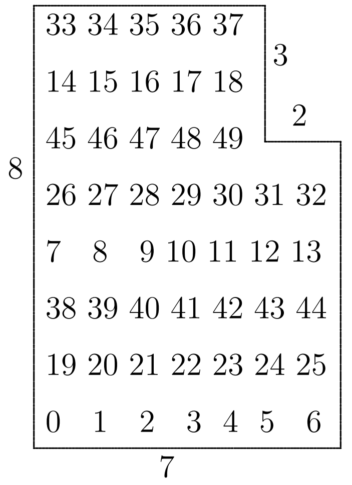
丙. 建立 $0 = \frac{\rho_{m+1}}{U_{m+1}} < \frac{\rho_m}{U_m} < \cdots < \frac{\rho_0}{U_0} < \frac{\rho_{-1}}{U_{-1}} = \infty$ 的單調上升序列。令 u 是最大滿足 $\frac{\rho_u}{U_u} > 1$ 的奇數, 定義 $\nu = \lceil \frac{\rho_u - U_u}{\rho_{u+1} + U_u} \rceil - 1$, 在例中, 序列爲

$$\frac{0}{50} < \frac{1}{21} < \frac{2}{8} < \frac{5}{5} < \frac{7}{3} < \frac{12}{2} < \frac{19}{1} < \frac{31}{1} < \frac{50}{0},$$

因爲 $\frac{\rho_3}{U_3} = \frac{7}{3}$ 是從左到右首次大於一, 且 3是奇數, 故 $u = 3$, $\nu = \lceil \frac{7-3}{5+3} \rceil - 1 = 0$

丁. 解 $l = \rho_u - \nu\rho_{u+1}$, $h = U_u + (\nu + 1) U_{u+1}$, $n = U_u - \nu U_{u+1}$, $p = \rho_u - (\nu + 1)\rho_{u+1}$, 在例中: $1 = 7 - 0 \cdot 5 = 7$, $h = 3 + 1 \cdot 5 = 8$, $n = 3 - 0 \cdot 5 = 3$, $p = 7 - 1 \cdot 5 = 2$

當我們用王的演算法畫出 L-形時, 邊長果然與上解相同。



神奇嗎? 如果不信的話, 請去查一下參考 [4]裡的證明。

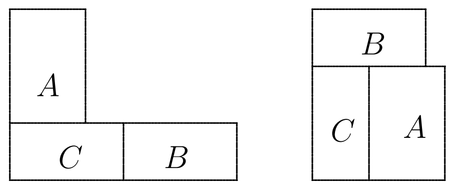
因爲雙環網路有點對稱的性質, 把出發點從0換成任何i, L-型不變。

3. L-型的等價變換

在 L-型離0-點最遠的點一定是兩個凸出角之一。把最遠的距離叫做雙環網路的直徑, 寫成 $D(N;a,b)$, 則

$$D(N;a,b)=\max\{h+m,l+g\}-2$$

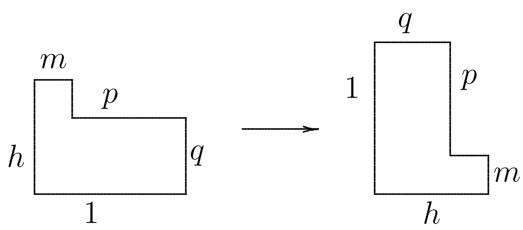
因此計算直徑, 只要用程和黃的 $O(\log N)$ 演算法求 h, l, n, p 即成。在 N 固定時, 我們要求得直徑最短 (代表傳輸最快, 最不易出錯) 的雙環網路, 因此比較他們的 L-型即可。黃和徐 [8]定義兩個 L-型如果距離0-點 d 步的點數, $d=1,2,\ldots,N-1$, 全部相同, 則爲等價。這時不但他們的直徑相等, 點間的平均距離及所有點對稱的距離函數値都會相等, 因此稱它們爲等價是很適合的。黃和徐提出一種幾何方法, 稱爲「三段法」來構造一個等價的 L-型。



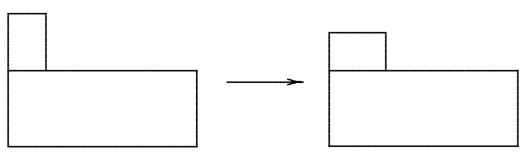
上圖中將 L-型切成三段 A,B,C,A-切把 L-型分成兩個長方形, 下面的長方形再切成 C,B 兩段, 其中 C 的長度要等於 A 的高度 (因爲 $l\geq p$, 所以做得到)。在構造等價的 L-型時, 把 C 豎起來, 把 A 平移到 C 的旁邊合成一長方形, 再把 B 平移到這長方形之上。以上的豎立、平移動作很容易驗證保全了每一段和0-點之間的距離, 因此由三段法造成的 L-型是等價的。

是否還有其他的造法呢? 很多! 但陳和黃 [3]證明所有的造法, 都可以寫成下列四個簡單幾何變換的一個串聯:

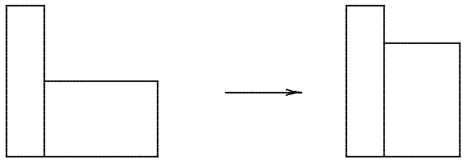
1. 翻轉 (F) 把右和上的方向對調。



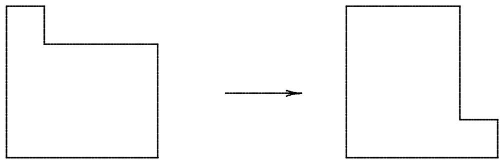
2.換頂 (T) 把頂上的一個長方形作 90° 轉換。



3. 換邊 (S) 把右邊的一個長方形作90° 轉換。



4. 換空 (E) 把 L-型右上方空缺的一個長方形作 $90°$ 轉換。



譬如三段法所得的等價 L-型可以經由換頂–翻轉–換頂得到。

對應於這些幾何變換, a 和 b 這兩個參數如何變換? 在翻轉時, 顯然 $a' = b, b' = a$, 其他三個轉換的 a' 和 b' 在理論上可由下列公式得到

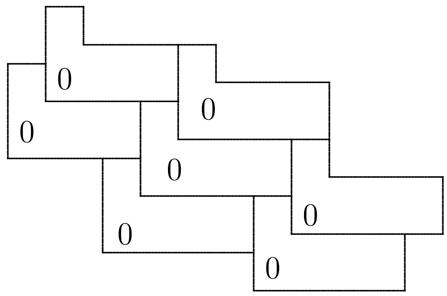
$$FH = TFE = BT = FEB$$

但實際去解並不容易, 陳和黃 [3]給出了 E 的 a' 和 b'。令 x 和 y 爲滿足 $bx - ay = 1$ 的整數, 則 $a' \equiv px - qy (\bmod N)$, $b' \equiv lx - ny (\bmod N)$。

注意找 x, y 也適用歐幾里得的輾轉相除法。

4. 從 L-形求 a, b

既然 L-型決定了直徑, 一個求有短直徑的雙環網路的方法就是先找到一面積爲 $N = hl - pn$ 且 $h + m$ 和 $l + q$ 都不太大的 L-型。下一步是求對應於這一個 L-型的 a 和 b。在第二節中, 我們看到了從 (a, b) 求 L-型是相當容易的事情, 王的演算法小學生也能照作。但反求的問題則難度甚大, 本節中我們介紹一個代數法和一個數論法。這兩種方法都從一個幾何的觀察開始: Fiol,Yebra,Alegre 和 Valero [5]指出任何一種 L-型都可舖滿二度空間, 即 L-型可以經由緊密但不重疊的併合, 無限的擴展它在二度空間裡無隙的含蓋 (見下圖)



下面兩個等式裡敍述了相鄰0-點位置的關係

$$la - nb \equiv 0 (\bmod N)$$
$$-pa + hb = 0 (\bmod N)$$

第一式說從一個0-點往右走 l 個 a-步, 再往下走 n 個 b-步就到達另一0-點, 第二式說往左走 p 個 a-步, 再往上走 h 個 b-步也到達一0-點。本來用兩個方程式解兩個未知數應該不是難事, 但這裡的方程式是在有限域裡, 就沒有標準解法了。Aguilo和 Fiol [1]用矩陣的 Smith 正則型轉換得到了一個解答。對

$$\begin{pmatrix} l & -p \\ -n & h \end{pmatrix}$$

這一矩陣求它的 Smith 正則型, 即用兩個 2×2 的矩形 L 和 R 其元素整數且行列式

值爲1使

$$L\begin{pmatrix} l & -p \\ -n & h \end{pmatrix}, R=\begin{pmatrix} l & 0 \\ 0 & N \end{pmatrix}$$

設 $L=\begin{pmatrix} w & x \\ y & z \end{pmatrix}$, 則 $a=y$, $b=z$。譬如當 $N=8$, $h=l=3$, $n=p=1$ 時

$$\begin{pmatrix} 0 & -1 \\ 1 & 3 \end{pmatrix}\begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix}\begin{pmatrix} -1 & 3 \\ 0 & 1 \end{pmatrix}=\begin{pmatrix} 1 & 0 \\ 0 & 8 \end{pmatrix}$$

所以$a=1$, $b=3$。上述作者也給出了一個相當複雜找L和R的方法。構造$DL(8;1,3)$的 L-型, 得

6	7	
3	4	5
0	1	2

h, l, n, p 值驗證無誤。陳和黃 [2]以數論的方法切入。定義

$$a_k = h + hn$$
$$b_k = p + kl$$

令 F 是 N 的質因數的集合, F_k 是 (a_k, b_k) 最大公約數的質因數的集合。在前例中 $N=18$, 故 $F=\{2,3\}$,

$$a_0=4, b_0=3 \text{最大公約數不存在故} F_0=\phi$$
$$a_1=8, b_1=6 \text{最大公約數是2故} F_1=\{2\}$$
$$\vdots$$

他們用數論中「篩」的理論證明了一定存在一個 k, 使得 F 中每一元素都不在 F_k 裡, 這時 $a=a_k$, $b=b_k$ 即爲解。在例題中 $F_0=\phi$, 自然不含 F 中元素, 所以 $a_0=4$, $b_0=3$ 即爲解。

構造 DL(18;4,3) 的 L-型, 得

2	5		
16	19		
12	15		
8	11	14	17
4	7	10	13
0	3	6	9

h, l, n, p 值驗證無誤。

5. L-形的訊息傳輸

假設網路上的每一點都有一訊息要送到另一點。當有兩個訊息要送到同一點時, 則兩條路線在終點附近很可能會發生爭線的情形, 所以我們假設終點全不同。當這 N 對 (起點, 終點) 最大的距離是 d 時, 則很顯然我們至少需要 d 步 (每步走一條線) 才能把這距離爲 d 的訊息送到。非常吃驚的一個結果是黃, 林和簡 [7]證明了所有 N 個訊息的同步輸送可以在 d 步完成 (一步中每根線只能送一個訊息)。假設 $a<b$, 則稱 b-步爲大步,a-步爲小步。

大步法

甲. 首先從 L-型上求得每一訊息從起點到終點需走多少大步, 多少小步。

乙. 每一訊息都先走大步, 走完再走小步。

丙. 當某些訊息需爭用同一線時, 離終點小步距離最大的先走, 其它訊息都在原地踏步等下一個機會。

一個訊息在走大步的階段時, 不會和其他訊息爭線, 因爲一開始時, 訊息都在不同點上, 所以走完大步後, 仍在不同點上。這時可能會有走小步的訊息落在同一點上, 但它要用的是a-線, 不和大步搶 b-線。一個訊息在走小步的階段, 可能會和其他走小步的訊息爭a-線, 這時小步距離最遠的先走。又因終點都不一樣, 所以同停在某點等待 a-線的訊息離終點的小步距離也都不同 (記住它們剩下的步都是小步) 。顯然的, 原距離爲 d 的訊息在圖中的任何點它都會是距離終點最遠的, 因此從來不須等。可以證明原距離爲 $d-k$ 的點最多只要等 k 次, 所以在 d 步後每個訊息都到達終點。

雙環網路 (當然越多環會越好) 是目前知道唯一能讓所有網路點同步輸送訊息而能保證在最短 d 步內完成的網路。

參考文獻

1. F. Aguilo and M. A. Fiol, An efficient algorithm to find optional double-loops network, Disc. Math. 138(1995) 5-14.
2. C. Chen and F. K. Hwang, The minimum distance diagram of double-loops networks, IEEE Trans. Comput, to appear.
3. C. Chen and F. K. Hwang, The spectrum of equivalent nondegenerate L-shapes of double-loops networks, Network, to appear.
4. Y. Cheng and F. K. Hwang, Diameters of weighted double-loops networks, J. Alg. 9(1998) 401-410.
5. M. A. Fiol, J. L. A. Yeba, I. Alegre and M. Valero, A discrete optimization problem in local networks and data alignment, IEEE Trans. Comput. 36(1987) 702-713.
6. F. K. Hwang, A complementary survey on double-loop networks, Theoretical Computer Science, to appear.
7. F. K. Hwang, T.-S. Lin and R.-H. Jan, A permutation routing algorithm for double-loop networks, Parall. Process. Lett. 7(1997) 259-265.
8. C. K. Wong and D. Coppersmith, A combinational problem related to multimodule memory organization, J. ACM 21(1974) 391-402.

—本文作者任教於交通大學應用數學系—