ELSEVIER

Discrete Applied Mathematics 121 (2002) 61–72

DISCRETE APPLIED MATHEMATICS

A simple algorithm to find the steps of double-loop networks☆

Robin Chi-Feng Chan, Chiuyuan Chen *, Zhi-Xin Hong

Department of Applied Mathematics, National Chiao Tung University, 1001 Ta-Hsueh Road, Hsinchu 300, Taiwan

Received 28 June 2000; received in revised form 20 March 2001; accepted 9 April 2001

Abstract

Double-loop networks have been widely studied as architecture for local area networks and it is well-known that the minimum distance diagram of a double-loop network yields an L-shape. Given an N, it is desirable to find a double-loop network $DL(N;s_1,s_2)$ with its diameter being the minimum among all double-loop networks with N stations. Since the diameter can be easily computed from an L-shape, one method is to start with a desirable L-shape and then asks whether there exist s_1 and s_2 (also called the steps of the double-loop network) to realize it. In this paper, we propose a simple and efficient algorithm to find s_1 and s_2, which is based on the Smith normalization method of Aguiló, Esqué and Fiol.

Keywords: Double-loop network; L-shape; Diameter; Algorithm

1. Introduction

A double-loop network $DL(N;s_1,s_2)$ has N nodes $0,1,\ldots,N-1$ and $2N$ links of two types:

$$s_1\text{-links} : i \to i+s_1 \pmod N, \quad i=0,1,\ldots,N-1,$$

$$s_2\text{-links} : i \to i+s_2 \pmod N, \quad i=0,1,\ldots,N-1.$$

Double-loop networks have been widely studied as architecture for local area networks. For surveys about these networks, see [2,10,11,14].

Fiol et al. [8] proved that $DL(N;s_1,s_2)$ is strongly connected if and only if $\gcd(N,s_1,s_2)=1$. When $DL(N;s_1,s_2)$ is strongly connected, then we can talk about

☆ This research was partially supported by the National Science Council of the Republic of China under the grant NSC89-2115-M009-026.

* Corresponding author.

E-mail address: cychen@cc.nctu.edu.tw (C. Chen).

0166-218X/02/$ - see front matter
PII: S0166-218X(01)00245-1

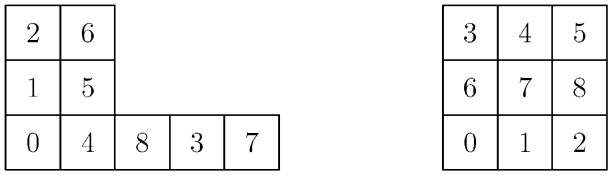

Fig. 1. Two examples of L-shapes.

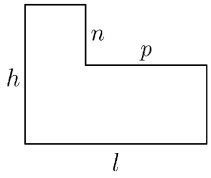

Fig. 2. An L-shape with parameters.

a minimum distance diagram. This diagram gives a shortest path from node u to node v for any u, v. Since a double-loop network is node-symmetric, it suffices to give a shortest path from node 0 to any other node. Let 0 occupy cell $(0,0)$. Then v occupies cell (i,j) if and only if $ia+jb \equiv v \pmod N$ and $i+j$ is the minimum among all (i',j') satisfying the congruence, where $\equiv$ means congruent modulo N. Namely, a shortest path from 0 to v is through taking i s_1-links and j s_2-links (in any order). Note that in a cell (i,j), i is the column index and j is the row index. A minimum distance diagram includes every node exactly once (in case of two shortest paths, the convention is to choose the cell with the smaller row index, i.e., the smaller j). Wong and Coppersmith [15] proved that the minimum distance diagram is always an L-shape (a rectangle is considered a degeneration). See Fig. 1 for two examples.

An L-shape is determined by four parameters l, h, p, n as shown in Fig. 2. These four parameters are the lengths of four of the six segments on the boundary of the L-shape. For example, $DL(9;4,1)$ in Fig. 1 has $l=5$, $h=3$, $p=3$, and $n=2$. Let $N=lh-pn$. Fiol et al. [8,9] and Chen and Hwang [3] proved that there exists a $DL(N;s_1,s_2)$ realizing the L-shape(l,h,p,n) if and only if $l>n$, $h \geqslant p$, and $\gcd(l,h,p,n)=1$.

The *diameter* $d(N;s_1,s_2)$ of a double-loop network $DL(N;s_1,s_2)$ is the largest distance between any pair of stations. It represents the maximum transmission delay between two stations. Therefore, it is desirable to minimize the diameter. This is the problem discussed by many authors; see [1,5–7,9,12,15]. Let $d(N)$ denote the best possible diameter of a double-loop network with N stations. Wong and Coppersmith [15] showed that $d(N) \geqslant \lceil \sqrt{3N} \rceil - 2$.

Given an N, it is desirable to find a double-loop network $DL(N;s_1,s_2)$ with its diameter being equal to $d(N)$. Since the diameter of a double-loop network $DL(N;s_1,s_2)$

can be readily computed from the dimensions of its L-shape, one method is to start with a desirable L-shape and then asks whether there exist s_1 and s_2 to realize it. Aguiló, Esqué and Fiol [1,7] proposed the Smith normalization method to find s_1 and s_2 for a given L-shape, but no explicit algorithm was given in their paper. In [3], Chen and Hwang proposed a simple method, based on the sieve method in number theory, to find s_1 and s_2 for a given L-shape.

In this paper, we propose a simple and efficient algorithm to find s_1 and s_2 for a given L-shape. Our algorithm is based on the Smith normalization method of Aguiló, Esqué and Fiol [1,7], but unlike their method, our algorithm does not require any matrix operation. Our algorithm takes at most $O((\log N)^2)$ time and if $\gcd(l,n)=1$ or $\gcd(l,p)=1$ or $\gcd(h,p)=1$ or $\gcd(h,n)=1$, then our algorithm could find the steps of a double-loop network in only $O(\log N)$ time.

2. Preliminary

It is well-known that

Lemma 1. *If a and b are integers, not both zero, then there exist integers α and β such that $\alpha a+\beta b=\gcd(a,b)$.*

We now prove that

Lemma 2. *If α, a, β, b are integers, not all zero, such that $\alpha a+\beta b=1$, then $\gcd(a,\beta)=1$.*

Proof. Assume that $\alpha a+\beta b=1$ and $\gcd(a,\beta)=k$. Then $k|a$ and $k|\beta$. Thus $k|\alpha a+\beta b=1$. So $k=1$. □

Theorem 3. *If a and b are integers, not both zero, then there exist integers x and y such that $xa+yb=\gcd(a,b)$ and $(y,\gcd(a,b))=1$.*

Proof. Set $r=\gcd(a,b)$ for easy writing. By Lemma 1, there exist integers α and β such that $\alpha a+\beta b=r$. If $\gcd(\beta,r)=1$, then we are done. In the following, assume that $\gcd(\beta,r)=k>1$. Suppose $k=p_1^{s_1}p_2^{s_2}\cdots p_m^{s_m}$, where $p_1<p_2<\cdots<p_m$ are the prime factors of k and suppose $r=p_1^{r_1}p_2^{r_2}\cdots p_m^{r_m}p_{m+1}^{r_{m+1}}p_{m+2}^{r_{m+2}}\cdots p_n^{r_n}$, where $p_1<p_2<\cdots<p_n$ are the prime factors of r. Since $k|r$, we have $r_i\geqslant s_i$ for all i, $1\leqslant i\leqslant m$. Let

$$r'=p_{m+1}^{r_{m+1}}p_{m+2}^{r_{m+2}}\cdots p_n^{r_n},\quad a'=a/r,\ \text{and}\ b'=b/r.$$

Note that $\gcd(r',\beta)=1$; otherwise, we will have $\gcd(\beta,r)>k$. Since $\gcd(r',\beta)=1$ and $k|\beta$, we have $\gcd(r',k)=1$. Since $\alpha a+\beta b=r$, we have $\alpha a'+\beta b'=1$. By Lemma 2, we have $\gcd(a',\beta)=1$. Since $\gcd(a',\beta)=1$ and $k|\beta$, we have $\gcd(a',k)=1$. Since $k|\beta$ and $\gcd(r',k)=1$ and $\gcd(a',k)=1$, we have $\gcd(\beta-r'a',k)=1$. Since $\gcd(r',\beta)=1$ and $r'|r'a'$, we have $\gcd(\beta-r'a',r')=1$. Since $\gcd(\beta-r'a',k)=1$ and

$\gcd(\beta - r'a', r') = 1$ and every prime factor of r is either a prime factor of k or a prime factor of r', we have $\gcd(\beta - r'a', r) = 1$. Consider

$$x = \alpha + r'b' \quad \text{and} \quad y = \beta - r'a'.$$

Then $xa + yb = (\alpha + r'b')a + (\beta - r'a')b = r$ and $\gcd(y, r) = \gcd(\beta - r'a', r) = 1$. We have this theorem. □

The proof of Theorem 3 leads to the following algorithm for finding x and y in Theorem 3.

ALGORITHM-MODIFIED-EUCLIDEAN

Input: Integers a and b, not both zero, and $r = \gcd(a, b)$.
Output: Integers x and y such that $xa + yb = r$ and $\gcd(y, r) = 1$.

1. Find integers α and β such that $\alpha a + \beta b = r$.
2. If $\gcd(\beta, r) = 1$, then let $x = \alpha$, $y = \beta$, return x, y and stop this algorithm.
3. Let $k = \gcd(\beta, r)$, $r' = r$, and $d = k$.
4. WHILE $(d > 1)$ DO
   BEGIN
   $r' = r'/d$;
   $d = \gcd(r', k)$;
   END
5. Let $a' = a/r$, $b' = b/r$, $x = \alpha + r'b'$ and $y = \beta - r'a'$. Return x, y.

We give an example to show how Step 4 is executed. Suppose before Step 4 is executed, $d = k = 2^3 \times 3^2 \times 7^4$ and $r' = r = 2^4 \times 3^8 \times 7^{15} \times 11 \times 23$. After the first iteration of the while-loop, $r' = 2 \times 3^6 \times 7^{11} \times 11 \times 23$ and $d = 2 \times 3^2 \times 7^4$. After the second iteration, $r' = 3^4 \times 7^7 \times 11 \times 23$ and $d = 3^2 \times 7^4$. After the third iteration, $r' = 3^2 \times 7^3 \times 11 \times 23$ and $d = 3^2 \times 7^3$. After the fourth iteration, $r' = 11 \times 23$ and $d = 1$. Since $d = 1$, we stop the iteration.

Theorem 4. *ALGORITHM-MODIFIED-EUCLIDEAN is correct and it takes at most* $O((\log N)^2)$ *time, where* $N = \max\{a, b\}$.

Proof. Note that Steps 1, 2, 3, and 5 of ALGORITHM-MODIFIED-EUCLIDEAN are translated directly from the proof of Theorem 3, so they are correct. Steps 1 and 2 take $O(\log N)$ time; Steps 3 and 5 take $O(1)$ time. It remains to consider Step 4. Let $k = p_1^{s_1} p_2^{s_2} \cdots p_m^{s_m}$ and $r = p_1^{r_1} p_2^{r_2} \cdots p_m^{r_m} p_{m+1}^{r_{m+1}} p_{m+2}^{r_{m+2}} \cdots p_n^{r_n}$ be defined as in the proof of Theorem 3. Note that $r_i \geqslant s_i$ for all i, $1 \leqslant i \leqslant m$. In the proof of Theorem 3, we need $r' = p_{m+1}^{r_{m+1}} p_{m+2}^{r_{m+2}} \cdots p_n^{r_n}$. The purpose of Step 4 is to derive $r' = p_{m+1}^{r_{m+1}} p_{m+2}^{r_{m+2}} \cdots p_n^{r_n}$.

Before Step 4 is executed, $r' = r = p_1^{r_1} p_2^{r_2} \cdots p_m^{r_m} p_{m+1}^{r_{m+1}} p_{m+2}^{r_{m+2}} \cdots p_n^{r_n}$. We then use a while-loop to remove $p_1^{r_1} p_2^{r_2} \cdots p_m^{r_m}$ from r'. Before an iteration of the while-loop, if p_i (where $1 \leqslant i \leqslant m$) still exists in the current r' and its power in the current r' is

r'_i, then after the iteration, the power of p_i is decreased by s_i or r'_i, whichever is smaller. At the end of Step 4, $d = \gcd(r', k) = 1$; that is, $r' = p_{m+1}^{r_{m+1}} p_{m+2}^{r_{m+2}} \cdots p_n^{r_n}$. Set $\rho = \max\{\lceil r_1/s_1 \rceil, \lceil r_2/s_2 \rceil, \ldots, \lceil r_m/s_m \rceil\}$ for easy writing. Step 4 iterates ρ times. Thus Step 4 takes $O(\rho \log N)$ time. Since $\rho = O(\log N)$, Step 4 takes at most $O((\log N)^2)$ time.

The above arguments show that ALGORITHM-MODIFIED-EUCLIDEAN is correct and it takes at most $O((\log N)^2)$ time. □

3. The Smith normalization method

Let $L(l, h, p, n)$ be an L-shape such that $l > n$, $h \geqslant p$, and $\gcd(l, h, p, n) = 1$. Aguiló and Fiol [1], and also Esqué et al. [7] proposed the following method of computing s_1 and s_2 such that $DL(N; s_1, s_2)$ realizes L. They considered the integral matrix

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

and computed the Smith normal form of $\mathscr{M}$,

$$\mathscr{S}(\mathscr{M}) = \begin{pmatrix} 1 & 0 \\ 0 & N \end{pmatrix}.$$

Then $\mathscr{S}(\mathscr{M}) = \mathscr{L}\mathscr{M}\mathscr{R}$, where $\mathscr{L}$ and $\mathscr{R}$ are two nonsingular unimodular (determinant ± 1) integral matrices. They proved that if

$$\mathscr{L} = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix},$$

then $s_1 = \gamma \pmod N$ and $s_2 = \delta \pmod N$ in $DL(N; s_1, s_2)$. No algorithm on computing the Smith normal form was actually given in their paper except a reference to [13]. In [13], the reader was referred to three theorems (Theorem II.1, Theorem II.2, and Theorem II.9) for learning how to compute the Smith normal form.

The following is a brief description of what the three theorems in [13] say. Let α_1 and α_2 be two integers, no both zero, and let $\delta = \gcd(\alpha_1, \alpha_2)$. Theorem II.1 says that there exists an integral matrix

$$\begin{pmatrix} \alpha_1 & \alpha_2 \\ \sigma & \rho \end{pmatrix}$$

with first row $[\alpha, \alpha_2]$ and determinant δ; note that the elements σ and ρ may be determined by the Euclidean algorithm. Theorem II.2 uses Theorem II.1 to show that the (1,1) element of a matrix may be replaced by the greatest common divisor of the first column of the matrix. Theorem II.9 uses Theorem II.2 to derive the Smith normal form. To make the readers easy to understand the Smith normalization method, we now give an explicit algorithm for it.

THE-SMITH-NORMALIZATION-METHOD

Input: l, h, p, n of an L-shape L, where $l > n$, $h \geqslant p$, and $\gcd(l, h, p, n) = 1$.
Output: s_1 and s_2 such that $DL(N; s_1, s_2)$ realizes the L-shape $L(l, h, p, n)$.

1. Let

$$\mathscr{M} = \begin{pmatrix} l & -p \\ -n & h \end{pmatrix},$$

   $\mathscr{M}_0 = \mathscr{M}$, $i = 0$, $j = 0$, $k = 0$.
2. Repeat sub-steps 2.1–2.2 until the (1,1) element of $\mathscr{M}_j$ divides both the (2,1) element and the (1,2) element of $\mathscr{M}_j$.
   2.1 If the (1,1) element of $\mathscr{M}_j$ does not divide the (2,1) element of $\mathscr{M}_j$, then let $i = i + 1$, $j = j + 1$, and find a nonsingular unimodular integral matrix $\mathscr{L}_i$ such that the (1,1) element of $\mathscr{M}_j = \mathscr{L}_i \mathscr{M}_{j-1}$ is the greatest common divisor of the first column of $\mathscr{M}_{j-1}$.
   2.2 If the (1,1) element of $\mathscr{M}_j$ does not divide the (1,2) element of $\mathscr{M}_j$, then let $j = j + 1$, $k = k + 1$, and find a nonsingular unimodular integral matrix $\mathscr{R}_k$ such that the (1,1) element of $\mathscr{M}_j = \mathscr{M}_{j-1} \mathscr{R}_k$ is the greatest common divisor of the first row of $\mathscr{M}_{j-1}$.
3. If the (2,1) element of $\mathscr{M}_j$ is not zero, then let $i = i+1$, $j = j+1$, and find a nonsingular unimodular integral matrix $\mathscr{L}_i$ to make the (2,1) element of $\mathscr{M}_j = \mathscr{L}_i \mathscr{M}_{j-1}$ zero.
4. If the (1,2) element of $\mathscr{M}_j$ is not zero, then let $j = j + 1$, $k = k + 1$, and find a nonsingular unimodular integral matrix $\mathscr{R}_k$ to make the (1,2) element of $\mathscr{M}_j = \mathscr{M}_{j-1} \mathscr{R}_k$ zero.
5. If the (1,1) element of $\mathscr{M}_j$ does not divide the (2,2) element of $\mathscr{M}_j$, then add column 2 of $\mathscr{M}_j$ to column 1 of $\mathscr{M}_j$ and go to Step 2.
6. Now $\mathscr{M}_j$ is the Smith normal form of $\mathscr{M}$, i.e.,

$$\mathscr{M}_j = \mathscr{L}_i \cdots \mathscr{L}_2 \mathscr{L}_1 \mathscr{M} \mathscr{R}_1 \mathscr{R}_2 \cdots \mathscr{R}_k = \mathscr{S}(\mathscr{M}) = \begin{pmatrix} 1 & 0 \\ 0 & N \end{pmatrix}.$$

   Let $\mathscr{L} = \mathscr{L}_i \cdots \mathscr{L}_2 \mathscr{L}_1$. If

$$\mathscr{L} = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix},$$

   then let $s_1 = \gamma \pmod N$ and let $s_2 = \delta \pmod N$. Return s_1, s_2.

Since [1,7] did not provide the time complexity analysis of the Smith normalization method, we now analyze its time complexity. Its time complexity is dominated by Step 2. Each execution of Step 2 takes $O(\log N)$ time. Since each execution of Step 2.1 and Step 2.2 makes the (1,1) element of $\mathscr{M}_j$ contains less prime factors than before, Step 2 is executed at most $O(\log N)$ times. Therefore the Smith normalization method takes at most $O((\log N)^2)$ time.

4. The sieve method

Let $L(l,h,p,n)$ be an L-shape such that $l > n$, $h \geqslant p$, and $\gcd(l,h,p,n) = 1$. Chen and Hwang [3] (see also [11]) proposed the following method, based on the sieve method in number theory, to find s_1 and s_2.

For $k = 0, 1, \ldots,$ define

$$a_k = kn + h,$$

$$b_k = kl + p.$$

Let F_k denote the set of prime factors of $\gcd(a_k, b_k)$ and F denote the set of prime factors of N. They proved that there exists a k such that $f \notin F_k$ for all $f \in F$; then $s_1 = a_k \pmod N$ and $s_2 = b_k \pmod N$ realize L. Note that if $f \in F$ appears in F_k for some k and k_f is the smallest such k, then f appears in every fth k after k_f.

For example, suppose $N = 2\times3\times5\times7\times11\times59$, and $L(l,h,p,n) = L(2^2\times107, 2^2\times3\times5\times7, 2\times3\times5\times7, 3^2\times23)$. Then $F_0 = \{2,3,5,7\}$, $F_1 = \{11\}$, and $F = \{2,3,5,7,11,59\}$. Thus

$2 \in F$ appears in $a_0, b_0,\ a_2, b_2,\ a_4, b_4,\ a_6, b_6,\ a_8, b_8$, etc.,

$3 \in F$ appears in $a_0, b_0,\ a_3, b_3,\ a_6, b_6,\ a_9, b_9$, etc.,

$5 \in F$ appears in $a_0, b_0,\ a_5, b_5,\ a_{10}, b_{10}, a_{15}, b_{15}$, etc.,

$7 \in F$ appears in $a_0, b_0,\ a_7, b_7,\ a_{14}, b_{14},\ a_{21}, b_{21}$, etc.,

$11 \in F$ appears in $a_1, b_1,\ a_{12}, b_{12},\ a_{23}, b_{23},\ a_{34}, b_{34}$, etc.

The first pair a_k, b_k that is not crossed out by the sieve method is a_{11}, b_{11}. Thus $s_1 = a_{11} \pmod N$ and $s_2 = b_{11} \pmod N$ realize L.

The sieve method is simple and easy to implement. Note that [3,11] did not give the time complexity analysis of the sieve method. Although we are also unable to give an exact time complexity analysis for the method, we give an upper bound for it. Let $\eta(N)$ denote the number of prime factors of N and let P_i denote the ith prime, i.e., $P_1 = 2, P_2 = 3$, etc. Since smaller primes cross out more pairs (a_k, b_k) than larger primes can cross out, the sieve method would take the longest time when N contains the smallest $\eta(N)$ primes. In this case, the final k is bounded above by $P_{\eta(N)}$. Since checking if $f \notin F_k$ for all $f \in F$ is equivalent to check if $\gcd(\gcd(a_k, b_k), N) = 1$ (which could be checked by using the Euclidean algorithm twice), the sieve method takes at most $O(P_{\eta(N)} \log N)$ time.

5. Our algorithm

Given an L-shape, we propose the following algorithm to find s_1 and s_2.

ALGORITHM-COMPUTING-STEPS

Input: l, h, p, n of an L-shape L, where $l > n$, $h \geqslant p$, and $\gcd(l,h,p,n) = 1$.

Output: s_1 and s_2 such that $DL(N; s_1, s_2)$ realizes L.

1. Find $r_1 = \gcd(l, -n)$.
2. Find integers α_1 and β_1 such that $\alpha_1 l + \beta_1(-n) = r_1$.
3. Find $r_2 = \gcd(r_1, -\alpha_1 p + \beta_1 h)$.
4. Find integers α_2 and β_2 such that $\alpha_2 r_1 + \beta_2(-\alpha_1 p + \beta_1 h) = r_2$ and $\gcd(\beta_2, r_2) = 1$.
5. $s_1 = \alpha_2 n - \beta_2 h \pmod N$ and $s_2 = \alpha_2 l - \beta_2 p \pmod N$.

For example, let $l = 5$, $h = 3$, $p = 3$, and $n = 2$. Then our algorithm derives $r_1 = 1$, $\alpha_1 = 1$, $\beta_1 = 2$, $r_2 = 1$, $\alpha_2 = -2$, and $\beta_2 = 1$. Thus $s_1 = -7 \pmod 9$ and $s_2 = -13 \pmod 9$, i.e., $s_1 = 2$ and $s_2 = 5$. It can be verified that $DL(9; 2, 5)$ realizes L-shape(5,3,3,2).

We now prove that

Theorem 5. *ALGORITHM-COMPUTING-STEPS is correct and it takes at most* $O((\log N)^2)$ *time.*

Proof. Note that $N = lh - pn$. Let

$$\mathscr{M} = \begin{pmatrix} l & -p \\ -n & h \end{pmatrix}.$$

Consider column 1 of $\mathscr{M}$: it contains l and $-n$. After Step 1 is performed, we have $r_1 = \gcd(l, -n)$ and $\alpha_1 l + \beta_1(-n) = r_1$. Let

$$\mathscr{L}_1 = \begin{pmatrix} \alpha_1 & \beta_1 \\ \frac{n}{r_1} & \frac{l}{r_1} \end{pmatrix}.$$

and let $\mathscr{M}_1 = \mathscr{L}_1 \mathscr{M}$. Then

$$\mathscr{M}_1 = \begin{pmatrix} \alpha_1 & \beta_1 \\ \frac{n}{r_1} & \frac{l}{r_1} \end{pmatrix} \begin{pmatrix} l & -p \\ -n & h \end{pmatrix} = \begin{pmatrix} r_1 & -\alpha_1 p + \beta_1 h \\ 0 & \frac{N}{r_1} \end{pmatrix}.$$

Consider row 1 of $\mathscr{M}_1$: it contains r_1 and $-\alpha_1 p + \beta_1 h$. After Step 2 is performed, we have $r_2 = \gcd(r_1, -\alpha_1 p + \beta_1 h)$, $\alpha_2 r_1 + \beta_2(-\alpha_1 p + \beta_1 h) = r_2$, and $\gcd(\beta_2, r_2) = 1$. Let

$$\mathscr{R}_1 = \begin{pmatrix} \alpha_2 & \frac{-(-\alpha_1 p + \beta_1 h)}{r_2} \\ \beta_2 & \frac{r_1}{r_2} \end{pmatrix}.$$

and let $\mathscr{M}_2 = \mathscr{M}_1 \mathscr{R}_1$. Then

$$\mathscr{M}_2 = \begin{pmatrix} r_1 & -\alpha_1 p + \beta_1 h \\ 0 & \frac{N}{r_1} \end{pmatrix} \begin{pmatrix} \alpha_2 & \frac{-(-\alpha_1 p + \beta_1 h)}{r_2} \\ \beta_2 & \frac{r_1}{r_2} \end{pmatrix} = \begin{pmatrix} r_2 & 0 \\ \frac{N\beta_2}{r_1} & \frac{N}{r_2} \end{pmatrix}.$$

Consider column 1 of $\mathscr{M}_2$: it contains r_2 and $N\beta_2/r_1$. Let $r_3 = \gcd(r_2, N\beta_2/r_1)$. Note that in Step 2 we choose $\gcd(\beta_2, r_2) = 1$. Thus

$$r_3 = \gcd\left(r_2, \frac{N\beta_2}{r_1}\right) = \gcd\left(r_2, \frac{N}{r_1}\right) = \gcd\left(r_1, -\alpha_1 p + \beta_1 h, \frac{N}{r_1}\right).$$

We claim that $r_3 = 1$. Suppose this is not true and $r_3 > 1$. Then every entry of $\mathscr{M}_1$ is a multiple of r_3. Since $\mathscr{M}_1 = \mathscr{L}_1\mathscr{M}$, we have

$$\mathscr{M} = \mathscr{L}_1^{-1}\mathscr{M}_1 = \frac{1}{\det(\mathscr{L}_1)}\begin{pmatrix} \frac{l}{r_1} & -\beta_1 \\ -\frac{n}{r_1} & \alpha_1 \end{pmatrix}\begin{pmatrix} r_1 & -\alpha_1 p + \beta_1 h \\ 0 & \frac{N}{r_1} \end{pmatrix}.$$

That is,

$$\mathscr{M} = \frac{1}{\det(\mathscr{L}_1)}\begin{pmatrix} \frac{l}{r_1} & -\beta_1 \\ -\frac{n}{r_1} & \alpha_1 \end{pmatrix} r_3 \begin{pmatrix} \frac{r_1}{r_3} & \frac{-\alpha_1 p+\beta_1 h}{r_3} \\ 0 & \frac{N}{r_1 r_3} \end{pmatrix}.$$

Since $r_3 = \gcd(r_1, -\alpha_1 p + \beta_1 h, \frac{N}{r_1})$,

$$\begin{pmatrix} \frac{r_1}{r_3} & \frac{-\alpha_1 p+\beta_1 h}{r_3} \\ 0 & \frac{N}{r_1 r_3} \end{pmatrix}$$

is integral. Since $\det(\mathscr{L}_1) = \pm 1$, every entry of $\mathscr{M}$ must be a multiple of r_3. Then $\gcd(l,h,p,n) \geqslant r_3 > 1$; this contradicts with the assumption that $\gcd(l,h,p,n) = 1$. Therefore $r_3 = 1$.

Since $r_3 = \gcd(r_2, N\beta_2/r_1)$ and $r_3 = 1$, by Lemma 1, there exist integers α_3 and β_3 such that $\alpha_3 r_2 + \beta_3(N\beta_2/r_1) = 1$. Let

$$\mathscr{L}_2 = \begin{pmatrix} \alpha_3 & \beta_3 \\ \frac{-N\beta_2}{r_1} & r_2 \end{pmatrix}$$

and let $\mathscr{M}_3 = \mathscr{L}_2\mathscr{M}_2$. Then

$$\mathscr{M}_3 = \begin{pmatrix} \alpha_3 & \beta_3 \\ \frac{-N\beta_2}{r_1} & r_2 \end{pmatrix}\begin{pmatrix} r_2 & 0 \\ \frac{N\beta_2}{r_1} & \frac{N}{r_2} \end{pmatrix} = \begin{pmatrix} 1 & \frac{\beta_3 N}{r_2} \\ 0 & N \end{pmatrix}.$$

Let

$$\mathscr{R}_2 = \begin{pmatrix} 1 & -\frac{\beta_3 N}{r_2} \\ 0 & 1 \end{pmatrix}.$$

and let $\mathscr{M}_4 = \mathscr{M}_3\mathscr{R}_2$. Then

$$\mathscr{M}_4 = \begin{pmatrix} 1 & \frac{\beta_3 N}{r_2} \\ 0 & N \end{pmatrix}\begin{pmatrix} 1 & -\frac{\beta_3 N}{r_2} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & N \end{pmatrix} = \mathscr{S}(\mathscr{M}).$$

From the above, $\mathscr{L}_2\mathscr{L}_1\mathscr{M}\mathscr{R}_1\mathscr{R}_2 = \mathscr{S}(\mathscr{M})$. Moreover, $\mathscr{L}_1$, $\mathscr{L}_2$, $\mathscr{R}_1$ and $\mathscr{R}_2$ are unimodular integral matrices. Let $\mathscr{L} = \mathscr{L}_1\mathscr{L}_2$. Then

$$\mathscr{L} = \begin{pmatrix} \alpha_3 & \beta_3 \\ \frac{-N\beta_2}{r_1} & r_2 \end{pmatrix}\begin{pmatrix} \alpha_1 & \beta_1 \\ \frac{n}{r_1} & \frac{l}{r_1} \end{pmatrix} = \begin{pmatrix} \alpha_3\alpha_1 + \frac{\beta_1 n}{r_1} & \alpha_3\beta_1 + \frac{\beta_3 l}{r_1} \\ \frac{-N\beta_2\alpha_1 + r_2 n}{r_1} & \frac{-N\beta_2\beta_1 + r_2 l}{r_1} \end{pmatrix}.$$

Using the facts that $N = lh - pn$ and $\alpha_1 l + \beta_1(-n) = r_1$ and $\alpha_2 r_1 + \beta_2(-\alpha_1 p + \beta_1 h) = r_2$, we have $(-N\beta_2\alpha_1 + r_2 n)/r_1 = \alpha_2 n - \beta_2 h$ and $(-N\beta_2\beta_1 + r_2 l)/r_1 = \alpha_2 l - \beta_2 p$. Thus if $S_1 = \alpha_2 n - \beta_2 h \pmod N$ and $s_2 = \alpha_2 l - \beta_2 p \pmod N$, then $DL(N; s_1, s_2)$ realizes L.

It is clear that Steps 1, 2, and 3 can be done in $O(\log N)$ time by using the Euclidean algorithm. Step 4 can be done in $O((\log N)^2)$ time by using ALGORITHM-MODIFIED-EUCLIDEAN. Step 5 can be done in $O(1)$ time. Thus ALGORITHM-COMPUTING-STEPS takes at most $O((\log N)^2)$. □

The following theorem will be used in the follow-up discussions.

Theorem 6 (Chen and Hwang [4]). *Suppose s'_1, s'_2 realize L-shape(l,h,n,p). Let x and y be integers such that $s'_2x - s'_1y = 1$. Then $s_1 = nx - hy \pmod N$ and $s_2 = lx - py \pmod N$ realize L-shape(l,h,p,n); moreover, s_1, s_2 can be derived from s'_1, s'_2 in $O(\log N)$ time.*

Theorem 7. *If $\gcd(l,n) = 1$ or $\gcd(l,p) = 1$ or $\gcd(h,p) = 1$ or $\gcd(h,n) = 1$, then we could use ALGORITHM-COMPUTING-STEPS to find the steps s_1 and s_2 of L-shape(l,h,p,n) in only $O(\log N)$ time.*

Proof. There are four cases:

Case 1: $\gcd(l,n) = 1$. Then, clearly, $r_1 = \gcd(l,-n) = 1$. Hence $r_2 = 1$ and Step 4 of ALGORITHM-COMPUTING-STEPS takes only $O(\log N)$ time. Thus ALGORITHM-COMPUTING-STEPS finds the steps s_1 and s_2 in only $O(\log N)$ time.

Case 2: $\gcd(l,p) = 1$. By an argument similar to that in Case 1, we could use ALGORITHM-COMPUTING-STEPS to find the steps s'_1 and s'_2 of L-shape(l,h,n,p) in only $O(\log N)$ time. Then, by Theorem 6, s_1, s_2 could be derived from s'_1, s'_2 in $O(\log N)$ time.

Case 3: $\gcd(h,p) = 1$. By an argument similar to that in Case 1, we could use ALGORITHM-COMPUTING-STEPS to find the steps s'_1 and s'_2 of L-shape(h,l,n,p) in only $O(\log N)$ time. Since L-shape(h,l,n,p) is the flipping of L-shape(l,h,p,n), $s_1 = s'_2$ and $s_2 = s'_1$.

Case 4: $\gcd(h,n) = 1$. Again, by an argument similar to that in Case 1, we could use ALGORITHM-COMPUTING-STEPS to find the steps s''_1 and s''_2 of L-shape(h,l,p,n) in only $O(\log N)$ time. Then, by Theorem 6, the steps s'_1, s'_2 of L-shape(h,l,n,p) could be derived from s''_1, s''_2 in $O(\log N)$ time. Since L-shape(h,l,n,p) is the flipping of L-shape(l,h,p,n), $s_1 = s'_2$ and $s_2 = s'_1$. □

We now compare the three existing algorithms for computing the steps of double-loop networks: the Smith normalization method [1,7], the sieve method [3,11], and our algorithm. Both the Smith normalization method and our algorithm take at most $O((\log N)^2)$ time. In the Smith normalization method, one needs to find nonsingular unimodular integral matrices $\mathscr{L}_i, \ldots, \mathscr{L}_2, \mathscr{L}_1, \mathscr{R}_1, \mathscr{R}_2, \ldots, \mathscr{R}_k$ such that

$$\mathscr{L}_i \cdots \mathscr{L}_2 \mathscr{L}_1 \mathscr{M} \mathscr{R}_1 \mathscr{R}_2 \cdots \mathscr{R}_k = \mathscr{S}(\mathscr{M}).$$

Our algorithm is based on the Smith normalization method, but our algorithm does not require any matrix operation; moreover, as could be seen from the proof of Theorem

5, we prove that there exist nonsingular unimodular integral matrices $\mathscr{L}_2, \mathscr{L}_1, \mathscr{R}_1, \mathscr{R}_2$ such that

$$\mathscr{L}_2\mathscr{L}_1\mathscr{M}\mathscr{R}_1\mathscr{R}_2 = \mathscr{S}(\mathscr{M}).$$

Therefore, our algorithm greatly simplifies the computation of the Smith normalization method.

Both the sieve method and our algorithm are very simple and easy to implement. The sieve method shows that the steps of a double-loop network are of the form

$$s_1 = kn + h \pmod N, \quad s_2 = kl + p \pmod N,$$

and our algorithm shows that the steps of a double-loop network are of the form

$$s_1 = \alpha_2 n - \beta_2 h \pmod N, \quad s_2 = \alpha_2 l - \beta_2 p \pmod N.$$

The sieve method takes at most $O(P_{\eta(N)}\log N)$ time. However, we are unable to predict the value of $P_{\eta(N)}$ and therefore unable to tell which algorithm is more efficient.

It is open whether the steps of a double-loop network can be found in $O(\log N)$ time. Note that Cheng and Hwang [5] gave an $O(\log N)$ time algorithm to compute the L-shape of a double-loop network $DL(N; s_1, s_2)$. It is also open whether we can find integers x and y such that $xa + yb = \gcd(a, b)$ and $(y, \gcd(a, b)) = 1$ in only $O(\log N)$ time, where a and b are integers, not both zero. If this is true, then ALGORITHM-COMPUTING-STEPS would take only $O(\log N)$ time and the steps of a double-loop network can be find in $O(\log N)$ time.

Acknowledgements

We thank Prof. Frank K. Hwang for many helpful comments. We also thank the referees for many constructive comments that greatly improve the presentation of this paper.

References

[1] F. Aguiló, M.A. Fiol, An efficient algorithm to find optimal double loop networks, Discrete Math. 138 (1995) 15–29.
[2] J.-C. Bermond, F. Comellas, D.F. Hsu, Distributed loop computer networks: a survey, J. Parallel Distribut. Comput. 24 (1995) 2–10.
[3] C. Chen, F.K. Hwang, The minimum distance diagram of double-loop networks, IEEE Trans. Comput. 49 (2000) 977–979.
[4] C. Chen, F.K. Hwang, Equivalent nondegenerate L-shapes of double-loop networks, Networks 36 (2000) 118–125.
[5] Y. Cheng, F.K. Hwang, Diameters of weighted double loop networks, J. Algorithms 9 (1988) 401–410.
[6] P. Erdös, D.F. Hsu, Distributed loop networks with minimum transmission delay, Theoret. Comput. Sci. 100 (1992) 223–241.
[7] P. Esqué, F. Aguiló, M.A. Fiol, Double commutative-step diagraphs with minimum diameters, Discrete Math. 114 (1993) 147–157.

[8] M.A. Fiol, M. Valero, J.L.A. Yebra, I. Alegre, T. Lang, Optimization of double-loop structures for local networks, Proceedings of the XIX International Symposium MIMI'82, Paris, France, 1982, pp. 37–41.
[9] M.A. Fiol, J.L.A. Yebra, I. Alegre, M. Valero, A discrete optimization problem in local networks and data alignment, IEEE Trans. Comput. C-36 (1987) 702–713.
[10] F.K. Hwang, A survey on double-loop networks, in: F. Roberts, F.K. Hwang, C. Monma (Eds.), Reliability of Computer and Communication Networks, AMS Series, 1991, pp. 143–151.
[11] F.K. Hwang, A complementary survey on double-loop networks, Theoret. Comput. Sci., to appear.
[12] F.K. Hwang, Y.H. Xu, Double loop networks with minimum delay, Discrete Math. 66 (1987) 109–118.
[13] M. Newman, Integral Matrices, Pure and Appl. Math. Series, Vol. 45, Academic Press, New York, 1972.
[14] J.M. Peha, F.A. Tobagi, Analyzing the fault tolerance of double-loop networks, IEEE Trans. Network 2 (1994) 363–373.
[15] C.K. Wong, D. Coppersmith, A combinatorial problem related to multimodule memory organizations, J. Assoc. Comput. Mach. 21 (1974) 392–402.