ELSEVIER Theoretical Computer Science 263 (2001) 211–229

Theoretical Computer Science

www.elsevier.com/locate/tcs

A complementary survey on double-loop networks

F.K. Hwang

Department of Applied Mathematics, National Chiao Tung University, Hsin Chu, Taiwan 300, ROC

Accepted April 2000

Abstract

We give a survey on double-loop networks with emphasis on new development since the surveys in 1986, 1991 and 1995.

1. Introduction

Wong and Coppersmith [29] introduced the multiloop networks $ML(N; s_1, \ldots, s_l)$ for organizing multimodule memory devices. The network can be viewed as a directed graph with N nodes $0, 1, \ldots, N-1$ and lN links of l types, where the type-i links, $i = 1, \ldots, l$, are

$$v \to v + s_i \,(\mathrm{mod}\, N), \quad v = 0, 1, \ldots, N-1.$$

Wong and Coppersmith set $s_1 = 1$. Thus $ML(s_1)$ is simply a ring, which is known to have long delay and low reliability. On the other hand, $ML(N; s_1, \ldots, s_l)$ for $l \geqslant 3$, besides consuming a lot of hardware, would require each node to have sophisticated control ability in switching the l inlinks to the l outlinks. Thus the double loop, which will be denoted by $DL(N; a, b)$ (with $s_1 = a, s_2 = b$), is a happy medium for most practical purposes. For examples, $DL(N; 1, N-1)$ was first proposed by Liu [24], called a *distributed double-loop computer network*, and is the topology of the SONET ring, the fiber distributed data interface network and distributed queue dual bus. The class $DL(N; 1, s)$ for $2 \leqslant s \leqslant N-1$ was proposed by Raghavendra et al. [26] for computer networks. In particular, $DL(N; 1, N-2)$, called a *daisy chain*, was highly acclaimed by Grnarov et al. [14]. Fiol et al. [13] proposed double-loop networks for data alignment in SIMD processors.

Several surveys [2, 17, 27] have been published. The current survey will minimize its overlapping with them by focusing on new material. These include not just updated versions of important results, but also results being somewhat overlooked before or

PII: S0304-3975(00)00243-7

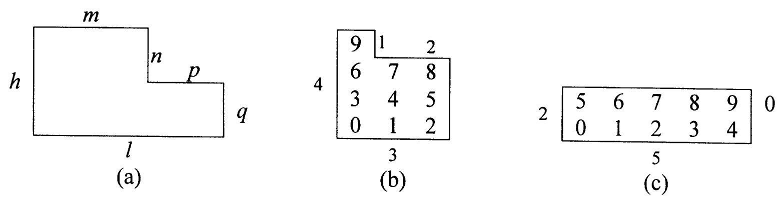


Fig. 1. Some L-shapes.

presented only in non-English literature, or simply results which we find a new way to organize.

2. The L-shape

It is well known that a regular digraph is strongly connected if and only if it is connected. Since $DL(N;a,b)$ is a 2-regular diagraph, we will substitute *connectivity* for *strong connectivity* throughout the paper for brevity.

Theorem 2.1. *$DL(N;a,b)$ is connected if and only if* $\gcd(N,a,b)=1$.

Proof. If $\gcd(N,a,b)=d>1$, then clearly a node i cannot reach a node j if $i\not\equiv j \pmod d$. On the other hand, suppose $\gcd(N,a,b)=1$. Then there exist nonnegative α and β such that

$$\alpha a+\beta b\equiv 1 \pmod N.$$

Assume

$$j-i\equiv k \pmod N,$$

where k is nonnegative. Then i can reach j by taking $k\alpha$ a-steps and $k\beta$ b-steps. □

When $DL(N;a,b)$ is connected, we can define a minimum distance diagram (MDD) as an array with node 0 in cell $(0,0)$ and node v in cell (i,j) (i is the column index and j the row index) if and only if $ia+jb\equiv v \pmod N$ and $i+j$ is minimum (in case of tie, take the minimum j). Since $DL(N;a,b)$ is node-symmetric, the minimum distance from u to v is same as from 0 to $v-u$.

Wong and Coppersmith proved that an MDD is always an L-shape (see Fig. 1(a)) which can be characterized by six parameters l, h, m, n, p, q (4 of them independent). Fig. 1(b) gives the MDD of $DL(10;1,3)$ in its L-shape.

Fig. 1(c) gives the MDD of $DL(10;1,5)$ in its L-shape which degenerates into a rectangle by having $p=0$. Unlike the nondegenerate case, the determination of m, n, p, q is not automatic. This will be discussed later.

Fiol et al. [13] observed that an L-shape always tessellates the plane (see Fig. 2) regardless of the L-shape is degenerate or not.

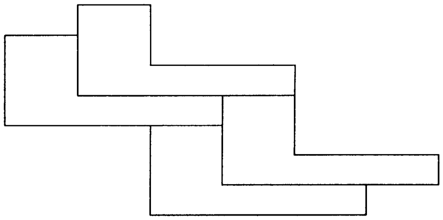


Fig. 2. Tessellation.

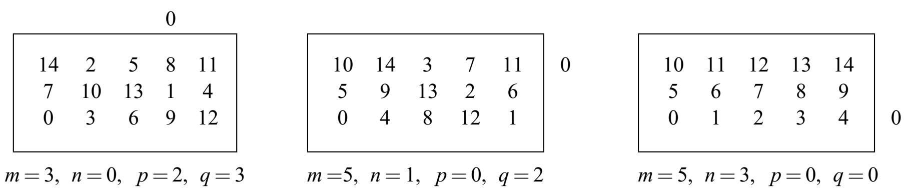


Fig. 3. Parameters for degenerate L-shapes.

By noting the location of cells containing element 0, Fiol et al. [13] obtained the following equations:

$$\begin{aligned} &la - nb \equiv 0 \,(\mathrm{mod}\, N), \\ &-pa + hb \equiv 0 \,(\mathrm{mod}\, N). \end{aligned} \tag{2.1}$$

Chen and Hwang [5] used this observation to prove that an L-shape is degenerate if and only if exactly one of the two congruences: $la \equiv 0 \,(\mathrm{mod}\, N)$ and $hb \equiv 0 \,(\mathrm{mod}\, N)$ is satisfied. They defined m, n, p, q using the following rules:

(i) Suppose $hb \not\equiv la \equiv 0 \,(\mathrm{mod}\, N)$. Let the zero immediately above the L-shape be at column j. Then $m = j$, $n = 0$, $p = l - j$, $q = h$.

(ii) Suppose $la \not\equiv hb \equiv 0 \,(\mathrm{mod}\, N)$. Let the zero immediately to the right of the L-shape be at row i. Then $m = l$, $n = h - i$, $p = 0$, $q = i$.

(iii) Suppose $la \equiv hb \equiv 0 \,(\mathrm{mod}\, N)$. If $h > l$, follow rule (i); otherwise, follow rule (ii).

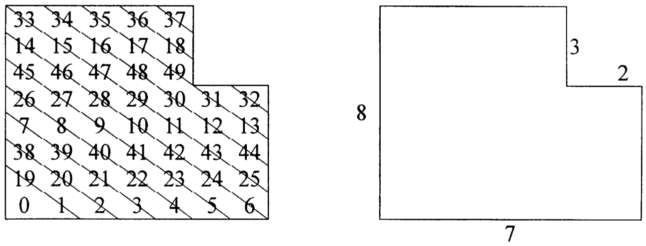
See Fig. 3 for examples.

This definition satisfies Eqs. (2.1) which are basic to many developments.

Wong and Coppersmith gave an $O(N)$ time algorithm to construct the MDD (hence the L-shape) diagonally starting from the $(0,0)$ cell. Namely, at step i we fill in the cells which are distance i away from node 0, unless the node to be filled in is already used. Fig. 4(a) shows how the MDD of $DL(50; 1, 19)$ is constructed this way.

It should be noted that any distance function can be obtained from the L-shape directly, without the MDD. For example, the diameter of $DL(N; a, b)$, written as $D(N; a, b)$, is $l + h - \min\{n, p\} - 2$, and the average distance $\bar{D}(N; a, b)$ is

$$\frac{1}{N}\left[\left(\frac{lh}{2} - np\right)(l + h - 2) + \frac{np(n + p - 2)}{2}\right].$$



(a) (b)

Fig. 4. The MDD of $DL(50;1,19)$ and its L-shape.

Cheng and Hwang [6] gave an $\mathrm{O}(\log N)$ time algorithm, based on the Euclidean algorithm, to compute the L-shape:

Assume $\gcd(N,a)=1$ (if $\gcd(N,a)=d>1$, replace N by $N/d, a$ by a/d and b by $b(\mathrm{mod}\, N/d)$).

Step 1. set $s_{-1}=N$. Let $0 \leqslant s_0 < N$ be the integer satisfying

$$as_0 + b \equiv 0 \pmod N,$$

and let $s_i,\ q_i,\ 1 \leqslant i \leqslant m+1$, be recursively defined by

$$s_i = s_{i-2} - q_i s_{i-1}, \quad 0 \leqslant s_i < s_{i-1},$$

where m is chosen such that $s_{m+1}=0$.

Step 2. Define $U_{-1}=0$, $U_0=1$ and

$$U_{i+1} = q_{i+1}U_i + U_{i-1}, \quad 0 \leqslant i \leqslant m.$$

Note that s_i is decreasing in i and U_i increasing. Hence

$$0 = \frac{s_{m+1}}{U_{m+1}} < \frac{s_m}{U_m} < \cdots < \frac{s_0}{U_0} < \frac{s_{-1}}{U_{-1}} = \infty.$$

Step 3. Let u be the largest odd integer such that $s_u/U_u > 1$. Define

$$v = \left\lceil \frac{s_u - U_u}{s_{u+1} + U_u} \right\rceil - 1,$$

Then $l = s_u - vs_{u+1},\ h = U_u + (v+1)U_{u+1},\ n = U_u - vU_{u+1},\ p = s_u - (v+1)s_{u+1}$.

J.C. Chang (private communication) observed that Cheng–Hwang algorithm actually only requires $\mathrm{O}(\log b)$ time since in the Euclidean algorithm, N is reduced to b in one step.

Example 1. For DL(50;1,19), we have

$$\begin{aligned}
50 &= 1\cdot 31+19, & U_1 &= 1\cdot 1+0=1,\\
31 &= 1\cdot 19+12, & U_2 &= 1\cdot 1+1=2,\\
19 &= 1\cdot 12+7, & U_3 &= 1\cdot 2+1=3,\\
12 &= 1\cdot 7+5, & U_4 &= 1\cdot 3+2=5,\\
7 &= 1\cdot 5+2, & U_5 &= 1\cdot 5+3=8,\\
5 &= 2\cdot 2+1, & U_6 &= 2\cdot 8+5=21,\\
2 &= 2\cdot 1+0, & U_7 &= 2\cdot 21+8=50,
\end{aligned}$$

From

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

we find $u=3$. Hence $v=0$, $l=7-0\cdot 5=7$, $h=3+1\cdot 5=8$, $n=3-0\cdot 5=3$, $p=7-1\cdot 5=2$.

Cheng and Hwang also extended their results to the weighted link case, i.e., the two types of edges have different costs.

There is the dual question of finding (a,b) from a given (l,h,n,p) L-shape. Fiol et al. [13] (also see Chen and Hwang [3]) proved

Theorem 2.2. *Necessary and sufficient conditions that $L(l,h,n,p)$ can be implemented is that $l>n$, $h\geqslant p$ and* $\gcd(l,h,n,p)=1$.

Note that Eqs. (2.1) can also be written as

$$\begin{pmatrix} l & -n \\ -p & h \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = N\begin{pmatrix} \alpha \\ \beta \end{pmatrix},$$

or

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} h & n \\ p & l \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

for some integers α, β. Fiol et al. [1, 11] proposed the Smith normalization method to solve for a and b. They proved:

Theorem 2.3. *There exists unimodular, integral 2×2 matrices L and R such that*

$$L\begin{pmatrix} l & -p \\ -n & h \end{pmatrix}R = S = \begin{pmatrix} 1 & 0 \\ 0 & N \end{pmatrix} \quad (\textit{the Smith normal form}).$$

Furthermore, let

$$L = \begin{pmatrix} w & x \\ y & z \end{pmatrix}.$$

Then DL$(N;y,z)$ implements $L(h,l,n,p)$ and (y,z) is unique up to isomorphism.

6	7	8		6	8	1
3	4	5		3	5	7
0	1	2		0	2	4

Fig. 5. Two nonisomorphic double loops.

The computation of L and R involves solving for q_1, q_2 in

$$q_1 u - q_2 v = 1$$

for various pairs of (u, v).

For general $L(l, h, n, p)$, Chen and Hwang [3] gave the following method to find a and b.

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

$$a_k = h + kn, \qquad b_k = p + kl.$$

Let F_k denote the set of prime factors of $\gcd(a_k, b_k)$ and F the set of prime factors of N. They used the sieve method in number theory to show the existence of a k such that $f \notin F_k$ for all $f \in F$. Then (a_k, b_k) is a solution of (2.1). For $L(6, 4, 3, 2)$, we easily find the solution $a = h = 4$ and $b = p = 3$. The following example shows that sometimes we have to explore a few k.

Example 2. Consider $L(187, 112, 22, 7)$ with $N = 20790$. Then $f \in \{2, 3, 5, 7, 11\}$ divides N.

k	0	1	2	3	4	5	6
a_k	112	134	156	178	200	222	244
b_k	7	194	381	568	755	942	1129
$f \in F \cap F_k$	7	2	3	2	5	2,3	ø

Hence $DL(20790; 244, 1129)$ implements $L(187, 112, 22, 7)$.

3. Isomorphism and equivalence

$DL(N; a, b)$ and $DL(N; a', b')$ are *isomorphic* (in the graph sense) if $\{a, b\} = h \cdot \{a', b'\}$ for some h prime to N. Clearly, isomorphic double loops have the same L-shape, but the converse is not true. Fig. 5 shows two nonisomorphic double loops with the same L-shape.

It is of interest to determine the necessary and sufficient conditions that two nonisomorphic double loops have the same L-shape.

Two L-shapes are called *equivalent* if they have the same number of cells which are distance-k away from cell $(0, 0)$ for every k. Clearly, equivalent L-shapes have the same diameter, same average distance and same value for any distance function. Hwang and Xu [22] first introduced this notion of equivalence and proved that $DL(N; 1, s)$ is

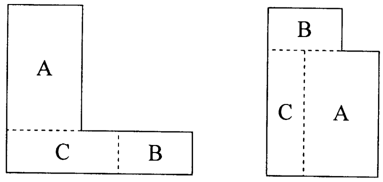


Fig. 6. The 3-rectangle transformation.

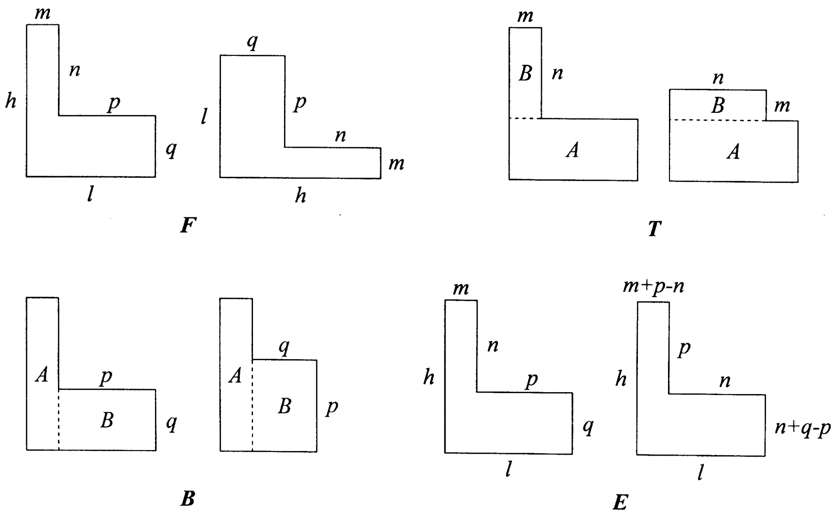


Fig. 7. Four geometrical transformations.

equivalent to $DL(N; 1, N+1-s)$. They showed that the two L-shapes can be obtained from each other through a 3-rectangle transformation as shown in Fig. 6.

Rödseth [28] gave an equivalence theorem for the multiloop. Its double-loop version is as follows:

Theorem 3.1. *$DL(N; a, b)$ is equivalent to $DL(N; N-a, b-a)$ and $DL(N; a-b, N-b)$.*

Note that the Hwang–Xu result is half of the special case $a = 1$. Recently, Chen and Hwang [4] characterized all equivalent nondegenerate L-shapes, and showed that they can be obtained through four geometric transformations F (flipping), T (top turning), B (bottom turning) and E (empty turning) (see Fig. 7).

It is of interest to find out the algebraic transformation (a, b) to (a', b') corresponding to a geometric transformation. Hwang et al. [16] showed that Rödeseth's transformations correspond to the 3-rectangle transformation, denoted by H. Chen and Hwang proved

$$FH = TFE = BT = FEB.$$

Since $(a',b')=(b,a)$ for F, a solution of algebraic transformation for any of T, B, V will solve the others through the above equations. Chen and Hwang gave such a solution for E.

Theorem 3.2. *Suppose* $1\leqslant a,\ b\leqslant N-1$. *Let* x *and* y *be integers such that* $bx-ay=1$. *Then* $DL(N,a',b')$ *with* $a'=px-hy\,(\mathrm{mod}\,N)$ *and* $b'=lx-ny\,(\mathrm{mod}\,N)$ *realizes* $E(L)$.

Proof. Suppose $1\leqslant a,\ b\leqslant N-1$. If $\gcd(a,b)=d$, take $a^*=a/d$ and $b^*=b/d$. Since $\gcd(N,a,b)=1$ implies $\gcd(d,N)=1$, $DL(N;a,b)$ is isomorphic to $DL(N;a^*,b^*)$. Therefore, we may assume $d=1$ since otherwise we could work with (a^*,b^*). Let

$$L=\begin{pmatrix}\frac{hb-pa}{N} & \frac{-la+nb}{N}\\ px-hy & lx-ny\end{pmatrix},\qquad M=\begin{pmatrix} l & -n\\ -p & h\end{pmatrix},\quad\text{and}\quad R=\begin{pmatrix} x & a\\ y & b\end{pmatrix}.$$

By Eqs. (2.1), $(hb-pa)/N$ and $(-la+nb)/N$ are integers. M is the corresponding matrix of $FV(L)$. It is easily verified that both L and R are nonsingular unimodular integral matrices and $LMR=diag(1,N)$, the Smith normal form $S(M)$ of M. By Theorem 3.3 given later, $DL(N;a',b')$ with $a'=px-hy\ (\mathrm{mod}\,N)$ and $b'=lx-ny\ (\mathrm{mod}\,N)$ realizes $FV(L)$. □

Note that (x,y) can be solved by the Euclidean algorithm which takes $O(\log N)$ time.

4. Diameter and average distance

The diameter represents the worst delay in the communication between two nodes, and the average distance the average delay. Wong and Coppersmith [29] proved

Theorem 4.1. $D(N;a,b)\geqslant\sqrt{3N}-2$ *and* $\bar{D}(N;a,b)\geqslant\sqrt{25N/27}-1$.

A double loop is called *tight* if it achieves the lower bound $\lceil\sqrt{3N}\,\rceil-2$.

Since the diameter and the average distance can be obtained from the L-shape directly, one approach to the problem of determining a and b such that $D(N;a,b)$ (or $\bar{D}(N;a,b)$) is minimized is to determine the desirable L-shapes first, then solving for (a,b). This actually motivated Theorem 2.2.

Note that by setting h and l as integers close to $2\sqrt{N/3}$, and n and p as integers close to $\sqrt{N/3}$, then the diameter would be close to the lower bound $\lceil\sqrt{3N}\,\rceil-2$. For many N, this approach can quickly find L-shapes with short diameters. In particular, Esqué et al. [11] developed a method to characterize all tight L-shapes. For general N, Aguiló and Fiol [1] gave an algorithm to search an L-shape with diameter $\lceil\sqrt{3N}\,\rceil-2+k$ in the order $k=0,1,2,\ldots$. The first-found L-shape must have minimum diameter. They estimated the time complexity of this algorithm to be $O(k^3)O(\log N)$ for each fixed k, where k is upper bounded by $O(N^{1/4})$ by a result of Hwang and Xu [22].

A second approach to find double loops with short diameters with a given N is to determine a and b directly, not via the determination of an L-shape first. Three heurisitics have been proposed.

Wong and Coppersmith [29] proposed setting $a=1$ and $b=\lceil\sqrt{N}\,\rceil$. The diameter is about $2\sqrt{N}$. Hwang and Xu [22] proposed setting $a=1$ and $b=\lfloor(N-1)/\alpha\rfloor$ initially, where $\alpha=\lfloor\sqrt{N/3}\rfloor$, then a calibration which can be computed in constant time results in a double loop whose diameter is upper bounded by

$$\sqrt{3N}+2(3N)^{1/4}+5 \quad \text{for } N\geqslant 6348.$$

Rödseth [28] gave a better calibration which upper bounds the diameter by

$$\sqrt{3N}+(3N)^{1/4}+\tfrac{5}{2}$$

and the average distance by

$$\tfrac{5}{9}\sqrt{3N}+\tfrac{5}{3}(3N)^{1/4}+\tfrac{21}{2}$$

for $N\geqslant 1200$.

Suppose the N nodes are arranged into a cycle. A routing algorithm is called a *k-pass algorithm* if a path (as seen going around the cycle) is allowed to pass the source at most k times. Write

$$N=\alpha b+\beta \quad \text{where } 0\leqslant\beta<b.$$

Erdös and Hsu [9] proved.

Theorem 4.2. *The diameter of an optimal* 1*-pass algorithm for* $DL(N;1,b)$ *is upper bounded by* $\max\{b+\alpha-\beta-2,\ 2\alpha+\beta-1\}$.

Proof. Without loss of generality, assume node 0 is the source and node t is the destination. Note that if the path takes $\alpha+i$ b-steps for $1\leqslant i\leqslant\alpha$, i.e., the path passes the source once, then it lands in the interval $[(i-1)b,ib)$ since $(i-1)b<(\alpha+i)b<ib$. Thus the interval is partitioned into two subintervals $[(i-1)b,(\alpha+i)b)$ and $[(\alpha+i)b,ib)$ with lengths $b-\beta$ and β. If t is in the first subinterval, then a 0-pass path has length

$$(i-1)+[t-(i-1)b]\leqslant\alpha-1+b-\beta-1;$$

if t is in the second subinterval, then a 1-pass has length

$$(\alpha+i)+\beta-1\leqslant 2\alpha+\beta-1. \qquad \square$$

Selecting $a=1$ and b to minimize $\max\{b+\alpha-\beta-2,2\alpha+\beta-1\}$ can be viewed as a heuristic for finding a double loop with short diameter. Erdös and Hsu claimed that by selecting $b=(1+o(1))\sqrt{3N}$, the bound in Theorem 4.2 is $(1+o(1))\sqrt{3N}$. They also quoted a private communication from Coppersmith who had proved that there exists an infinite number of N for which the diameter is lower bounded by $\sqrt{3N}+c(\log N)^{1/4}$.

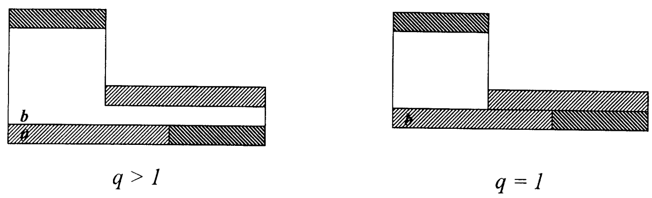


Fig. 8. The change of origin.

Cheng et al. [7] gave the *fault diameter* D_F when either a node or a link is the only failed component. A revision was recently given by Hwang [18]. Let $d_f(s,t)$ denote the distance from node s to node t when node f is faulty.

Lemma 4.3. *$d_f(s,t)=d(s,t)$ except when $0, f, t$ are collinear in the L-shape (with origin s) and f precedes t. In the exception case, for $q \geqslant 2$,*

$$d_f(s,t) = 1 + d(s+b,t) \quad \begin{cases} h+m-l+t/a & \text{if } l-t/a \leqslant m, \\ 1+m+l-t/a & \text{if } l-t/a > m. \end{cases}$$

For $q=1$ let

$$l - t/a = xm - y \quad \text{where } 0 \leqslant y < m.$$

Then

$$d_f(s,t) = x + h - 1 + y.$$

Proof. If the case is not the exception, then clearly there exists a path in the L-shape from s to t by passing f. In the exception case, assume without loss of generality that s, f, t are in the same row. Then the path has to take a b-step and we may assume the first step is a b-step. Therefore, after the first step, the new source is the node $s+b$. Notice that the L-shape with origin $s+b$ can be obtained from the L-shape with origin s by moving the first p entries of the bottom row to the top of the last p columns, and the last m entries to the top of the first m columns (see Fig. 8). For $q \geqslant 2$, $s+b, f, t$ are no longer collinear. By tracking whether t is moved to the top and its exact location, we obtain $d_f(s,t)$. For $q=1$ (see Fig. 9), repeat such moves until t is moved to the top of the first m columns; then there exists a shortest path from the origin to t not blocked by f. In Lemma 4.3, x counts the number of moves, and y the column index of t at the end. □

Corollary 4.4. $D_F(N;a,b) = \max\{\alpha + h + m - 2,\ \beta + l + q - 2\}$, *where*

$$\alpha = \begin{cases} \left\lceil \dfrac{l-2}{m} \right\rceil & \text{if } q = 1, \\ 1 & \text{otherwise} \end{cases}$$

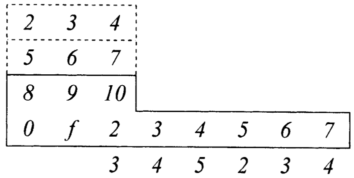


Fig. 9. Fault-tolerant distance with $q=1$.

and

$$\beta = \begin{cases} \left\lceil \dfrac{h-2}{q} \right\rceil & \text{if } m=1, \\ 1 & \text{otherwise.} \end{cases}$$

Proof. For $q=1$, $\max_t d_f(s,t)$ occurs when t is the node located at cell (h,m) after $\lceil (l-2)/m \rceil$ b-steps. By Lemma 4.3,

$$D_F(N;a,b) \geqslant \alpha + h + m - 2.$$

By symmetry of the vertical and the horizontal directions, we also have

$$D_F(N;a,b) \geqslant \beta + l + q - 2. \qquad \square$$

Example 3. Consider $DL(10;1,8)$. Suppose $f=1$. $d_f(0,i)$ for $2 \leqslant i \leqslant 7$ is marked under node i (Fig. 9).

Corollary 4.5. *$DL(N;a,b)$ is 2-connected if and only if* $\gcd(N,a,b)=1$.

Proof. Since $DL(N;a,b)$ is a 2-regular digraph, its connectivity is at most 2. On the other hand, since the fault diameter is not infinity, $DL(N;a,b)$ is at least 2-connected. $\square$

5. Embedding

Certain data structures are preferred by certain classes of algorithms. For example, the existence of a Hamiltonian path facilitates the running of a pipeline algorithm; while the existence of a Hamiltonian circuit preserves the facility even when there is a faulty element.

In the following, the y in $\gcd(x,y)$ will be interpreted as the residue of $y \pmod{x}$. Define $\gcd(N;b-a)=d$. Fiol and Yebra [12] proved.

Theorem 5.1. *$DL(N;a,b)$ contains a Hamiltonian circuit if and only if there exists a k, $0 \leqslant k \leqslant d$, such that* $\gcd(N/d, a+k(b-a)/d)=1$.

Proof. Let F denote a 1-factor (a 1-regular digraph) of $DL(N;a,b)$. Suppose the b-link $i \to i+b$ is in F. Then the a-link $i+b-a \to i+b$ is not in F, which in turn forces the b-link $i+b-a \to i+2b-a$ to be in F. Repeating this argument N/d times, we obtain the statement that the a-link $i \to i+a$ is not in F. At this time, we obtain a subset of N/d b-links of F, called a *module*. If F has other links, we choose an arbitrary link to start and obtain another module (could consist of a-links now). Thus F consists of d modules each having N/d links.

Let p be a path which starts from node x of module M, visits every other module once and ends at node $y \neq x$ of module M. Suppose p consists of k b-links and $d-k$ a-links. Then

$$y \equiv x + kb + (d-k)a \equiv x + da + k(b-a) \pmod{N}.$$

Hence

$$y - x = da + k(b-a).$$

If $(y-x)/d$ is prime to N/d, then running p N/d times with $x+(j-1)(y-x)$ as the starting point for the jth time yields a Hamiltonian circuit. □

A digraph is said to be LFT (link fault-tolerant) Hamiltonian if it is Hamiltonian with any link fault. Similarly we can define NFT (node fault-tolerant). Recently, Lin [23] proved the following two results concerning LFT and NFT Hamiltonian circuits of double loops.

Theorem 5.2. *$DL(N;a,b)$ is LFT-Hamiltonian if and only if either* (a) *there exists a k, $0 \leqslant k \leqslant d$, such that* $\gcd(N/d,\ a+k(b-a)/d)=1$ *or* (b) $\gcd(N,a)=\gcd(N,b)=1$.

Proof. *Necessity*: Clearly, the necessary condition given in Theorem 5.1 of being Hamiltonian is also necessary for being LFT Hamiltonian. But if no Hamiltonian circuit using both types of links exists, then both a-links and b-links must form Hamiltonian circuits to be LFT.

Sufficiency. (b) is trivial. Suppose (a) is satisfied. Then there exists a Hamiltonian circuit H using both types of link. Since $DL(N;a,b)$ is node-symmetric, the Hamiltonian property is preserved by rotating the nodes on H. Without loss of generality, assume a type-a link $(i, i+a)$ is faulty. Let $(j, j+b)$ be a type-b link in H. Then we can obtain H' from H by rotating j to i (hence $j+b$ to $i+b$). Note that H' does not contain the faulty link. □

To discuss *NFT*, we assume without loss of generality that node 0 is the faulty node. Construct two sets $A=(a_0,a_1,\ldots)$ and $B=(b_0,b_1,\ldots)$ such that

$$a_i \equiv N-b+i(a-b) \pmod{N}, \quad i=0,1,\ldots,\min\{N-2, j-1\colon\ j(a-b)\equiv 0 \text{ or } b \pmod{N}\},$$

$$b_i \equiv N - a - i(a-b) \,(\mathrm{mod}\, N), \quad i = 0,1,\ldots,\min\{N-2, j-1:\ j(a-b)\equiv 0 \text{ or } -a \,(\mathrm{mod}\, N)\}.$$

For example, for $N=8$, $a=1$, $b=2$, $A=(6,5,4,3,2,1)$, $B=(7)$.

Lemma 5.3. *Suppose* $\gcd(N, a-b)=1$. *Then* (i) $|A|(a-b)\equiv b \,(\mathrm{mod}\, N)$, $|B|(a-b)\equiv -a \,(\mathrm{mod}\, N)$, (ii) $|A|+|B| = N-1$, (iii) $A\cap B = \emptyset$ *and* (iv) $A\cup B = \{1,\ldots,N-1\}$.

Proof. (i) $\gcd(N, a-b)=1$ implies $j(a-b)\equiv 0 \,(\mathrm{mod}\, N)$ has no solution for $j\geqslant 1$.
(ii) From (i),

$$(|A|+|B|)(a-b)\equiv b-a \,(\mathrm{mod}\, N)$$

or

$$(|A|+|B|)\equiv -1 \,(\mathrm{mod}\, N).$$

But

$$2\leqslant |A|+|B| \leqslant 2(N-1).$$

(ii) follows immediately.
(iii) Suppose to the contrary that $a_i = b_j$. Then

$$N - b + i(a-b) \equiv N - a - j(a-b) \,(\mathrm{mod}\, N)$$

or

$$(i+j+1)(a-b)\equiv 0 \,(\mathrm{mod}\, N).$$

Since $\gcd(N, a-b)=1$, we have

$$i+j+1\equiv 0 \,(\mathrm{mod}\, N), \quad \text{contradicting the fact } 2\leqslant i+j \leqslant N-4.$$

(iv) From (ii) and (iii), it suffices to prove $0\notin A\cup B$. But $a_i = 0$ implies

$$i(a-b)\equiv b \,(\mathrm{mod}\, N), \quad \text{contradicting the fact } A \text{ sequence stops before } i.$$

Similarly we can prove $0\notin B$. □

Theorem 5.4. *$DL(N; a, b)$ is NFT-Hamiltonian if and only if* (a) $\gcd(N, b-a)=1$, (b) $\gcd(|A|, N-1)=1$.

Proof. *Necessity*. For any Hamiltonian circuit C on $\{1,\ldots,N-1\}$, we prove by induction that a_i uses a-links and b_i uses b-links. Since $a_0 + b\equiv 0 \,(\mathrm{mod}\, N)\notin C$, a_0 must use an a-link. Suppose a_{k-1} uses an a-link. We show a_k must use an a-link:

$$a_k + b = [a_{k-1} + (a-b)] + b = a_{k-1} + a.$$

So the b-link of a_k ends at the node $a_{k-1}+a$, which already received an a-link from a_{k-1}, contradicting the fact that C is Hamiltonian.

(a) Suppose to the contrary that $\gcd(N, a-b)=d>1$. Then $a\equiv b \pmod d$. Consequently,

$$a_i \equiv b_j \pmod d \quad \text{for all } i, j,$$

contradicting the fact that a_i use a-links and b_j b-links on C.

(b) Define $(C_0,\ldots,C_{N-2})=(a_0,\ldots,a_{|A|-1},b_{|B|-1},\ldots,b_0)$ where the subscripts of C_i are modulo $N-1$. Then

$$\begin{aligned} C_i + a &\equiv N - b + i(a-b) + a \\ &\equiv N + (i+1)(a-b) - a - |B|(a-b) \quad \text{by Lemma 5.3(i)} \\ &\equiv N - a - (|B| - i - 1)(a-b) = b_{|B|-i-1} = C_{i+|A|} \quad \text{for } 0 \leqslant i \leqslant |A|-1 \\ C_i + b &\equiv b_{|B|-(i-|A|+1)} + b \equiv N - a - (|B| - i + |A| - 1)(a-b) + b \\ &\equiv N - a - (|B| - i - 1)(a-b) \equiv N - a + (i - |B|)(a-b) = a_{i-|B|} \\ &= a_{i+|A|-(N-1)} = C_{i+|A|-(N-1)} = C_{i+|A|} \quad \text{for all } |A| \leqslant i \leqslant N-2-|A|. \end{aligned}$$

Therefore $C_0 \to C_{|A|} \to C_{2|A|} \to \cdots \to C_{(N-2)|A|} \to C_0$, i.e., $\gcd(|A|, N-1)=1$.

Sufficiency. Since $\gcd(N, a-b)=1$, $A\cup B=\{1,\ldots,N-1\}$ by Lemma 5.3(iv). Define $(C_0,\ldots,C_{N-2})$ as before. Since $\gcd(|A|, N-1)=1$, $C_0 \to C_{|A|} \to C_{2|A|} \to \cdots \to C_{(N-2)|A|} \to C_0$ is a Hamiltonian circuit. □

6. Routing

First we discuss 2-terminal routing. Since the double loop is node symmetric, we may assume the routing is from node 0 to node t. The Cheng–Hwang algorithm, as discussed in Section 2, determines the L-shape by first finding a shortest path from node 0 to node 0. The same procedure can be used to compute a shortest path from node 0 to node t by simply changing the starting congruence to

$$as_0 + b \equiv t \pmod N.$$

Therefore a shortest path can be found in $O(\log N)$ time.

Cheng et al. [7] followed the above reasoning to give a routing algorithm which requires $O(\log N)$ time for preprocessing and constant processing time at each node on the route.

Guan [15] simplified the above procedure by noticing that there is no need to compute the shortest path to t. Call a path a-path (b-path) if it involves only a-steps (b-steps). As long as the a-path from s to t is not longer than the b-path, there exists a shortest path which takes an a-step at s, and vice versa. Computing the length of a-path and b-path also involves Euclidean algorithm and requires $O(\log N)$ time. Guan also extended his algorithm to the weighted case.

In case of a node failure known at f, the algorithm of Cheng et al. can be applied twice to locate both t and f. If 0, f, t are not collinear and lined up in that order in $L(N;a,b)$, then the routing is same as before except that any path through f should be avoided (one of the two *basic paths*, one taking all a-steps first and the other taking all b-steps first, will bypass f).

Suppose 0, f, t are collinear in that order. Without loss of generality, assume they are in the same row. Then the path will start with b-steps such that 0, f, t are no longer collinear. Then the non-collinear routing applies. Note that the time complexity is still O(log N).

Guan's algorithm is well suited to the case that f is unknown. Follow the algorithm until the next step, say, an a-step, hits f. Then take a b-step instead. If the length of the a-path is at least l, then t is not at the bottom row of the current L-shape; hence after the b-step, all shortest paths to t does not encounter f (which is at the bottom row). If the length of the a-path is less than l, then the algorithm enters the *fault mode* in which the location of f is known. In this mode, in addition to computing the lengths of the a-path and the b-path to t, the length of the a-path to f is also computed. Take a b-step unless the first length is shorter than the other two.

A *survival graph* has the N modes as the vertex set and an edge from u to v if there exists a shortest path from u to v not containing f. Escudero et al. [10] proved that the survival graph of a double-loop network with one faulty node has diameter 2.

The case of a simple faulty edge is analogous to the single faulty node case.

A *permutation routing* is to route N pairs of source-destination where the sources are all distinct and so are the destinations. A (permutation) routing is called *minimum* if every path is a shortest path, called *tight* if the number of steps required equal to the maximum distance of the N pairs, and called *oblivious* if the routing of each pair assumes no knowledge of other pairs. Hwang et al. [19] gave a surprising result that there exists a minimum, tight, oblivious permutation routing, which they called a big-foot algorithm because each path is basic with the big-steps first.

The Big-foot Algorithm

1. Construct the L-shape to determine the number of b-steps b_i and the number of a-steps a_i in a shortest path for the ith pair.
2. Each path is basic with the b-steps first.
3. If several paths compete for the same a-link, the path with the longest distance to go gets the priority. All other paths stay put during this step.

Theorem 6.1. *The big-foot algorithm works.*

Proof. Since all paths start from distinct sources, they end at distinct nodes after a big step. Therefore, all paths to take a big step next are at distinct nodes and do not compete with each other.

Paths competing for an a-link have only small steps left. Thus their remaining distance to their destinations are all different since the destinations are distinct. Therefore there exists a unique path with the longest remaining distance to go. □

Theorem 6.2. *The big-foot algorithm is minimum, tight and oblivious.*

Proof. "Minimality" follows from rule 1. "Obliviousness" follows from rule 2. "Tightness" follows from rule 3 since a path with the maximum distance moves at each step while a path with a distance k less than the maximum distance stays put in at most k steps. □

We can construct the MDD (with base 0) in $O(N)$ time and obtain (a_i, b_i) for all i by changing bases in another $O(N)$ time. Or if we have N processors, then we can simultaneously compute (a_i, b_i) using the 2-terminal routing in $O(\log b)$ time. On the other hand, each path can contain at most $b-1$ small steps (any b small steps can be replaced with big steps). Hence at most $b-1$ paths traverse an a-link since all destinations are distinct and bounded by $b-1$ small steps. In the worst scenario (which actually cannot happen) that all traversings of an a-link occur simultaneously, it takes $O(b \log b)$ time to order b distances at each node. So the ordering time for N nodes in $O(Nb \log b)$ with a single processor and $O(b \log b)$ for N processors.

A fault-tolerant big-foot algorithm remains an open problem.

7. Reliability

In the general reliability model, each node and each link has an individual probability to fail. A simplification in expression (though not in theory) is achieved by assuming all nodes have the same failure probability p, all a-links p_a and all b-links p_b which is called the *node-link model*. Some special cases are

(i) The *uniform node-link model*: $p_a = p_b$.
(ii) The *node model*: $p_a = p_b = 0$.
(iii) The *link model*: $p = 0$.
(iv) The *uniform link model*: $p = 0, p_a = p_b$.

The node model and the uniform link model are the most-studied in the literature.

We define *reliability* $R(N; a, b)$ as the probability that all working nodes are strongly connected. Computing the exact reliability of a double loop is a difficult problem, even for the node model or the uniform link model. We summarize all known results in the following (see [20, 21]):

(i) $DL(N; 1, N-1)$ under the general model.
(ii) $DL(N; 1, 2)$ under the general model.
(iii) $DL(N; 1, N-2)$ under the general model.
(iv) $DL(N; 1, 1+N/2)$ for N even under the general model.
(v) $DL(N; 2, 3)$ for N odd under the uniform node-link model with $p_a = 0$.

Hwang and Wright [20] gave a general approach to compute exact reliability under the general model. However, there is still a part of the computation which is specific to each individual double loop.

An *element* is either a node or a link. The state of an element is either *working* or *failed*. A part is simply a subgraph of $DL(N;a,b)$. The state of a part is the set of states of its elements.

Consider a state S of $DL(N;a,b)$. Let $G(S)$ denote the digraph obtained from $DL(N;a,b)$ by deleting all failed nodes and links, as well as links to and from failed nodes. A node is called an *island* if it has neither inlink nor outlink.

Lemma 7.1. *S is a working state for the double loop if and only if $G(S)$ contains neither an island nor two disconnected circuits.*

Proof. Suppose to the contrary that there exist two nodes u and v such that u cannot reach v. Let U be the set of nodes u can reach, and V the set of nodes which can reach v. Then U and V are disconnected. Since every node has an inlink and an outlink, both U and V contain a circuit. Then these two circuits are disconnected. □

Let $\boldsymbol{S}_1$ denote the set of states S containing no island and $\boldsymbol{S}_2$ the set containing no two disconnected circuits. Define

$$P_1(N;a,b) = \sum_{S\in \boldsymbol{S}_1} \text{Prob}(S),$$

$$P_2(N;a,b) = \sum_{S\in \boldsymbol{S}_2} \text{Prob}(S).$$

Then

$$R(N;a,b) = P_1(N;a,b) - P_2(N;a,b).$$

P_1 can be computed by a divide-and-conquer method.

Let $\Pi_1,\dots,\Pi_l$, denote a partition of $DL(N;a,b)$ into l parts such that the outlinks of Π_j go to either Π_j or Π_{j+1} for $1\leqslant j\leqslant l$ ($\Pi_{l+1}\equiv\Pi_1$). Let $S_j=\{S_{ji}\}$ denote the set of states of Π_j containing no island. If $\Pi_j\cup\Pi_{j+1}$ contains no island under the joint state $(S_{ji},S_{(j+1)i'})$ for all i and i', then $G(S)$ contains no island where $S=(S_1,\dots,S_l)$. Define a $|S_j|\times|S_{j+1}|$ matrix M_j be setting the element in row S_{ji} and column $S_{(j+1)i'}$

$$m_j(S_{ji},S_{(j+1)i'}) = \begin{cases} \text{Prob}(S_{ji}) & \text{if } (S_{ji},S_{(j+1)i'}) \text{ contains no island,} \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$\begin{aligned} P_1(N;a,b) &= \sum_{S_{1i}\in S_1}\dots\sum_{S_{li}\in S_l} m_1(S_{1i},S_{2i})m_2(S_{2i},S_{3i})\dots m_l(S_{li},S_{1i}) \\ &= Trace\left(\prod_{i=1}^{l} M_i\right). \end{aligned}$$

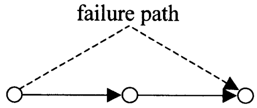


Fig. 10. A self-healing node.

If a and b are of order O(1), then Π_j can be chosen such that $|S_j| = \mathrm{O}(1)$ (and $l = \mathrm{O}(N)$). Hence each M_j can be computed in O(1) time, and $P_1(N; a, b)$ in O(N) time. If $M_j = M$, then $P_1(N; a, b)$ can be computed in O(log N) time. Note that the complexity does not change if a constant number of Π_j are nonisomorphic to or have different failure probabilities than the rest.

Therefore, the problem of computing the reliability for a double-loop is reduced to finding a method to compute $P_2(N; a, b)$.

Reliability can of course be measured in other ways. Peha and Tobagi [25] considered the expected number of nodes reachable from a working node, and gave lower and upper bounds. Dao and Silio [8] introduced *circular connectivity*, u and v are circularly connected if there exists a circuit containing u and v, which is the relevant measure in SONET. A fault-tolerant node has the capacity of self-avoiding in a path when being faulty (see Fig. 10). They gave the probability that all working nodes are circularly connected for $DL(N; 1, N-1)$ with fault-tolerant nodes.

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. Para. Dist. Comput. 24 (1995) 2–10.
[3] C. Chen, F.K. Hwang, The minimum distance diagram of double-loop networks, preprint 1997, IEEE Trans. Comput., to appear.
[4] C. Chen, F.K. Hwang, The spectrum of equivalent nondegenerate L-shapes of double-loop networks, Networks, to appear.
[5] C. Chen, F.K. Hwang, Equivalent degenerate L-shapes of double-loop networks, preprint, 1999, COIN, to appear.
[6] Y. Cheng, F.K. Hwang, Diameters of weighted double loop networks, J. Algebra 9 (1988) 401–410.
[7] Y. Cheng, F.K. Hwang, I.F. Akyildiz, D.F. Hsu, Routing algorithms for double loop networks, Int. J. Foundations Comput. Sci. 3 (1992) 323–331.
[8] H.M. Dao, C.B. Silio, Jr., Ring-network reliability with a constrained number of consecutively-bypassed stations, IEEE Trans. Rel., to appear.
[9] P. Erdös, D.F. Hsu, Distributed loop network with minimum transmission delay, Theoret. Comput. Sci. 100 (1992) 223–241.
[10] M. Escudero, J. Fiàbrea, P. Morillo, Fault-tolerant routing in double-loop networks, Ars Combin. 25A (1998) 187–198.
[11] P. Esqué, F. Aguiló, M.A. Fiol, Double commutative-step digraphs with minimum diameters, Discrete Math. 114 (1993) 147–157.
[12] M.A. Fiol, J.L.A. Yebra, Ciclos de Hamilton en redes de pasos commutativos y de paso fijo, Stochastica XII-2 (1988) 113–129.

[13] M.A. Fiol, J.L.A. Yebra, I. Alegre, M. Valero, A discrete optimization problem in local networks and data alignment, IEEE Trans. Comput. 36 (1987) 702–713.
[14] A. Grnarov, L. Kleinrock, M. Gerla, A highly reliable distributed loop network architecture, Proc. Int. Symp. Fault-Tolerant Comput., Kyoto, Japan, 1980, pp. 319–324.
[15] D.J. Guan, An optimal message routing algorithm for double-loop networks, Inform. Process Lett. 65 (1998) 255–260.
[16] S.H. Huang, F.K. Hwang, Y.H. Liu, Equivalent double loop networks, preprint, 1998.
[17] 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.
[18] F.K. Hwang, The 1.5-loop network, preprint, 1998.
[19] F.K. Hwang, T.-S. Lin, R.-H. Jan, A permutation routing algorithm for double loop networks, Para. Proc. Lett. 7 (1997) 259–265.
[20] F.K. Hwang, P.E. Wright, Survival reliability of some double-loop networks and chordal rings, IEEE Trans. Comput. 44 (1995) 1468–1471.
[21] F.K. Hwang, P.E. Wright, X.D. Hu, Exact reliabilities of most reliable double-loop networks, Networks 30 (1997) 81–90.
[22] F.K. Hwang, Y.H. Xu, Double loop networks with minimum delay, Discrete Math. 66 (1987) 109–118.
[23] C.-Y. Lin, Fault tolerant token ring embedding in double loop networks, Master Thesis, Department of Comput. Inform. Sci., National Chiao Tung Univ., 1997.
[24] M.T. Liu, Distributed loop computer networks, Adv. Comput. 17 Academic, New York (1981) 163–221.
[25] J.M. Peha, F.A. Tobagi, Analyzing the fault tolerance of double-loop networks, IEEE Trans. Networking 2 (1994) 363–373.
[26] C.S. Raghavendra, M. Gerla, A. Avizienis, Reliable loop topologies for large local computer networks, IEEE Trans. Comput. 34 (1985) 46–55.
[27] C.S. Raghavendra, J.A. Sylvester, A survey of multi-connected loop topologies for local computer networks, Comput. Netw. ISDN Syst. 11 (1986) 29–42.
[28] O.J. Rödseth, Weighted multi-connected loop networks, Discrete Math. 148 (1996) 161–173.
[29] C.K. Wong, D. Coppersmith, A combinatorial problem related to multimodule memory organizations, J. ACM 21 (1974) 392–402.