國 立 交 通 大 學

應 用 數 學 系

博 士 論 文

混合式弦環網路之距離相關問題

On Distance-related Problems of

Mixed Chordal Ring Networks

研 究 生：藍國元

指導教授：陳秋媛 博士

中 華 民 國 九 十 九 年 六 月

混合式弦環網路之距離相關問題

On Distance-related Problems of Mixed Chordal Ring Networks

研 究 生：藍國元　　Student: James K. Lan
指導教授：陳秋媛 博士　Advisor: Chiuyuan Chen

國 立 交 通 大 學
應 用 數 學 系
博 士 論 文

A Dissertation
Submitted to Department of Applied Mathematics
College of Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in
Applied Mathematics
June 2010
Hsinchu, Taiwan, Republic of China

中 華 民 國 九 十 九 年 六 月

Abstract

This research covers the less trodden field of mixed chordal ring networks, in the attempt to discover the existence of efficient algorithms on distance-related problems, including the minimum distance diagram construction, the diameter computation, and the node-to-node shortest path routing. The extensively studied double-loop network has proven to hold efficient algorithms on the above specified distance-related problems. The significance of this research lies in mixed chordal ring network's achievement of a better diameter, as well as the in-vertex-transitive feature of it, which makes its exploration on distance-related problems a lot more sophisticated.

We first study and investigate the minimum distance diagram problem. We find that the minimum distance diagram of a mixed chordal ring network can be obtained by reassembling the PSEUDOMDD. This observation can be used to study other distance-related problems. For the diameter computation problem, we proposed an efficient algorithm to compute the diameter of a given mixed chordal ring network. For the optimization problem of finding optimal networks, we improve previous lower and upper bounds and successfully obtain a class of optimal mixed chordal ring networks. For the routing problem, two node-to-node routing algorithms are presented for flexible applications: the shortest-path-based routing algorithm and the dynamic routing algorithm. In addition, we also present an optimal fault-tolerant routing algorithm for mixed chordal ring networks in the presence of up to one node or link failure. All the routing algorithms presented do not require routing tables and only very little computational overhead is needed.

Keywords: Mixed chordal ring network; Double-loop network; Algorithm; Diameter; Optimal routing; Fault-tolerant routing; Minimum distance diagram; Interconnection network; Parallel processing.

中文摘要

本研究涵蓋混合式弦環網路中較少被討論的部份，並試圖發掘混合式弦環網路在與距離相關的問題中，是否存在有效率的演算法。這些問題包括最短距離圖的建造、直徑的計算、以及點與點之間的最優路由連線設計。上述與距離相關之問題在已被廣泛研究的雙環式網絡中，已經找得到有效率的演算法。本研究的重要性在於混合式弦環網路的直徑比雙環式網絡來得小，以及，混合式弦環網路沒有點對稱性質，因此使得上述與距離相關之問題複雜很多。

我們首先研究混合式弦環網路的最短距離圖建造問題。我們發現混合式弦環網路的最短距離圖可經由重新組合「虛擬距離圖」得到。這樣的觀察可以讓我們研究其它與距離相關之問題。針對直徑計算問題，我們提出一個有效率的演算法可以計算出任一給定的混合式弦環網路的直徑。關於找出混合式弦網環路中最小直徑的最佳化問題，我們改進了前人針對此問題所提出的上下限，並且成功地得到一個無限最優混合式弦環網路族。針對網路路由連線設計問題，我們提出兩個可彈性應用的點對點最優路由連線設計演算法：基於最短路徑路由演算法及動態路由演算法。此外，我們也提出了一個最優容錯路由演算法。此演算法在網路壞掉一個點或一個邊時可以執行正確。上述所有路由演算法都不需要路由表格，並且只需要非常小的額外計算花費。

關鍵字：混合式弦環網路；雙環式網路；演算法；直徑；最優路由；容錯路由；最短距離圖；連接網絡；平行處理。

Acknowledgments

首先，我要以最誠摯的心意感謝我的指導老師：陳秋媛老師。當初會想攻讀博士有很大一部分因素是因為老師的關係。從碩士到博士跟老師這麼多年，深深地被老師對學生的熱心給感動到！任何疑難雜症都可以找老師商量、指點。而且不論對象是誰，她都很樂意幫助學生，實在是一位難能可貴的好老師。與老師 meeting 時，學習到老師對相關論文的看法及觀點；有論文上的想法或問題請益老師時，她都會非常用心的審視我的問題。此外當投稿不順利時，老師會鼓勵我、支持我，並幫助我渡過低潮。當與學弟妹有新的研究發現時，老師會很積極讓我們能團結一致，全力寫出結果。不僅如此，老師鼓勵我參加國際型研討會去發表論文，讓我增廣見聞，也學習別人是怎麼做研究的。有任何機會，老師都會主動詢問我。我只能說，能夠遇到這樣的老師，我真是幸運！

另外也要感謝在攻讀博士期間所認識朋友。我們家族的學弟妹：柏澍、威雄、鈺傑、志文、子鴻、信菖、松育、宜君、士慶、慧蓁、思賢、思綸、健峰、詩妤、恭毅，由於你們在每週 Group Meeting 的精彩表現，讓我獲益匪淺。還有學長姊學弟妹：君逸、宏賓、文祥、惠蘭、飛黃、業忠。有你們的參與下，讓我的生活多采多姿！！在此也要感謝攻讀博士期間打工家教的家長：陸媽媽及黃媽媽。由於家裡經濟的關係，平時需要在外面打工。而陸媽媽及黃媽媽對我非常地好，讓我比較沒有經濟上的壓力。

此外，很幸運地在這段期間也遇到我人生中的另一半。人的一生不可能一輩子都只靠一個人活下去，感謝她的出現，讓我在人生的重要階段有了支持及陪伴，讓我的人生充滿彩色！！就讀外文系的她當然是我寫作請益的最佳對象。不僅如此，當有任何想法時，她能不厭其煩的聽我述說；遇到不順利的時候，能夠傾聽我、幫助我，與我一起分擔！！而且理工的我以及人文的她剛好形成最大的互補，讓我攻讀博士期間能夠有更美好的生活及無窮的支持！！我只能說，謝謝你，baby ～～

最後當然最重要的還是感謝我的父母，沒有他們就沒有我。從小呵護我、栽培我唸書，始終在背後支持我。平常回家會一直進補我，讓我有健狀的身心。你們是我任何成就的最大推手！感謝的心，不止於此，僅以微薄紙筆，代表我心！

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Interconnection Networks

In recent years, interconnection networks are applicable in many different fields, ranging from internal buses in very large-scale integration (VLSI) circuits to wide area computer networks. Among others, these applications include parallel computing, backplane buses and system area networks, telephone switches, internal networks for asynchronous transfer mode (ATM) and Internet Protocol (IP) switches, processor/memory interconnects for vector supercomputers, interconnection networks for multi-computers and distributed shared-memory multiprocessors, clusters of workstations and personal computers, local area networks, metropolitan area networks, wide area computer networks, and networks for industrial applications [26, 31, 37, 61].

To implement high performance parallel and distributed systems by designing interconnection architectures is a task both significant and challenging. [55, 56]. The choice of the interconnection network may affect several characteristics of the final system, including implementation cost (node complexity, VLSI area, wiring density), performance, ease of programming, reliability, and scalability. Throughout times, many different interconnection networks had been applied in commercially available concurrent systems and numerous research prototypes [46, 54]; other alternatives are proposed and evaluated in theoretical studies [56].

Interconnection networks have been traditionally classified according to the operating mode (synchronous or asynchronous) and network control (centralized, decentralized or distributed) [31]. According to [31], there are *four* major classes based primarily on network topology: *shared-medium networks*, *direct networks* (router-based networks), *indirect networks* (switch-based Networks) and *hybrid networks*. In this research, our target networks, *double-loop networks* and *mixed chordal ring networks*, belong to direct networks.

The *direct network* or *point-to-point network* is a popular interconnection network architecture that scales well to a large number of processors [31]. A direct network consists of a set of *nodes*, each node being directly connected to a subset of other nodes in the network. Theses nodes may have different functional capabilities. One common component of theses nodes is a *router*, which handles message communication among nodes. Direct networks have been a popular interconnection architecture for constructing large-scale parallel computers.

Almost all direct network topologies studied in the literature have some degree of symmetry. Such a symmetric topology has many advantages: First, it allows the network to be constructed from simple building blocks and expanded in a modular fashion. Second, the regular topology facilitates the use of simple routing algorithms. Third, it is easier to develop efficient computational algorithms for multiprocessors interconnected by a symmetric network. Finally, it makes the network easier to model and analyze. For example, in a *ring network* of N nodes labeled from 0 to $N-1$, each processor i is directly connected to processors $(i-1) \bmod N$ and $(i+1) \bmod N$.

Mathematical models for interconnection networks have played important roles in understanding, synthesizing, and comparing a multitude of network architectures. The architecture of an interconnection network can be represented by a graph or a digraph, where vertices

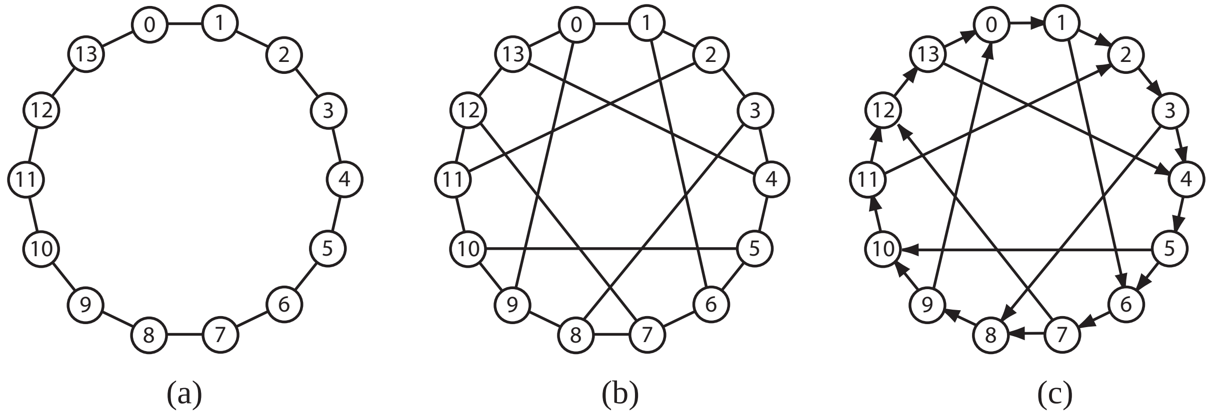


Figure 1.1: Examples of direct network topologies: (a) (Undirected) Ring network (b) Chordal ring network (c) Directed chordal ring network.

represent processors/nodes and edges represent links/channels between processors/nodes. Fig. 1.1 shows some direct network topologies.

1.2 Evaluation Criteria for Networks

The topology of a direct network determines many architecture features of the network and affects several performance metrics. Although the actual performance of a network depends on many technology and implementations factors, several topological properties and metrics can be used to evaluate and compare different topologies in a technology-independent manner. Most of these properties are derived from the graph model of the network topology.

- **Symmetry and Regularity**

  A *regular* network is defined as a network in which each node connects to the same number of other nodes. A *symmetric* network is a network in which the topology looks identical when viewed from every node or every edge. There are two types of symmetric: Node symmetric and edge symmetric. In graph-theoretic terms, a graph is *node-symmetric* (*vertex-transitive*) if, for every pair of vertices u and v, there is an automorphism which maps u to v. The definition of edge-symmetric is identical to the node-symmetric, except that the automorphism maps edges among themselves. References to symmetry without qualification usually imply node-symmetry.

The main advantage of symmetric in a network lies in the ease of routing data in the network. This allows all nodes to use the same routing algorithm. The task of path-selection is also often simplified. Many popular direct interconnection networks are regular and symmetric. Clearly, all networks in Fig. 1.1 are regular. In addition, networks in Figs. 1.1(a) and 1.1(b) are also symmetric.

- **Connectivity**

  The primary factor relating directly to the robustness of a graph-modeled interconnection structure is its *connectivity* or *edge connectivity*. From the graph theory viewpoint, the *connectivity* (resp., *edge connectivity*) of an undirected graph is the minimum number of vertices (resp., edges) whose removal causes the graph to be disconnected or to contain only one vertex. A digraph is *strongly connected* if for each *ordered pair* u, v of vertices, there is a path from u to v. In a directed graph, the *connectivity* (resp., *edge connectivity*) is defined as the minimum number of vertices (resp., edges) whose removal causes the graph to be non-strongly connected. For some symmetric networks, the connectivity is usually the same as the degree of a node.

- **Distance Measures**

  In a direct network, communication between two nodes that are not directly connected must take place through other nodes. The *network diameter* (diameter for short) D, defined as the longest of the internode distances, is an important figure of merit for networks. The diameter D indicates the worst-case number of hops in sending a message from one node to another. If the message delay is proportional to the number of links traversed, this provides an *upper bound* on the delay in the absence of any interfering traffic. The diameter D may also be viewed as a *lower bound* on the delay between two nodes that are located farthest from each other. Although diameter does not completely characterizes the performance of an interconnection network, it is still useful in comparing networks with respect to their power to perform certain operations.

Although the diameter is useful in comparing two interconnection networks with identical node degrees, it may not always be indicative of the actual performance of the networks. Since two nodes in a network do not always communicate with each other by traversing the length of the diameter D, it is more important to measure the average distance traveled by a message in practice. *Average internode distance* $\overline{D}$ is defined as the average lengths of the distance between all N^2 pairs of nodes. The average distance is representative of average or expected communication latencies, whereas D represents the worst case.

- **Efficient Routing**

As interconnection networks differ in the way they accommodate message traffic, routing performance is a primary indicator of the overall benefits of a particular topology. Efficient message routing can improve the network utilization. Many parameters including the length of the route, the computational overhead, the memory requirement at each node and the extra overhead information included in the message, can affect the routing performance. The first issue in the algorithmic aspect is to design efficient algorithms such that every message is sent along a shortest path from its source node to its destination node. Thus one of the most important features to be taken into account in the design of an interconnection network is the existence of efficient algorithms for routing messages.

When some nodes or links in the network fail, some routes become unavailable. However, assuming that the network remains connected, communication is still possible by sending affected message along a sequence of surviving routes. Therefore, the design of algorithms for sending messages along the shortest route after detecting the faulty element is also an important issue.

1.3 Distributed Loop Networks

Loop networks have been widely considered in recent years as good network models for interconnection or communication networks due to their regularity, simple structure and symmetry; see Bermond et al. [9] for an exhaustive survey on this topic. The *ring network* (i.e., the *single-loop network*) is one of the most simple and frequently used loop network for interconnection networks, and has many attractive properties such as simplicity, extendibility, low degree, and ease of implementation. Although the ring network has many attractive properties, it has poor reliability (any failure in an interface or communication link destroys the function of the network) and it has high transmission delay. As a result, a lot of hybrid topologies utilizing the ring network as a basis for synthesizing richer interconnection schemes have been proposed to improve the reliability and reduce the transmission delay [6, 20, 27, 64].

One example of the commonly used extensions for the ring network is the *multi-loop network* $ML(N; s_1, s_2, \ldots, s_\ell)$, which was first proposed by Wong and Coppersmith in [64] for organizing multi-module memory services. The most widely studied multi-loop network is perhaps the *double-loop network* (DLN for short). A DLN $DL(N; s_1, s_2)$ can be modeled by using a digraph with N nodes $0, 1, \ldots, N-1$ and $2N$ links as follows

$$i \to (i + s_1) \bmod N, \quad i = 0, 1, 2, \ldots, N-1,$$
$$i \to (i + s_2) \bmod N, \quad i = 0, 1, 2, \ldots, N-1,$$

where $0 < s_1 \neq s_2 < N$. The double-loop network has been used for local area network [47] as well as the large local area optical network as SONET [7].

Another example of the commonly used extensions for the ring network is the *chordal ring network*, which is constructed by adding *chords* to the ring topology [6, 41]. Arden and Lee [6] first proposed and studied the chordal ring network. More specifically, an (undirected) chordal ring network $CR(N;w)$, where N is even and w is odd, can be modeled by using a graph with N nodes $0,1,\ldots,N-1$ and $3N/2$ links:

$$(i,(i+1) \bmod N), \quad i=0,1,2,\ldots,N-1,$$
$$(i,(i+w) \bmod N), \quad i=1,3,5,\ldots,N-1.$$

See Fig. 1.1(b) for an example of $CR(14;5)$. Since then, more than one hundred papers have been published on the topic of the chordal ring network and its variants. Especially, the chordal ring networks of degree 3, 4, and 6 have been widely discussed in the literature [8, 11, 24, 50, 65]. As was pointed out in [20], the chordal ring network is a 3-regular graph and it offers a happy medium between the (undirected) ring network and the undirected double-loop network in the amount of hardware. Also, it preserves the Hamiltonian cycle from the ring network and has a better diameter than the undirected ring network.

In [41], Hwang and Wright considered the directed version of the chordal ring network and made a slight generalization on the ring links. More specifically, a *directed chordal ring network* $DCR(N;s,w)$, where N is even and both s and w are odd, can be modeled by using a digraph with N nodes $0,1,\ldots,N-1$ and $3N/2$ links:

$$i \to (i+s) \bmod N, \quad i=0,1,2,\ldots,N-1,$$
$$i \to (i+w) \bmod N, \quad i=1,3,5,\ldots,N-1.$$

For an example, Fig. 1.1(c) is $DCR(14;1,5)$.

Recently, Chen et al. [20] introduced the *mixed chordal ring network* (MCRN for short) as a topology of interconnection networks. An MCRN $MCR(N;s,w)$ can be modeled by

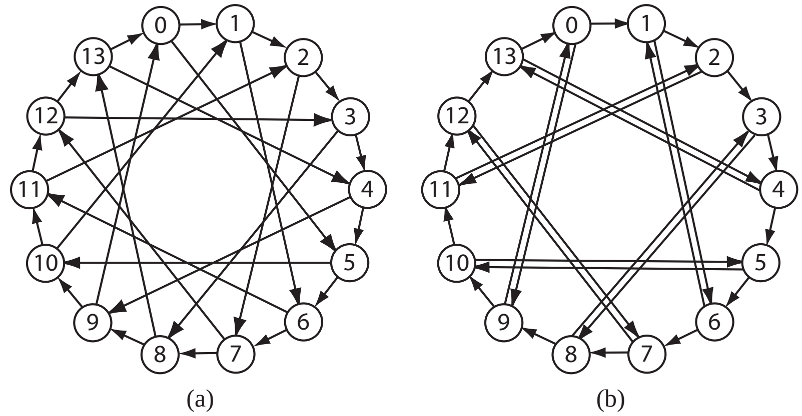


Figure 1.2: A DLN and an MCRN.

using a digraph with N nodes $0, 1, \ldots, N-1$ and $2N$ links of the following types

$$\begin{array}{lll} \text{ring-links:} & i \to (i+s) \bmod N, & i=0,1,2,\ldots,N-1, \\ \text{chordal-links:} & i \to (i-w) \bmod N, & i=0,2,4,\ldots,N-2, \\ \text{chordal-links:} & i \to (i+w) \bmod N, & i=1,3,5,\ldots,N-1, \end{array}$$

where N is even, both s and w are odd. Figs. 1.2(a) and 1.2(b) illustrate $DL(14; 1, 5)$ and $MCR(14; 1, 5)$, respectively.

1.4 Motivation

Since each node in the DLN or MCRN has two in-links and two out-links, the DLN and MCRN are very comparable[1]. Throughout this thesis, N denotes the number of nodes in a communication network. For a fixed N, let $D_{DL}(N)$ and $D_{MCR}(N)$ denote the *optimal* (i.e., smallest) diameter of all DLNs and all MCRNs with N nodes, respectively. A well-known

[1]When comparing the mixed chordal ring network with the double-loop network, we assume both networks have the same number of nodes.

lower bound on $D_{DL}(N)$ is as follows [64]:

$$D_{DL}(N) \geq \lceil \sqrt{3N}\, \rceil - 2. \tag{1.4.1}$$

For upper bounds on $D_{DL}(N)$, Hwang and Xu [42] managed to prove, using a heuristic method, that

$$D_{DL}(N) \leq \sqrt{3N} + 2(3N)^{1/4} + 5 \quad \text{for} \quad N \geq 6348. \tag{1.4.2}$$

In [57], Rödeseth further improved the above upper bound to be

$$D_{DL}(N) \leq \sqrt{3N} + (3N)^{1/4} + \tfrac{5}{2} \quad \text{for } N \geq 1200. \tag{1.4.3}$$

For MCRNs, Chen et al. [20] showed the following result:

Theorem 1.4.1. [20] *There exists a choice of s and w such that the diameter of $MCR(N; s, w)$ is no larger than $\sqrt{2N} + 3$. In other words, $D_{MCR}(N) \leq \sqrt{2N} + 3$.*

Since $\sqrt{2N} + 3$ is severed as an upper bound, we have

$$D_{MCR}(N) \leq \lfloor \sqrt{2N}\, \rfloor + 3. \tag{1.4.4}$$

Note that there exist some erroneous cases in the proof of Theorem 1.4.1 and thus it is not known whether or not MCRNs can achieve a better diameter than DLNs. In spite of the erratum in the proof of Theorem 1.4.1, we confirm that MCRNs can achieve a better diameter than DLNs by giving an improved upper bound on $D_{MCR}(N)$ in Section 5.2 as

$$D_{MCR}(N) \leq 2 \lceil \sqrt{N/2}\, \rceil + 1. \tag{1.4.5}$$

From equations (1.4.1), (1.4.4) and (1.4.5), we can conclude definitely that the MCRN can achieve a better diameter than the DLN.

One of the most important and fundamental optimization problem in designing interconnection networks is, for a given number of nodes N, how to find an *optimal network* with the smallest diameter and to give the construction of such a network. More precisely, for double-loop networks, $DL(N; s_1, s_2)$ is *optimal* if the diameter of $DL(N, s_1, s_2)$ is equal to $D_{DL}(N)$. This optimization problem for the double-loop network has been widely studied in the literature [2, 9, 10, 14, 15, 16, 30, 32, 42, 59]. However, to the best of our knowledge, there is no result about the exact value of $D_{MCR}(N)$ in the literature.

Message routing is a fundamental and important function in interconnection networks. Efficient message routing not only can reduce the transmission delay but also can improve the network utilization. A routing algorithm is said to be *optimal* if every message is sent along a shortest path from its source node to its destination node. There has been a numerous amount of work on message routing in DLNs [22, 23, 35, 36, 40, 49]. In particular, it has been studied with respect to network applications such as message routing [35, 36, 49], permutation routing [40] and fault-tolerant routing [23, 49].

The *minimum distance diagram* (MDD for short), also called *optimal routing region* in [27], is a tool to encode distance-related information such as diameter and shortest route for multi-loop networks. It is well-known that the MDD of a DLN always forms an *L-shape* and one can compute the diameter and the average distance of a DLN from the lengths of segments on the boundary of an L-shape in constant time [33]. Cheng and Hwang [21] proposed an $O(\log N)$-time algorithm to derive the lengths of segments on the boundary of the L-shape of $DL(N; s_1, s_2)$. Furthermore, many researchers addressed designing efficient routing algorithms or fault-tolerant routing by using the L-shapes [22, 23, 36, 49]. For further results of the DLN; see the excellent survey papers [9, 38, 39].

In contrast to the DLN, there has been little work reported in the literature on distance-related problems of MCRNs. To the best of our knowledge, neither the diameter-computating strategy nor the message-routing strategy was found in the literature. A natural question arises, namely, whether the diameter computation and the message routing in MCRNs can be done efficiently as in DLNs. Table 1.1 shows a comparison of previous results between

DLNs and MCRNs.

Table 1.1: Previous results on double-loop networks and mixed chordal ring networks.

	DLN	MCRN
MDD construction	[64]	?
Diameter computation	[21, 66]	?
Optimal networks	[10, 14, 15, 59]	?
Node-to-node routing	[22, 35, 36]	?
Fault-tolerant routing	[23, 36, 49]	?

1.5 Summary of the Contribution of This Research

In this section, we present a summary of the specific problems analyzed and the results derived in this thesis. The contribution of our research will be introduced in Chapters 3-7.

In Chapter 3, we consider the problem of exploring and constructing the MDD of a MCRN $MCR(N; s, w)$. Specifically, we introduce the PSEUDOMDD that helps study the distance-related problems in MCRNs. By mapping the nodes of a MCRN to the two-dimensional integer lattice, one can study the distance properties between the nodes of a MCRN. Due to the tessellation of the plane formed by PSEUDOMDD, we successfully obtain the MDD of a given MCRN from the PSEUDOMDD in a simple manner. In the last section of this chapter, we give an algorithm to construct the MDD of a MCRN. The visualization tool established in this chapter will be used throughout this thesis.

In Chapter 4, we consider the problem of computing the diameter of an MCRN. Instead of constructing the MDD of an MCRN first, we present a subroutine that can compute the maximum of distances of the nodes in the MDD to the node at the origin in constant time as long as we have the L-shape of the PSEUDOMDD. As an application, we obtain an algorithm that can compute the diameter of a given MCRN in $O(\log N)$ worst-case time in Section 4.2.

In Chapter 5, we discuss the problem of finding optimal MCRNs. In other words, we are interested in finding MCRNs which achieve the smallest diameter among all MCRNs with the same number of nodes. Due to the difficulty of this optimization problem, we aim at looking for bounds on $D_{MCR}(N)$ instead of finding optimal MCRNs directly. In Section 5.3, we successfully obtains a class of optimal MCRNs which matches the upper and lower bounds presented in Sections 5.1 and 5.2.

In Chapter 6, we consider the problem of routing in MCRNs. In particular, routing of node-to-node message with at most one faulty element in MCRNs is considered. We design and present two optimal node-to-node routing algorithms and an optimal fault-tolerant routing algorithm for MCRNs.

The two optimal node-to-node routing algorithms presented are *shortest-path-based routing* and *dynamic routing*. The shortest-path-based routing algorithm computes the *routing parameter* that can be used to determine a routing path. This algorithm takes $O(\log N)$-time for a source node to compute the routing parameter, and each node on the routing path can take constant time to determine the link (and therefore the node) to send messages according to the routing parameter. On the other hand, for the dynamic routing algorithm, after an $O(\log N)$-time precalculation to determine the network parameters (only computed once and stored them in all nodes), it can route messages using constant time at each node along the routing path. The routing path is augmented on-the-fly at each routing step. A shortest-path-based routing algorithm is presented in Section 6.1. A dynamic routing algorithm is presented in Section 6.2.

In Section 6.3, we present an optimal fault-tolerant routing algorithm for MCRNs. The algorithm does not require routing tables; it is efficient and it requires very little computational overhead. After an $O(\log N)$-time precalculation, the algorithm can route messages to the destination using a constant time at each node along the route. Moreover, the fault-tolerant algorithm presented is guaranteed to find the optimal route at the presence of up to one node or link failure.

Here we summarize the contribution of this thesis: it proposes

(a) an algorithm to construct the MDDs of a mixed chordal ring network,

(b) an efficient algorithm to compute the diameter of a mixed chordal ring network,

(c) improved upper and lower bounds on $D_{MCR}(N)$,

(d) two optimal node-to-node routing algorithms for mixed chordal ring networks,

(e) an optimal fault-tolerant routing algorithm for mixed chordal ring networks.

Chapter 2

Background Material

In this chapter, we present some background material on the double-loop network and the mixed chordal ring network, as well as some previous results related to both networks. In addition, some fundamental concepts of graph theory are given first. Our terminologies and notations of graph theory are standard; see [63] and also [13].

2.1 Fundamental Concepts of Graph Theory

A graph G with n vertices and m edges consists of the *vertex set* $V(G) = \{ v_1, v_2, \ldots, v_n \}$ and *edge set* $E(G) = \{ e_1, e_2, \ldots, e_m \}$, where each edge consists of two (possibly equal) vertices called, *endpoints*. An element in $V(G)$ is called a *vertex* of G. An element in $E(G)$ is called an *edge* of G. When vertices u and v are the endpoints of an edge e, they are *adjacent* and are *neighbors*. We write (u, v) when $\{ u, v \} \in E(G)$. A *loop* is an edge whose endpoints are equal. *Multiple edges* are edges having the same pair of endpoints. A *simple graph* is a graph having no loops or multiple edges.

A *directed graph* or *digraph* G consists of a vertex set $V(G)$ and an edge set (or *arc set*) $E(G)$, where each edge is an ordered pair of vertices. The first vertex of the ordered pair is the *tail* of the edge, and the second is the *head*; together, they are the endpoints. We say that an edge is an edge from its tail to its head. We write $u \to v$ when there is an edge from u to v. In a digraph, a *loop* is an edge whose endpoints are equal. *Multiple edges* are edges having the same ordered pair of endpoints. A digraph is *simple* if each ordered pair is the head and tail of at most one edge. For a vertex v of a digraph G, the *outdegree* $d^+(v)$ the number of edges with tail v. The *indegree* $d^-(v)$ is the number of edges with head v.

Unless otherwise specified, the following definitions and terms hold for both graphs and digraphs. A *separating set* or *vertex cut* of a graph G is a set $S \subseteq V(G)$ such that $G \setminus S$ has more than one component. A graph is k-connected if every separating set has at least k vertices. A digraph G is *strongly connected* or *strong* if there is a path from u to v in G for every ordered pair $u, v \in V(G)$. A digraph G is *strongly k-connected* if $|V(G)| \geq k+1$ and every separating set of G has at least k vertices.

An *isomorphism* from a simple graph G to a simple graph H is a bijection $f : V(G) \to V(H)$ such that $\{ u, v \} \in E(G)$ if and only if $\{ f(u), f(v) \} \in E(H)$. An *automorphism* of G is an isomorphism from G into G. A graph G is *vertex-transitive* if for every pair $u, v \in V(G)$, there is an automorphism that maps u to v.

2.2 The Double-loop Network

A *double-loop network* (DLN for short) $DL(N; s_1, s_2)$ can be modeled by using a digraph with N nodes $0, 1, \ldots, N-1$ and $2N$ links

$$\begin{aligned} &i \to (i + s_1) \bmod N, \quad i = 0, 1, 2, \ldots, N-1, \\ &i \to (i + s_2) \bmod N, \quad i = 0, 1, 2, \ldots, N-1, \end{aligned}$$

where $0 < s_1 \neq s_2 < N$. The integers s_1, s_2 are called *steps* or *hops* or *jumps*. The connectivity of the DLN has been determined by Doorn [60] (or see [38]):

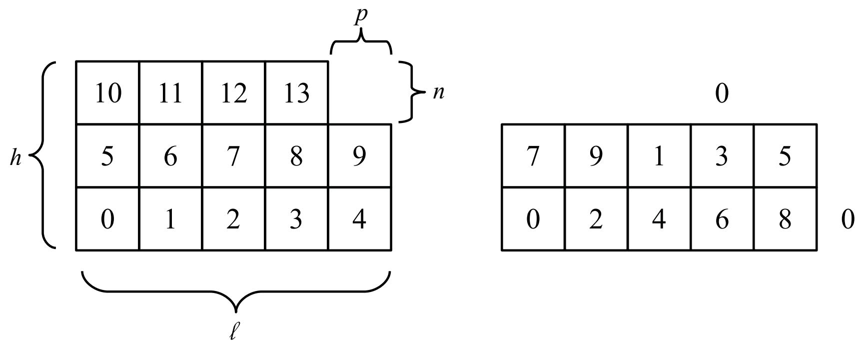
Theorem 2.2.1. [38] *$DL(N; s_1, s_2)$ is strongly 2-connected if and only if* $\gcd(N, s_1, s_2) = 1$.

It is well-known [34] that $DL(N; s_1, s_2)$ is a Cayley digraph of the cyclic group $\mathbb{Z}_N$ with the set of generators $\{s_1, s_2\}$. Since Cayley digraphs are vertex-transitive, the distance-related problems of DLNs can be reduced to the problem of studying paths originated at a fixed vertex[1], usually node 0. A visualization tool that allows studying distance-related problems of DLNs from a geometric point of view is set up as follows: Consider the *two-dimensional integer lattice* $\mathbb{Z} \times \mathbb{Z}$. Given $DL(N; s_1, s_2)$, label each lattice point (x, y) (i.e., x and y being integers) of $\mathbb{Z} \times \mathbb{Z}$ by $(xs_1 + ys_2) \bmod N$. Unless otherwise specified, we refer to a *point* as a lattice point.

A *minimum distance diagram* (MDD) of $DL(N; s_1, s_2)$ is an array with node 0 at point $(0, 0)$ and node u at point (x, y) if and only if $xs_1 + ys_2 \equiv u \pmod N$ and $x + y$ is the minimum among all (x', y') satisfying the congruence. Namely, a shortest path from node 0 to node u is through taking x s_1-steps and y s_2-steps (in any order). Note that an MDD includes every node exactly once. Most authors [2, 12, 18, 19, 21, 33, 38] always "break ties" lexicographically (choose with smaller y) whenever there are two (x, y)'s satisfying $xs_1 + ys_2 \equiv u \pmod N$. Without this convention, Sabariego and Santos [58] showed that every DLN has at most two MDD's. Throughout this thesis, we follow the convention used in the literature, i.e., we assume a DLN has only one MDD constructed by using the convention. Fig. 2.1(a) illustrates the MDD of $DL(14; 1, 5)$.

It is well-known [64] that the MDD of a DLN is of a definite form: an *L-shape*. The L-shape is determined by four parameters (ℓ, h, p, n); these four parameters are the lengths of four of the six segments on the boundary of the L-shape; see Fig. 2.1(a). For example, the MDD in Fig. 2.1(a) has an L-shape $(\ell, h, p, n) = (5, 3, 1, 1)$. An L-shape is *degenerate* if its shape is a rectangle; for example, the MDD in Fig. 2.1(b) is degenerate.

[1] Although a network and the graph modeling it are conceptually distinct, we shall use the terms "node" and "verex" interchangeably when there is no ambiguity.



(a) An MDD of $DL(14;1,5)$.

(b) An MDD of $DL(10;2,7)$.

Figure 2.1: MDDs of DLNs.

Fiol et al. [33] observed that the distribution of all points with the same label repeat periodically and an MDD always tessellates the plane regardless of whether its L-shape is degenerate or not. By considering the relative positions of point with the label 0, Fiol et al. derived the following congruences:

$$\begin{aligned} \ell s_1 - n s_2 &\equiv 0 \pmod N \\ -p s_1 + h s_2 &\equiv 0 \pmod N. \end{aligned} \tag{2.2.1}$$

Let vectors $\boldsymbol{\alpha} = (\ell, -n)$ and $\boldsymbol{\beta} = (-p, h)$. It is known that all the points with the label 0 can be generated by repeatedly adding $\pm\boldsymbol{\alpha}$ and $\pm\boldsymbol{\beta}$ to each new point with the label 0. Moreover, if one location of node u is known, then the positions of all other points with the label u can be expressed in terms of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ [27].

Chen and Hwang [17] used the observation (2.2.1) to prove that an L-shape is degenerate if and only if exactly one of the two congruences: $\ell s_1 \equiv 0 \bmod N$ and $h s_2 \equiv 0 \bmod N$ is satisfied. They introduced the *Chen-Hwang-Rules* [17] to define the lengths of segments on the boundary of the L-shape when an L-shape is degenerate; see Fig. 2.2. As an example, the L-shape of $DL(10; 2, 7)$ in Fig. 2.1(b) is $(5, 2, 2, 0)$.

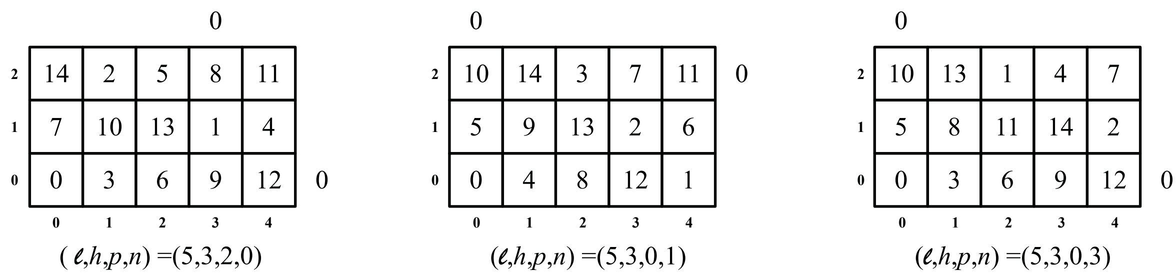
Wong and Coppersmith [64] gave an $O(N)$-time algorithm to construct an MDD (hence the L-shape) diagonally starting from point $(0, 0)$. Specifically, consider filling numbers in

Rule 1. Suppose $hs_2 \not\equiv \ell s_1 \equiv 0 \pmod{N}$. Let the zero immediately above the L-shape be at point (i, h). Then $p = \ell - i,\ n = 0$.

Rule 2. Suppose $\ell s_1 \not\equiv hs_2 \equiv 0 \pmod{N}$. Let the zero immediately to the right of the L-shape be at point (ℓ, j). Then $p = 0,\ n = h - j$.

Rule 3. Suppose $\ell s_1 \equiv hs_2 \equiv 0 \pmod{N}$. If $h > \ell$, follow Rule 1; otherwise, follow Rule 2. Note that for an L-shape (ℓ, h, p, n), we have $\ell > 0, h > 0, p \geq 0, n \geq 0$, p and n not both zero.

Figure 2.2: Chen-Hwang-Rules



(a) Rule 1 (b) Rule 2 (c) Rule 3

Figure 2.3: The (ℓ, h, p, n) determined by the Chen-Hwang-Rules in [17].

$\{ (x, y) \mid x \geq 0, y \geq 0, x \in \mathbb{Z}, y \in \mathbb{Z} \}$. Start from the origin (0, 0), then the line (1, 0), (0, 1), and then the line (2, 0), (1, 1), (0, 2), and so on. At each lattice point (x, y) (i.e., x, y being integers), if the value u, where $xs_1 + ys_2 \equiv u \pmod{N}$, has not appeared so far, we fill u at point (x, y), otherwise we just leave a blank. We stop when all values of u, i.e. $u = 0, 1, \ldots, N-1$, have been accounted for.

Cheng and Hwang [21] gavn an $O(\log N)$-time algorithm, we call it *Cheng-Hwang-Algorithm*, based on the Euclidean algorithm, to compute the L-shape (ℓ, h, p, n). For the completeness of this thesis, the Cheng-Hwang-Algorithm is given in Appendix A.

However, when an L-shape is degenerate, the solution of (ℓ, h, p, n) determined by Chen-Hwang-Rules [17] does not always coincide with the values determined by the Cheng-Hwang-Algorithm [21]. One such example is that for $DL(15; 4, 5)$, Chen-Hwang-Rules determines the L-shape $(\ell, h, p, n) = (5, 3, 0, 1)$, whereas Cheng-Hwang-Algorithm determines the L-

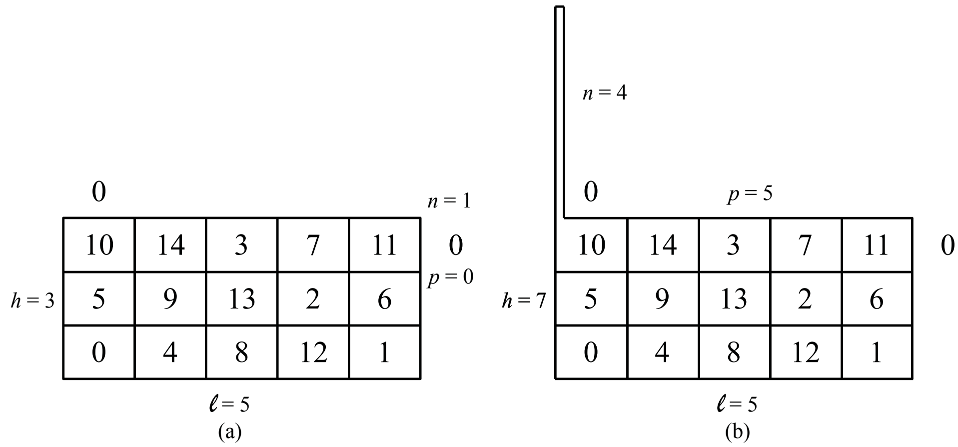


Figure 2.4: The inconsistency between Chen-Hwang-Rules and Cheng-Hwang-Algorithm.

shape $(\ell, h, p, n) = (5, 7, 5, 4)$; see Fig. 2.4.

Clearly, the result determined by Chen-Hwang-Rules is more accurate and precise. In addition, our algorithms (diameter-computing algorithm, routing algorithm) for the distance-related problems on the MCRNs highly rely on the correct information of the L-shapes. Thus, to overcome this problem, Lee, Lan and Chen [45] proposed a simple modification to the Cheng-Hwang-Algorithm as follows: Let $(\hat{\ell}, \hat{h}, \hat{p}, \hat{n})$ denote the solution of Cheng-Hwang-Algorithm and $(\bar{\ell}, \bar{h}, \bar{p}, \bar{n})$, the solution of Chen-Hwang-Rules.

Theorem 2.2.2. [45] *Given $DL(N; s_1, s_2)$, let $d = \gcd(N, s_1)$, $d' = \gcd(N, s_2)$. Then*

1. *If $DL(N; s_1, s_2)$ satisfies $d' > 1$ and there exists $1 \leq j \leq \min\{d' - 1, \frac{N}{d'} - 1\}$ such that $d's_1 \equiv js_2 \pmod N$ with $j < \frac{N}{2d'}$, then $\bar{\ell} = \hat{\ell}$, $\bar{h} = \hat{h} - \hat{n}$, $\bar{p} = 0$, $\bar{n} = j$.*

2. *If $DL(N; s_1, s_2)$ satisfies $d > 1$, $d' > 1$ and $d's_1 \equiv ds_2 \equiv 0 \pmod N$ and $d < d'$, then $\bar{\ell} = \hat{\ell}$, $\bar{h} = \bar{n} = \hat{h} - \hat{n}$, $\bar{p} = 0$.*

2.3 The Mixed Chordal Ring Network

A *mixed chordal ring network* (MCRN for short) $MCR(N; s, w)$ can be modeled by using a digraph with N nodes $0, 1, \ldots, N-1$ and $2N$ links of the following types

$$\begin{array}{lll} \text{ring-links:} & i \to (i+s) \bmod N, & i=0,1,2,\ldots,N-1, \\ \text{chordal-links:} & i \to (i-w) \bmod N, & i=0,2,4,\ldots,N-2, \\ \text{chordal-links:} & i \to (i+w) \bmod N, & i=1,3,5,\ldots,N-1, \end{array}$$

where N is even, both s and w are odd, and $0 < s \neq w < N$. It should be noted that the parameters s and w should satisfy $s + w \neq N$ in order to prevent the multiple links between two nodes of the digraph, which means a waste of the hardware. Chen, Hwang and Liu [20] proved the following theorem.

Theorem 2.3.1. [20] *$MCR(N; s, w)$ is strongly 2-connected if and only if* $\gcd(N, s, w) = 1$.

The proofs of equation (1.4.4) and Theorem 2.3.1 are based on the idea of embedding a MCRN into a DLN. Specifically, Chen, Hwang and Liu [20] showed that the MCRN $MCR(N; s, w)$ can be embedded into the DLN $DL(\frac{N}{2}; \frac{s-w}{2}, \frac{s+w}{2})$ by combining nodes $2k+1$ and $2k+1+w$ as *supernode* k^* for all $k = 0, 1, \ldots, N/2–1$, where $\frac{s-w}{2} = \left(\frac{s-w}{2}\right) \bmod \frac{N}{2}$, $\frac{s+w}{2} = \left(\frac{s+w}{2}\right) \bmod \frac{N}{2}$. They used this idea to obtain the connectivity and diameter information of the MCRNs. However, we observe that this embedding sometimes fails. Take $MCR(10; 1, 5)$ as an example; its corresponding DLN is $DL(\frac{10}{2}; \frac{1-5}{2}, \frac{1+5}{2})$, i.e., $DL(5; 3, 3)$, which is clearly not a valid DLN, yet $MCR(10; 1, 5)$ is a valid mixed chordal ring network. In general, $MCR(2(2k+1); 1, 2k+1)$ can not be embedded into a valid DLN. The idea used in [20] to prove Theorem 2.3.1 is to show that $MCR(N; s, w)$ is strongly 2-connected if and only if its corresponding DLN $DL(\frac{N}{2}; \frac{s-w}{2}, \frac{s+w}{2})$ is strongly 2-connected. We now correct the proof. First, a lemma is needed.

Lemma 2.3.2. *For $MCR(N; s, w)$,*

1. *if $w \neq \frac{N}{2}$, then $DL(\frac{N}{2}; \frac{s-w}{2}, \frac{s+w}{2})$ is a double-loop network;*

2. *if $w = \frac{N}{2}$, then $DL(\frac{N}{2}; \frac{s-w}{2}, \frac{s+w}{2})$ is not a double-loop network and $MCR(N; s, \frac{N}{2})$ is itself the double-loop network $DL(N; s, \frac{N}{2})$.*

Proof. $DL(\frac{N}{2}; \frac{s-w}{2}, \frac{s+w}{2})$ is not a valid double-loop network whenever $\frac{s-w}{2} \equiv 0 \pmod{\frac{N}{2}}$ or $\frac{s+w}{2} \equiv 0 \pmod{\frac{N}{2}}$ or $\frac{s-w}{2} \equiv \frac{s+w}{2} \pmod{\frac{N}{2}}$ or $\gcd(\frac{N}{2}, \frac{s-w}{2}, \frac{s+w}{2}) \neq 1$. Since we assume $s \neq w$ and $s + w \neq N$, it is impossible that $\frac{s-w}{2} \equiv 0 \pmod{\frac{N}{2}}$ or $\frac{s+w}{2} \equiv 0 \pmod{\frac{N}{2}}$. Also, $\frac{s-w}{2} \equiv \frac{s+w}{2} \pmod{\frac{N}{2}}$ if and only if $w = \frac{N}{2}$. In addition, we have assumed $\gcd(N, s, w) = 1$; therefore $\gcd(\frac{N}{2}, \frac{s-w}{2}, \frac{s+w}{2}) = 1$. Thus we have the first if-statement. When $w = \frac{N}{2}$, $\frac{N}{2} \equiv -\frac{N}{2} \pmod{N}$ occurs and the chordal-links of $MCR(N; s, w)$ become:

$$i \to (i + \tfrac{N}{2}) \bmod N, \; i = 0, 1, \ldots, N-1.$$

Thus $MCR(N; s, \frac{N}{2})$ is itself the double-loop network $DL(N; s, \frac{N}{2})$ with steps s and $N/2$, and we have the second if-statement. □

Lemma 2.3.2 shows that $DL(\frac{N}{2}; \frac{s-w}{2}, \frac{s+w}{2})$ is a valid embedding if and only if $w \neq \frac{N}{2}$. In [20], the following lemma is proved.

Lemma 2.3.3. ([20]) *$MCR(N; s, w)$ is strongly connected if and only if* $\gcd(N, s, w) = 1$.

Now we give a correct proof for Theorem 2.3.1.

Proof of Theorem 2.3.1: *Necessity.* Since $MCR(N; s, w)$ is strongly 2-connected, it is also strongly connected. Thus, this part follows directly from Lemma 2.3.3.

Sufficiency. There are two cases.

Case 1: $w \neq \frac{N}{2}$. Then by Lemma 2.3.2, $DL(\frac{N}{2}; \frac{s-w}{2}, \frac{s+w}{2})$ is a double-loop network. Since $w \neq \frac{N}{2}$, $\frac{s-w}{2} \neq \frac{s+w}{2}$. Since $\gcd(N, s, w) = 1$, $\gcd(\frac{N}{2}, \frac{s-w}{2}, \frac{s+w}{2}) = 1$. Thus by Theorem 2.2.1,

$DL(\frac{N}{2}; \frac{s-w}{2}, \frac{s+w}{2})$ is strongly 2-connected. Since the two nodes in each super-node can reach each other through the chordal-links between them, $MCR(N; s, w)$ is strongly 2-connected. Case 2: $w = \frac{N}{2}$. By Lemma 2.3.2, $MCR(N; s, w)$ is itself the double-loop network $DL(N; s, w)$. Thus by Theorem 2.2.1 and by the assumption that $\gcd(N, s, w) = 1$, $MCR(N; s, w)$ is strongly 2-connected. □

Being *vertex-transitive* (or *vertex symmetric*) is a desirable property of an efficient network topology. Intuitively, a vertex-transitive network looks the same from any node. This property reduces the complexity of distance-related problems. For example, it allows the use of an identical routing algorithm at every node. However, as was pointed out in [44], an MCRN may fail to be vertex-transitive. One such example is $MCR(12; 3, 5)$, in which node 0 can reach any node within 4 steps, while it takes 5 steps for node 1 to reach node 8.

Although an MCRN may fail to be vertex-transitive, it does satisfy the *even-odd-vertex-transitive* property: for every pair of vertices $u, v \in \{0, 1, \ldots, N-1\}$ with the same parity, there is an automorphism φ that maps u to v. In other words, in an MCRN, all even-numbered nodes are symmetric and all odd-numbered nodes are symmetric. By using this property, we may pay our attention to node 0 and node 1 without loss of generality. In Theorem 2.3.4, we further prove that node 1 can be regarded as an even-numbered node in another MCRN.

Two MCRNs $MCR(N; s, w)$ and $MCR(N; s', w')$ are said to be *strongly isomorphic* if there is a bijection φ from the nodes of $MCR(N; s, w)$ to the nodes of $MCR(N; s', w')$ such that $\varphi(v + s) = \varphi(v) + s'$ for all nodes v and either

$$\begin{cases} \varphi(v - w) = \varphi(v) - w', & \text{for even } v \text{ and even } \varphi(v); \\ \varphi(v + w) = \varphi(v) + w', & \text{for odd } v \text{ and odd } \varphi(v). \end{cases}$$

or

$$\begin{cases} \varphi(v - w) = \varphi(v) + w', & \text{for even } v \text{ and odd } \varphi(v); \\ \varphi(v + w) = \varphi(v) - w', & \text{for odd } v \text{ and even } \varphi(v). \end{cases}$$

Theorem 2.3.4. *$MCR(N;s,w)$ and $MCR(N;s,N-w)$ are strongly isomorphic.*

Proof. Let the bijection from the nodes of $MCR(N;s,w)$ to the nodes of $MCR(N;s,N-w)$ be

$$\varphi(v) = (v+w) \mod N. \tag{2.3.1}$$

It is not difficult to check that $\varphi(v+s) = \varphi(v)+s$ for all nodes v and $\varphi(v-w) = \varphi(v)+N-w$ for even v and odd $\varphi(v)$; $\varphi(v+w) = \varphi(v) - N + w$ for odd v and even $\varphi(v)$. Therefore $MCR(N;s,w)$ and $MCR(N;s,N-w)$ are strongly isomorphic. □

For convenience, the function in (2.3.1) is called the *renaming function*. From the above discussion, throughout this thesis, we will assume that $MCR(N;s,w)$ satisfies the following conditions:

$$s \neq w, \; s+w \neq N, \; w \neq N/2, \text{ and } \gcd(N,s,w) = 1. \tag{2.3.2}$$

The first two assumptions are from the definition of the MCRN in order to prevent multiple links between two nodes. The reason for the assumption $w \neq N/2$ is that since $MCR(N;s,\frac{N}{2})$ is $DL(N;s,\frac{N}{2})$ and many previous results of DLNs can apply on it. Besides, since we only consider connected graph, the last assumption ensure the MCRN being strongly connected. Furthermore, by the even-odd-vertex-transitive property of MCRNs, without loss of generality, we may restrict our discussion on node 0 and node 1 (to obtain the diameter and to obtain a routing path). Moreover, by Theorem 2.3.4, node 1 of $MCR(N;s,w)$ can be regarded as the even-numbered node $(1+w)$ mod N in $MCR(N;s,N-w)$; the node $(1+w)$ mod N can be further regarded as node 0 in $MCR(N;s,N-w)$.

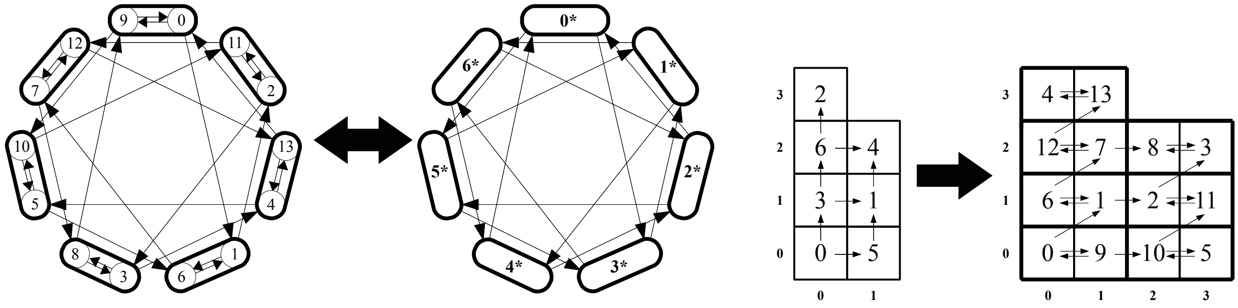
Chapter 3

The Minimum Distance Diagrams of Mixed Chordal Ring Networks

The purpose of this chapter is to explore and to investigate the minimum distance diagrams of mixed chordal ring networks. Results derived from this chapter have been submitted to [43]. The definition of the minimum distance diagrams of a mixed chordal ring network is given in Section 3.3.

3.1 The Two-Dimensional Integer Lattice Environment

One approach to study the distance-related problems of MCRNs is as that of in DLNs: Maps (or labels) each point of the two-dimensional integer lattice $\mathbb{Z}^+ \times \mathbb{Z}^+$ to a node of a given MCRN. However, since an MCRN is only even-odd-vertex-transitive, it is not clear how to label each point of $\mathbb{Z}^+ \times \mathbb{Z}^+$ for a given MCRN. In other words, how to define the *labeling function* from the points of $\mathbb{Z}^+ \times \mathbb{Z}^+$ to the nodes of a given MCRN is our first issue. In the following, the labeling function we defined is based on the PSEUDOMDD introduced in Section 3.1.1.



(a) $MCR(14;1,5)$ (b) $DL(7;5,3)$ (c) An MDD of $DL(7;5,3)$ (d) PSEUDOMDD

Figure 3.1: Embedding a mixed chordal ring network into a double-loop network.

3.1.1 The Embedding Technique and the PseudoMDD

Graph embedding is an important technique as we can take the advantage of all the known results about the host graph and apply these results on the guest graph. Given an $MCR(N;s,w)$ with $w \neq N/2$, we can embed $MCR(N;s,w)$ into $DL\left(\frac{N}{2};\frac{s-w}{2},\frac{s+w}{2}\right)$ by combining nodes $2k$ and $2k-w$ as *supernode* k^* for all $k = 0,1,\ldots,N/2-1$. Note that, unless otherwise specified, $\frac{s-w}{2}$ means $\left(\frac{s-w}{2}\right) \bmod \frac{N}{2}$, $\frac{s+w}{2}$ means $\left(\frac{s+w}{2}\right) \bmod \frac{N}{2}$, nodes of an MCRN are taken modulo N (thus node u means node $u \bmod N$), and nodes of a DLN with $N/2$ nodes are taken modulo $N/2$ (thus node v means node $v \bmod N/2$). Figs. 3.1(a) and 3.1(b) illustrate the embedding of $MCR(14;1,5)$ into $DL(7;5,3)$ and the bold rounded rectangles indicate the supernodes (host nodes). Since we can embed an MCRN into a DLN, we can embed an MCRN into the MDD of the corresponding DLN. Given $MCR(N;s,w)$, the PSEUDOMDD is constructed as follows: (see Figs. 3.1(c) and 3.1(d)):

pseudoMDD: Replace each node u in the MDD of $DL\left(\frac{N}{2};\frac{s-w}{2},\frac{s+w}{2}\right)$ with two nodes $2u$ and $2u-w$. If u is at point (x,y), then $2u$ and $2u-w$ are at points $(2x,y)$ and $(2x+1,y)$, respectively.

Recall that the MDD of a DLN always forms an L-shape, and this MDD tessellates the plane. Since a PSEUDOMDD provides a one-to-one correspondence between the corresponding DLN's MDD's and the PSEUDOMDD's, it is obvious that a PSEUDOMDD is also an

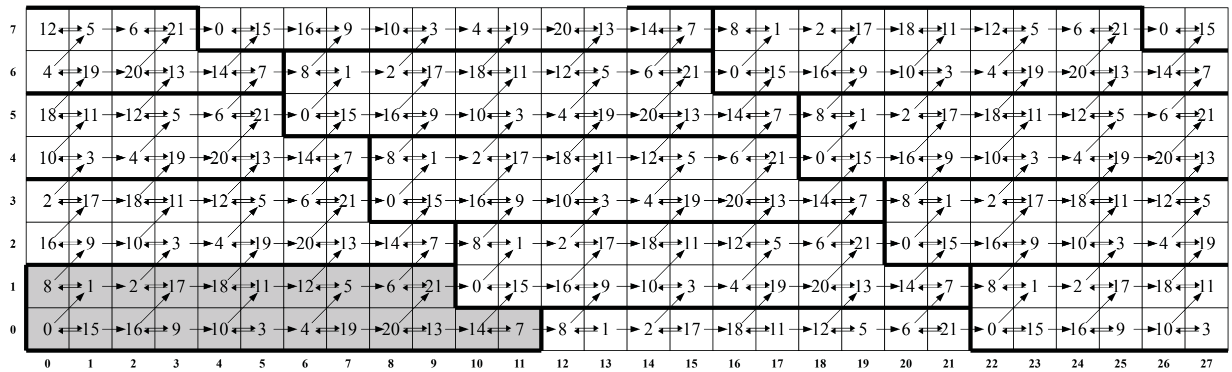


Figure 3.2: The tessellation of the plane formed by the PSEUDOMDD of $MCR(22; 1, 7)$.

L-shape, but the length of the horizontal segment on the boundary of the PSEUDOMDD is twice of that of the corresponding DLN's MDD. For example in Figs. 3.1(c)(d), the PSEUDOMDD has an L-shape $(\ell, h, p, n) = (4, 4, 2, 1)$, whereas the corresponding DLN's MDD has an L-shape $(\ell, h, p, n) = (2, 4, 1, 1)$. We have the following fact.

Fact. A PSEUDOMDD has the following properties.

(i) It contains every node of the MCRN exactly once.

(ii) The shape is always an L-shape with parameters $(2\ell, h, 2p, n)$ whenever the corresponding DLN's MDD has an L-shape (ℓ, h, p, n).

(iii) It always tessellates the plane (see Fig. 3.2 for an example).

The name "PSEUDOMDD" comes from the reason that a PSEUDOMDD may *fail* to be a "minimum" distance diagram. For example, consider Fig. 3.2. Both points $(8, 0)$ and $(6, 2)$ represent node 20. However, the distance (minimum number of links) from point $(0, 0)$ to $(8, 0)$ is 8 (the unique shortest path is $0 \to 15 \to 16 \to 9 \to 10 \to 3 \to 4 \to 19 \to 20$) while the distance from point $(0, 0)$ to $(6, 2)$ is 6 (one of the shortest path is $0 \to 1 \to 2 \to 3 \to 4 \to 19 \to 20$), yet point $(8, 0)$ is inside the PSEUDOMDD. Note that some PSEUDOMDD's are indeed MDD's. See Section 3.3 for more further discussion.

3.1.2 The Labeling Function

Recall that a node u at point (x, y) of the MDD of $DL\left(\frac{N}{2}; \frac{s-w}{2}, \frac{s+w}{2}\right)$ satisfies

$$x\left(\frac{s-w}{2}\right) + y\left(\frac{s+w}{2}\right) \equiv u \pmod{\frac{N}{2}}.$$

By the construction of the PSEUDOMDD of $MCR(N; s, w)$, nodes $2u$ and $2u - w$ of $MCR(N; s, w)$ are at points $(2x, y)$ and $(2x + 1, y)$, respectively. As a result, the labeling function for point $\boldsymbol{z} = (x, y)$ is

$$l(\boldsymbol{z}) = \begin{cases} \frac{x}{2}(s-w) + y(s+w) & \pmod{N} \quad \text{if } x \text{ is even;} \\ \left(\frac{x-1}{2}\right)(s-w) + y(s+w) - w & \pmod{N} \quad \text{if } x \text{ is odd.} \end{cases} \tag{3.1.1}$$

Or, equivalently

$$l(\boldsymbol{z}) = \left(\left\lfloor \frac{x}{2} \right\rfloor + y\right) s - \left(\left\lceil \frac{x}{2} \right\rceil - y\right) w \mod N. \tag{3.1.2}$$

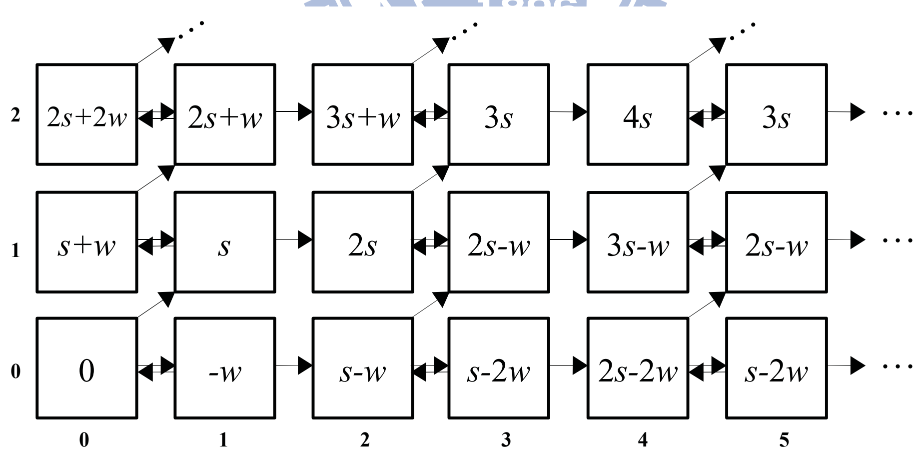


Figure 3.3: The labels for each point in the plane.

Table 3.1: The nodes that can be reached from node u by using one link.

Node u at point (x, y)	node	at point
x is even	$u+s$	$(x+1, y+1)$
	$u-w$	$(x+1, y)$
x is odd	$u+s$	$(x+1, y)$
	$u+w$	$(x-1, y)$

3.1.3 The Interconnection Rules

It should be noticed that the interconnection rules between adjacent points in the two-dimensional integer lattice are quite different from those of DLNs (recall that in DLNs, each point can reach either an east or a north point). Roughly speaking, a point (x, y) can reach either a) *east or northeast points* or b) *east or west points*, depending on the parity of x. Nodes that can be reached from node u at point (x, y) are shown in Fig. 3.4 and Table 3.1. Note that we will only consider points in $\mathbb{Z}^+ \times \mathbb{Z}^+ = \{ (x, y) \in \mathbb{Z} \times \mathbb{Z} \mid x \geq 0, y \geq 0 \}$.

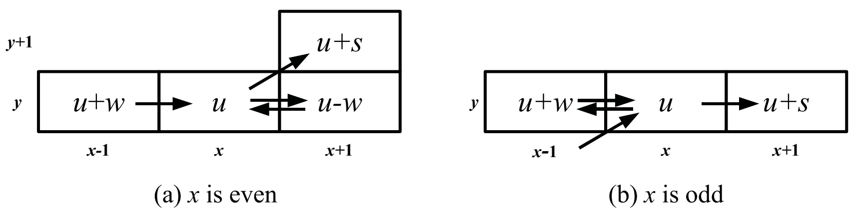


Figure 3.4: The interconnection rules.

3.1.4 The Distance-related Properties

Some distance-related properties will be investigated in this section. For convenience, some notations will be introduced first. Define the *parity* of an integer x to be 0 if $(x \bmod 2)$ equals to 0 and 1 if $(x \bmod 2)$ equals to 1. The parity of an integer x is denoted by $parity(x)$.

Partition $\mathbb{Z}^+ \times \mathbb{Z}^+$ into $\mathbf{\Gamma}^+$ and $\mathbf{\Gamma}^-$ as follows:

$$\mathbf{\Gamma}^+ \stackrel{\text{def}}{=} \{\, (x,y) \in \mathbb{Z}^+ \times \mathbb{Z}^+ \mid x \geq 2y \geq 0 \,\} \text{ and } \mathbf{\Gamma}^- \stackrel{\text{def}}{=} \{\, (x,y) \in \mathbb{Z}^+ \times \mathbb{Z}^+ \mid 0 \leq x < 2y \,\}.$$

Each point $\boldsymbol{z} = (x,y)$ of $\mathbb{Z}^+ \times \mathbb{Z}^+$ is associated with a *distance* (or *norm*), denoted by $\Delta(\boldsymbol{z})$, which is the minimum number of links that needs to be traversed from point $(0,0)$ to (x,y). The distance for each point can be determined as follows.

Lemma 3.1.1. *The distance of point $\boldsymbol{z} = (x,y)$ is*

$$\Delta(\boldsymbol{z}) = \begin{cases} x & \text{if } \boldsymbol{z} \in \mathbf{\Gamma}^+, \\ 2y - parity(x) & \text{if } \boldsymbol{z} \in \mathbf{\Gamma}^-. \end{cases} \tag{3.1.3}$$

Proof. We prove this lemma by induction on x and y. For the basis step, clearly, $\Delta((0,0)) = 0$, $\Delta((x,0)) = x$, $\Delta((0,y)) = 2y$ and thus (3.1.3) holds. For the induction step, suppose (3.1.3) holds for points $(x-1,y)$, $(x-1,y-1)$ and $(x,y-1)$. Now consider point (x,y), where $x \geq 1$ and $y \geq 1$.

Case 1: x is even. Then $\Delta((x,y)) = \min\{\, d_1, d_2 \,\}$, where $d_1 = \Delta((x-1,y)) + 1$ and $d_2 = \Delta((x,y-1)) + 2$.

Subcase 1.1: $x \geq 2y$. By the induction hypothesis, $d_2 = x + 2$. If $x - 1 \geq 2y$, then by the induction hypothesis, $d_1 = x$; if $x - 1 < 2y$, then we have $x = 2y$ and hence, by the induction hypothesis, $d_1 = x$. Therefore $\Delta((x,y)) = x$ and (3.1.3) holds.

Subcase 1.2: $x < 2y$. By the induction hypothesis, $d_1 = 2y$. If $x \geq 2(y-1)$, then $x = 2y - 2$. By the induction hypothesis $d_2 = 2y$; if $x < 2(y-1)$, then by the induction hypothesis $d_2 = 2y$. Therefore $\Delta((x,y)) = 2y$ and (3.1.3) holds.

Case 2: x is odd. Then $\Delta((x,y)) = \min\{\, d_1, d_2 \,\}$, where $d_1 = \Delta((x-1,y)) + 1$ and $d_2 = \Delta((x-1,y-1)) + 1$.

Subcase 2.1: $x \geq 2y$. If $x-1 \geq 2y$, then by the induction hypothesis, $d_1 = d_2 = x$; if $x-1 < 2y$, then we have $x = 2y$, a contradiction to odd x. Hence $\Delta((x,y)) = x$ and (3.1.3) holds.

Subcase 2.2: $x < 2y$. Clearly by the induction hypothesis, $d_1 = 2y+1$. If $x-1 \geq 2(y-1)$ then $x = 2y-1$ and, by the induction hypothesis, $d_2 = x = 2y-1$; if $x-1 < 2(y-1)$ then by the induction hypothesis, $d_2 = 2y-1$. Therefore $\Delta((x,y)) = 2y-1$ and (3.1.3) holds. □

Note that the distance function for point (x,y) in the two-dimensional integer lattice is quite different from the standard one (i.e., $|x|+|y|$). A tool that can compare the distances of the two points is given as follows. For point $\boldsymbol{z} = (x,y)$ and vector $\boldsymbol{v} = (v_1, v_2)$ with v_1, v_2 being integers, let $\boldsymbol{z}+\boldsymbol{v}$ denote the point $(x+v_1, y+v_2)$. Then:

Lemma 3.1.2. *Suppose $\boldsymbol{v} = (v_1, v_2)$ with even v_1. Then we have $\Delta(\boldsymbol{z}) \leq \Delta(\boldsymbol{z}+\boldsymbol{v})$ if*

(i) $\boldsymbol{z} \in \boldsymbol{\Gamma}^+$ *and* $v_1 \geq 0, v_2 \leq 0$ *or*

(ii) $\boldsymbol{z} \in \boldsymbol{\Gamma}^-$ *and* $v_1 \leq 0, v_2 \geq 0$ *or*

(iii) $v_1 \geq 0$ *and* $v_2 \geq 0$.

Proof. Since v_1 is even, $parity(x) = parity(x+v_1)$. The first two cases (i) and (ii) come from (3.1.3) undoubtedly. Now consider case (iii). If $\boldsymbol{z}$ and $\boldsymbol{z}+\boldsymbol{v}$ are both in $\boldsymbol{\Gamma}^+$ (or $\boldsymbol{\Gamma}^-$), then the result is easy to see. Suppose $\boldsymbol{z} \in \boldsymbol{\Gamma}^+$ and $\boldsymbol{z}+\boldsymbol{v} \in \boldsymbol{\Gamma}^-$. Let $\boldsymbol{z'} = (x+v_1, y)$. Since $v_1 \geq 0$, clearly $\boldsymbol{z'} \in \boldsymbol{\Gamma}^+$. By (3.1.3), $\Delta(\boldsymbol{z}) = x$, $\Delta(\boldsymbol{z'}) = x+v_1$ and $\Delta(\boldsymbol{z}+\boldsymbol{v}) = 2y+2v_2-parity(x)$. Since $\boldsymbol{z}+\boldsymbol{v} \in \boldsymbol{\Gamma}^-$, we have $x+v_1 < 2y+2v_2$. Therefore $\Delta(\boldsymbol{z}) \leq \Delta(\boldsymbol{z'}) \leq \Delta(\boldsymbol{z}+\boldsymbol{v})$ holds. The case of $\boldsymbol{z} \in \boldsymbol{\Gamma}^-$ and $\boldsymbol{z}+\boldsymbol{v} \in \boldsymbol{\Gamma}^+$ is similar to obtain. □

Note that if point $\boldsymbol{z}+\boldsymbol{v}$ is outside $\mathbb{Z}^+ \times \mathbb{Z}^+$, then we may simply let $\Delta(\boldsymbol{z}+\boldsymbol{v}) = \infty$ to ensure the correctness of Lemma 3.1.2.

3.2 Finding an Optimal Copy

Suppose the PSEUDOMDD of $MCR(N; s, w)$ has an L-shape $(2\ell, h, 2p, n)$. The following two vectors that characterize the shape of the PSEUDOMDD are crucial in the remaining discussion and are defined by

$$\boldsymbol{\alpha} \stackrel{\text{def}}{=} (2\ell, -n),\ \boldsymbol{\beta} \stackrel{\text{def}}{=} (-2p, h).$$

Since a PSEUDOMDD consists of N points, for each node $u \in \{0, 1, \ldots, N-1\}$, there is exactly one point of the PSEUDOMDD with label u and we denote this point by $\boldsymbol{\pi}_u$. In $\mathbb{Z}^+ \times \mathbb{Z}^+$, points having the same label as $\boldsymbol{\pi}_u$ are called *copies* (or *relocations*) of $\boldsymbol{\pi}_u$. The set of all points with label u is denoted by $\boldsymbol{\Pi}_u$. Since a PSEUDOMDD can tessellate the plane, by considering all points with label 0, we perspicuously have that all the other copies of $\boldsymbol{\pi}_0$ can be expressed in terms of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. More generally, point $\boldsymbol{\pi}$ is a copy of point $\boldsymbol{\pi}_u$ if and only if $\boldsymbol{\pi} = \boldsymbol{\pi}_u + i\boldsymbol{\alpha} + j\boldsymbol{\beta}$ for some integers i, j; see Fig. 3.5. Given a $\boldsymbol{\pi}_u$, define $\boldsymbol{R}_u^{\boldsymbol{\alpha}}$ and $\boldsymbol{R}_u^{\boldsymbol{\beta}}$ as follows:

$$\boldsymbol{R}_u^{\boldsymbol{\alpha}} \stackrel{\text{def}}{=} \{\, \boldsymbol{\pi}_u + k\boldsymbol{\alpha} \mid k \in \mathbb{Z}, k \geq 0 \,\} \quad \text{and} \quad \boldsymbol{R}_u^{\boldsymbol{\beta}} \stackrel{\text{def}}{=} \{\, \boldsymbol{\pi}_u + k\boldsymbol{\beta} \mid k \in \mathbb{Z}, k \geq 0 \,\}.$$

Example. The PSEUDOMDD of $MCR(22; 1, 7)$ in Fig. 3.5 has an L-shape $(2\ell, h, 2p, n) = (12, 2, 2, 1)$ and its shape is characterized by vectors $\boldsymbol{\alpha} = (12, -1)$ and $\boldsymbol{\beta} = (-2, -2)$. For node $u = 20$, $\boldsymbol{\pi}_u$ is the point $(8, 0)$ and the copies of $\boldsymbol{\pi}_u$ are enclosed by a circle. $\boldsymbol{R}_u^{\boldsymbol{\alpha}} = \emptyset$ (since the points in $\boldsymbol{R}_u^{\boldsymbol{\alpha}}$ are outside $\mathbb{Z}^+ \times \mathbb{Z}^+$) and $\boldsymbol{R}_u^{\boldsymbol{\beta}} = \{\, (8, 0), (6, 2), (4, 4), (2, 6), (0, 8) \,\}$.

The purpose of this section is to look for an *optimal copy* of $\boldsymbol{\pi}_u$ for each node $u \in \{0, 1, \ldots, N-1\}$ of a given MCRN. We denote the optimal copy of $\boldsymbol{\pi}_u$ by $\boldsymbol{\pi}_u^*$. Clearly, $\Delta(\boldsymbol{\pi}_u^*) \leq \Delta(\boldsymbol{\pi})$ for all $\boldsymbol{\pi} \in \boldsymbol{\Pi}_u$. Although there are infinite number of copies of $\boldsymbol{\pi}_u$ in the two-dimensional integer lattice, in fact, we only need to consider those copies in $\boldsymbol{R}_u^{\boldsymbol{\alpha}}$ and $\boldsymbol{R}_u^{\boldsymbol{\beta}}$

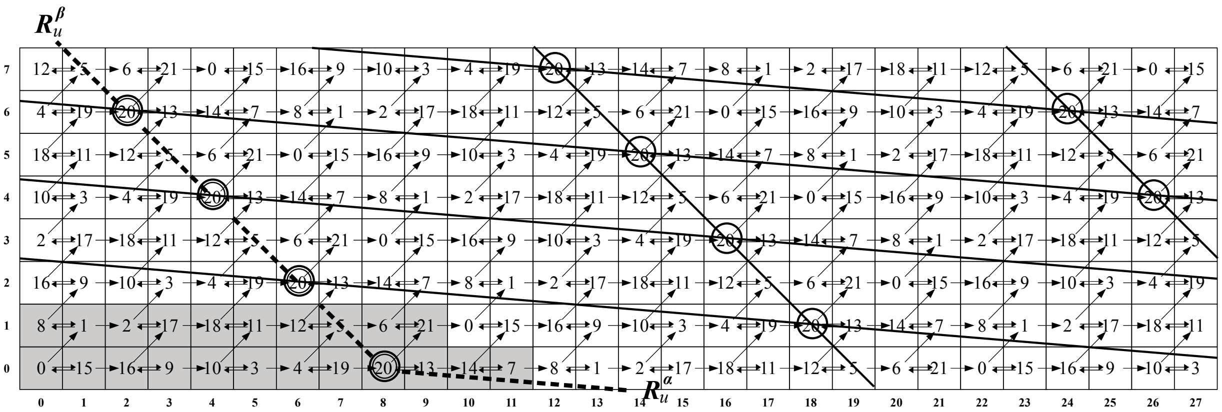


Figure 3.5: The illustrations of $\boldsymbol{\pi}_u$, copies of $\boldsymbol{\pi}_u$, $\boldsymbol{R}_u^{\boldsymbol{\alpha}}$ and $\boldsymbol{R}_u^{\boldsymbol{\beta}}$ for $u = 20$.

since a copy $\boldsymbol{\pi} \in \boldsymbol{\Pi}_u \setminus \{ \boldsymbol{R}_u^{\boldsymbol{\alpha}} \cup \boldsymbol{R}_u^{\boldsymbol{\beta}} \}$ is either outside $\mathbb{Z}^+ \times \mathbb{Z}^+$ or, by Lemma 3.1.2(iii), has a larger distance than that of $\boldsymbol{\pi}_u$.

Each $\boldsymbol{\pi}_u$ is associated with two points $\boldsymbol{\pi}_u^+$ and $\boldsymbol{\pi}_u^-$ defined as follows: If $\boldsymbol{\pi_u} \in \boldsymbol{\Gamma}^+$, let $\boldsymbol{\pi}_u^+$ and $\boldsymbol{\pi}_u^-$ denote the point in $\boldsymbol{R}_u^{\boldsymbol{\beta}}$ such that $\boldsymbol{\pi}_u^+ \in \boldsymbol{\Gamma}^+$, $\boldsymbol{\pi}_u^- \in \boldsymbol{\Gamma}^-$ and $\boldsymbol{\pi}_u^- = \boldsymbol{\pi}_u^+ + \boldsymbol{\beta}$. Similarly, If $\boldsymbol{\pi_u} \in \boldsymbol{\Gamma}^-$, let $\boldsymbol{\pi}_u^+$ and $\boldsymbol{\pi}_u^-$ denote the point in $\boldsymbol{R}_u^{\boldsymbol{\alpha}}$ such that $\boldsymbol{\pi}_u^+ \in \boldsymbol{\Gamma}^+$, $\boldsymbol{\pi}_u^- \in \boldsymbol{\Gamma}^-$ and $\boldsymbol{\pi}_u^+ = \boldsymbol{\pi}_u^- + \boldsymbol{\alpha}$; see Fig. 3.6 for illustrations. Take Fig. 3.5 for an example. Suppose $\boldsymbol{\pi}_u = (8, 0) \in \boldsymbol{\Gamma}^+$, then $\boldsymbol{\pi}_u^+$ and $\boldsymbol{\pi}_u^-$ are the point $(6, 2)$ and $(4, 4)$, respectively; suppose $\boldsymbol{\pi}_u = (0, 1) \in \boldsymbol{\Gamma}^-$, then $\boldsymbol{\pi}_u^- = (0, 1)$ and $\boldsymbol{\pi}_u^+ = (12, 0)$. Note that for $\boldsymbol{\pi}_u$, its $\boldsymbol{\pi}_u^+$ or $\boldsymbol{\pi}_u^-$ may not exist. For example, suppose $\boldsymbol{\pi}_u = (1, 0) \in \boldsymbol{\Gamma}^+$ in Fig. 3.5, then $\boldsymbol{\pi}_u + \boldsymbol{\beta} = (-1, 2)$ which is outside $\mathbb{Z}^+ \times \mathbb{Z}^+$. In this case, we have $\Delta(\boldsymbol{\pi}_u + \boldsymbol{\beta}) = \infty$. The following lemma tells that for each $\boldsymbol{\pi}_u$, $\boldsymbol{\pi}_u^*$ can be found by only considering $\boldsymbol{\pi}_u^+$ and $\boldsymbol{\pi}_u^-$.

Lemma 3.2.1. $\Delta(\boldsymbol{\pi}_u^*) = \min \{ \Delta(\boldsymbol{\pi}_u^+), \Delta(\boldsymbol{\pi}_u^-) \}$.

Proof. Suppose $\boldsymbol{\pi}_u \in \boldsymbol{\Gamma}^+$. Since $\boldsymbol{\pi}_u - \boldsymbol{\alpha}$ is outside the first quadrant and by Lemma 3.1.2(i), $\Delta(\boldsymbol{\pi}_u) \le \Delta(\boldsymbol{\pi}_u + \boldsymbol{\alpha}) \le \Delta(\boldsymbol{\pi}_u + 2\boldsymbol{\alpha}) \le \cdots$, we only need to consider points in $\boldsymbol{R}_u^{\boldsymbol{\beta}}$. By Lemma 3.1.2(i), $\Delta(\boldsymbol{\pi}_u^+) \le \Delta(\boldsymbol{\pi}_u^+ - \boldsymbol{\beta}) \le \Delta(\boldsymbol{\pi}_u^+ - 2\boldsymbol{\beta}) \le \cdots$. By Lemma 3.1.2(ii), $\Delta(\boldsymbol{\pi}_u^-) \le \Delta(\boldsymbol{\pi}_u^- + \boldsymbol{\beta}) \le \Delta(\boldsymbol{\pi}_u^- + 2\boldsymbol{\beta}) \le \cdots$. Hence $\Delta(\boldsymbol{\pi}_u^*) = \min \{ \Delta(\boldsymbol{\pi}_u^-), \Delta(\boldsymbol{\pi}_u^-) \}$. The case of $\boldsymbol{\pi}_u \in \boldsymbol{\Gamma}^-$ is similar to prove. $\square$

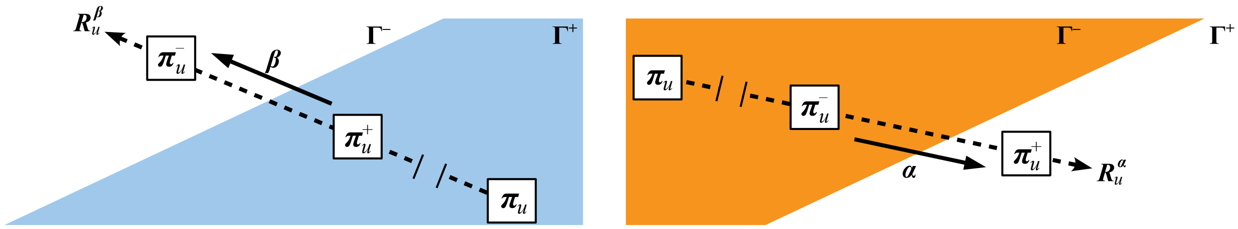


Figure 3.6: Two possible ways to find an optimal copy of $\boldsymbol{\pi}_u$. The left figure is for the $\boldsymbol{\pi_u} \in \boldsymbol{\Gamma}^+$ case; the right figure is for the $\boldsymbol{\pi_u} \in \boldsymbol{\Gamma}^-$ case.

By using the lengths $(2\ell, h, 2p, n)$ in a PSEUDOMDD, we partition $\boldsymbol{\Gamma}^+$ and $\boldsymbol{\Gamma}^-$ as follows (see Fig. 3.7 for an illustration):

$$\boldsymbol{\Gamma}^+ = \bigcup_{i=0}^{\infty} \boldsymbol{\Gamma}_i^+ \quad \text{and} \quad \boldsymbol{\Gamma}^- = \bigcup_{i=0}^{\infty} \boldsymbol{\Gamma}_i^-,$$

where

$$\begin{aligned} \boldsymbol{\Gamma}_0^+ &= \left\{ (x,y) \in \mathbb{Z}^+ \times \mathbb{Z}^+ \mid 0 \le x - 2y < 2h \right\}, \\ \boldsymbol{\Gamma}_0^- &= \left\{ (x,y) \in \mathbb{Z}^+ \times \mathbb{Z}^+ \mid -2\ell \le x - 2y < 0 \right\}, \end{aligned} \tag{3.2.1}$$

and for $i \in \mathbb{Z}, i \ge 1$

$$\begin{aligned} \boldsymbol{\Gamma}_i^+ &= \left\{ (x,y) \in \mathbb{Z}^+ \times \mathbb{Z}^+ \mid 2h + (i-1)\cdot(2h+2p) \le x - 2y < 2h + i\cdot(2h+2p) \right\}, \\ \boldsymbol{\Gamma}_i^- &= \left\{ (x,y) \in \mathbb{Z}^+ \times \mathbb{Z}^+ \mid -2\ell - i\cdot(2\ell+2n) \le x - 2y < -2\ell - (i-1)(2\ell+2n) \right\}. \end{aligned} \tag{3.2.2}$$

According to the relative position of $\boldsymbol{\pi}_u$ in $\boldsymbol{\Gamma}^+$ or $\boldsymbol{\Gamma}^-$, we can find $\boldsymbol{\pi}_u^*$ by using the following three lemmas (Lemmas 3.2.2, 3.2.3 and 3.2.4). For convenience, the equal sign followed $\boldsymbol{\pi}_u^*$ means "can be chosen as".

Lemma 3.2.2. *If $\boldsymbol{\pi}_u$ belongs to $\boldsymbol{\Gamma}_0^+$ or $\boldsymbol{\Gamma}_0^-$, then $\boldsymbol{\pi}_u^* = \boldsymbol{\pi}_u$.*

Proof. Let $\boldsymbol{\pi}_u = (x,y)$. If $\boldsymbol{\pi}_u \in \boldsymbol{\Gamma}_0^+$, then $\boldsymbol{\pi}_u + \boldsymbol{\beta} = (x-2p, y+h) \in \boldsymbol{\Gamma}^-$. This implies

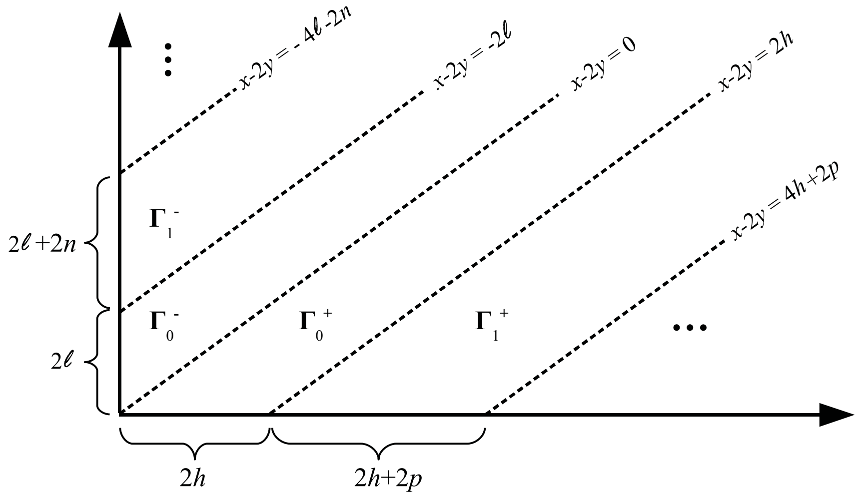


Figure 3.7: The partitions of $\mathbb{Z}^+ \times \mathbb{Z}^+$.

$\boldsymbol{\pi}_u^+ = \boldsymbol{\pi}_u$ and $\boldsymbol{\pi}_u^- = \boldsymbol{\pi}_u + \boldsymbol{\beta}$. By (3.1.3), $\Delta(\boldsymbol{\pi}_u^+) = x$, $\Delta(\boldsymbol{\pi}_u^-) = 2y + 2h - parity(x)$ and $\Delta(\boldsymbol{\pi}_u^+) \le \Delta(\boldsymbol{\pi}_u^-)$ holds. By Lemma 3.2.1, $\boldsymbol{\pi}_u^* = \boldsymbol{\pi}_u$. The case of $\boldsymbol{\pi}_u \in \boldsymbol{\Gamma}_0^-$ is similar and we omit the proof. □

Lemma 3.2.3. *If $\boldsymbol{\pi}_u$ belongs to $\boldsymbol{\Gamma}_i^+$ for some positive integer i, then $\boldsymbol{\pi}_u^* = \boldsymbol{\pi}_u + i \cdot \boldsymbol{\beta}$.*

Proof. Let $\boldsymbol{\pi}_u = (x, y)$, then $\boldsymbol{\pi}_u + (i-1) \cdot \boldsymbol{\beta} = \Big(x - 2(i-1)p, y + (i-1)h\Big)$, $\boldsymbol{\pi}_u + i \cdot \boldsymbol{\beta} = (x - 2ip, y + ih)$ and $\boldsymbol{\pi}_u + (i+1) \cdot \boldsymbol{\beta} = \Big(x - 2(i+1)p, y + (i+1)h\Big)$. It is not difficult to check that points $\boldsymbol{\pi}_u + (i-1) \cdot \boldsymbol{\beta}$, $\boldsymbol{\pi}_u + i \cdot \boldsymbol{\beta}$ and $\boldsymbol{\pi}_u + (i+1) \cdot \boldsymbol{\beta}$ are inside $\mathbb{Z}^+ \times \mathbb{Z}^+$. Since $\boldsymbol{\pi}_u \in \boldsymbol{\Gamma}_i^+$, we clearly have $\boldsymbol{\pi}_u \in \boldsymbol{\Gamma}^+$.

Now we further partition $\boldsymbol{\Gamma}_i^+$ into two smaller parts (possibly empty):

$$
\begin{aligned}
{}_{\mathrm{L}}\boldsymbol{\Gamma_i}^+ &= \Big\{ (x, y) \in \mathbb{Z}^+ \times \mathbb{Z}^+ \mid 2h + (i-1) \cdot (2h + 2p) \le x - 2y < 2h + (i-1) \cdot (2h + 2p) + 2p \Big\}, \\
{}_{\mathrm{R}}\boldsymbol{\Gamma_i}^+ &= \Big\{ (x, y) \in \mathbb{Z}^+ \times \mathbb{Z}^+ \mid 2h + (i-1) \cdot (2h + 2p) + 2p \le x - 2y < 2h + i \cdot (2h + 2p). \Big\}
\end{aligned}
$$

Suppose $\boldsymbol{\pi}_u \in {}_{\mathrm{L}}\boldsymbol{\Gamma_i}^+$, then $\boldsymbol{\pi}_u + (i-1) \cdot \boldsymbol{\beta} \in \boldsymbol{\Gamma}^+$ and $\boldsymbol{\pi}_u + i \cdot \boldsymbol{\beta} \in \boldsymbol{\Gamma}^-$. Hence $\boldsymbol{\pi}_u^+ = \boldsymbol{\pi}_u + (i-1) \cdot \boldsymbol{\beta}$ and $\boldsymbol{\pi}_u^- = \boldsymbol{\pi}_u + i \cdot \boldsymbol{\beta}$. By (3.1.3), $\Delta(\boldsymbol{\pi}_u^+) = x - 2(i-1)p$, $\Delta(\boldsymbol{\pi}_u^-) = 2y + 2ih - parity(x)$ and

$\Delta(\boldsymbol{\pi}_u^-) \leq \Delta(\boldsymbol{\pi}_u^+)$ holds. By Lemma 3.2.1, $\boldsymbol{\pi}_u^* = \boldsymbol{\pi}_u + i \cdot \boldsymbol{\beta}$ and $\Delta(\boldsymbol{\pi}_u^*) = 2y + 2ih - parity(x)$ hold.

Now suppose $\boldsymbol{\pi}_u \in {}_{\mathrm{R}}\boldsymbol{\Gamma_i}^+$, then $\boldsymbol{\pi}_u + i \cdot \boldsymbol{\beta} \in \boldsymbol{\Gamma}^+$ and $\boldsymbol{\pi}_u + (i+1) \cdot \boldsymbol{\beta} \in \boldsymbol{\Gamma}^-$. Therefore $\boldsymbol{\pi}_u^+ = \boldsymbol{\pi}_u + i \cdot \boldsymbol{\beta}$ and $\boldsymbol{\pi}_u^- = \boldsymbol{\pi}_u + (i+1) \cdot \boldsymbol{\beta}$. By (3.1.3), $\Delta(\boldsymbol{\pi}_u^+) = x - 2ip$, $\Delta(\boldsymbol{\pi}_u^-) = 2y + 2(i+1)h - parity(x)$ and $\Delta(\boldsymbol{\pi}_u^+) \leq \Delta(\boldsymbol{\pi}_u^-)$ holds. Again, by Lemma 3.2.1, we have $\boldsymbol{\pi}_u^* = \boldsymbol{\pi}_u + i \cdot \boldsymbol{\beta}$ and $\Delta(\boldsymbol{\pi}_u^*) = x - 2ip$. □

By using similar arguments, we can obtain the following lemma and we omit the proof.

Lemma 3.2.4. *If $\boldsymbol{\pi}_u$ belongs to $\boldsymbol{\Gamma}_i^-$ for some positive integer i, then $\boldsymbol{\pi}_u^* = \boldsymbol{\pi}_u + i \cdot \boldsymbol{\alpha}$.*

By combining Lemmas 3.2.2, 3.2.3 and 3.2.4, we can find an optimal copy of $\boldsymbol{\pi}_u$ as follows.

Theorem 3.2.5. *If $\boldsymbol{\pi}_u \in \boldsymbol{\Gamma}_i^+$ (resp., $\boldsymbol{\pi}_u \in \boldsymbol{\Gamma}_i^-$) for some non-negative integer i, then $\boldsymbol{\pi}_u^* = \boldsymbol{\pi}_u + i \cdot \boldsymbol{\beta}$ (resp., $\boldsymbol{\pi}_u + i \cdot \boldsymbol{\alpha}$).*

Theorem 3.2.5 states that an optimal copy of $\boldsymbol{\pi}_u$ can be obtained by "moving" $\boldsymbol{\pi}_u$ to some other copy in $\boldsymbol{R}_u^{\boldsymbol{\alpha}}$ or $\boldsymbol{R}_u^{\boldsymbol{\beta}}$. In particular, if $\boldsymbol{\pi}_u \in \boldsymbol{\Gamma}_i^+$ (resp., $\boldsymbol{\Gamma}_i^-$), then $\boldsymbol{\pi}_u^*$ can be obtained by moving $\boldsymbol{\pi}_u$ i steps in $\boldsymbol{R}_u^{\boldsymbol{\beta}}$ (resp., $\boldsymbol{R}_u^{\boldsymbol{\alpha}}$).

Example. Take $MCR(22; 1, 7)$ in Fig. 3.5 for an example. For $u = 19$, we have $\boldsymbol{\pi}_u = (7, 0) \in \boldsymbol{\Gamma}_1^+$. Hence $\boldsymbol{\pi}_u^* = \boldsymbol{\pi}_u + 1 \cdot \boldsymbol{\beta} = (5, 2)$ and $\boldsymbol{\pi}_u + 1 \cdot \boldsymbol{\beta}$ has the minimum distance (note that $\Delta(\boldsymbol{\pi}_u + 1 \cdot \boldsymbol{\beta}) = 5$) among all points in $\boldsymbol{\Pi}_u$. For $u = 14$, we have $\boldsymbol{\pi}_u = (10, 0) \in \boldsymbol{\Gamma}_2^+$. Hence $\boldsymbol{\pi}_u^* = \boldsymbol{\pi}_u + 2 \cdot \boldsymbol{\beta} = (6, 4)$ and $\boldsymbol{\pi}_u + 2 \cdot \boldsymbol{\beta}$ has the minimum distance (note that $\Delta(\boldsymbol{\pi}_u + 2 \cdot \boldsymbol{\beta}) = 8$) among all points in $\boldsymbol{\Pi}_u$.

Theorem 3.2.6. *Given the L-shape $(2\ell, h, 2p, n)$ of the* PSEUDOMDD *and an arbitrary node u of a mixed chordal ring network, the location of an optimal copy $\boldsymbol{\pi}_u^*$ of $\boldsymbol{\pi}_u$ can be computed in constant time if $\boldsymbol{\pi}_u$ is known in advance.*

Proof. Suppose $\boldsymbol{\pi}_u = (x, y)$. Our main issue is to determine the part $\boldsymbol{M}_i$ which $\boldsymbol{\pi}_u$ belongs to. After that, we can apply Theorem 3.2.5 to obtain $\boldsymbol{\pi}_u^*$ in constant time. Clearly, we can determine whether $\boldsymbol{\pi}_u$ belongs to $\boldsymbol{\Gamma}^+$ or $\boldsymbol{\Gamma}^-$ by comparing $x - 2y$ with 0. By (3.2.1) and (3.2.2), if $x - 2y - 2h < 0$ or $x - 2y + 2l >= 0$ then we have $i = 0$. Otherwise, if $\boldsymbol{\pi}_u \in \boldsymbol{\Gamma}^+$, then i can be determined by

$$i = \left\lfloor \frac{x - 2y - 2h}{2h + 2p + 1} \right\rfloor ;$$

if $\boldsymbol{\pi}_u \in \boldsymbol{\Gamma}^-$, then i can be determined by

$$i = \left\lceil \frac{-x + 2y - 2l}{2l + 2n} \right\rceil .$$

Thus, given the L-shape of the PSEUDOMDD of a MCRN, one can determine the part $\boldsymbol{M}_i$ which $\boldsymbol{\pi}_u$ belongs to in constant time. Obviously, the overall process can be done in constant time, and therefore we have this result. □

3.3 The Minimum Distance Diagrams of the Mixed Chordal Ring Network

Since $MCR(N; s, w)$ satisfies the even-odd-vertex-transitive property, there are two minimum distance diagrams associated with an MCRN: MDD_0 and MDD_1. The formal definition is given as follows.

Definition. The *minimum distance diagram* MDD_λ, $\lambda \in \{0, 1\}$ of $MCR(N; s, w)$ is an array with node λ at point $(0, 0)$ and node u at point $\boldsymbol{z} = (x, y)$ satisfying $l(\boldsymbol{z}) \equiv u \pmod N$ with $x \geq 0, y \geq 0$ and the minimum $\Delta(\boldsymbol{z})$, where label $l(\boldsymbol{z})$ is defined in (3.1.1).

By Theorem 2.3.4, the MDD_1 of $MCR(N; s, w)$ can be constructed by considering MDD_0 of $MCR(N; s, N - w)$. By the renaming function in (2.3.1), in the following, we can focus on the problem of converting a PSEUDOMDD into MDD_0. For convenience, denote the given PSEUDOMDD by symbol $\boldsymbol{M}$. Suppose $\boldsymbol{M}$ has an L-shape $(2\ell, h, 2p, n)$. Let κ and γ be defined by

$$\kappa \stackrel{\text{def}}{=} \begin{cases} \left\lceil \frac{\ell-h}{h+p} \right\rceil & \text{if } \ell \geq h; \\ \left\lceil \frac{h-\ell-1}{\ell+n} \right\rceil & \text{if } \ell < h. \end{cases}$$

$$\gamma \stackrel{\text{def}}{=} \begin{cases} (\ell-h) \bmod (h+p) & \text{if } \ell \geq h \\ (h-\ell-1) \bmod (\ell+n) & \text{if } \ell < h \end{cases}$$

The intersections of $\boldsymbol{\Gamma}_i^+$ (or $\boldsymbol{\Gamma}_i^-$) and $\boldsymbol{M}$ partition $\boldsymbol{M}$ into $\kappa+1$ parts: Let

$$\boldsymbol{M}_0 = \boldsymbol{M} \cap (\boldsymbol{\Gamma}_0^+ \cup \boldsymbol{\Gamma}_0^-),$$

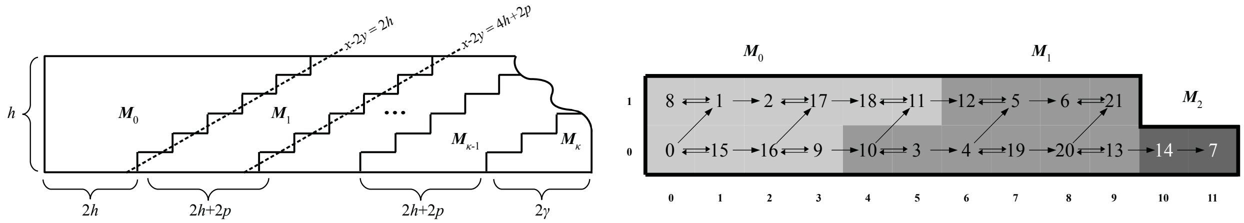
and for $1 \leq i \leq \kappa$,

$$\boldsymbol{M}_i = \begin{cases} \boldsymbol{M} \cap \boldsymbol{\Gamma}_i^+ & \text{if } \ell \geq h \\ \boldsymbol{M} \cap \boldsymbol{\Gamma}_i^- & \text{if } \ell < h \end{cases}$$

Example. Take the PSEUDOMDD in Fig. 3.8(b) for an illustration. Clearly $\boldsymbol{M}$ has an L-shape $(12, 2, 2, 1)$ and $\boldsymbol{M} = \{ (x, y) \mid 0 \leq x < 12, 0 \leq y < 2 \text{ and either } x < 10 \text{ or } y < 1 \}$. By (3.3) and (3.3), we have $\kappa = 2$ and $\gamma = 2$. Thus partition $\boldsymbol{M}$ into three parts as follows: $\boldsymbol{M}_0 = \{(0,0), (1,0), (2,0), (3,0), (0,1), (1,1), (2,1), (3,1), (4,1), (5,1)\}$; $\boldsymbol{M}_1 = \{(4,0), (5,0), (6,0), (7,0), (8,0), (9,0), (6,1), (7,1), (8,1), (9,1)\}$; $\boldsymbol{M}_2 = \{(10,0), (11,0)\}$.

Theorem 3.3.1. *Suppose the* PSEUDOMDD *of* $MCR(N; s, w)$ *has an L-shape* $(2\ell, h, 2p, n)$. *Then* MDD_0 *can be constructed by replacing* $\boldsymbol{\pi}_u$ *with* $\boldsymbol{\pi}_u + i \cdot \boldsymbol{\beta}$ *(resp.,* $\boldsymbol{\pi}_u + i \cdot \boldsymbol{\alpha}$*) for each point* $\boldsymbol{\pi}_u$ *in* $\boldsymbol{M}_i, 0 \leq i \leq \kappa$ *if* $\ell \geq h$ *(resp.,* $\ell < h$*).*

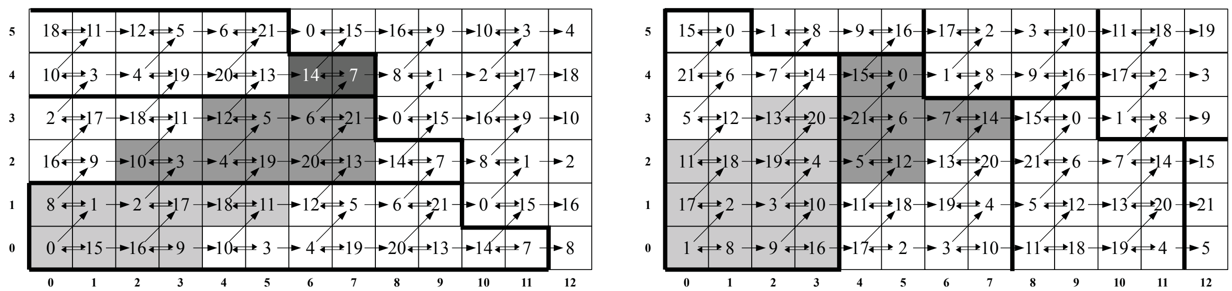
Proof. If $\ell > h$ (resp., $\ell < h$), then no point of the PSEUDOMDD is inside $\boldsymbol{\Gamma}_i^-$ (resp., $\boldsymbol{\Gamma}_i^+$) for $i \geq 1$. If $\ell = h$, then all points of the PSEUDOMDD is inside $\boldsymbol{\Gamma}_0^+ \cup \boldsymbol{\Gamma}_0^-$. This theorem



(a) The partition of PSEUDOMDD. The case of $\ell \geq h$.

(b) The partition of PSEUDOMDD of $MCR(22;1,7)$.

Figure 3.8: The partition of the PSEUDOMDD.



(a) MDD_0 of $MCR(22;1,7)$.

(b) MDD_1 of $MCR(22;1,7)$.

Figure 3.9: The MDD_0 and MDD_1 of $MCR(22;1,7)$.

comes from Theorem 3.2.5 obviously. □

Figs. 3.9(a) and 3.9(b) illustrate the MDD_0 and MDD_1 of $MCR(22;1,7)$, respectively. Note that since a PSEUDOMDD always tessellates the plane and all points in $\boldsymbol{M}_i$ move i steps in $\boldsymbol{R}_u^\alpha$ or in $\boldsymbol{R}_u^\beta$ to find their optimal copy, we conclude that MDD_0 and MDD_1 of $MCR(N;s,w)$ can be obtained by "reassembling" the PSEUDOMDDs of $MCR(N;s,w)$ and $MCR(N;s,N-w)$, respectively, according to the rules stated in Theorem 3.3.1. Figs 3.8(b) and 3.9(a) illustrate the reassembling of the PSEUDOMDD of $MCR(22;1,7)$ into MDD_0. As a conclusion, the MDD's of an MCRN can tessellate the plane; see Fig. 3.10.

Theorem 3.3.2. *The MDD_λ, $\lambda \in \{0,1\}$, of $MCR(N;s,w)$ can tessellate the plane.*

Now we characterize the shape of MDD_λ. Assume the given PSEUDOMDD has an L-shape $(2\ell,h,2p,n)$. According to Theorem 3.3.1, the shape of MDD_λ of the MCRN is shown in Fig. 3.11. The dashed-curves in this figure indicate the remainder part of the MDD_λ as it

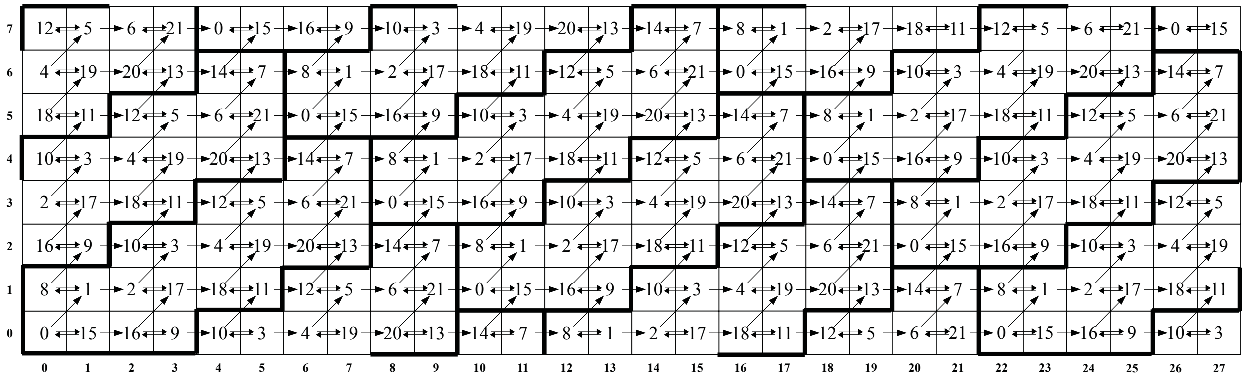
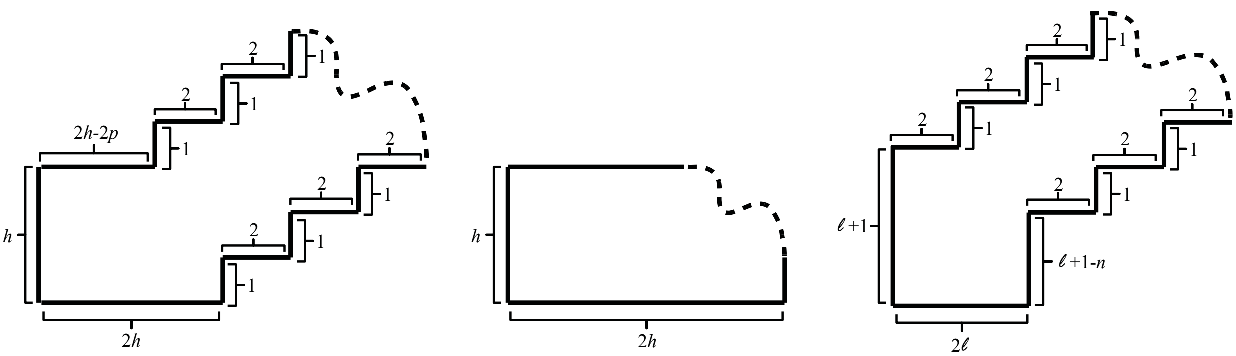


Figure 3.10: The tessellation of the plane formed by the MDD_0 of $MCR(22; 1, 7)$.



(a) $\ell > h$ (b) $\ell = h$ (c) $\ell < h$

Figure 3.11: The dimension of MDD_λ.

depends on the given L-shape. The only situation that the given PSEUDOMDD is eventually an MDD is the case of $\ell = h$.

3.4 MDD Construction Algorithm for MCRNs

By the discussion in Sections 3.1 to 3.3, we present our algorithm, called *MCRN-MDD-Algorithm*, to construct MDD_0 and MDD_1 of a given $MCR(N; s, w)$. This algorithm works as follows. For the construction of MDD_λ, $\lambda \in \{0, 1\}$, the algorithm first computes the L-shape of PSEUDOMDD of $MCR(N; s, w)$ and $MCR(N; s, N-w)$. Once we have the L-shape of the PSEUDOMDD, the MDD_λ can be constructed by sequentially examining (row-by-row

Algorithm 1 ***MCRN-MDD-Algorithm***

Input: N, s, w.
Output: MDD_0 and MDD_1 of $MCR(N; s, w)$.

1: **for** $\lambda \leftarrow 0$ **to** 1 **do**
2:     **if** $\lambda = 1$ **then** ▷ Consider $MCR(N; s, N-w)$
3:         $w \leftarrow (N-w) \bmod N$
4:     **end if**
5:     $(2\ell_\lambda, h_\lambda, 2p_\lambda, n_\lambda) \leftarrow$ the L-shape of the PSEUDOMDD of $MCR(N; s, w)$
6:     $\boldsymbol{\alpha} \leftarrow (2\ell_\lambda, -n_\lambda), \boldsymbol{\beta} \leftarrow (-2p_\lambda, h_\lambda)$
7:     **for** $x \leftarrow 0$ **to** $2\ell_\lambda - 1$ **do** ▷ Row-by-row fashion
8:         **for** $y \leftarrow 0$ **to** $h_\lambda - 1$ **do**
9:             **if** $x < 2\ell_\lambda - 2p_\lambda$ or $y < h_\lambda - n_\lambda$ **then** ▷ (x, y) is inside PSEUDOMDD
10:                 $u \leftarrow \left(\left(\left\lfloor \frac{x}{2} \right\rfloor + y\right) s - \left(\left\lceil \frac{x}{2} \right\rceil - y\right) w\right) \bmod N$ ▷ The labeling function (3.1.2)
11:                 $(x^*, y^*) \leftarrow$ an optimal copy of (x, y) ▷ By Theorems 3.2.5 and 3.2.6
12:                 $\text{MDD}_\lambda[u] \leftarrow (x^*, y^*)$
13:             **end if**
14:         **end for**
15:     **end for**
16: **end for**

or column-by-column or diagonal-by-diagonal) each point inside the PSEUDOMDD. After that, for each point (x, y) in the PSEUDOMDD, the algorithm determine the node u such that (x, y) has label u. Then, by applying Theorems 3.2.5 and 3.2.6, we can obtain the location of an optimal copy of $\boldsymbol{\pi}_u$.

Theorem 3.4.1. *MCRN-MDD-Algorithm is correct, and it takes* $\Theta(N)$*-time.*

Proof. The correctness comes from the fact that the PSEUDOMDD contains every node of a MCRN exactly once and Theorem 3.2.5. Now we analyze the time complexity. It takes $O(\log N)$-time (by using the Cheng-Hwang-Algorithm [38]) to derive the L-shapes of PSEUDOMDD in line 5. By Theorem 3.2.6, line 11 can be done in constant time. Since lines 7-15 examine every point of the PSEUDOMDD exactly once, it takes $O(2\ell h - 2pn) = O(N)$-time in lines 7-15. Therefore, the total time needed to construct MDD_λ is $O(N)$. Since it takes $\Omega(N)$-time to construct MDD_λ, we can construct MDD_λ in time $\Theta(N)$. $\square$

Chapter 4

The Diameter

In this section, we consider the problem of determining the diameter of a given MCRN. This problem can be solved straightforwardly by first constructing MDD_0 and MDD_1 of $MCR(N; s, w)$, and then, finding the point in MDD_0 and MDD_1 that has the maximum distance. However, this approach takes $\Omega(N)$-time, which is exponential in the input size (each of the three integers N, s, w takes at most $\log N$ bits). Instead of constructing MDD_0 and MDD_1 first, we give an efficient algorithm to compute the diameter of an MCRN that takes $O(\log N)$ worst-case time, which is polynomial in the input size. Results derived from this chapter have been submitted to [43].

4.1 The **MAXDIST** Subroutine

Let $d(N; s, w)$ denote the diameter of $MCR(N; s, w)$. Let d_λ, $\lambda \in \{ 0, 1 \}$, denote the maximum distance over all points in MDD_λ, i.e., $d_\lambda = \max \{ d(\lambda, u) \mid u \in \{ 0, 1, \ldots, N-1 \} \}$, where $d(u, v)$ denote the distance between nodes u and v. Clearly

$$d(N; s, w) = \max \{ d_0, d_1 \} .$$

Suppose for some u, point $\boldsymbol{\pi}_u^*$ in MDD_λ achieves the distance d_λ. Then $d_\lambda = \Delta(\boldsymbol{\pi}_u^*)$. Given the L-shape of a PSEUDOMDD, the subroutine MAXDIST shown in Algorithm 2 can compute d_λ in constant time.

Theorem 4.1.1. *The subroutine MAXDIST is correct and takes constant time to compute d_λ.*

Proof. The time complexity is easy to see and we now verify the correctness. Given a PSEUDOMDD with L-shape $(2\ell, h, 2p, n)$, there are three cases: (1) $\ell = h$, (2) $\ell > h$, and (3) $\ell < h$. We will only show the details of the first two cases; the case of $\ell < h$ is similar to the case of $\ell > h$ and is omitted. Suppose $\ell = h$. By Theorem 3.3.1, we know that this PSEUDOMDD is an MDD. That is, $\boldsymbol{\pi}_u = \boldsymbol{\pi}_u^*$ for all nodes u. Clearly, if $\boldsymbol{\pi}_u \in \boldsymbol{\Gamma}^+$, then $\Delta(\boldsymbol{\pi}_u) \leq 2\ell - 1$; if $\boldsymbol{\pi}_u \in \boldsymbol{\Gamma}^-$, then $\Delta(\boldsymbol{\pi}_u) \leq 2(h-1)$. Choose $\boldsymbol{\pi} = (2\ell - 1, 0)$, then $d_\lambda = \Delta(\boldsymbol{\pi}) = 2\ell - 1 = 2h - 1$. Since MAXDIST returns $2h - 1$ in line 10, MAXDIST is correct.

Now consider the case of $\ell > h$. By Lemma 3.1.2, we only need to consider the *corner point* $\boldsymbol{\pi}_u^*$, which is at the uppermost of the rightmost points in MDD_λ if $\boldsymbol{\pi}_u^* \in \boldsymbol{\Gamma}^+$, and is at the point left to the rightmost of the uppermost points in MDD_λ if $\boldsymbol{\pi}_u^* \in \boldsymbol{\Gamma}^-$ (since the rightmost of the uppermost point in MDD_λ has odd x-coordinate, the point left to this point has a larger distance by (3.1.3)). Let the *pseudo corner point* $\boldsymbol{\pi}_u$ be the point in the PSEUDOMDD such that $\boldsymbol{\pi}_u^*$ (an optimal copy of $\boldsymbol{\pi}_u$ by Theorem 3.2.5) is a corner point. Note that since MDD_λ is a reassembling of the PSEUDOMDD, the pseudo corner points must occur in $\boldsymbol{M}_{\kappa-1}$ or $\boldsymbol{M}_\kappa$. For convenience, set $m = \ell - p, q = h - n$. According to γ, we have the following subcases; see Fig. 4.1.

Subcase 1: $\gamma = 0$. By (3.3), we have $\ell - h = \kappa(h + p)$. The pseudo corner points are $\boldsymbol{\pi}_0 = (2\ell - 1, q - 1)$ and $\boldsymbol{\pi}_1 = (2m - 2, h - 1)$. Then $\boldsymbol{\pi}_0 \in \boldsymbol{M}_\kappa$ and $\boldsymbol{\pi}_1 \in \boldsymbol{M}_\kappa$. By Theorem 3.2.5 and by (3.1.3), $\Delta(\boldsymbol{\pi}_0^*) = 2(\kappa+1)h - 1$ and $\Delta(\boldsymbol{\pi}_1^*) = 2(\kappa+1)h - 2$. Therefore

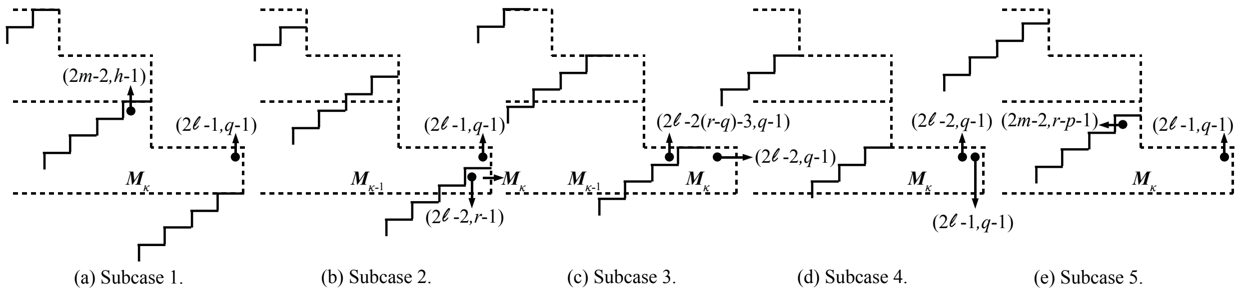


Figure 4.1: The five subcases of the case $\ell > h$.

$d_\lambda = \Delta(\boldsymbol{\pi}_0^*)$. Since MAXDIST returns $2(\kappa+1)h-1$ in line 10, MAXDIST is correct.

In the following, we will assume $\gamma > 0$. By (3.3), we have $\ell - h = (\kappa-1)(h+p)+\gamma$.

Subcase 2: $0 < \gamma < h-n$. The pseudo corner points are $\boldsymbol{\pi}_0 = (2\ell-1, q-1)$ and $\boldsymbol{\pi}_1 = (2\ell-2, \gamma-1)$. Then $\boldsymbol{\pi}_0 \in \boldsymbol{M}_\kappa$ and $\boldsymbol{\pi}_1 \in \boldsymbol{M}_{\kappa-1}$. By Theorem 3.2.5 and (3.1.3), $\Delta(\boldsymbol{\pi}_0^*) = 2\kappa h + 2\gamma - 1$ and $\Delta(\boldsymbol{\pi}_1^*) = 2\kappa h + 2\gamma - 2$. Thus $d_\lambda = \Delta(\boldsymbol{\pi}_0^*)$. Since MAXDIST returns $2\kappa h + 2\gamma - 1$ in line 12, MAXDIST is correct.

Subcase 3: $h-n \le \gamma \le h-n+p-2$. The pseudo corner points are $\boldsymbol{\pi}_0 = (2\ell-2, q-1)$ and $\boldsymbol{\pi}_1 = \Big(2\ell-2(\gamma-q)-3, q-1\Big)$. Then $\boldsymbol{\pi}_0 \in \boldsymbol{M}_\kappa$ and $\boldsymbol{\pi}_1 \in \boldsymbol{M}_{\kappa-1}$. By Theorem 3.2.5 and by (3.1.3), $\Delta(\boldsymbol{\pi}_0^*) = 2(\kappa+1)h-2n-3$ and $\Delta(\boldsymbol{\pi}_1^*) = 2(\kappa+1)h-2n-2$. Thus $d_\gamma = \Delta(\boldsymbol{\pi}_1)$. Since MAXDIST returns $2(\kappa+1)h-2n-2$ in line 14, MAXDIST is correct.

Subcase 4: $\gamma = h-n+p-1$ or $h-n+p$. The pseudo corner points are $\boldsymbol{\pi}_0 = (2\ell-2, q-1)$, $\boldsymbol{\pi}_1 = (2\ell-1, q-1)$. Then $\boldsymbol{\pi}_0 \in \boldsymbol{M}_\kappa$ and $\boldsymbol{\pi}_1 \in \boldsymbol{M}_\kappa$. By Theorem 3.2.5 and by (3.1.3), $\Delta(\boldsymbol{\pi}_0^*) = 2(\kappa+1)h-2n-2$ and

$$\Delta(\boldsymbol{\pi}_1^*) = \begin{cases} 2(\kappa+1)h-2n-3 & \text{if } r = q+p-1, \\ 2(\kappa+1)h-2n-1 & \text{if } r = q+p. \end{cases}$$

Thus $d_\lambda = \Delta(\boldsymbol{\pi}_0)$ if $\gamma = q+p-1$ and $d_\lambda = \Delta(\boldsymbol{\pi}_1)$ if $\gamma = q+p$. Since MAXDIST returns $2(\kappa+1)h-2n-2$ in line 14 if $\gamma = q+p-1$ and returns $2(\kappa+1)h-2n-1$ in line 14 if $r = q+p$, MAXDIST is correct.

Subcase 5: $h - n + p < \gamma < h + p$. The pseudo corner points are $\boldsymbol{\pi}_0 = (2\ell - 1, q - 1)$ and $\boldsymbol{\pi}_1 = (2m - 2, \gamma - p - 1)$. Then $\boldsymbol{\pi}_0 \in \boldsymbol{M}_\kappa$ and $\boldsymbol{\pi}_1 \in \boldsymbol{M}_\kappa$. By Theorem 3.2.5 and by (3.1.3), $\Delta(\boldsymbol{\pi}_0^*) = 2\kappa h - 2p + 2\gamma - 1$ and $\Delta(\boldsymbol{\pi}_1^*) = 2\kappa h - 2p + 2\gamma - 2$. Thus $d_\lambda = \Delta(\boldsymbol{\pi}_0)$. Since MAXDIST returns $2\kappa h - 2p + 2\gamma - 1$ in line 16, MAXDIST is correct.

Note that when $\gamma = h - n + p$, the value $2(\kappa + 1)h - 2n - 1$ is equal to $2\kappa h - 2p + 2\gamma - 1$. Thus we can combine the cases of $\gamma = h - n + p$ and $h - n + p < \gamma < h + p$. From the above discussions, MAXDIST is correct. □

4.2 An Efficient Diameter-computing Algorithm

We present our diameter-computing algorithm, called *MCRN-Diameter-Algorithm*, in Algorithm 2. We now prove and analyze this algorithm.

Theorem 4.2.1. *MCRN-Diameter-Algorithm is correct and takes* $O(\log N)$*-time.*

Proof. The values of d_0 and d_1 can be obtained by finding the maximum distance over all points in MDD_0 and MDD_1, respectively. Thus MDD_0 and MDD_1 can be obtained from the PSEUDOMDD's of $MCR(N; s, w)$ and $MCR(N; s, N - w)$, respectively. Consequently, the correctness of Algorithm 2 follows from the correctness of subroutine MAXDIST obviously (see Theorem 4.1.1). We now analyze the time complexity of MCRN-Diameter-Algorithm. Lines 3-5 take only constant time as the subroutine MAXDIST takes only constant time. For lines 1-2, the algorithm proposed in [21] can be used to obtain the L-shape of a DLN and therefore the L-shape of the PSEUDOMDD. Since the algorithm in [21] takes $O(\log N)$-time, lines 1-2 takes $O(\log N)$-time. As a consequence, MCRN-Diameter-Algorithm takes $O(\log N)$-time. □

Algorithm 2 ***MCRN-Diameter-Algorithm***

Input: N, s, w.
Output: The diameter of $MCR(N; s, w)$.

$(2\ell_0, h_0, 2p_0, n_0) \leftarrow$ the L-shape of the PSEUDOMDD of $MCR(N; s, w)$
$(2\ell_1, h_1, 2p_1, n_1) \leftarrow$ the L-shape of the PSEUDOMDD of $MCR(N; s, N - w)$
$d_0 \leftarrow \mathsf{MAXDIST}(2\ell_0, h_0, 2p_0, n_0)$
$d_1 \leftarrow \mathsf{MAXDIST}(2\ell_1, h_1, 2p_1, n_1)$
return $\max\{ d_0, d_1 \}$

Subroutine $\mathsf{MAXDIST}(2\ell, h, 2p, n)$
if $\ell \geq h$ **then**
    $\kappa \leftarrow \left\lceil \frac{\ell - h}{h + p} \right\rceil$
    $\gamma \leftarrow (\ell - h) \bmod (h + p)$
    if $\gamma = 0$ **then**
        return $2(\kappa + 1)h - 1$
    else if $0 < \gamma < h - n$ **then**
        return $2\kappa h + 2\gamma - 1$
    else if $h - n \leq \gamma < h - n + p$ **then**
        return $2(\kappa + 1)h - 2n - 2$
    else
        return $2\kappa h - 2p + 2\gamma - 1$
    end if
else ▷ the $\ell < h$ case
    $\kappa \leftarrow \left\lceil \frac{h - \ell - 1}{\ell + n} \right\rceil$
    $\gamma \leftarrow (h - \ell - 1) \bmod (\ell + n)$
    if $\gamma = 0$ **then**
        return $2(\kappa + 1)\ell$
    else if $0 < \gamma < \ell - p$ **then**
        return $2\kappa\ell + 2\gamma$
    else if $\ell - p \leq \gamma < \ell - p + n$ **then**
        return $2(\kappa + 1)\ell - 2p - 1$
    else
        return $2\kappa\ell - 2n + 2\gamma$
    end if
end if

Example. Take $MCR(22;1,7)$ as an example. The L-shapes of the PSEUDOMDD of $MCR(22;1,7)$ and $MCR(22;1,22-7)$ are (12,2,2,1) and (4,6,2,1), respectively. First consider d_0. We have $\kappa = 2, \gamma = 1$ and MAXDIST returns $d_0 = 8$. Now consider d_1. We have $\kappa = 1, \gamma = 0$ and MAXDIST returns $d_1 = 8$. As a conclusion, $d(22;1,7) = 8$.

Example. Take $MCR(12;3,5)$ as another example. The L-shapes of the PSEUDOMDD of $MCR(12;3,5)$ and $MCR(12;3,12-5)$ are (4,3,0,1) and (6,2,2,0), respectively. First consider d_0. We have $\kappa = 0, \gamma = 0$ and MAXDIST returns $d_0 = 4$. Now consider d_1. We have $\kappa = 1, \gamma = 1$ and MAXDIST returns $d_1 = 5$. Thus, $d(12;3,5) = 5$.

Remark 4.2.2. By the definition of a PSEUDOMDD, the L-shapes of the PSEUDOMDD of $MCR(N;s,w)$ and $MCR(N;s,N-w)$ can be obtained by first deriving the L-shapes of $DL\left(\frac{N}{2};\frac{s-w}{2},\frac{s+w}{2}\right)$ and $DL\left(\frac{N}{2};\frac{s+w}{2},\frac{s-w}{2}\right)$, respectively, and then doubling the lengths of ℓ and p. Note that if $DL(N;s_1,s_2)$ has an L-shape (ℓ,h,p,n), then there exists an MDD of $DL(N;s_2,s_1)$ such that the MDD has an L-shape (h,l,n,p). This is because there is a one-to-one correspondence between the node at point (x,y) of $\mathbb{Z}^+ \times \mathbb{Z}^+$ formed by the vertex set of $DL(N;s_1,s_2)$ and the node at point (y,x) of $\mathbb{Z}^+ \times \mathbb{Z}^+$ formed by the vertex set $DL(N;s_2,s_1)$. Therefore, if $(2\ell_0,h_0,2p_0,n_0)$ is the L-shape of the PSEUDOMDD of $MCR(N;s,w)$, then $(2\ell_1,h_1,2p_1,n_1)$ can be set as $(2h_0,\ell_0,2n_0,p_0)$ in line 2 of MCRN-Diameter-Algorithm.

Remark 4.2.3. The computation of MCRN-Diameter-Algorithm highly relies on the computation of DLN's L-shape. To the best of our knowledge, the fastest algorithm that can compute the L-shape of a DLN has the same time complexity as MCRN-Diameter-Algorithm, which is $O(\log N)$-time. Therefore, any algorithm that can compute the DLN's L-shape in $o(\log N)$-time can also improve the time complexity of MCRN-Diameter-Algorithm.

Remark 4.2.4. We test MCRN-Diameter-Algorithm for a considerable range of N's ($N = 6, 8, \ldots, 10004$, a total of 5000 N's) with all possible parameters s and w. Compared with the naive diameter-computing algorithm, Bread-First-Search (BFS), all results of MCRN-Diameter-Algorithm match with the BFS results.

Chapter 5

Optimal Mixed Chordal Ring Networks

Results derived from this chapter have been published in [44]. Recall that $D_{MCR}(N)$ denotes the smallest diameter among all MCRNs with N nodes and $d(N; s, w)$ denotes the diameter of $MCR(N; s, w)$. One of the most important and fundamental optimization problem in designing interconnection networks is, for a given number of nodes N, how to find an *optimal network* with the smallest diameter and to give the construction of such a network. Specifically, given an N, we are interested in finding $D_{MCR}(N)$ and in finding $MCR(N; s, w)$ with $d(N; s, w) = D_{MCR}(N)$. $MCR(N; s, w)$ is said to be *optimal* if $d(N; s, w) = D_{MCR}(N)$.

However, finding optimal MCRNs is a very difficult problem. The difficulty is due to the fact that the diameter of MCRNs does not increase monotonically with N. For example, $D_{MCR}(16) = 6 > 5 = D_{MCR}(18)$ and $D_{MCR}(44) = 10 > 9 = D_{MCR}(46)$. Thus, there is no closed formula for $D_{MCR}(N)$ up to now. Double-loop networks also have the same difficulty; see [2, 32, 33]. By taking another approach, we aim at looking for bounds on $D_{MCR}(N)$.

5.1 Lower Bound

Given a $MCR(N; s, w)$, let n_k denote the number of additional nodes that node 0 can reach in k moves. Clearly, $n_0 = 0$, $n_1 = 2$ and $n_2 = 3$. Chen et al. [20] had proven that

$$n_k \leq n_{k-1} + 1 \quad \text{for } 2 \leq k \leq d(N; s, w). \tag{5.1.1}$$

In other words, for $k \geq 2$, the number of additional nodes that node 0 can reach at the kth move increases by at most 1. We now have the following result.

Theorem 5.1.1. *$D_{MCR}(N) \geq \lceil \sqrt{2N} - 3/2 \rceil$ and this bound is tight.*

Proof. By (5.1.1),

$$N \leq \sum_{k=0}^{d(N;s,w)} (k+1) = \frac{(d(N;s,w)+2)(d(N;s,w)+1)}{2}.$$

Therefore, $(d(N; s, w))^2 + 3d(N; s, w) + (2-2N) \geq 0$. Since $d(N; s, w)$ is positive, $d(N; s, w) \geq (\sqrt{8N+1} - 3)/2 > \sqrt{2N} - 3/2$. Since $d(N; s, w)$ is an integer, $d(N; s, w) \geq \lceil \sqrt{2N} - 3/2 \rceil$. This bound is tight since $d(8; 1, 3) = 3 \geq D_{MCR}(8) \geq \lceil \sqrt{2 \cdot 8} - 3/2 \rceil = 3$. □

5.2 Upper Bounds

Although Chen et al [20] proposed an upper bound on $D_{MCR}(N)$ (see Theorem 1.4.1), we find that there exist some erroneous cases in their proof. We first indicate the erroneous part in their proof as follows. Consider $N = 38$. To obtain an upper bound of $D_{MCR}(38)$, Chen *et al.* [20] will use $MCR(38; 7, 5)$ and embed $MCR(38; 7, 5)$ into $DL(19; 1, 6)$. For convenience, define $\hat{N}$ to be a function of N as follows:

$$\hat{N} = \left\lceil \sqrt{\frac{N}{2}} \right\rceil. \tag{5.2.1}$$

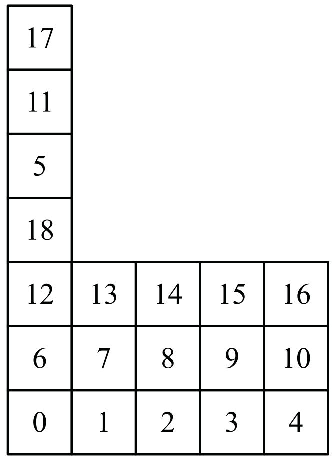


Figure 5.1: A counterexample to the proof of Theorem 1.4.1.

The L-shape of $DL(19; 1, 6)$ has $\ell = 5$ and $h = 7$ (see Fig. 5.1), which has $h > \hat{N} = 6$ and violates

$$\ell \leq \hat{N} \text{ and } h = \hat{N} \tag{5.2.2}$$

needed in the proof of $D_{MCR}(38) \leq \sqrt{2N} + 3$. In fact, we can construct infinite many N's that violates (5.2.2); specifically, let $N = 2(4t^2 + 2t - 1)$ for some positive integer t, then the corresponding DLN of $MCR(N; \hat{N} + 1, \hat{N} - 1)$ has an L-shape $(\ell, h, p, n) = (\hat{N} - 1, \hat{N} + 1, \hat{N} - 2, \hat{N} - 2)$, which clearly violates (5.2.2); see Theorem 4.5 in [20] for more details. Instead of correcting Theorem 1.4.1, in the following we give an improved upper bound on $D_{MCR}(N)$.

The following lemma had been proven in [20] and it follows from the fact that each move in the MDD of $DL(\frac{N}{2}; \frac{s-w}{2}, \frac{s+w}{2})$ corresponds to either one or two moves in $MCR(N; s, w)$ (depending on which node in the supernode we start from).

Lemma 5.2.1. [20] *Suppose $DL(\frac{N}{2}; \frac{s-w}{2}, \frac{s+w}{2})$ has an L-shape (ℓ, h, p, n), then $d(N; s, w) \leq$ $2 \cdot \max\{\ell, h\} - 1$.*

We now obtain an upper bound on $D_{MCR}(N)$. The main idea used in obtaining the upper bound is, for each N, to choose s and w suitably so that the corresponding double-loop network $DL(\frac{N}{2}; \frac{s-w}{2}, \frac{s+w}{2})$ has an L-shape(ℓ, h, p, n) with ℓ and h being as small as possible and to apply Lemma 5.2.1.

According to the parity of $\hat{N}$, define M as follows:

$$M = \begin{cases} \hat{N} & \text{if } \hat{N} \text{ is even,} \\ \hat{N}+1 & \text{if otherwise.} \end{cases} \tag{5.2.3}$$

Lemma 5.2.2. *Suppose* $N \neq 2(4t^2 + 2t - 1)$ *for any positive integer* t *and let* M *be defined as in* (5.2.3)*. Then the* L*-shape*(ℓ, h, p, n) *of* $DL(\frac{N}{2}; 1, M)$ *satisfies* $\ell \leq M$ *and* $h \leq M$*.*

Proof. Consider $\mathbb{N} = \bigcup_{t=0}^{\infty}[4t^2 + 1, 4(t+1)^2]$. Then $\frac{N}{2} \in [4t^2 + 1, 4(t+1)^2]$ for some non-negative integer t. Thus $M = 2t + 2$. Consider the L-shape(ℓ, h, p, n) of $DL(\frac{N}{2}; 1, M)$. Since

$$M \cdot 1 \equiv 1 \cdot M \pmod{\tfrac{N}{2}},$$

cell $(M, 0)$ and cell $(0, 1)$ contain the same node. Since $M > 1$, cell $(M, 0)$ is outside the L-shape. Consequently, $\ell \leq M$. Now let $N_0(t) = [4t^2 + 1, 4t^2 + 2t - 2]$, $N_1(t) = [4t^2 + 2t - 1, 4t^2 + 4t]$, $N_2(t) = [4t^2 + 4t + 1, 4t^2 + 6t + 2]$, and $N_3(t) = [4t^2 + 6t + 3, 4t^2 + 8t + 4]$. Note that $N_0(0)$, $N_1(0)$, and $N_0(1)$ are empty. Then $\mathbb{N} = \bigcup_{t=0}^{\infty}(N_0(t) \cup N_1(t) \cup N_2(t) \cup N_3(t))$. Suppose $\frac{N}{2} \in N_k(t)$, where $0 \leq k \leq 3$. Define $N_k^*(t)$ to be the maximum integer in $N_k(t)$. Clearly, $N_k^*(t) = 4t^2 + 2t - 2 + (2t + 2)k$. Suppose $\frac{N}{2} = N_k^*(t) - j$ for some non-negative integer j. Then $0 \leq j \leq 2t - 3$ if $k = 0$ and $0 \leq j \leq 2t + 1$ if $1 \leq k \leq 3$. Again, consider the L-shape(ℓ, h, p, n) of $DL(\frac{N}{2}; 1, M)$. Since

$$\begin{aligned} j \cdot 1 &= N_k^*(t) - \tfrac{N}{2} \\ &= (4t^2 + 2t - 2 + (2t+2)k) - \tfrac{N}{2} \\ &\equiv (2t - 1 + k)(2t + 2) \pmod{\tfrac{N}{2}} \\ &= (2t - 1 + k)M \pmod{\tfrac{N}{2}}, \end{aligned}$$

cell $(j, 0)$ and cell $(0, 2t-1+k)$ contain the same node. Note that $j \le 2t-1+k$ except when $k = 1$ and $j = 2t + 1$, that is, except when $\frac{N}{2} = 4t^2 + 2t - 1$. Hence if $N \neq 2(4t^2 + 2t - 1)$ for any positive integer t, then cell $(0, 2t - 1 + k)$ is outside the L-shape. Consequently, $h \le 2t - 1 + k \le 2t + 2 = M$. □

Lemma 5.2.3. *Suppose $N = 2(4t^2 + 2t - 1)$ for some positive integer t and let M be defined as in* (5.2.3). *Then the L-shape(ℓ, h, p, n) of $DL(\frac{N}{2}; 2, M - 1)$ satisfies $\ell \le M - 1$ and $h \le M - 1$.*

Proof. Since $N = 2(4t^2 + 2t - 1)$ for some positive integer t, we have $M = 2t + 2$. Consider the L-shape(ℓ, h, p, n) of $DL(\frac{N}{2}; 2, M - 1)$. Since

$$(2t + 1) \cdot 2 \equiv 2 \cdot (2t + 1) \pmod{\tfrac{N}{2}},$$

cell $(2t + 1, 0)$ and cell $(0, 2)$ contain the same node. Since t is a positive integer, we have $2t + 1 > 2$. Thus cell $(2t + 1, 0)$ is outside the L-shape. Consequently, $\ell \le 2t + 1 \le M - 1$. Similarly, since

$$(t + 1) \cdot 2 \equiv (2t + 1)(2t + 1) \pmod{\tfrac{N}{2}},$$

cell $(t + 1, 0)$ and cell $(0, 2t + 1)$ contain the same node. Clearly, $2t + 1 > t + 1$ for $t > 0$; thus cell $(0, 2t + 1)$ is outside the L-shape. Thus $h \le 2t + 1 \le M - 1$. □

Lemma 5.2.4. *Let M be defined as in* (5.2.3)*. Then:*

1. *If $N \neq 2(4t^2+2t-1)$ for any positive integer t, then $d(N; M+1, M-1) \leq 2M-1$.*

2. *If $N = 2(4t^2+2t-1)$ for some positive integer t, then $d(N; M+1, M-3) \leq 2M-3$.*

Proof. Consider the first statement. It is not difficult to verify that both $M+1$ and $M-1$ are positive odd integers and $\gcd(N, M+1, M-1) = 1$. Thus $MCR(N; M+1, M-1)$ is a valid mixed chordal ring network. Since we can embed $MCR(N; M+1, M-1)$ into $DL(\frac{N}{2}; 1, M)$, this statement follows directly from Lemmas 5.2.1 and 5.2.2. The second statement can be proven similarly except that Lemma 5.2.2 is replaced with Lemma 5.2.3. □

Theorem 5.2.5. *Let $\hat{N}$ be defined as in* (5.2.1)*.*

1. *If $\hat{N}$ is even, then $D_{MCR}(N) \leq 2\lceil\sqrt{N/2}\,\rceil - 1$.*

2. *If $\hat{N}$ is odd and $N = 2(4t^2+2t-1)$ for some positive integer t, then $D_{MCR}(N) \leq 2\lceil\sqrt{N/2}\,\rceil - 1$.*

3. *If $\hat{N}$ is odd and $N \neq 2(4t^2+2t-1)$ for any positive integer t, then $D_{MCR}(N) \leq 2\lceil\sqrt{N/2}\,\rceil + 1$.*

Moreover, these bounds are tight.

Proof. Note that if $N = 2(4t^2+2t-1)$ for some positive integer t, then $\hat{N}$ is odd. Thus if $\hat{N}$ is even, then $N \neq 2(4t^2+2t-1)$ for any positive integer t; consequently, $M = \hat{N}$. If $\hat{N}$ is odd and $N = 2(4t^2+2t-1)$ for some positive integer t, then $M = \hat{N}+1$. If $\hat{N}$ is odd and $N \neq 2(4t^2+2t-1)$ for any positive integer t, then $M = \hat{N}+1$. Statements 1, 2 and 3 in this theorem now follow from Lemma 5.2.4. By the aid of a computer program, we obtain $D_{MCR}(20) = 7$, $D_{MCR}(38) = 9$ and $D_{MCR}(48) = 11$. Thus the bound in statement 1 is tight since $D_{MCR}(20) = 7$ and $2\lceil\sqrt{20/2}\,\rceil - 1 = 7$. The bound in statement 2 is tight

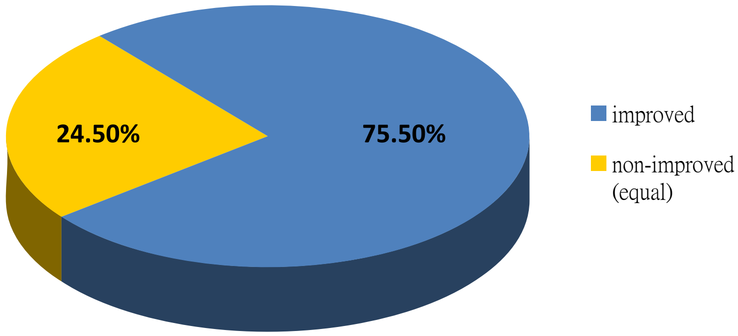


Figure 5.2: The improved ratio of our upper bound as compared to the previous upper bound for $N = 6, 8, 10, \ldots, 10004$ (total 5000 N's).

since $D_{MCR}(38) = 9$ and $2\lceil\sqrt{38/2}\,\rceil - 1 = 9$. Similarly, the bound in statement 3 is tight since $D_{MCR}(48) = 11$ and $2\lceil\sqrt{48/2}\,\rceil + 1 = 11$. □

Remark 5.2.6. The previous upper bound on $D_{MCR}(N)$ is $\sqrt{2N} + 3$ [20]. Since $\sqrt{2N} + 3$ is served as an upper bound, we replace it with $\lfloor\sqrt{2N}+3\rfloor$. The largest upper bound in Theorem 5.2.5 is $2\lceil\sqrt{N/2}\,\rceil+1$ and it is always no larger than $\lfloor\sqrt{2N}+3\rfloor$. To see how good our upper bound $2\lceil\sqrt{N/2}\,\rceil+1$ is, we use a computer to obtain results for $N = 6, 8, 10, \ldots, 10004$. Among these 5000 N's, for 3775 (about 75.50%) out of them, our upper bound $2\lceil\sqrt{N/2}\,\rceil+1$ improves the previous upper bound $\lfloor\sqrt{2N}+3\rfloor$; see Fig. 5.2.

5.3 Optimal Mixed Chordal Ring Networks

It should be noticed that the upper bound $2\lceil\sqrt{N/2}\,\rceil - 1$ in Theorem 5.2.5 is no larger than the upper bound $\lceil\sqrt{2N}\,\rceil + 1$ in Theorem 5.2.5 and is very close to the lower bound $\lceil\sqrt{2N} - 3/2\rceil$ in Theorem 5.1.1. In the following, we show that there exist infinite number of N's such that the upper bound $2\lceil\sqrt{N/2}\,\rceil-1$ matches the lower bound $\lceil\sqrt{2N} - 3/2\rceil$; in other words, we determine the exact value of $D_{MCR}(N)$ for these N's.

Theorem 5.3.1. *Suppose* $N = 2(4t^2 - t + k)$ *for some positive integers* t *and* k*, where* $1 \leq k \leq t$*. Then*

$$D_{MCR}(N) = 2\lceil\sqrt{N/2}\,\rceil - 1.$$

Moreover, $d(N; \lceil\sqrt{N/2}\,\rceil + 1, \lceil\sqrt{N/2}\,\rceil - 1) = D_{MCR}(N)$.

Proof. Suppose $N = 2(4t^2 - t + k)$ for some positive integer t and k, where $1 \leq k \leq t$. Then $2(4t^2 - 4t + 1) < N \leq 2 \cdot 4t^2$; therefore, $M = \hat{N} = \lceil\sqrt{N/2}\,\rceil = 2t$. By Lemma 5.2.4 and Theorem 5.2.5, $D_{MCR}(N) \leq d(N; \lceil\sqrt{N/2}\,\rceil + 1, \lceil\sqrt{N/2}\,\rceil - 1) \leq 2\lceil\sqrt{N/2}\,\rceil - 1$. Since $2(4t^2 - t + \frac{1}{4}) < N \leq 2(4t^2 + t + \frac{1}{4})$, we have $D_{MCR}(N) \geq \lceil\sqrt{2N} - 3/2\,\rceil = 4t - 1 = 2\lceil\sqrt{N/2}\,\rceil - 1$. We now have this theorem. □

The N's that satisfy Theorem 5.3.1 are: 8, 30, 32, 68, 70, 72, 122, ..., and so on. For $N = 6, 8, 10, \ldots, 10004$ (total 5000 N's), about 12.60% out of them satisfy Theorem 5.3.1 and their optimal diameter can be determined by Theorem 5.3.1.

Chapter 6

Routing

In this chapter, we discuss the routing problem in MCRNs. Particularly, routing of node-to-node message with at most a single faulty element in MCRNs is considered. Results derived from Sections 6.1 and 6.2 have been submitted to [43].

A routing algorithm is said to be *optimal* if every message is sent along a shortest path from its source node to its destination node. A fault-tolerant routing algorithm is said to be *optimal* if every message is sent along a shortest path from its source node to its destination node after detecting a faulty element. In Sections 6.1 and 6.2, we design and present two optimal node-to-node shortest path routing algorithms for MCRNs for flexible applications. In Section 6.3, we present an optimal fault-tolerant routing algorithm for MCRNs.

The two optimal node-to-node routing algorithms presented are *shortest-path-based routing* and *dynamic routing*. The shortest-path-based routing algorithm computes the *routing parameter* that can be used to determine a routing path. After an $O(\log N)$-time preprocessing, this algorithm takes $O(\log N)$-time for a source node to compute the routing parameter, and each node on the routing path takes constant time to determine the link (and therefore the node) to send messages according to the routing parameter. It was pointed out in [28] that a shortest-path-based routing algorithm has the advantage that it can often choose from

Table 6.1: Comparing the SP-based routing algorithm with the dynamic routing algorithm.

	SP-Based routing	Dynamic routing
Preprocessing	$O(\log N)$	$O(\log N)$
Computation time for source node	$O(\log N)$	$O(1)$
Computation time for other nodes	$O(1)$	$O(1)$
Number of paths can choose	as many as in the graph	1

a larger set of candidates for the next node to be visited and can avoid potential routing problems that arise from congestion or node/link faults.

On the other hand, for the dynamic routing algorithm, after an $O(\log N)$-time precalculation to determine the network parameters (only computed once and stored them in all nodes), it can route messages using constant time at each node (includes the source node) along the routing path. The routing path is augmented on-the-fly at each routing step. It was pointed out in [36] that the dynamic routing algorithm can be efficiently implemented even if the computation ability of nodes is very limited. Table 6.1 illustrates a comparison between the shortest-path-based (SP-Based) routing algorithm and the dynamic routing algorithm.

Suppose we are sending a message from source node u to destination node v. The even-odd-vertex-transitive property of the MCRN indicates that for even u, a path from u to v in $MCR(N; s, w)$ can be deduced to a path from 0 to $v - u$ in $MCR(N; s, w)$. By the renaming function in (2.3.1), for odd u, nodes u and v of $MCR(N; s, w)$ are mapped to nodes $u + w$ and $v + w$ in $MCR(N; s, N - w)$, respectively. Since $u + w$ is even, a path from $u + w$ to $v + w$ in $MCR(N; s, N - w)$ can be deduced to a path from 0 to $v - u$ in $MCR(N; s, N - w)$. Let $\mu = v - u \pmod N$. As a consequence, a path from u to v in $MCR(N; s, w)$ can be deduced to a path from 0 to μ in $MCR(N; s, w)$ if u is even and a path from 0 to μ in $MCR(N; s, N - w)$ if u is odd. In the rest of this chapter, without loss of generality, we assume that the routing request is from node 0 to node μ ($\neq 0$) in $MCR(N; s, w)$ if the source node is even-numbered, and in $MCR(N; s, N-w)$ if the source node is odd-numbered.

6.1 A Shortest-Path-Based Routing Algorithm

6.1.1 Routing Parameter

A routing path can be viewed as a sequence of links. In the MCRN, there are three types of links: $+s$ link, $+w$ link and $-w$ link. Clearly, a shortest path cannot have both $+w$ and $-w$ links and therefore it consists of either a combination of $+s$ and $+w$ links or a combination of $+s$ and $-w$ links. In addition, two $+w$ links (or $-w$ links) cannot appear consecutively in a shortest path as the parity of the node-number changes at each routing step (because s, w are odd integers). Let $[n_s, n_w]$ denote the *routing parameter* of a path, where n_s and n_w are integers with $n_s \geq 0$ and the sign of the term n_w indicates which w link ($+w$ or $-w$) is used in this path. For example, consider routing in $MCR(22; 1, 7)$ in Fig. 3.9. A shortest path from node 0 to node 18 is $0 \xrightarrow{+s} 1 \xrightarrow{+s} 2 \xrightarrow{-w} 17 \xrightarrow{+s} 18$, which consists of three $+s$ links and one $-w$ links. Thus the routing parameter of this path is therefore $[3, -1]$.

The routing parameter can be appended to the header of a message. Each node in a routing step chooses one of its out-links to deliver the messages, according to the routing parameter, and then updates the routing parameter. The update can be implemented as follows.

Update( )

if $+s$ link is used **then**

$n_s \leftarrow n_s - 1$

else

$n_w \leftarrow n_w - \text{sign}(n_w) \cdot 1$

end if

The routing is done when the routing parameter becomes $[0, 0]$. There are several possible routing algorithms for a given routing parameter. The simplest way are given by greedy algorithms defined by the following rules:

- $\pm w$ *link first*: Use $\pm w$ link whenever it can use $\pm w$ link and $|n_w| \geq 1$.
- $\pm w$ *link last* ($+s$ *link first*): Use $+s$ link whenever $n_s > |n_w|$.

In the rest of this section, we aim at finding the routing parameter of a shortest path from node 0 to destination node μ.

6.1.2 Computing the Routing Parameter

By using the visualization tool established in Chapter 3, the main steps to obtain the routing parameter of a path are shown in Fig. 6.1. The detailed version of our shortest-path-based routing algorithm, called *SP-Based-Routing-Algorithm* (SPBRA for short), is presented in Algorithm 3.

Step 1. Find the location of $\boldsymbol{\pi}_\mu$.

Step 2. Apply Theorems 3.2.5 and 3.2.6 to find $\boldsymbol{\pi}_\mu^*$.

Step 3. Convert the location of $\boldsymbol{\pi}_\mu^*$ into the routing parameter.

Figure 6.1: Steps of finding the routing parameter.

Theorem 6.1.1. *SP-Based-Routing-Algorithm is correct and takes $O(\log N)$-time.*

Proof. We first prove the correctness. Since the correctness of Step 2 follows from Theorems 3.2.5 and 3.2.6, it is sufficient to prove the correctness of Step 1 and Step 3. Recall the following notations introduced in Section 3.1: $\boldsymbol{\pi}_\mu$ is the unique point in PSEUDOMDD that has label μ; $\boldsymbol{\pi}_\mu^*$ is an optimal copy of $\boldsymbol{\pi}_\mu$; two vectors that characterize the L-shape of the PSEUDOMDD are $\boldsymbol{\alpha} = (2\ell, -n)$ and $\boldsymbol{\beta} = (-2p, h)$.

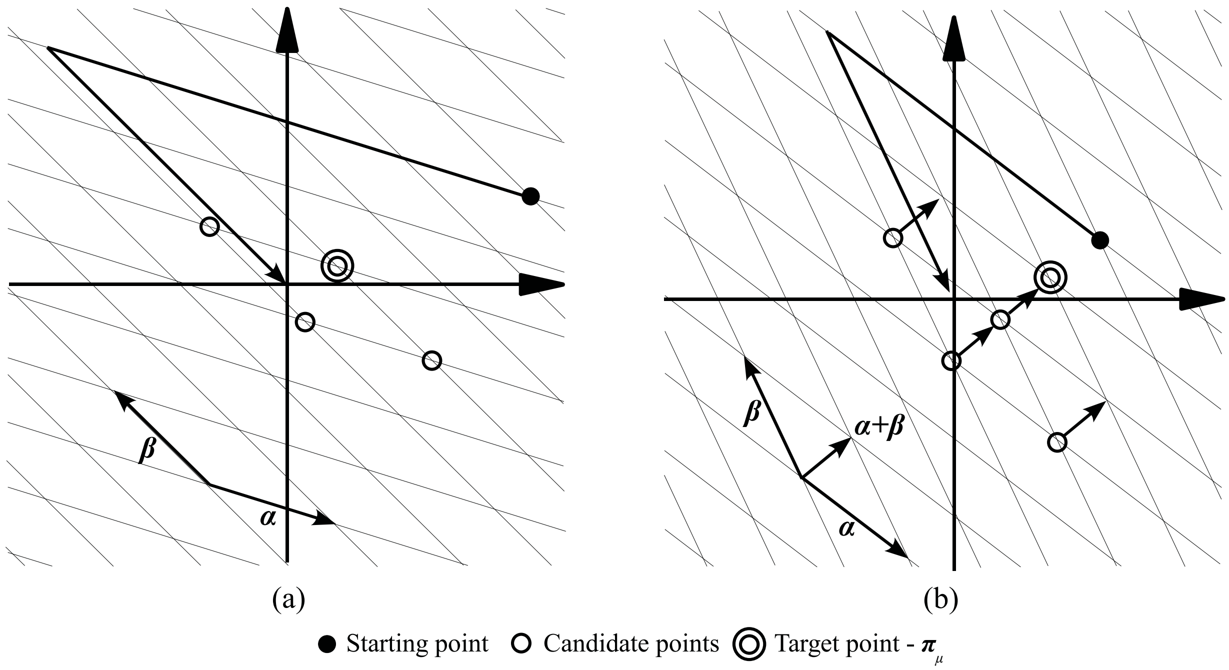


Figure 6.2: Finding the location of π_μ.

Correctness of Step 1: The SPBRA first uses the Euclidean algorithm to find a solution (x_0, y_0) of equation (3.1.1) such that point (x_0, y_0) has label $l((x_0, y_0)) = \mu$. This solution always exists as $MCR(N; s, w)$ is assumed satisfying $\gcd(N, s, w) = 1$. Note that in equation (3.1.1), a point (x, y) with label $l((x, y)) = \mu$ satisfies $parity(x) = parity(l((x, y)))$. Hence we only need to choose one of the two equations in (3.1.1) to find a solution according to the parity of μ. We regard the point (x_0, y_0) as the *starting point*. Consider a path from the starting point to $(0, 0)$ through an integer number of vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$: $(x_0, y_0) + x\boldsymbol{\alpha} + y\boldsymbol{\beta} = (0, 0)$, i.e.,

$$\begin{cases} x_0 + 2\ell x - 2py = 0 \\ y_0 - nx + hy = 0 \end{cases} \tag{6.1.1}$$

The solution $x = \frac{-(hx_0+2py_0)}{2\ell h-2pn}$, $y = \frac{-(nx_0+2\ell y_0)}{2\ell h-2pn}$ to (6.1.1) indicates a path (through an integer number of $\boldsymbol{\alpha}, \boldsymbol{\beta}$) from (x_0, y_0) to $(0, 0)$; see Fig. 6.2(a). Then the four points (we regard them as candidate points) $(x, y) + a\boldsymbol{\alpha} + b\boldsymbol{\beta}$, where

$$(a, b) \in \{ (\lfloor x \rfloor, \lfloor y \rfloor), (\lfloor x \rfloor, \lceil y \rceil), (\lceil x \rceil, \lfloor y \rfloor), (\lceil x \rceil, \lceil y \rceil) \},$$

(not necessary distinct) are copies of (x_0, y_0) that surround $(0, 0)$. Note that both a and b corresponding to this path can not be integers simultaneously since the destination node μ is not the same as the source node. Then target point $\boldsymbol{\pi}_\mu$ can be determined by checking which candidate point is inside the PSEUDOMDD

$$\left\{ (x, y) \in \mathbb{Z}^+ \times \mathbb{Z}^+ \;\middle|\; 0 \le x < 2\ell, 0 \le y < h, \text{ and either } x < 2\ell - 2p \text{ or } y < h - n \right\}.$$

However, it could happen that none of the four candidates points is inside the PSEUDOMDD; Fig. 6.2(b) illustrates such a situation. In this case, the target point can be determined by checking the four new candidate points $\boldsymbol{c}'_i = \boldsymbol{c}_i + \boldsymbol{\alpha} + \boldsymbol{\beta}$ for all candidate point $\boldsymbol{c}_i$. This is because $\boldsymbol{\pi}_\mu$ is the unique point that has label μ inside the PSEUDOMDD and therefore can reach some $\boldsymbol{c}_i$ through an integer number of a vector $\boldsymbol{\alpha}$ and a vector $\boldsymbol{\beta}$.

Correctness of Step 3: After the execution of Step 2, we have $\boldsymbol{\pi}^*_\mu = (x^*_\mu, y^*_\mu)$. By equation (3.1.2), a point (x, y) in $\mathbb{Z}^+ \times \mathbb{Z}^+$ has label

$$l((x, y)) = \left(y + \left\lfloor \frac{x}{2} \right\rfloor\right) s + \left(y - \left\lceil \frac{x}{2} \right\rceil\right) w \mod N.$$

Thus, the routing parameter can be obtained by

$$\begin{aligned} n_s &= y^*_\mu + \left\lfloor \tfrac{x^*_\mu}{2} \right\rfloor, \\ n_w &= y^*_\mu - \left\lceil \tfrac{x^*_\mu}{2} \right\rceil. \end{aligned} \qquad (6.1.2)$$

Now we analyze the time complexity. It takes $O(\log N)$-time to derive the L-shapes of the PSEUDOMDD's of $MCR(N; s, w)$ and $MCR(N; s, N - w)$ in the preprocessing phase. Each line of SPBRA takes constant time except line 10. In line 10, a solution can be found by using the Euclidean algorithm, which takes at most $O(\log N)$-time. As a result, SPBRA takes $O(\log N)$-time. $\square$

Algorithm 3 ***SP-Based-Routing-Algorithm*** **(SPBRA)**

input: N, s, w, u: source node, v: destination node.
output: The routing parameter $[n_s, n_w]$.

begin preprocessing

$(2\ell_0, h_0, 2p_0, n_0) \leftarrow$ the L-shape of the PSEUDOMDD of $MCR(N; s, w)$

$(2\ell_1, h_1, 2p_1, n_1) \leftarrow$ the L-shape of the PSEUDOMDD of $MCR(N; s, N - w)$

end preprocessing

begin SPBRA

$\mu \leftarrow (v - u) \mod N$

$\lambda \leftarrow u \bmod 2$ ▷ The parity of the source node.

$(2\ell, h, 2p, n) \leftarrow (2\ell_\lambda, h_\lambda, 2p_\lambda, n_\lambda)$

$\boldsymbol{\alpha} \leftarrow (2\ell, -n), \boldsymbol{\beta} \leftarrow (-2p, h)$

if $\lambda = 1$ **then** ▷ Consider $MCR(N; s, N - w)$ if u is odd-numbered.

  $w \leftarrow N - w$

end if

▷ *Step 1.*

Use the Euclidean algorithm to find a solution (x_0, y_0) of

$$\mu \equiv \begin{cases} \frac{x}{2}(s - w) + y(s + w) \pmod N & \text{if } \lambda = 0 \\ \left(\frac{x-1}{2}\right)(s - w) + y(s + w) - w \pmod N & \text{if } \lambda = 1 \end{cases}$$

$x \leftarrow \frac{-(hx_0 + 2py_0)}{2\ell h - 2pn}$, $y \leftarrow \frac{-(nx_0 + 2\ell y_0)}{2\ell h - 2pn}$

$i \leftarrow 1$

for each $(a, b) \in \{(\lfloor x \rfloor, \lfloor y \rfloor), (\lfloor x \rfloor, \lceil y \rceil), (\lceil x \rceil, \lfloor y \rfloor), (\lceil x \rceil, \lceil y \rceil)\}$ **do**

  $\boldsymbol{c}_i \leftarrow (x, y) + a\boldsymbol{\alpha} + b\boldsymbol{\beta}$

  $\boldsymbol{c}'_i \leftarrow \boldsymbol{c}_i + \boldsymbol{\alpha} + \boldsymbol{\beta}$

  $i \leftarrow i + 1$

end for

$\boldsymbol{\pi}_\mu \leftarrow$ the point of $\{ \boldsymbol{c}_i, \boldsymbol{c}'_i \mid 1 \le i \le 4 \}$ that is inside the PSEUDOMDD

▷ *Step 2.*

$\boldsymbol{\pi}^*_\mu \leftarrow (x^*_\mu, y^*_\mu)$ (by applying Theorems 3.2.5 and 3.2.6)

▷ *Step 3.*

return $[n_s, n_w] \leftarrow \left[y^*_\mu + \left\lfloor \frac{x^*_\mu}{2} \right\rfloor, y^*_\mu - \left\lceil \frac{x^*_\mu}{2} \right\rceil \right]$

end of SPBRA

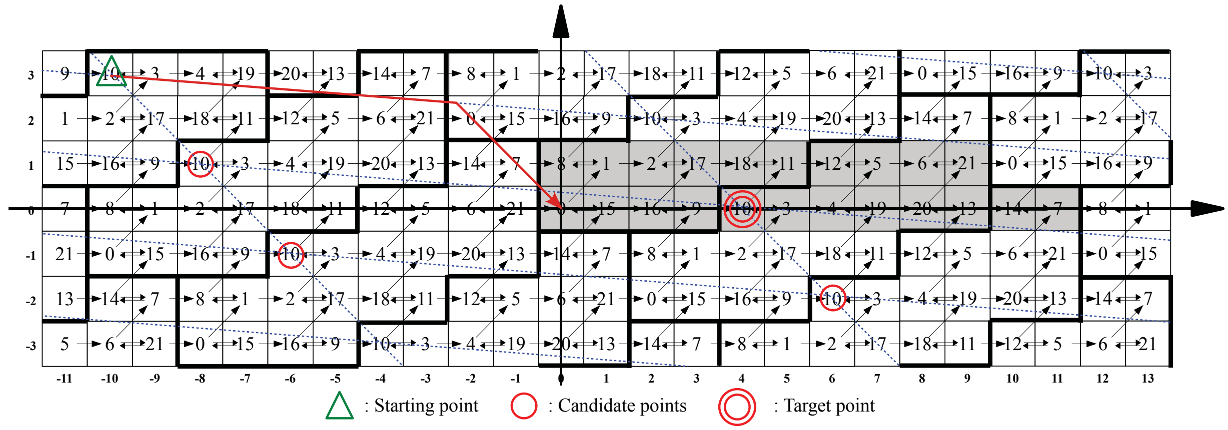


Figure 6.3: An example of shortest path based routing.

Example. Consider routing in $MCR(22;1,7)$; see Fig. 6.3. The preprocessing phase of SPBRA computes the L-shapes of the PSEUDOMDD of $MCR(22;1,7)$ and $MCR(22;1,15)$ and obtains $(12,2,2,1)$ and $(4,6,2,1)$, respectively. Suppose we are sending a message from node $u=2$ to node $v=12$. Then SPBRA derives $\mu=10$, $(2\ell,h,2p,n)=(12,2,2,1)$, $\boldsymbol{\alpha}=(12,-1)$ and $\boldsymbol{\beta}=(-2,2)$. Use the Euclidean algorithm to find a solution, for example $(-10,3)$, of

$$\tfrac{x}{2}(1-7)+y(1+7)\equiv 10 \pmod N).$$

Then SPBRA derives $(x,y)=(\frac{14}{22},-\frac{26}{22})$. The candidate points $\boldsymbol{c}_i$ are $(-6,-1)$, $(-8,1)$, $(6,-2)$ and $(4,0)$; the new candidate points $\boldsymbol{c}'_i$ are $(4,0)$, $(2,2)$, $(16,-1)$ and $(14,1)$. The unique one among $\{\,\boldsymbol{c}_i,\boldsymbol{c}'_i \mid 1\le i\le 4\,\}$ that is inside the PSEUDOMDD is $\boldsymbol{\pi}_\mu=(4,0)$. Since $\boldsymbol{\pi}_\mu\in\boldsymbol{\Gamma}_1^+$, by Theorems 3.2.5 and 3.2.6, we have $\boldsymbol{\pi}^*_\mu=\boldsymbol{\pi}_\mu+1\cdot\boldsymbol{\beta}=(2,2)$. Finally, SPBRA returns the routing parameter $[n_s,n_w]=[3,1]$. If the "$\pm w$ link first" strategy is applied, the routing path is

$$2\xrightarrow{+s}3\xrightarrow{+w}10\xrightarrow{+s}11\xrightarrow{+s}12.$$

On the other hand, if the "$\pm w$ link last" strategy is applied, the routing path is

$$2\xrightarrow{+s}3\xrightarrow{+s}4\xrightarrow{+s}5\xrightarrow{+w}12.$$

6.2 A Dynamic Routing Algorithm

In this section, we present an optimal dynamic routing algorithm for MCRNs. Specifically, after an $O(\log N)$-time to compute the network parameters (computed them once and stored in all nodes), each node can take constant time to determine the link (and hence the node) along the shortest path.

Suppose we are sending messages from source node u to destination node v. Recall that a shortest u, v-path in $MCR(N; s, w)$ can be deduced to a path from node 0 to node $\mu = v - u \bmod N$ in $MCR(N; s, w)$ if u is even-numbered, and in $MCR(N; s, N - w)$ if u is odd-numbered. Also, recall that $\boldsymbol{\pi}^*_\mu$ is the point with the label μ in the MDD of $MCR(N; s, w)$.

6.2.1 Finding a Shortest Route in the Plane

In this section, we construct a shortest path from $(0, 0)$ to $\boldsymbol{\pi}^*_\mu$ in the plane with path length $\Delta(\boldsymbol{\pi}^*_\mu)$. Define

$$\begin{aligned}\boldsymbol{S} &= \{\, (0,0), (1,1), (2,1), (3,2), (4,2), \ldots \,\} \\ &= \left\{\, (x, y) \in \mathbb{Z}^+ \times \mathbb{Z}^+ \mid y = \left\lceil \tfrac{x}{2} \right\rceil, y \geq 0 \,\right\}.\end{aligned}$$

Namely, the points in the plane that can be reached by $(0, 0)$ by using only $+s$-links. Now given $\boldsymbol{\pi}^*_\mu = (x^*_\mu, y^*_\mu)$, let $A = (A_x, A_y)$ be the point in $\boldsymbol{S}$ such that

$$A_x \text{ is } \textit{even}, \text{ and } A_y = \begin{cases} y^*_\mu & \text{if } \boldsymbol{\pi}^*_\mu \in \boldsymbol{\Gamma}^+, \\ \lfloor x^*_\mu - 0.5 \rfloor & \text{if } \boldsymbol{\pi}^*_\mu \in \boldsymbol{\Gamma}^-. \end{cases} \tag{6.2.1}$$

Let P be the path from $(0, 0)$ to point $\boldsymbol{\pi}^*_\mu$ constructed as follows. This path consists of two subpaths P_1 and P_2.

$$P = \underbrace{(0,0),\ \ldots,\ (A_x,}_{P_1}\underbrace{A_y),\ \ldots,\ (x^*_\mu, y^*_\mu)}_{P_2}. \tag{6.2.2}$$

P_1 is from point $(0,0)$ to point A along the points in $\boldsymbol{S}$:

$$(0,0),\ (1,1),\ (2,1),\ (3,2),\ (4,2),\ \ldots,\ (A_x, A_y).$$

P_2 is from point A to point $\boldsymbol{\pi}^*_\mu$ and

- if $\boldsymbol{\pi}^*_\mu \in \boldsymbol{\Gamma}^+$ (for example, the point B in Fig. 6.4), then $A_y = y^*_\mu$, P_2 keeps going east. Thus P_2 is: $(A_x, y^*_\mu),\ (A_x+1, y^*_\mu),\ (A_x+2, y^*_\mu),\ \ldots,\ (x^*_\mu, y^*_\mu)$.
- if $\boldsymbol{\pi}^*_\mu \in \boldsymbol{\Gamma}^-$ and x^*_μ is even (for example, the point C in Fig. 6.4), then $A_x = x^*_\mu$, P_2 repeatedly goes northeast and then west. Thus P_2 is (x^*_μ, A_y), $(x^*_\mu+1, A_y+1)$, (x^*_μ, A_y+1), $(x^*_\mu+1, A_y+2)$, (x^*_μ, A_y+2), $\ldots$, $(x^*_\mu+1, y^*_\mu)$, (x^*_μ, y^*_μ);
- if $\boldsymbol{\pi}^*_\mu \in \boldsymbol{\Gamma}^-$ and x^*_μ is odd (for example, the point D in Fig. 6.4), then $A_x = x^*_\mu - 1$, P_2 goes northeast first, and then repeatedly goes west and then northeast. Thus P_2 is $(x^*_\mu-1, A_y)$, (x^*_μ, A_y+1), $(x^*_\mu-1, A_y+1)$, (x^*_μ, A_y+2), $(x^*_\mu-1, A_y+2)$, (x^*_μ, A_y+3), $\ldots$, $(x^*_\mu-1, y^*_\mu-1)$, (x^*_μ, y^*_μ).

Lemma 6.2.1. *The path P from $(0,0)$ to $\boldsymbol{\pi}^*_\mu = (x^*_\mu, y^*_\mu)$ is of length $\Delta(\boldsymbol{\pi}^*_\mu)$.*

Proof. Let $|P|$ denote the length of P and $A = (A_x, A_y)$ denote the point defined in (6.2.1). Clearly, $|P| = |P_1| + |P_2|$ and $|P_1| = A_x$. If $\boldsymbol{\pi}^*_\mu \in \boldsymbol{\Gamma}^+$, then we have $|P_2| = x^*_\mu - A_x$ and $|P| = x^*_\mu$. By Lemma 3.1.1, we have $\Delta(\boldsymbol{\pi}^*_\mu) = x^*_\mu = |P|$.

Now suppose $\boldsymbol{\pi}^*_\mu \in \boldsymbol{\Gamma}^+$. Then $|P_2| = 2(y^*_\mu - A_y)$ if x^*_μ is even and $|P_2| = 2(y^*_\mu - A_y) - 1$ if x^*_μ is odd. In addition, $A_x = x^*_\mu$ if x^*_μ is even and $A_x = x^*_\mu - 1$ if x^*_μ is odd. In either case, $|P_2| = 2(y^*_\mu - A_y) - parity(x^*_\mu)$ and $A_x = x^*_\mu - parity(x^*_\mu)$. Since point A is in S, we have

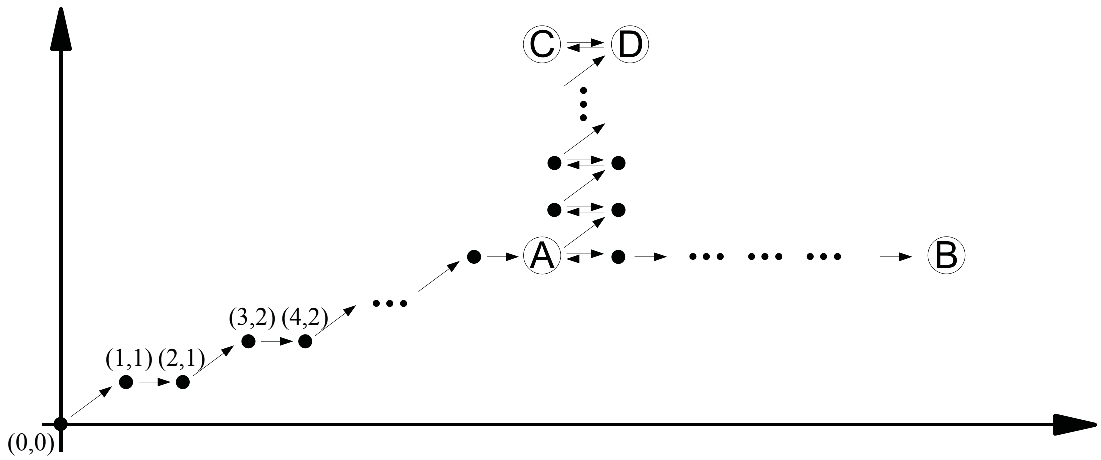


Figure 6.4: A shortest routing path.

$A_y = \lfloor \frac{A_x+1}{2} \rfloor$, or equivalently, $A_x = 2A_y - parity(A_x)$. Since A_x is even, $parity(A_x) = 0$ and thus $|P_2| = 2y^*_\mu - A_x - parity(x^*_\mu)$. By Lemma 3.1.1, we have $\Delta(\boldsymbol{\pi}^*_\mu) = 2y^*_\mu - parity(x^*_\mu) = |P|$. □

The basic idea of designing the dynamic routing algorithm is to choose a link according to the relative position of $\boldsymbol{\pi}^*_\mu$ such that the link is contained in P. Let $\boldsymbol{B}$ denote the set of points in the bottommost row of MDD_0, i.e.,

$$\boldsymbol{B} = \{\, \boldsymbol{\pi} \in \text{MDD}_0 \mid \boldsymbol{\pi} = (x, y) \text{ with } y = 0 \,\}\,.$$

For example, consider $MCR(22; 1, 7)$ in Fig. 3.9, $\boldsymbol{B} = \{\,(0,0),(1,0),(2,0),(3,0)\,\}$. The cardinality of $\boldsymbol{B}$ can be determined from Fig. 3.11 as

$$|\boldsymbol{B}| = 2\min\{\,\ell, h\,\}\,.$$

Lemma 6.2.2. *Suppose $\mu \neq 0$. If $\boldsymbol{\pi}^*_\mu \in \boldsymbol{B}$ (resp., $\boldsymbol{\pi}^*_\mu \notin \boldsymbol{B}$), then there exists a shortest path from node 0 to node μ whose first link is through $(0,0)$ to $(1,0)$, i.e., "$-w$ link" (resp., through $(0,0)$ to $(1,1)$, i.e., "$+s$ link").*

Proof. It should be noted that a shortest path from node 0 to node μ in the MCRN corresponds to a shortest path from point $(0, 0)$ to point $\boldsymbol{\pi}^*_\mu$ in the MDD. Suppose $\boldsymbol{\pi}^*_\mu = (x^*_\mu, y^*_\mu)$. Let P denote the path defined in (6.2.2). If $\boldsymbol{\pi}^*_\mu \in \boldsymbol{B}$, then $y^*_\mu = 0$ and P contains the link from $(0, 0)$ to $(1, 0)$; if $\boldsymbol{\pi}^*_\mu \notin \boldsymbol{B}$, then $y^*_\mu > 0$ and P contains the link from $(0, 0)$ to $(1, 1)$. Note that points in P can not have the label μ, except the end point of P, since $\boldsymbol{\pi}^*_\mu$ is unique point in MDD_0 with the label μ and, by Lemma 6.2.1, the length of P is $\Delta(\boldsymbol{\pi}^*_\mu)$. Thus we have the lemma. □

6.2.2 A Dynamic Routing Algorithm

Now we are ready to present a dynamic routing algorithm, called *Dynamic-Routing-Algorithm* (DRA for short), in Algorithm 4. The main idea of DRA is to determine whether or not $\boldsymbol{\pi}^*_\mu$ belongs to $\boldsymbol{B}$, and then applies Lemma 6.2.2.

Theorem 6.2.3. *Dynamic-Routing-Algorithm is correct. After an $O(\log N)$-time preprocessing phase, Dynamic-Routing-Algorithm takes only constant time to determine the next node on the shortest path to which the message should be sent.*

Proof. We first prove the correctness. The main issue is to decide whether or not $\boldsymbol{\pi}^*_\mu$ belongs to $\boldsymbol{B}$. One naive way to solve this problem is to examine each point sequentially in $\boldsymbol{B}$. However, this method takes $O(\min\{\ell, h\})$-time (can be as worse as $O(\sqrt{N})$). In the following, we show that deciding whether $\boldsymbol{\pi}^*_\mu$ belongs to $\boldsymbol{B}$ can be done in constant time if the L-shapes of the PSEUDOMDD of $MCR(N; s, w)$ and $MCR(N; s, N - w)$ are known in advance.

By the labeling function in (3.1.1), the set of points in $\boldsymbol{B}$ corresponds to the following set of nodes:

$$l(\boldsymbol{B}) = \left\{ 0, -w, s-w, s-2w, \ldots, (\tfrac{|\boldsymbol{B}|}{2}-1)(s-w), (\tfrac{|\boldsymbol{B}|}{2}-1)(s-w)-w \right\} \quad (6.2.3)$$

$$= \left\{ t(s-w),\ t(s-w)-w \mid 0 \le t < \tfrac{|\boldsymbol{B}|}{2} \right\}. \quad (6.2.4)$$

Therefore, $\boldsymbol{\pi}^*_\mu \in \boldsymbol{B}$ if and only if there exists an integer t, $0 \le t < \frac{|\boldsymbol{B}|}{2}$ such that

$$t(s-w) \equiv \mu \pmod N, \quad (6.2.5)$$

or

$$t(s-w) - w \equiv \mu \pmod N. \quad (6.2.6)$$

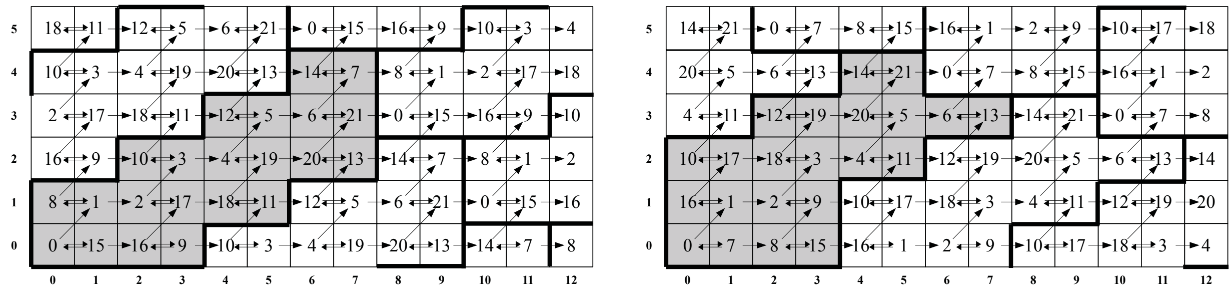
Note that equations (6.2.5) and (6.2.6) can be transferred into the following general modular equation:

$$ax \equiv b \pmod N. \quad (6.2.7)$$

Let $g = \gcd(a, N)$. Equation (6.2.7) has a solution if and only if b is divisible by g. If b is divisible by g, then the solution to (6.2.7) is $\frac{b \bmod N}{g}\left(\frac{a}{g}\right)^{-1} \bmod \frac{N}{g}$, where $\left(\frac{a}{g}\right)^{-1}$ is the inverse of $\frac{a}{g}$ in $\mathbb{Z}_{N/g}$. The subroutine $\mathsf{SOLVE}(a, b, N, g, inv)$ is used to find the smallest positive integer x of the congruence equation (6.2.7). Note that the values g and $\left(\frac{a}{g}\right)^{-1}$ can be obtained by using the Euclidean algorithm and Extended Euclidean algorithm [25], respectively, and we only need to compute these values once and store them in all nodes.

The subroutine inB determines whether $\boldsymbol{\pi}^*_\mu \in \boldsymbol{B}$ by using the subroutine SOLVE to find a non-negative integer t, $0 \le t < \frac{|\boldsymbol{B}|}{2}$, which satisfies equations (6.2.5) and (6.2.6). Thus subroutine inB can determine whether $\boldsymbol{\pi}^*_\mu \in \boldsymbol{B}$ correctly. Finally, if $\boldsymbol{\pi}^*_\mu \in \boldsymbol{B}$, then DRA will send messages to node $(u - w) \bmod N$ by using the "$-w$"-link, i.e., the link between (0,0) and (1,0). Otherwise, DRA will send messages to node $(u + s) \bmod N$ by using the "+s"-link, i.e., the link between (0,0) and (1,1). By Lemma 6.2.2, DRA is correct.

We now analyze the time complexity. It takes $O(\log N)$-time to derive the L-shapes of the PSEUDOMDD of $MCR(N; s, w)$ and $MCR(N; s, N - w)$. By using the Euclidean



(a) MDD_0 of $MCR(22;1,7)$. (b) MDD_0 of $MCR(22;1,15)$.

Figure 6.5: Routing in $MCR(22; 1, 7)$.

algorithm and Extended Euclidean algorithm [25], lines 3-4 take $O(\log N)$-time. Thus, the preprocessing phase totally takes $O(\log N)$-time. Once the preprocessing phase is done, each line of DRA, subroutines SOLVE and inB take only constant time. Consequently, after an $O(\log N)$-time preprocessing phase, DRA takes only constant time to determine the next node on the shortest path to which the message should be sent. □

Example. Suppose we are sending messages from node 1 to node 11 in $MCR(22; 1, 7)$; see Fig.6.5. The preprocessing phase of DRA computes the L-shapes of the PSEUDOMDD of $MCR(22; 1, 7)$ and $MCR(22; 1, 22 - 7)$ and obtains $(12, 2, 2, 1)$ and $(4, 6, 2, 1)$, respectively. Thus $g_0 = g_1 = 2$, $inv_0 = 7, inv_1 = 3$. Then DRA derives $\mu = 10, \lambda = 1, (2\ell, h, 2p, n) = (4, 6, 2, 1), |\boldsymbol{B}| = 4$ and $w = 15$. After that, subroutine $\mathsf{inB}(22, 1, 15, 10, 4, 2, 3)$ returns false. Then DRA returns node $1 + 1 \equiv 2 \pmod N$. Now the problem becomes sending a message from node 2 to destination node 11. Then DRA sets $\mu = 9, \lambda = 0, (2\ell, h, 2p, n) = (12, 2, 2, 1), |\boldsymbol{B}| = 4$ and $w = 7$. Then subroutine $\mathsf{inB}(22, 1, 7, 9, 4, 2, 7)$ returns true and DRA returns node $2 - 7 \equiv 17 \pmod{22}$. Continuing in this way, the routing path from source node 1 to destination node 11 will be

$$1 \xrightarrow{+s} 2 \xrightarrow{-w} 17 \xrightarrow{+s} 18 \xrightarrow{-w} 11.$$

Algorithm 4 ***Dynamic-Routing-Algorithm*** **(DRA)**

input: N, s, w,u: source node, v: destination node.
output: The next node on a shortest u, v-path.

begin preprocessing
$(2\ell_0, h_0, 2p_0, n_0) \leftarrow$ the L-shape of the PSEUDOMDD of $MCR(N; s, w)$
$(2\ell_1, h_1, 2p_1, n_1) \leftarrow$ the L-shape of the PSEUDOMDD of $MCR(N; s, N-w)$
$g_0 \leftarrow \gcd(s-w, N), g_1 \leftarrow \gcd(s+w, N)$
$inv_0 \leftarrow \left(\frac{s-w}{g_0}\right)^{-1}$ in $\mathbb{Z}_{N/g_0}, inv_1 \leftarrow \left(\frac{s+w}{g_1}\right)^{-1}$ in $\mathbb{Z}_{N/g_1}$
end preprocessing

begin DRA
$\mu \leftarrow v - u \pmod N$
$\lambda \leftarrow u \bmod 2$ ▷ The parity of the source node.
$(2\ell, h, 2p, n) \leftarrow (2\ell_\lambda, h_\lambda, 2p_\lambda, n_\lambda)$
$|\boldsymbol{B}| \leftarrow 2 \cdot \min\{\ell, h\}$
$w \leftarrow N - w$ **if** $\lambda = 1$ ▷ Consider $MCR(N; s, N-w)$ if u is odd-numbered.
if $\mu = 0$ **then**
    receive the message and stop the algorithm
else
    if inB$(N, s, w, \mu, |\boldsymbol{B}|, g_\lambda, inv_\lambda)$ = **true then**
        return $(u - w) \bmod N$ ▷ $\boldsymbol{\pi}^*_\mu \in \boldsymbol{B}$
    else
        return $(u + s) \bmod N$ ▷ $\boldsymbol{\pi}^*_\mu \notin \boldsymbol{B}$
    end if
end if
end DRA

Subroutine SOLVE(a, b, N, g, inv)
return $\left(\frac{b \bmod N}{g}\right) \cdot inv \mod \frac{N}{g}$

Subroutine inB$(N, s, w, \mu, |\boldsymbol{B}|, g, inv)$
if $g \nmid b$ **then**
    return false
else
    $t \leftarrow$ SOLVE$(s - w, \mu, N, g, inv)$
    $t' \leftarrow$ SOLVE$(s - w, \mu + w, N, g, inv)$
    if $0 \le t < \frac{|\boldsymbol{B}|}{2}$ or $0 \le t' < \frac{|\boldsymbol{B}|}{2}$ **then**
        return true
    else
        return false
    end if
end if

6.3 Fault-tolerant Routing in MCRNs

In this section, we consider the problem of routing messages in MCRNs in the presence of up to one node or link failure (note that more than one fault can isolate a node). We present an optimal fault-tolerant routing algorithm for MCRNs. The fault-tolerant algorithm do not require routing tables and only very little computational overhead is needed. After an $O(\log N)$-time preprocessing, the algorithm can route messages to the destination node using a constant time at each node along the route. The fault-tolerant routing algorithm presented is guaranteed to find an optimal route after a faulty element is detected.

We assume that in each node there is no global information of the network and thus a faulty element is detected only when a node tries to send messages by using it. Our fault-tolerant routing algorithm is based on the shortest-path-based routing algorithm (SPBRA) presented in Sections 6.1. The SPBRA computes the *routing parameter*, which can be used to determine a routing path. Once we have this information, a node receiving a message examines it and if it is not the receiver, then it can decide which link to use to send messages toward the destination. More specifically, given the routing parameter $[n_s, n_w]$, each node on the routing path can decide the link ($+s$ link or $\pm w$ link) to send messages by the *S-Link-First-Algorithm* or the *W-Link-First-Algorithm* shown in Algorithms 5 and 6, respectively. Note that for a routing parameter $[n_s, n_w]$, n_s and n_w are integers with $n_s \geq 0$, and the sign of n_w indicates which w link ($+w$ link or $-w$ link) to use. In S-Link-First-Algorithm, nodes use $+s$ link as long as the number of remaining $+s$ links is larger than the number of remaining $\pm w$ links. On the other hand, in W-Link-First-Algorithm, nodes use the $\pm w$ link as long as they are applicable. The routing is done when $[n_s, n_w]$ becomes $[0, 0]$. Since we assume each node is only aware of the states of its two immediate links and the nodes connected to these links, a link is considered faulty if it is actually faulty or connected to a faulty node.

Algorithm 5 ***S-Link-First-Algorithm***

input: Routing parameter $[n_s, n_w]$.
output: The output link e.

if $[n_s, n_w] = [0, 0]$ **then**
    receive the message and stop the algorithm
else
    if $n_s > |n_w|$ **then**
        $e \leftarrow +s$ link
        if e is not faulty **then**
            $n_s \leftarrow n_s - 1$
        end if
    else
        if $n_w \neq 0$ and the current node can use $\text{sign}(n_w) \cdot w$ link **then**
            $e \leftarrow \text{sign}(n_w) \cdot w$ link
            if e is not faulty **then**
                $n_w \leftarrow n_w - \text{sign}(n_w) \cdot 1$
            end if
        end if
    end if
end if

Algorithm 6 ***W-Link-First-Algorithm***

input: Routing parameter $[n_s, n_w]$.
output: The output link e.

if $[n_s, n_w] = [0, 0]$ **then**
    receive the message and stop the algorithm
else
    if $n_w \neq 0$ and the current node can use $\text{sign}(n_w) \cdot w$ link **then**
        $e \leftarrow \text{sign}(n_w) \cdot w$ link
        if e is not faulty **then**
            $n_w \leftarrow n_w - \text{sign}(n_w) \cdot 1$
        end if
    else
        $e \leftarrow +s$ link
        if e is not faulty **then**
            $n_s \leftarrow n_s - 1$
        end if
    end if
end if

6.3.1 Finding Alternative Paths

Our fault-tolerant routing algorithm will first call the S-Link-First-Algorithm and each node along the route executes it when forwarding a message. In this section, we consider the problem of finding an alternative path after a faulty link made by the S-Link-First-Algorithm is detected. Let the source and destination nodes be u and v, respectively. Recall that a shortest u, v-path in $MCR(N; s, w)$ can be deduced to a path from node 0 to node $\mu = v - u \bmod N$ in $MCR(N; s, w)$ if u is even and in $MCR(N; s, N - w)$ if u is odd. Since node 0 is mapped to $(0, 0)$ in the plane, we may assume routing is from $(0, 0)$ to $\boldsymbol{\pi}^*_\mu$ in the rest of this section, where $\boldsymbol{\pi}^*_\mu$ is the location of node μ in the MDD_0 of $MCR(N; s, w)$ or $MCR(N; s, N - w)$ (depends on the parity of node u). Since point $(0, 0)$ can only reach either $(1, 1)$ or $(1, 0)$, a faulty link is detected at the $(0, 0)$ to $(1, 1)$ link (resp., $(0, 0)$ to $(1, 0)$ link) when the node wants to send messages by using the $+s$ link (resp., $-w$ link). For convenience, denote the $(0, 0)$ to $(1, 1)$ link (resp., $(0, 0)$ to $(1, 0)$ link) by the e_s link (resp., e_w link).

When a faulty link is detected, we need to convert the routing parameter to the position of $\boldsymbol{\pi}^*_\mu$. This conversion can be done in constant time shown as follows. Note that there is an one-to-one correspondence between the routing parameters and the points in the plane. By equation (3.1.2), a point (x, y) in $\mathbb{Z}^+ \times \mathbb{Z}^+$ has label

$$l((x, y)) = \left(y + \left\lfloor \frac{x}{2} \right\rfloor\right) s + \left(y - \left\lceil \frac{x}{2} \right\rceil\right) w \mod N. \tag{6.3.1}$$

Thus, a point (x, y) can have the routing parameter

$$\left[y + \left\lfloor \frac{x}{2} \right\rfloor, y - \left\lceil \frac{x}{2} \right\rceil\right], \tag{6.3.2}$$

and which point (x, y) having the routing parameter $[a, b]$ can be determined by solving the equation:

$$\begin{cases} y + \left\lfloor \frac{x}{2} \right\rfloor & = & a \\ y - \left\lceil \frac{x}{2} \right\rceil & = & b. \end{cases} \tag{6.3.3}$$

By (6.3.1), it is clear that all shortest paths from $(0,0)$ to (x,y) in the plane consist of $+s$ links and $+w$ links if $y > \lceil \frac{x}{2} \rceil$; only $+s$ links if $y = \lceil \frac{x}{2} \rceil$; and $+s$ links and $-w$ links if $y < \lceil \frac{x}{2} \rceil$. As a result, the paths from $(0,0)$ to $\boldsymbol{\pi}^*_\mu$ corresponding to the applying of the S-Link-First-Algorithm (resp., W-Link-First-Algorithm) are shown in Fig. 6.6 (resp., Fig. 6.7).

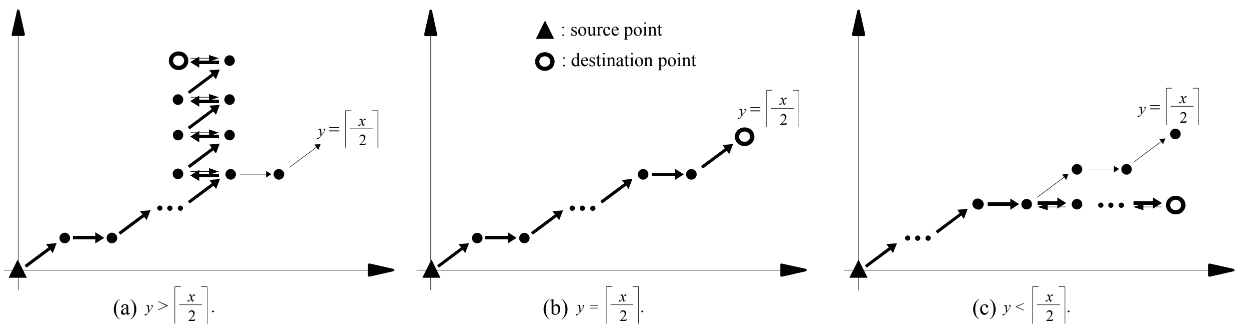


Figure 6.6: The paths correspond to the applying of the S-Link-First-Algorithm.

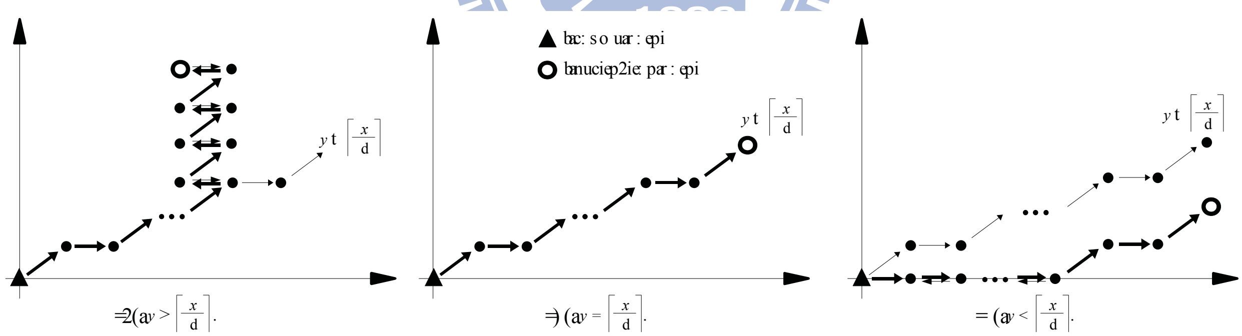


Figure 6.7: The paths correspond to the applying of the W-Link-First-Algorithm.

In most cases, a fault can not block all shortest pathes from source node to destination node. The following lemma provides an alternative shortest path when a faulty link is detected.

Lemma 6.3.1. *Suppose* $\boldsymbol{\pi}_\mu^* = (x_\mu^*, y_\mu^*)$ *with* $1 \leq y_\mu^* < \left\lceil \frac{x_\mu^*}{2} \right\rceil$. *If the* e_s *link is faulty, then there exists a shortest path from* $(0,0)$ *to* $\boldsymbol{\pi}_\mu^*$ *by using the W-Link-First-Algorithm.*

Proof. Since $y_\mu^* < \left\lceil \frac{x_\mu^*}{2} \right\rceil$, all paths from $(0,0)$ to $\boldsymbol{\pi}_\mu^*$ consist of only $+s$ links and $-w$ links, and $|n_w| = \left| y_\mu^* - \left\lceil \frac{x_\mu^*}{2} \right\rceil \right| > 0$. Since node at $(0,0)$ can use the $-w$ link, there exists a shortest path from $(0,0)$ to $\boldsymbol{\pi}_\mu^*$ by using the W-Link-First-Algorithm; see Fig. 6.7(c). □

Note that in Lemma 6.3.1, we exclude the case of $y_\mu^* = 0$. This is because when $y_\mu^* = 0$, the S-Link-First-Algorithm will use the $-w$ link and therefore the e_sd link will not be detected as a faulty link. If $\boldsymbol{\pi}_\mu^*$ does not satisfy Lemma 6.3.1, then we try to find a route to a copy of $\boldsymbol{\pi}_\mu^*$ or make an estimate as to the minimum link increment necessary to route to avoid the fault. The following lemma provides a detour from $(0,0)$ to $\boldsymbol{\pi}_\mu^*$ to avoid the fault by adding two more links.

Lemma 6.3.2. *Suppose* $\boldsymbol{\pi}_\mu^* = (x_\mu^*, y_\mu^*)$ *with* $y_\mu^* \geq \left\lceil \frac{x_\mu^*}{2} \right\rceil$ *and* $x_\mu^* \geq 2$. *If the* e_s *link is faulty, then there exists a path from* $(0,0)$ *to* $\boldsymbol{\pi}_\mu^*$ *with length* $\Delta(\boldsymbol{\pi}_\mu^*)+2$: $(0,0),(1,0),(2,0)$, *followed by using the S-Link-First-Algorithm. In addition, the two links increment is the minimum link increase necessary to reach the same destination if* $\boldsymbol{\pi}_\mu^*$ *has no other copy* $\boldsymbol{z}$ *such that* $\Delta(\boldsymbol{\pi}_\mu^*) = \Delta(\boldsymbol{z})$.

Proof. Since $y_\mu^* \geq \left\lceil \frac{x_\mu^*}{2} \right\rceil$, $\Delta(\boldsymbol{\pi}_\mu^*) = 2y_\mu^* - parity(x_\mu^*)$. Let $(2,0)$ be the new origin and thus $\boldsymbol{\pi}_\mu^*$ corresponds to point $\boldsymbol{z} = (x_\mu^* - 2, y_\mu^*)$ in the new coordinate system. It is clear that routing from $(2,0)$ to $\boldsymbol{\pi}_\mu^*$ in the original coordinate system is equivalent to route from $(0,0)$ to $\boldsymbol{z}$ in the new coordinate system; see Fig. 6.8. Since $\boldsymbol{z} \in \boldsymbol{\Gamma}^-$, $\Delta(\boldsymbol{z}) = 2y_\mu^* - parity(x_\mu^*)$ in the new coordinate system. Thus, a detour from $(0,0)$ to $\boldsymbol{\pi}_\mu^*$ can be constructed by $P_1 \cup P_2$, where subpath P_1 is $(0,0),(1,0),(2,0)$ and subpath P_2 is from $(2,0)$ to $\boldsymbol{\pi}_\mu^*$ by shifting the coordinates of nodes of a shortest path (constructed by using the S-Link-First-Algorithm) from $(0,0)$ to $\boldsymbol{z}$ in the new coordinate system. Clearly, $|P_1|+|P_2| = \Delta(\boldsymbol{\pi}_\mu^*)+2$ and this path clearly contains no e_s link. Moreover, since point $(1,0)$ can only reach either $(0,0)$ or $(2,0)$

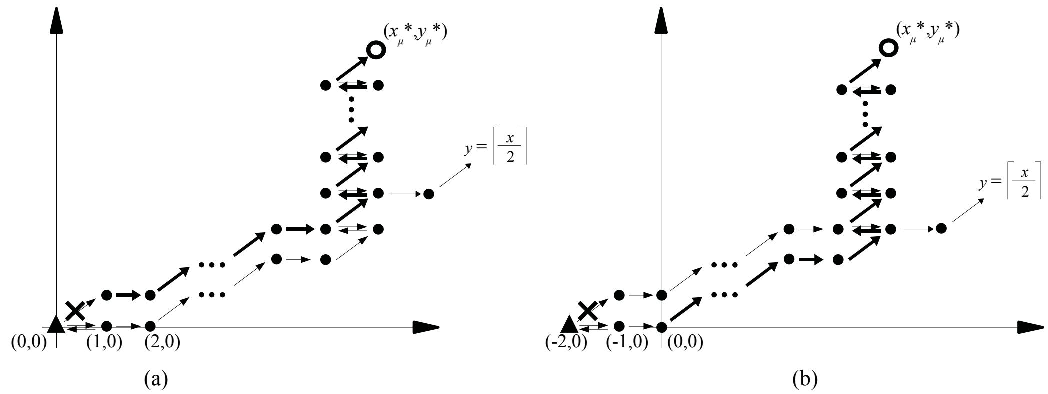


Figure 6.8: A detour: adding two more links to avoid a fault.

and the e_s link is faulty, point $(1,0)$ can get to $\boldsymbol{\pi}_\mu^*$ only by reaching point $(2,0)$. Thus, the two links increment is the minimum link increase necessary to reach the same destination point if $\boldsymbol{\pi}_\mu^*$ has no other copy $\boldsymbol{z}$ such that $\Delta(\boldsymbol{\pi}_\mu^*) = \Delta(\boldsymbol{z})$. □

In the following, we aim at finding a route to a copy of $\boldsymbol{\pi}_\mu^*$. Suppose $\boldsymbol{z}$ is a copy of $\boldsymbol{\pi}_\mu^*$. Define the *cost function* $\boldsymbol{z} \to \mathbb{Z}^+$ as follows.

- If the e_s link is faulty, then

$$cost(\boldsymbol{z}) = \begin{cases} \Delta(\boldsymbol{z}) & \text{if } y < \left\lceil \frac{x}{2} \right\rceil, \\ \Delta(\boldsymbol{z}) + 2 & \text{if } \ell - p \neq 1,\ y \geq \left\lceil \frac{x}{2} \right\rceil \text{ and} \\ & \text{either } x_\mu^* \geq 2 \text{ and } x \geq 2 \\ & \text{or } x_\mu^* < 2 \text{ and } \left\lfloor \frac{x}{2} \right\rfloor \geq 2, \\ \infty & \text{if otherwise.} \end{cases} \tag{6.3.4}$$

- If the e_w link is faulty, then

$$cost(\boldsymbol{z}) = \begin{cases} \Delta(\boldsymbol{z}) & \text{if either } y \geq \left\lceil \frac{x}{2} \right\rceil, \\ & \text{or } h - n \neq 1,\ 2 \leq y < \left\lceil \frac{x}{2} \right\rceil \text{ and } x \neq 2, \\ \Delta(\boldsymbol{z}) + 2 & \text{if } h - n \neq 1,\ 2 \leq y < \left\lceil \frac{x}{2} \right\rceil \text{ and } x = 2, \\ \infty & \text{if otherwise.} \end{cases} \tag{6.3.5}$$

Lemma 6.3.3. *Suppose* $\boldsymbol{z}$ $(\neq \boldsymbol{\pi}^*_\mu)$ *is a copy of* $\boldsymbol{\pi}^*_\mu$. *If the* e_s *link is faulty and* $cost(\boldsymbol{z}) \neq \infty$, *then there exists a path from* $(0,0)$ *to* $\boldsymbol{z}$ *with length* $cost(\boldsymbol{z})$. *Moreover, if* $\boldsymbol{z}$ *has the smallest cost among all copies of* $\boldsymbol{\pi}^*_\mu$, *then the path from* $(0,0)$ *to* $\boldsymbol{z}$ *contains no* e_s *link.*

Proof. If $y < \lceil \frac{x}{2} \rceil$, then by Lemma 6.3.1, there exists a path from $(0,0)$ to $\boldsymbol{z}$ with length $\Delta(\boldsymbol{z}) = cost(\boldsymbol{z})$: $(0,0),(1,0),(2,0)$, followed by using the S-Link-First-Algorithm. In the following, we assume $y \geq \lceil \frac{x}{2} \rceil$. In this case, all shortest paths from $(0,0)$ to $\boldsymbol{z}$ must pass through the e_s link and therefore it has to make a detour to route. Note that for every point $\boldsymbol{z}$, the point $\boldsymbol{z} + \boldsymbol{\alpha} + \boldsymbol{\beta}$, where $\boldsymbol{\alpha} + \boldsymbol{\beta} = (2\ell - 2p, h - n)$, is always in $\mathbb{Z}^+ \times \mathbb{Z}^+$. If $\ell - p = 1$, then every copy of $\boldsymbol{\pi}^*_\mu$ will be blocked by the faulty link. Thus, suppose $\ell - p \neq 1$ and $x^*_\mu \geq 2$. If $x < 2$, then the path from $(0,0)$ to $\boldsymbol{z}$ is clearly blocked by the e_s link. If $x \geq 2$, then by Lemma 6.3.2, a path from $(0,0)$ to $\boldsymbol{z}$ can be found by adding two more links: $(0,0),(1,0),(2,0)$, followed by using the S-Link-First-Algorithm; the path length is $\Delta(\boldsymbol{z}) + 2 = cost(\boldsymbol{z})$.

Now we suppose $x^*_\mu < 2$. According to the parity of x, we construct a path from $(0,0)$ to $\boldsymbol{z}$ as follows. If x is even, then the path is $(0,0),(1,0),(2,0)$, followed by using the S-Link-First-Algorithm to point $(x-1,y)$, then to point (x,y); see Fig. 6.9(a). If x is odd, then the path is $(0,0),(1,0),(2,0)$, followed by using the S-Link-First-Algorithm to point $(x-2,y-1)$, then $(x-1,y-1)$, (x,y); see Fig. 6.9(b). Clearly, the path length is $\Delta(\boldsymbol{z}) + 2 = cost(\boldsymbol{z})$.

Suppose $\boldsymbol{z}$ has the smallest cost among all copies of $\boldsymbol{\pi}^*_\mu$ and the path from $(0,0)$ to $\boldsymbol{z}$ contains e_s link. Let the copy of $(0,0)$ in this path be (x',y') with $x' > 0, y' > 0$. Then the point $(x-x', y-y')$ is also a copy of $\boldsymbol{\pi}^*_\mu$, yet has a smaller cost than $\boldsymbol{z}$, a contradiction. □

Lemma 6.3.4. *Suppose* $\boldsymbol{z}$ $(\neq \boldsymbol{\pi}^*_\mu)$ *is a copy of* $\boldsymbol{\pi}^*_\mu$. *If the* e_w *link is faulty and* $cost(\boldsymbol{z}) \neq \infty$, *then there exists a path from* $(0,0)$ *to* $\boldsymbol{z}$ *with length* $cost(\boldsymbol{z})$. *Moreover, if* $\boldsymbol{z}$ *has the smallest cost among all copies of* $\boldsymbol{\pi}^*_\mu$, *then the path from* $(0,0)$ *to* $\boldsymbol{z}$ *contains no* e_s *link.*

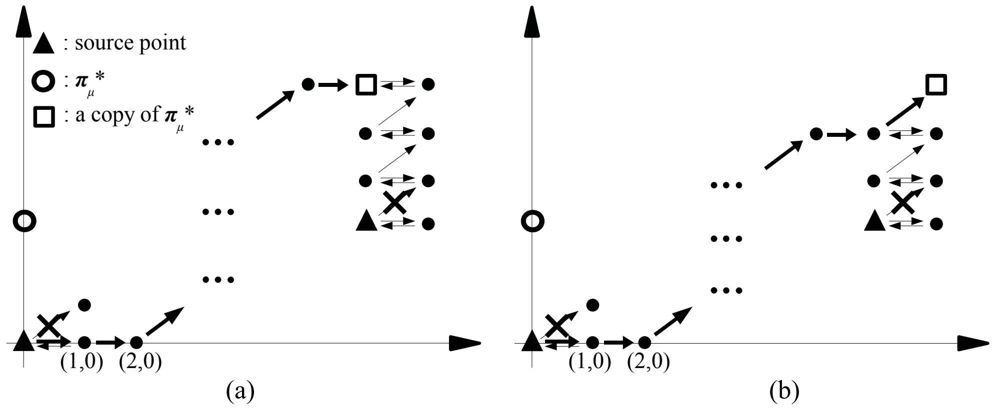


Figure 6.9: The illustrations of the cases to the proof in Lemma 6.3.3.

Proof. If $y \geq \lceil \frac{x}{2} \rceil$, then clearly there exists a path from $(0, 0)$ to $\boldsymbol{z}$ by using the S-Link-First-Algorithm. In the following, we assume $y < \lceil \frac{x}{2} \rceil$. Note that for every point $\boldsymbol{z}$, the point $\boldsymbol{z}+\boldsymbol{\alpha}+\boldsymbol{\beta}$, where $\boldsymbol{\alpha}+\boldsymbol{\beta} = (2\ell-2p, h-n)$, is always in $\mathbb{Z}^+ \times \mathbb{Z}^+$. If $h-n = 1$, then every copy of $\boldsymbol{\pi}_\mu^*$ will be blocked by the faulty link. In addition, if $y < 2$, then any path from $(0, 0)$ to $\boldsymbol{z}$ will be blocked by the faulty link. Thus, suppose $h - n \neq 1$ and $2 \leq y < \lceil \frac{x}{2} \rceil$. If $x \neq 2$ and x is even, then a path from $(0, 0)$ to $\boldsymbol{z}$ can be constructed as $(0, 0), (1, 1), (2, 1)$, followed by using the W-Link-First-Algorithm to point $(x - 2, y - 1)$, then to point $(x - 1, y)$, (x, y); if $x \neq 2$ and x is is odd, then a path from $(0, 0)$ to $\boldsymbol{z}$ can be constructed as $(0, 0), (1, 1), (2, 1)$, followed by using the W-Link-First-Algorithm to point $(x - 1, y - 1)$, then to (x, y); see Fig. 6.10(a) for an illustration. Clearly, these paths are of length $\Delta(\boldsymbol{z}) = cost(\boldsymbol{z})$. If $x = 2$, then a path from $(0, 0)$ to $\boldsymbol{z}$ can be constructed as $(0, 0), (1, 1), (2, 1)$, followed by using the W-Link-First-Algorithm to point $(x, y - 1)$, then to point $(x + 1, y)$, (x, y). Clearly, this path is of length $\Delta(\boldsymbol{z}) = cost(\boldsymbol{z})$; see Fig. 6.10(b).

Suppose $\boldsymbol{z}$ has the smallest cost among all copies of $\boldsymbol{\pi}_\mu^*$ and the path from $(0, 0)$ to $\boldsymbol{z}$ contains e_s link. Let the copy of $(0, 0)$ in this path be (x', y') with $x' > 0, y' > 0$. Then the point $(x - x', y - y')$ is also a copy of $\boldsymbol{\pi}_\mu^*$, yet has a smaller cost than $\boldsymbol{z}$, a contradiction. □

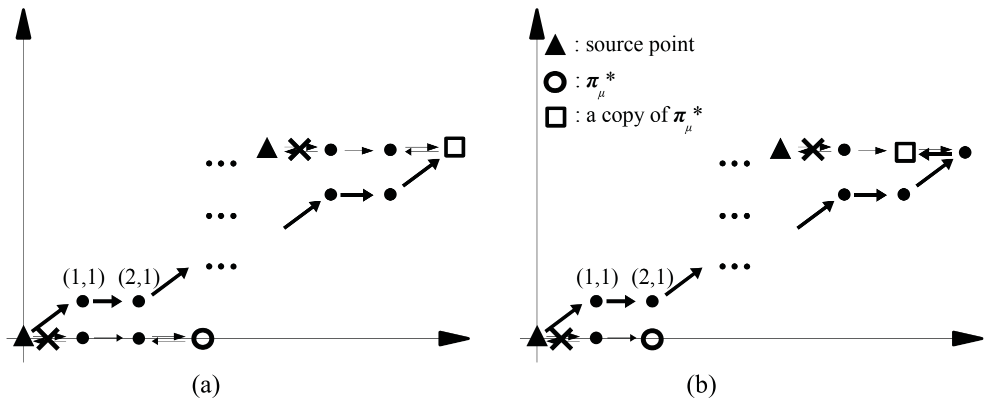


Figure 6.10: The illustrations of the cases to the proof in Lemma 6.3.4.

6.3.2 Finding the Lowest Cost Point

Our aim in this section is to find the lowest cost copy of $\boldsymbol{\pi}_\mu^*$ (other than $\boldsymbol{\pi}_\mu^*$). For convenience, some notations will be introduced first. Given a point $\boldsymbol{z}$ and a vector $\boldsymbol{v}$, define $L_{\boldsymbol{v}}(\boldsymbol{z}) = \{\, \boldsymbol{z} + t\boldsymbol{v} \mid t \in \mathbb{Z} \,\}$. Namely, points in $L_{\boldsymbol{v}}(\boldsymbol{z})$ are reachable by $\boldsymbol{z}$ through an integer number of $\boldsymbol{v}$. The set of the two points in $L_{\boldsymbol{v}}(\boldsymbol{z})$ that are around the $x - 2y = 0$ line is denoted by $P_{\boldsymbol{v}}(\boldsymbol{z})$, i.e.,

$$P_{\boldsymbol{v}}(\boldsymbol{z}) = \{\, \boldsymbol{z}_1, \boldsymbol{z}_2 \in L_{\boldsymbol{v}}(\boldsymbol{z}) \mid \boldsymbol{z}_1 \in \boldsymbol{\Gamma}^-, \boldsymbol{z}_2 \in \boldsymbol{\Gamma}^+ \text{ and } \boldsymbol{z}_1 = \boldsymbol{z}_2 + \boldsymbol{v} \,\}.$$

Note that the cardinality of $P_{\boldsymbol{v}}(\boldsymbol{z})$ may be less than 2 as we only consider points in $\mathbb{Z}^+ \times \mathbb{Z}^+$.

Lemma 6.3.5. *Suppose vector $\boldsymbol{v} = (v_1, v_2)$ with even v_1 and $v_1 \cdot v_2 \leq 0$. Then the two points in $P_{\boldsymbol{v}}(\boldsymbol{z})$ have the smallest distance (to $(0,0)$) among all points in $L_{\boldsymbol{v}}(\boldsymbol{z})$.*

Proof. Without loss of generality, assume $v_1 \geq 0, v_2 < 0$. Let $\boldsymbol{z}_1, \boldsymbol{z}_2$ be two points in $P_{\boldsymbol{v}}(\boldsymbol{z})$ such that $\boldsymbol{z}_1 \in \boldsymbol{\Gamma}^-, \boldsymbol{z}_2 \in \boldsymbol{\Gamma}^+$ and $\boldsymbol{z}_1 = \boldsymbol{z}_2 + \boldsymbol{v}$. If $v_1 > 0$, then by Lemma 3.1.2, $\Delta(v_1) < \Delta(v_1 - \boldsymbol{v}) < \Delta(v_1 - 2\boldsymbol{v}) < \cdots$ and $\Delta(v_2) < \Delta(v_2 + \boldsymbol{v}) < \Delta(v_2 + 2\boldsymbol{v}) < \cdots$ hold. If $v_1 = 0$, then $\Delta(v_1) < \Delta(v_1 - \boldsymbol{v}) < \Delta(v_1 - 2\boldsymbol{v}) < \cdots$ and $\Delta(v_2) = \Delta(v_2 + \boldsymbol{v}) = \Delta(v_2 + 2\boldsymbol{v}) = \cdots$. Thus we have this lemma. $\square$

Suppose the PSEUDOMDD has an L-shape $(2\ell, h, 2p, n)$. Two vectors characterizing the L-shape of the PSEUDOMDD are $\boldsymbol{\alpha} = (2\ell, -n)$ and $\boldsymbol{\beta} = (-2p, h)$. Recall that copies of $\boldsymbol{\pi}^*_\mu$ can be reach from $\boldsymbol{\pi}^*_\mu$ through an integer number of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Consider the parallelogram formed by $\boldsymbol{\pi}^*_\mu$ and $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$; see Fig. 6.11. The four lines $L_{\boldsymbol{\alpha}}(\boldsymbol{\pi}^*_\mu)$, $L_{\boldsymbol{\beta}}(\boldsymbol{\pi}^*_\mu)$, $L_{\boldsymbol{\alpha}}(\boldsymbol{\pi}^*_\mu + \boldsymbol{\alpha} + \boldsymbol{\beta})$, $L_{\boldsymbol{\beta}}(\boldsymbol{\pi}^*_\mu + \boldsymbol{\alpha} + \boldsymbol{\beta})$ consist of copies of $\boldsymbol{\pi}^*_\mu$ that are as close as to $(0, 0)$. Thus the lowest cost copy of $\boldsymbol{\pi}^*_\mu$ may appear in one of these lines. By Lemma 6.3.5, we only need to consider $P_{\boldsymbol{\alpha}}(\boldsymbol{\pi}^*_\mu) \cup P_{\boldsymbol{\beta}}(\boldsymbol{\pi}^*_\mu) \cup P_{\boldsymbol{\alpha}}(\boldsymbol{\pi}^*_\mu + \boldsymbol{\alpha} + \boldsymbol{\beta}) \cup P_{\boldsymbol{\beta}}(\boldsymbol{\pi}^*_\mu + \boldsymbol{\alpha} + \boldsymbol{\beta}) \setminus \{ \boldsymbol{\pi}^*_\mu \}$. In other words, we need to examine at most eight points to determine the lowest cost copy of $\boldsymbol{\pi}^*_\mu$. In fact, most of them are the same point or are not inside $\mathbb{Z}^+ \times \mathbb{Z}^+$. Note that given $\boldsymbol{z}$ and $\boldsymbol{v}$, finding $P_{\boldsymbol{v}}(\boldsymbol{z})$ can be done in constant time shown as follows. Let t be an integer such that $\boldsymbol{z} + t\boldsymbol{v} \in \boldsymbol{\Gamma}^-$ and $\boldsymbol{z} + (t+1)\boldsymbol{v} \in \boldsymbol{\Gamma}^+$. Then $P_{\boldsymbol{v}}(\boldsymbol{z}) = \{ \boldsymbol{z} + t\boldsymbol{v}, \boldsymbol{z} + (t+1)\boldsymbol{v} \}$.

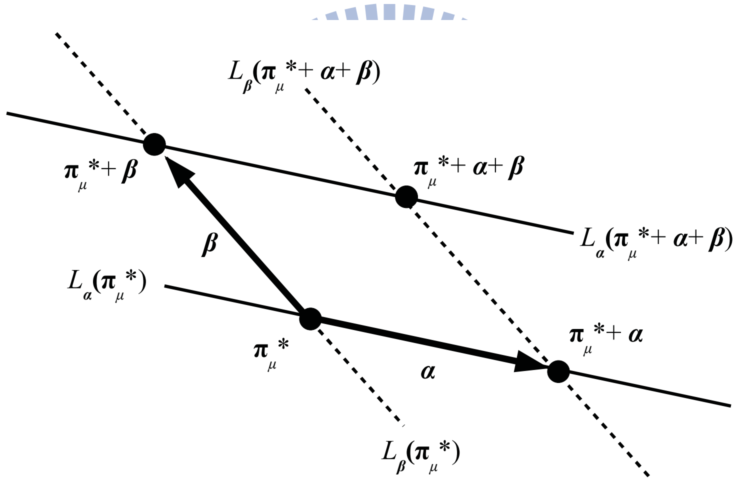


Figure 6.11: Find the lowest cost point.

For convenience, set $P_{\boldsymbol{\alpha}}(\boldsymbol{\pi}^*_\mu) \cup P_{\boldsymbol{\beta}}(\boldsymbol{\pi}^*_\mu) \cup P_{\boldsymbol{\alpha}}(\boldsymbol{\pi}^*_\mu + \boldsymbol{\alpha} + \boldsymbol{\beta}) \cup P_{\boldsymbol{\beta}}(\boldsymbol{\pi}^*_\mu + \boldsymbol{\alpha} + \boldsymbol{\beta}) \setminus \{ \boldsymbol{\pi}^*_\mu \} = PARA(\boldsymbol{\pi}^*_\mu)$. It should be noticed that the set $PARA(\boldsymbol{\pi}^*_\mu)$ cannot be empty. This is because point $\boldsymbol{\pi}^*_\mu + \boldsymbol{\alpha} + \boldsymbol{\beta} = (x^*_\mu + 2\ell - 2p, y^*_\mu + h - n)$ is inside $\mathbb{Z}^+ \times \mathbb{Z}^+$ and is always contained in $P_{\boldsymbol{\alpha}}(\boldsymbol{\pi}^*_\mu + \boldsymbol{\alpha} + \boldsymbol{\beta}) \cup P_{\boldsymbol{\beta}}(\boldsymbol{\pi}^*_\mu + \boldsymbol{\alpha} + \boldsymbol{\beta})$. However, in some cases, the lowest cost copy of $\boldsymbol{\pi}^*_\mu$ may not exist in $PARA(\boldsymbol{\pi}^*_\mu)$. In this case, we must have all points (except $\boldsymbol{\pi}^*_\mu + \boldsymbol{\alpha} + \boldsymbol{\beta}$) of $PARA(\boldsymbol{\pi}^*_\mu)$ to be outside $\mathbb{Z}^+ \times \mathbb{Z}^+$ and either (i) the e_w link is faulty and $h - p = 1$ or (ii) the

e_s is faulty and $\ell - p = 1$. In other words, $\boldsymbol{\pi}^*_\mu + \boldsymbol{\alpha} + \boldsymbol{\beta}$ is still be blocked by the faulty link. For example, in Fig. 6.12, suppose $\mu = 32$ and the e_w link is faulty. We have $\boldsymbol{\pi}^*_\mu = (2, 0)$ and thus $PARA(\boldsymbol{\pi}^*_\mu)$=$\{\ \boldsymbol{\pi}^*_\mu + \boldsymbol{\alpha} + \boldsymbol{\beta}\ \} = \{\ (6, 1)\ \}$. Clearly $(6, 1)$ is still be blocked by the faulty link. This problem can be solved by the following result proposed by Liu [49]. We modify their results to fit our notations.

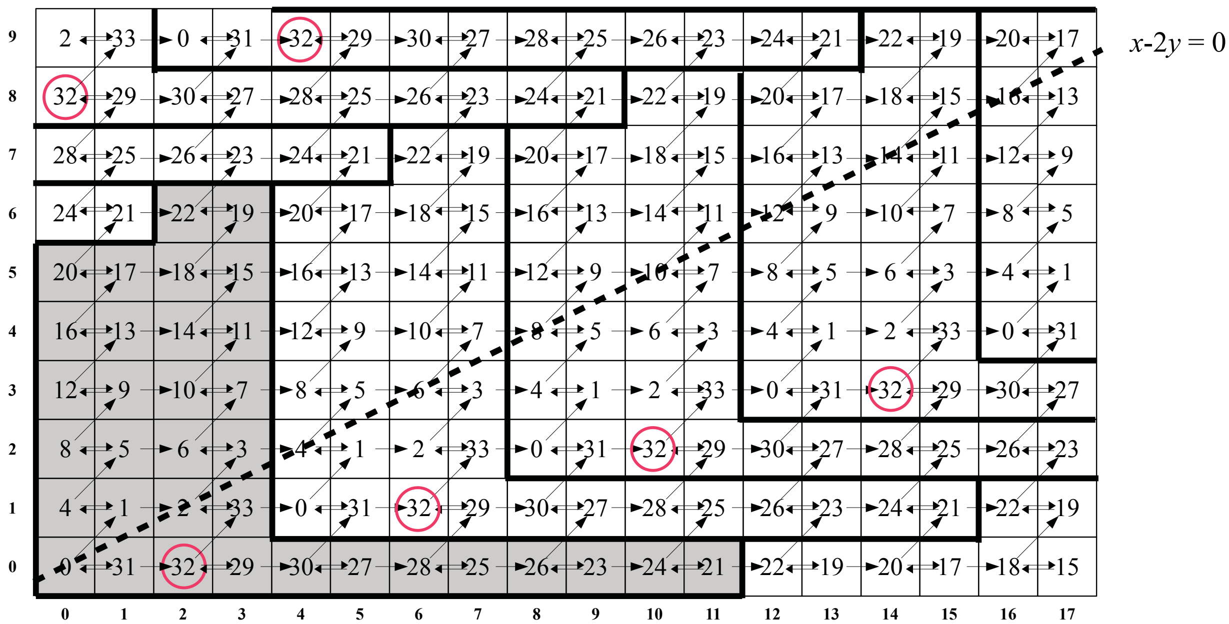


Figure 6.12: Optimal fault-tolerant in $MCR(34; 1, 3)$. Two vectors that characterizing the PSEUDOMDD are $\boldsymbol{\alpha} = (14, -5), \boldsymbol{\beta} = (-10, 6)$.

Lemma 6.3.6. [49] *Suppose* $\boldsymbol{\pi}^*_\mu = (x^*_\mu, y^*_\mu)$ *with* $y^*_\mu = 0$*. If the* e_w *link is faulty,* $PARA(\boldsymbol{\pi}^*_\mu) = \{\ \boldsymbol{\pi}^*_\mu + \boldsymbol{\alpha} + \boldsymbol{\beta}\ \}$ *and* $h - n = 1$*, then* $\boldsymbol{z} = (r, h + k - 1)$ *is the lowest copy of* $\boldsymbol{\pi}^*_\mu$*, where* $k = \left\lceil \frac{2\ell - x^*_\mu}{2\ell - 2p} \right\rceil$ *and* $r = (x^*_\mu + k(2\ell - 2p)) \bmod 2\ell$*. Moreover, the path from* $(0, 0)$ *to* $\boldsymbol{z}$ *by using the S-Link-First-Algorithm contains no faulty link.*

Proof. Since $h - n = 1$, every copy of $\boldsymbol{\pi}^*_\mu$ in $\boldsymbol{\Gamma}^+$ will be blocked by the faulty link. In this situation, the lowest cost copy of $\boldsymbol{\pi}^*_\mu$ must appear in $\boldsymbol{\Gamma}^-$. In [49], Liu et al. find that point $\boldsymbol{z} = (r, h + k - 1)$ is the closest copy of $\boldsymbol{\pi}^*_\mu$ (in ℓ_1-norm). It is not difficult to check that $L_{\boldsymbol{\alpha}+\boldsymbol{\beta}}(\boldsymbol{z})$ consisting of copies of $\boldsymbol{\pi}^*_\mu$ is the closest line (to $(0, 0)$) in $\boldsymbol{\Gamma}^-$, and thus $\boldsymbol{z}$ is the lowest copy of $\boldsymbol{\pi}_\mu^*$ in $\boldsymbol{\Gamma}^-$. Moreover, the path from $(0,0)$ to $\boldsymbol{z}$ by using the S-Link-First-Algorithm contains no e_w link; if not, then we can easily find another copy of $\boldsymbol{\pi}_\mu^*$ with smaller cost than $\boldsymbol{z}$, a contradiction. □

Lemma 6.3.7. [49] *Suppose $\boldsymbol{\pi}_\mu^* = (x_\mu^*, y_\mu^*)$ with $x_\mu^* < 2$. If the e_s link is faulty, $PARA(\boldsymbol{\pi}_\mu^*) = \{\, \boldsymbol{\pi}_\mu^* + \boldsymbol{\alpha} + \boldsymbol{\beta} \,\}$ and $\ell - p = 1$, then $\boldsymbol{z} = (r, 2\ell + k - 1)$ is the lowest cost copy of $\boldsymbol{\pi}_\mu^*$, where $k = \left\lceil \frac{h-y_\mu^*}{h-n} \right\rceil$ and $t = (y_\mu^* + k(h-n)) \bmod h$. Moreover, the path from $(0,0)$ to $\boldsymbol{z}$ by using the S-Link-First-Algorithm contains no faulty link.*

Proof. Since the proof is similar to that of Lemma 6.3.6, we omit it. □

Now we are ready to present the fault-tolerant routing algorithm for MCRNs, call *FTRA*, in Algorithm 7. Since the correctness of FTRA follows from Lemmas 6.3.1, 6.3.2, 6.3.3, 6.3.4, 6.3.6 and 6.3.7 directly, we omit it.

Theorem 6.3.8. *FTRA is an optimal fault-tolerant routing algorithm for MCRNs. After an $O(\log N)$-time preprocessing, FTRA takes constant time to execute at each node along the route.*

Example. Suppose we are sending a message from node 0 to node 8 in $MCR(22;1,7)$; see Fig. 6.13(a). The preprocessing phase of FTRA computes the L-shapes of the PSEUDOMDD of $MCR(22;1,7)$ and $MCR(22;1,15)$ and obtains $(12,2,2,1)$ and $(4,6,2,1)$, respectively, and derives the routing parameter as $[1,1]$. Suppose the e_s link is detected as a faulty link at node 0. Then FTRA converts the routing parameter to $\boldsymbol{\pi}_\mu^* = (0,1)$. After that, it computes $PARA(\boldsymbol{\pi}_\mu^*) = \{\,(6,6),(8,4),(10,2),(12,0)\,\}$ and obtains $cost((6,6)) = 14, cost((8,4)) = 10, cost((10,2)) = 10, cost((12,0)) = 12$. Since point $(10,2)$ has the smallest cost, FTRA construct a route from $(0,0)$ to $(10,2)$ as follows: $(0,0) \to (1,0) \to (2,0)$, followed by using S-Link-First-Algorithm. In other words, the whole path will be $(0,0),(1,0),(2,0),(3,1),(4,1),(5,2),(6,2),(7,2),(8,2),(9,2),(10,2)$.

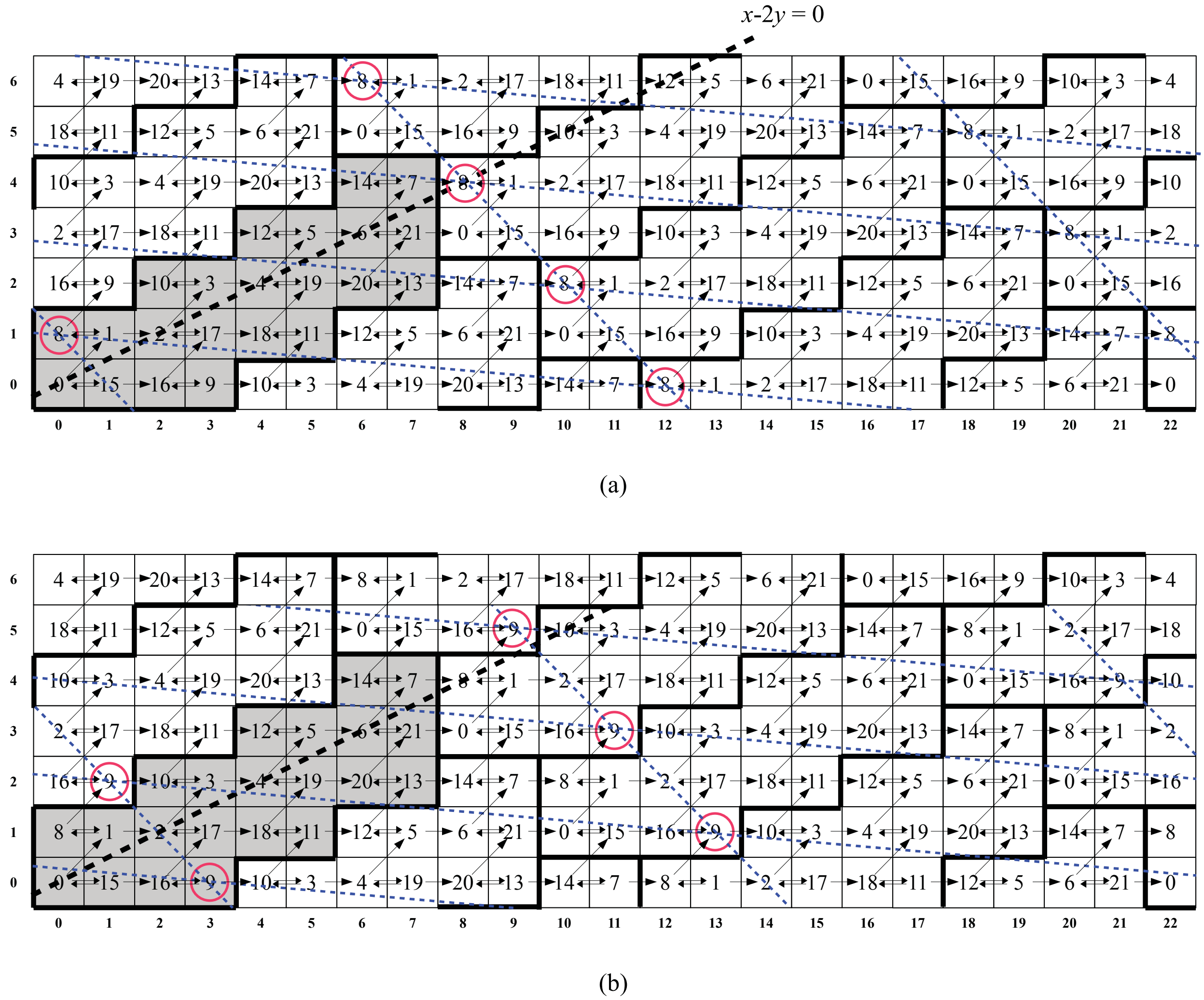


Figure 6.13: Optimal fault-tolerant in $MCR(22;1,7)$.

Example. Suppose we are sending a message from node 0 to node 8 in $MCR(22;1,7)$; see Fig. 6.13(b). The preprocessing phase of FTRA derives the routing parameter as $[1,-2]$. Suppose the e_w link is detected as a faulty link at node 0. Then FTRA converts the routing parameter to $\boldsymbol{\pi}^*_\mu = (3,0)$. After that, it computes $PARA(\boldsymbol{\pi}^*_\mu) = \{\,(1,2),(9,5),(11,3),(13,1)\,\}$ and obtains $cost((1,2)) = 3, cost((9,5)) = 9, cost((11,3)) = 11, cost((13,1)) = \infty$. Since point $(1,2)$ has the smallest cost, FTRA construct a route from $(0,0)$ to $(1,2)$ by using S-Link-First-Algorithm as follows: $(0,0),(1,1),(0,1),(1,2)$.

Algorithm 7 ***Fault-Tolerant-Routing-Algorithm*** **(FTRA)**

input: N, s, w, u: source, v: destination.
output: The output link e.

begin preprocessing

$(2\ell_0, h_0, 2p_0, n_0) \leftarrow$ the L-shape of the PSEUDOMDD of $MCR(N; s, w)$
$(2\ell_1, h_1, 2p_1, n_1) \leftarrow$ the L-shape of the PSEUDOMDD of $MCR(N; s, N-w)$
if u is odd-numbered **then** ▷ Consider $MCR(N; s, N-w)$
  $w \leftarrow N - w \mod N$
end if
$\mu \leftarrow v - u \mod N$
$[n_s, n_w] \leftarrow$ the routing parameter of node μ in the MDD_0 of $MCR(N; s, w)$

end preprocessing

begin FTRA

Call the S-Link-First-Algorithm
if e is faulty **then**
  if u is odd-numbered **then** ▷ Consider $MCR(N; s, N-w)$
    $w \leftarrow N - w \mod N$
    $n_w \leftarrow -n_w$
  end if
  $\mu \leftarrow v - u \pmod N$
  $\boldsymbol{\pi}^*_\mu = (x^*_\mu, y^*_\mu) \leftarrow$ the point having the routing parameter $[n_s, n_w]$
  if $1 \le y^*_\mu < \left\lceil \frac{x^*_\mu}{2} \right\rceil$ **then**
    route to $\boldsymbol{\pi}^*_\mu$ by using the W-Link-First-Algorithm
  else
    $\lambda \leftarrow u \bmod 2$
    $\boldsymbol{\alpha} \leftarrow (2\ell_\lambda, -n_\lambda), \boldsymbol{\beta} \leftarrow (-2p_\lambda, h_\lambda)$
    let $\boldsymbol{z}$ be a point in $PARA(\boldsymbol{\pi}^*_\mu)$ with the smallest cost, where the cost of a point is defined in (6.3.4), (6.3.5)
    if $e = e_s$, $y^*_\mu \ge \left\lceil \frac{x^*_\mu}{2} \right\rceil$, $x^*_\mu \ge 2$ and $\Delta(\boldsymbol{\pi}^*_\mu) + 2 \le \Delta(\boldsymbol{z})$ **then**
      route to $\boldsymbol{\pi}^*_\mu$ by Lemma 6.3.2
    end if
    if $e = e_w$, $PARA(\boldsymbol{\pi}^*_\mu) = \left\{ \boldsymbol{\pi}^*_\mu + \boldsymbol{\alpha} + \boldsymbol{\beta} \right\}$ and $h - n = 1$ **then**
      route to $(r, h + k - 1)$, defined in Lemma 6.3.6
      break
    end if
    if $e = e_s$, $PARA(\boldsymbol{\pi}^*_\mu) = \left\{ \boldsymbol{\pi}^*_\mu + \boldsymbol{\alpha} + \boldsymbol{\beta} \right\}$ and $\ell - p = 1$ **then**
      route to $(r, 2\ell + k - 1)$, defined in Lemma 6.3.7
      break
    end if
    route to $\boldsymbol{z}$ by using the route illustrated in Lemmas 6.3.3 and 6.3.4
  end if
else
  $u \leftarrow u + e \mod N$
  send messages to the node by using e
end if

end FTRA

Chapter 7

Experimental Results

7.1 Experimental Results

Although Theorem 5.3.1 provides a class of optimal MCRNs, to find optimal MCRNs is extremely difficult to solve analytically for all values of N. In addition, to find MCRNs that minimize the *average distance* for all values of N is another difficult problem, where the average distance of $MCR(N; s, w)$ is defined by

$$\overline{d}(N, s, w) = \frac{1}{N^2} \sum_{u,v \in V(G)} d(u, v),$$

and the *optimal average distance* $\overline{D}(N)$ is the smallest average distance among all MCRNs with N nodes.

Both of the above two discrete problems turn out to be difficult due to the following reason: neither the diameter nor the average distance between vertices will always increase with N. The discrete nature of the problem may prevent the statement of the optimal results in closed form. For example, $D_{MCR}(16) = 6 > 5 = D_{MCR}(18)$ and $\overline{D}_{MCR}(16) = 3.0625 > 3.0555 = \overline{D}_{MCR}(18)$; see Table B.1 for more other examples.

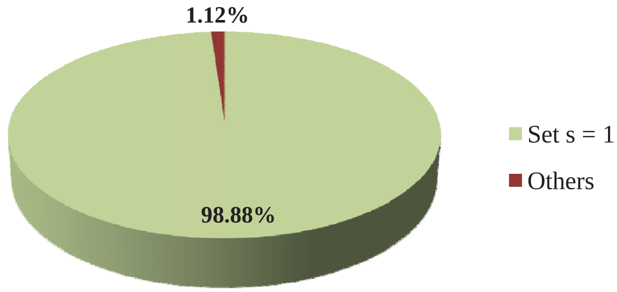


Figure 7.1: An exhaustive computer search shows that 98.88% of optimal MCRNs $MCR(N; s, w)$ can be obtained by setting $s = 1$ when $N \leq 5000$.

Nevertheless, we obtain optimal MCRNs by an exhaustive computer search for $N \leq 5000$. Among them we find that 98.88% of optimal MCRNs can be obtained by setting $s = 1$ (see Fig. 7.1). Namely, among all pairs (s, w) that minimize the diameter for a given value of N, there is one pair $(1, w)$ except for some exceptional values of N. In other words, there is no additional advantage in letting s be different from 1.

By the experiment result, the first N such that the optimal MCRN cannot be achieved by setting $s = 1$ is 30. Let $D^1_{MCR}(N)$ denote the smallest diameter of MCRN with N nodes and $s = 1$. When $N = 30$, the optimal MCRN is achieved by setting $s = 3, w = 5$ (see Theorem 5.3.1) and gives

$$d(30; 3, 5) = D_{MCR}(30) = 7,$$

while the best solution with $s = 1$ gives

$$D^1_{MCR}(30) = 9 \quad (\text{with } w = 5).$$

The first N that is not satisfying Theorem 5.3.1 and the optimal MCRN cannot be achieved by setting $s = 1$ is 1320. When $N = 1320$, the optimal MCRN is achieved by setting $s = 3, w = 95$ and gives

$$d(1320; 3, 95) = D_{MCR}(1320) = 51,$$

while the best solution with $s = 1$ gives

$$D^1_{MCR}(1320) = 53 \quad (\text{with } w = 135).$$

The optimal MCRNs that are not achieved by setting $s = 1$ when $N \leq 5000$ are shown in Table 7.1. Moreover, for all values of $N \leq 5000$ there are optimal MCRNs that minimize both the diameter and the average distance between nodes simultaneously; see Table B.1 for $N \leq 256$. Note that there are DLNs that minimize either the diameter or the average distance between nodes, but not both simultaneously; see Table B.1 for examples.

Table 7.1: The optimal MCRNs that are not achieved by setting $s = 1$ when $N \leq 5000$.

N	$D_{MCR}(N)$	s	w	$D^1_{MCR}(N)$	s'	w'
30	7 (by Theorem 5.3.1)	3	5	9	1	5
70	11 (by Theorem 5.3.1)	5	7	13	1	9
126	15 (by Theorem 5.3.1)	7	9	17	1	11
198	19 (by Theorem 5.3.1)	9	11	21	1	17
286	23 (by Theorem 5.3.1)	11	13	25	1	21
390	27 (by Theorem 5.3.1)	13	15	29	1	19
510	31 (by Theorem 5.3.1)	15	17	33	1	29
646	35 (by Theorem 5.3.1)	17	19	37	1	33
798	39 (by Theorem 5.3.1)	19	21	41	1	29
966	43 (by Theorem 5.3.1)	21	23	45	1	41
1150	47 (by Theorem 5.3.1)	23	25	49	1	39
1320	51	50	3	95	53	135
1350	51 (by Theorem 5.3.1)	25	27	53	1	49
1566	55 (by Theorem 5.3.1)	27	29	57	1	53
1798	59 (by Theorem 5.3.1)	29	31	61	1	57
2046	63 (by Theorem 5.3.1)	31	33	65	1	61
2250	67	66	3	65	69	57
2280	67	67	3	625	69	309
2310	67 (by Theorem 5.3.1)	33	35	69	1	57
2590	71 (by Theorem 5.3.1)	35	37	73	1	69
2886	75 (by Theorem 5.3.1)	37	39	77	1	73
3198	79 (by Theorem 5.3.1)	39	41	81	1	77
3526	83 (by Theorem 5.3.1)	41	43	85	1	81
3870	87 (by Theorem 5.3.1)	43	45	89	1	71
4230	91 (by Theorem 5.3.1)	45	47	93	1	89
4606	95 (by Theorem 5.3.1)	47	49	97	1	83
4914	99	98	3	581	101	87
4998	99 (by Theorem 5.3.1)	49	51	101	1	97

7.2 Performance Evaluation

In this section, we compare the MCRNs with DLNs in terms of performance parameters including the network diameter and the average distance. For the network diameter part, although the MCRN can achieve a better diameter than the DLN (see equations (1.4.1), (1.4.4) and (1.4.5)), however, the exact values of the diameter of the optimal MCRNs and the optimal DLNs are not known so far for every N. Thus, it is interesting to compare the minimum diameter as well as the minimum average distance between MCRNs and DLNs with the same number of nodes. Fig. 7.2 shows a comparison of the minimum diameter between MCRNs and DLNs. It is clear to see that the minimum diameter of the MCRN is always smaller than that of the DLN, and the gap between these two values increases markedly when number of nodes increases. This tells that the MCRN performs a better performance than the DLN in worst case of the transmission delay.

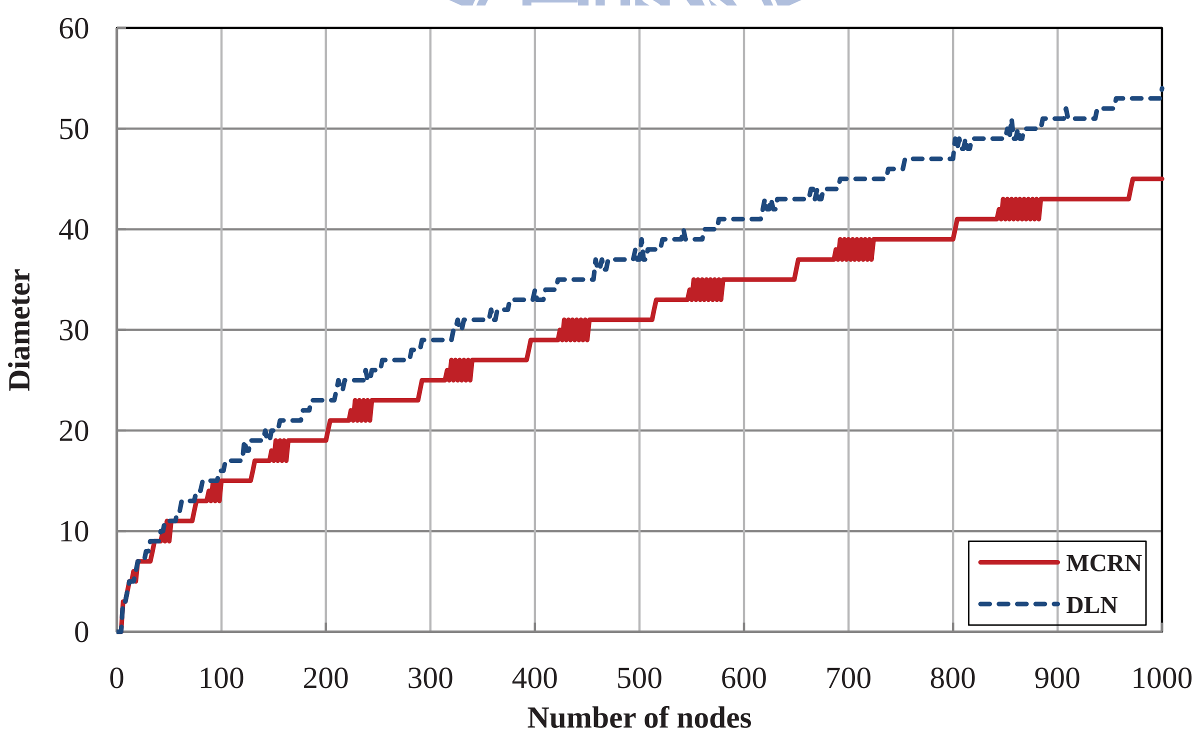


Figure 7.2: Comparing the minimum diameter between MCRNs and DLNs.

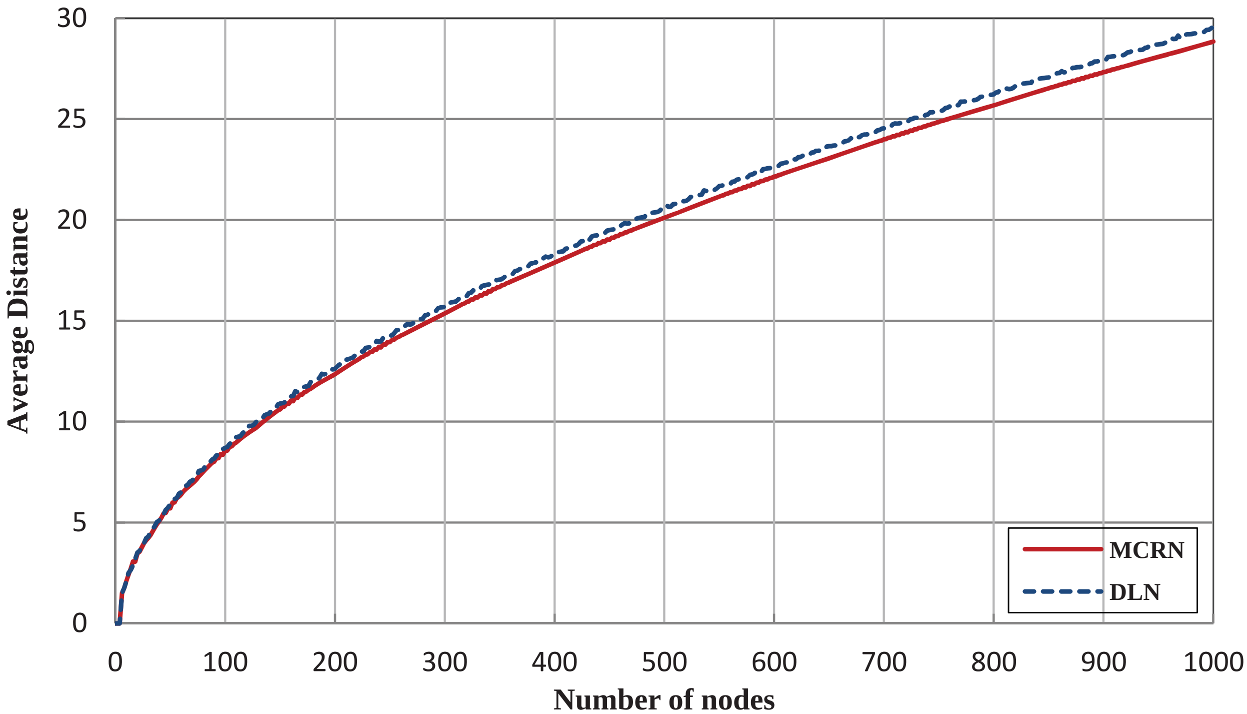


Figure 7.3: Comparing the minimum average distance between MCRNs and DLNs.

For the average distance part, Fig. 7.3 shows a comparison of the minimum average distance between MCRNs and DLNs. In this case, the minimum average distance of the MCRN is also smaller than that of the DLN, but the gap between these two values increases in a slow fashion when number of nodes increases. We can conclude that the MCRN performs a slightly better performance than the DLN in average case of the transmission delay.

Chapter 8

Conclusions

In this chapter, we present a summary of this thesis, and we discuss some directions for further research.

8.1 Summary of This Research

The double-loop network has been extensively studied in many aspects such as the minimum distance diagram, the diameter and the routing. Efficient algorithms exist for distance-related problems of the double-loop network. However, compared with the double-loop network, neither diameter-computing algorithms nor routing algorithms for the mixed chordal ring network have been addressed in the literature. In this thesis, our research goal is to improve the knowledge of the mixed chordal ring networks and is focused on solving the distance-related problems of the mixed chordal ring network: The minimum distance diagram problem, the diameter problem and the shortest path routing problem.

We first study and investigate the minimum distance diagram of the mixed chordal ring network. Specifically, we find that the minimum distance diagram of a mixed chordal ring network can be obtained easily by reassembling the PSEUDOMDD in a particular way. The tool we developed can be used to study other distance-related problems.

For the diameter-related problem, we proposed an efficient algorithm to compute the diameter of a given mixed chordal ring network. For the optimization problem of finding the optimal mixed chordal ring network, we improve previous bounds on this problem and successfully obtain a class of optimal mixed chordal ring networks. By using the presented diameter-computing algorithm, an exhaustive computer search suggests that most of the optimal mixed chordal ring networks can be achieved by setting the ring-parameter to be 1.

For the routing problem, two node-to-node routing algorithms are presented for flexible applications: shortest-path-based routing algorithm and dynamic routing algorithm. Both routing algorithms do not use routing tables and always use a shortest path to route. The shortest-path-based routing algorithm takes $O(\log N)$-time for the source node and takes constant time for the other nodes in the routing path. In the dynamic routing algorithm, after an $O(\log N)$-time to determine the network parameters, each node (including the source node) takes constant time to determine the next node on the routing path to which the message should be sent.

In addition, we also present an optimal fault-tolerant routing algorithm for MCRNs in the presence of up to one node or link failure. The fault-tolerant algorithm presented do not require routing tables and requires very little computational overhead. After an $O(\log N)$-time preprocessing, the algorithm can route messages to the destination node using a constant time at each node along the route. The fault-tolerant routing algorithm presented is guaranteed to find an optimal route after a faulty element is detected.

We believe that these results will benefit further researches on mixed chordal ring networks. In the following, a comparison between the double-loop network with the mixed chordal ring network is shown in Table 8.1. In the next section, we discuss some directions for further research on MCRNs.

Table 8.1: Comparing the double-loop network with the mixed chordal ring network.

	double-loop network	mixed chordal ring network
restriction on order	a positive integer	an even positive integer
degree	in-degree: 2, out-degree: 2	in-degree: 2, out-degree: 2
connectivity	strongly 2-connected [60]	strongly 2-connected (this thesis)
optimal diameter (lower bound)	$D_{DL}(N) \geq \lceil \sqrt{3N} \rceil - 2$ [64]	$D_{MCR}(N) \geq \lceil \sqrt{2N} - 3/2 \rceil$ (this thesis)
optimal diameter (upper bound)	$D_{DL}(N) \leq \sqrt{3N} + (3N)^{1/4} + \frac{5}{2}$ for $N \geq 1200$ [57]	$D_{MCR}(N) \leq 2 \lceil \sqrt{N/2} \rceil + 1$ (this thesis)
optimal networks	many classes [10, 14, 15, 16, 32, 59]	a class (this thesis)
computing diameter	$O(\log N)$-time [21]	$O(\log N)$-time (this thesis)
optimal routing	$O(\log N)$-time [2, 36]	$O(\log N)$-time (this thesis)
optimal fault-tolerant routing	$O(\log N)$-time [23, 49]	$O(\log N)$-time (this thesis)

8.2 Directions for Future Research

One of the most important and fundamental optimization problems in designing interconnection networks is, for a given number of nodes N, how to find an *optimal network* with the smallest diameter and to give the construction of such a network. For the double-loop network, determining the exact value of $D_{DL}(N)$ is a hard problem and even determining $\overline{D}_{DL}(N) = \min_{s_2}\{d_{DL}(N; 1, s_2)\}$, where $d_{DL}(N; 1, s_2)$ is the diameter of $DL(N; 1, s_2)$, is a hard problem, too; see [9] for more detail. By (1.4.1), (1.4.2) and (1.4.3), the gap between the upper and the lower bounds on $D_{DL}(N)$ increases by a factor of $(3N)^{1/4}$ and it seems that there is no closed form for $D_{DL}(N)$. However, for the mixed chordal ring network, we have successfully narrowed the gap between the upper and the lower bounds on $D_{MCR}(N)$ as $2\left\lceil\sqrt{N/2}\,\right\rceil + 1$ and $\left\lceil\sqrt{2N} - 3/2\,\right\rceil$. It has a great probability to determine $D_{MCR}(N)$ and therefore solve this optimization problem in the near future.

Another research perspective may take into the *weighted version* of the mixed chordal ring network for consideration. Related research results on the weighted double-loop network can be found in the literature. For example, the diameter computation in [21]; bounds on the minmum diameter and average distance in [57]; optimal fault-tolerant routing in [49]. Thus it is interesting to know whether the proposed results on mixed chordal ring networks in this thesis, including the minimum distance diagram construction, the diameter computation, and the node-to-node routing, can be easily translated into the weighted version.

From the graph theoretical viewpoint, we are also interested in the isomorphism problem. A large number of papers are devoted to the isomorphism problem for circulant graphs [1, 5, 29, 51, 52]. In addition, Barrière gave a polynomial-time algorithm to decide isomorphism between two chordal rings. The necessary and sufficient condition for two double-loop networks to be strongly isomorphic is characterized in [41]. Moreover, Hwang and Wright [41] studied the reliability of some double-loop networks by considering the nonstrongly isomorphic networks. Thus, given two mixed chordal ring networks with the same number of nodes, it is interesting to check whether or not these two networks are strongly isomorphic. For example, when $N = 20$, the nonstrongly isomorphic mixed chordal ring networks are $MCR(20; 1, 3)$, $MCR(20; 1, 5)$, $MCR(20; 1, 7)$, $MCR(20; 1, 9)$, $MCR(20; 5, 1)$. Theorem 2.3.4 provides a sufficient condition for two mixed chordal ring networks to be strongly isomorphic. However, this result does not cover all networks with the same number of nodes. Thus, determining the necessary and sufficient condition for two mixed chordal ring network to be strongly connected is another challenging direction for further research. Moreover, if the necessary and sufficient condition is not easy to obtain, then we search for an efficient algorithm to determine the isomorphism between mixed chordal ring networks.

Other research direction may take into the collective communication for consideration. The most important among these are one-to-all broadcasting (a source node sending a message to every other node), all-to-all broadcasting, all-to-all personalized exchange (every node sending a unique message to each of the other nodes), and a number of permutation routing patterns whereby each of the N nodes sends a message to a distinct node (so that N messages initially at their respective source nodes are permuted, each ending up at its destination node). Obradovic et. al. [53] studied the one-to-all broadcasting problem on the undirected double-loop networks $UDL(N; \pm a, \pm b)$ and gave the construction of optimal broadcast trees for i-port undirected double-loop networks. Hwang [40] showed that double-loop networks have parallel processing capability by giving the first permutation routing algorithm, and the number of routing steps required is equal to the diameter of the network, which is the best bound one can get. In our opinion, analyzing the collective communication problem on mixed chordal ring networks seems to be an interesting and challenging direction for further research.

Appendices

Appendix A

Cheng-Hwang-Algorithm

Input: N, s_1, s_2.

Output: The L-shape (ℓ, h, p, n) of $DL(N; s_1, s_2)$.

Step 1. Let $d = \gcd(N, s_1),\ d' = \gcd(N, s_2),\ N' = N/d,\ s_1' = s_1/d,$ and $s_2' = s_2 \pmod{N'}$.

Let $t_{-1} = N'$. Let t_0 be the integer with

$$s_1' t_0 + s_2' \equiv 0 \pmod{N'},\ 0 \le t_0 < N'.$$

Define q_i, t_i, recursively (by the Euclidean algorithm) as follows:

$$\begin{aligned}
t_{-1} &= q_1 t_0 + t_1, & 0 \le t_1 < t_0 \\
t_0 &= q_2 t_1 + t_2, & 0 \le t_2 < t_1 \\
t_1 &= q_3 t_2 + t_3, & 0 \le t_3 < t_2 \\
& \dots & \\
t_{k-2} &= q_k t_{k-1} + t_k, & 0 \le t_k < t_{k-1} \\
t_{k-1} &= q_{k+1} t_k, & 0 = t_{k+1} < t_k.
\end{aligned}$$

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

$$U_{i+1} = q_{i+1}U_i + U_{i-1},\ i = 0, 1, \cdots, k.$$

By induction,

$$t_iU_{i+1} + t_{i+1}U_i = N',\ i = 0, 1, \cdots, k.$$

Regard $t_{-1}/U_{-1} = \infty > x$ for real number x. Since $\{t_i\}_{i=-1}^{k+1}$ and $\{U_i\}_{i=-1}^{k+1}$ are strictly decreasing and increasing, respectively, we have

$$0 = \frac{t_{k+1}}{U_{k+1}} < \frac{t_k}{U_k} < \cdots < \frac{t_0}{U_0} < \frac{t_{-1}}{U_{-1}} = \infty.$$

Step 3. Let u be the largest odd integer such that $d < t_u/U_u$. Define

$$v = \left\lceil \frac{t_u - dU_u}{t_{u+1} + dU_{u+1}} \right\rceil - 1.$$

return

$$\begin{aligned} \ell &= t_u - vt_{u+1}, \\ h &= d(U_u + (v+1)U_{u+1}), \\ p &= t_u - (v+1)t_{u+1}, \\ n &= d(U_u + vU_{u+1}). \end{aligned}$$

end Cheng-Hwang-Algorithm

Appendix B

Optimal Mixed Chordal Ring Networks and Double-Loop Networks

Table B.1: Optimal MCRNs and DLNs for $N = 6, 8, \ldots, 256$.

N	$D_{MCR}(N)$	$\overline{D}_{MCR}(N)$	s	w	$D_{DL}(N)$	$\overline{D}_{DL}(N)$	a	b
6	3	1.5	1	3	3	1.5	1	2
8	3	1.75	1	3	3	1.75	1	3
10	4	2.1	1	3	4	2.1	1	3
12	5	2.416666667	1	3	5	2.5	1	3
14	5	2.714285714	1	3	5	2.642857143	1	4
16	6	3.0625	1	3	5	2.875	1	7
18	5	3.055555556	1	5	6	3.166666667	1	4
20	7	3.45	1	5	7	3.5	1	4
22	7	3.545454545	1	5	7	3.590909091	1	5
24	7	3.75	1	5	7	3.75	1	10
26	7	3.961538462	1	7	7	3.961538462	1	8
28	7	4.107142857	1	5	8	4.214285714	1	5
30	7	4.233333333	3	5	8	4.3	1	9
32	7	4.375	1	7	9	4.5625	1	6
34	8	4.558823529	1	13	9	4.676470588	1	10
36	9	4.75	1	15	9	4.833333333	1	11
38	9	4.921052632	1	7	9	5.026315789	1	9

N	$D_{MCR}(N)$	$\overline{D}_{MCR}(N)$	s	w	$D_{DL}(N)$	$\overline{D}_{DL}(N)$	a	b
40	9	5.075	1	7	9	5.1	1	12
42	9	5.238095238	1	9	10	5.357142857	1	10
44	10	5.431818182	1	13	10	5.409090909	1	13
46	9	5.47826087	1	7	11	5.630434783	1	18
48	11	5.729166667	1	21	11	5.75	1	11
50	9	5.7	1	9	11	5.9	1	15
52	11	5.980769231	1	7	11	5.961538462	1	12
54	11	6	1	15	11	6.166666667	1	16
56	11	6.196428571	1	21	11	6.214285714	1	13
58	11	6.275862069	1	9	12	6.431034483	1	11
60	11	6.4	1	9	12	6.5	1	14
62	11	6.532258065	1	11	13	6.693548387	1	14
64	11	6.640625	1	19	13	6.8125	1	12
66	11	6.742424242	1	25	13	6.863636364	1	15
68	11	6.838235294	1	9	13	7.029411765	1	13
70	11	6.928571429	5	7	13	7.071428571	1	16
72	11	7.027777778	1	11	13	7.277777778	1	20
					14	7.25	1	22
74	12	7.148648649	1	31	13	7.310810811	1	14
76	13	7.276315789	1	21	14	7.552631579	1	21
78	13	7.397435897	1	17	14	7.576923077	1	18
80	13	7.5125	1	35	14	7.625	1	15
82	13	7.62195122	1	11	15	7.792682927	1	23
84	13	7.726190476	1	11	15	7.928571429	1	16
86	13	7.837209302	1	13	15	7.965116279	1	16
88	14	7.965909091	1	19	15	8.113636364	1	14
90	13	8.022222222	1	33	15	8.166666667	1	17
92	15	8.184782609	1	21	15	8.369565217	1	21
					16	8.326086957	1	17
94	13	8.191489362	1	11	15	8.393617021	1	15
96	15	8.385416667	1	21	15	8.4375	1	18
98	13	8.357142857	1	13	16	8.642857143	1	27
100	15	8.57	1	45	16	8.7	1	16
102	15	8.568627451	1	39	16	8.735294118	1	19
104	15	8.740384615	1	11	17	8.884615385	1	29
106	15	8.773584906	1	19	17	9.009433962	1	20

N	$D_{MCR}(N)$	$\overline{D}_{MCR}(N)$	s	w	$D_{DL}(N)$	$\overline{D}_{DL}(N)$	a	b
108	15	8.898148148	1	45	17	9.055555556	1	17
110	15	8.963636364	1	13	17	9.227272727	1	24
112	15	9.053571429	1	13	17	9.25	1	21
114	15	9.149122807	1	15	17	9.289473684	1	18
116	15	9.232758621	1	25	17	9.465517241	1	16
					18	9.431034483	1	45
118	15	9.313559322	1	27	17	9.516949153	1	22
120	15	9.391666667	1	33	17	9.55	1	19
122	15	9.467213115	1	51	19	9.795081967	1	19
124	15	9.540322581	1	13	18	9.790322581	1	17
126	15	9.611111111	7	9	18	9.833333333	1	20
128	15	9.6875	1	15	19	9.96875	1	20
130	16	9.776923077	1	57	19	10.11538462	1	18
132	17	9.871212121	1	39	19	10.13636364	1	18
134	17	9.962686567	1	29	19	10.17164179	1	21
136	17	10.05147059	1	31	19	10.32352941	1	32
138	17	10.13768116	1	21	19	10.36956522	1	19
140	17	10.22142857	1	63	19	10.4	1	22
142	17	10.3028169	1	15	20	10.52816901	1	31
144	17	10.38194444	1	15	19	10.61111111	1	56
146	17	10.46575342	1	17	19	10.65068493	1	20
148	18	10.56081081	1	41	20	10.85135135	1	18
150	17	10.61333333	1	55	20	10.9	1	63
152	19	10.73026316	1	27	20	10.92105263	1	24
154	17	10.75324675	1	47	20	10.95454545	1	21
156	19	10.89102564	1	29	21	11.07692308	1	34
158	17	10.88607595	1	15	21	11.20886076	1	19
160	19	11.04375	1	43	21	11.25	1	22
162	17	11.01851852	1	17	21	11.27777778	1	22
164	19	11.18902439	1	25	21	11.57317073	1	44
					22	11.51219512	1	26
166	19	11.18072289	1	49	21	11.46385542	1	20
168	19	11.32738095	1	77	21	11.5	1	23
170	19	11.34117647	1	65	21	11.67647059	1	66
					22	11.61764706	1	23
172	19	11.45930233	1	15	21	11.72093023	1	27

N	$D_{MCR}(N)$	$\overline{D}_{MCR}(N)$	s	w	$D_{DL}(N)$	$\overline{D}_{DL}(N)$	a	b
174	19	11.49425287	1	23	21	11.74137931	1	21
176	19	11.58522727	1	77	21	11.77272727	1	24
178	19	11.64044944	1	17	22	11.98314607	1	28
					23	11.97191011	1	24
180	19	11.71111111	1	17	22	12	1	39
182	19	11.78571429	1	19	22	12.03846154	1	22
184	19	11.85326087	1	51	22	12.06521739	1	25
186	19	11.91935484	1	33	23	12.17741935	1	82
188	19	11.98404255	1	35	23	12.43617021	1	36
					24	12.37234043	1	41
190	19	12.04736842	1	41	23	12.34210526	1	26
192	19	12.109375	1	69	23	12.375	1	23
194	19	12.17010309	1	85	23	12.53092784	1	21
196	19	12.22959184	1	17	23	12.57142857	1	35
198	19	12.28787879	9	11	23	12.59090909	1	27
200	19	12.35	1	19	23	12.62	1	24
202	20	12.42079208	1	91	23	12.80693069	1	36
					24	12.72772277	1	60
204	21	12.49509804	1	75	23	12.82352941	1	22
206	21	12.56796117	1	47	23	12.8592233	1	28
208	21	12.63942308	1	37	23	12.88461538	1	25
210	21	12.70952381	1	39	24	13.07142857	1	40
212	21	12.77830189	1	57	25	13.09433962	1	38
214	21	12.84579439	1	25	24	13.13551402	1	23
216	21	12.91203704	1	99	24	13.16666667	1	26
218	21	12.97706422	1	19	25	13.2706422	1	26
220	21	13.04090909	1	19	25	13.40909091	1	30
222	21	13.10810811	1	21	25	13.44594595	1	24
224	22	13.18303571	1	51	25	13.46428571	1	24
226	21	13.2300885	1	69	25	13.49115044	1	27
228	23	13.32017544	1	87	25	13.65789474	1	31
230	21	13.34782609	1	95	25	13.67391304	1	41
232	23	13.45258621	1	35	25	13.70689655	1	25
234	21	13.46153846	1	69	25	13.73076923	1	28
236	23	13.58050847	1	37	25	13.89830508	1	23

N	$D_{MCR}(N)$	$\overline{D}_{MCR}(N)$	s	w	$D_{DL}(N)$	$\overline{D}_{DL}(N)$	a	b
					26	13.83050847	1	66
238	21	13.57142857	1	19	26	14.14705882	1	36
					27	14.00420168	1	38
240	23	13.70416667	1	93	25	13.95833333	1	71
242	21	13.68181818	1	21	25	13.98760331	1	26
244	23	13.82377049	1	57	26	14.20491803	1	33
					27	14.1557377	1	29
246	23	13.81300813	1	75	26	14.20731707	1	24
					27	14.18292683	1	44
248	23	13.93951613	1	29	26	14.24193548	1	92
250	23	13.944	1	105	26	14.26	1	30
252	23	14.0515873	1	117	26	14.28571429	1	27
254	23	14.07086614	1	75	27	14.38188976	1	71
256	23	14.16015625	1	19	27	14.53125	1	25

References

[1] A. Adam. Research problem 2-10. *J. Combin. Theory*, 2:393, 1967.

[2] F. Aguiló and M. A. Fiol. An efficient algorithm to find optimal double loop networks. *Discrete Mathematics*, 138:15–29, 1995.

[3] F. Aguiló and A. Mirallés. Pomdig: Construcción de l-formas implementables. In *IV Jornadas de Matemática Discreta y Algorítmica. Universida Polite'cnica de Madrid*, pages 109–116, 2004.

[4] F. Aguilo, E. Simo, and M. Zaragoza. Optimal double-loop networks with non-unit steps. *The Electronic Journal of Combinatorics*, 10(1), 2003.

[5] B. Alspach and T. D. Parsons. Isomorphism of circulant graphs and digraphs. *Discrete Mathematics*, 25(2):97–108, 1979.

[6] B. W. Arden and H. Lee. Analysis of chordal ring network. *IEEE Transactions on Computers*, 30(4):291–295, 1981.

[7] J. B. Babcock. Sonet: a practical perspective. *Business Communications Review*, pages 59–63, SEP 1990.

[8] L. Barriere. Symmetry properties of chordal rings of degree 3. *Discrete Applied Mathematics*, 129(2-3):211–232, 2003.

[9] J. C. Bermond, F. Comellas, and D. F. Hsu. Distributed loop computer-networks - a survey. *Journal of Parallel and Distributed Computing*, 24(1):2–10, 1995.

[10] J. C. Bermond and D. Tzvieli. Minimal diameter double-loop networks: dense optimal families. *Networks*, 24(1):1–9, 1991.

[11] R. F. Browne and R. M. Hodgson. Symmetrical degree-4 chordal ring networks. *IEE Proceedings-E Computers and Digital Techniques*, 137(4):310–318, 1990.

[12] R. Chan, C. Chen, and Z. Hong. A simple algorithm to find the steps of double-loop networks. *Discrete Applied Mathematics*, 121(1-3):61–72, 2002.

[13] G. Chartrand and L. Lensniak. *Graph and digraphs*. Wadsworth, Monterey, CA, 1981.

[14] B. X. Chen, X. B. Chen, J. X. Meng, and W. J. Xiao. The construction of infinite families of any k-tight optimal and singular k-tight optimal directed double loop networks. *Science in China Series A-Mathematics*, 50(7):1055–1064, 2007.

[15] B. X. Chen, J. X. Meng, and W. J. Xiao. Some new optimal and suboptimal infinite families of undirected double-loop networks. *Discrete Mathematics and Theoretical Computer Science*, 8(1):299–311, 2006.

[16] B. X. Chen and W. J. Xiao. Optimal designs of directed double-loop networks. In *Computational and Information Science, Proceedings*, volume 3314 of *Lecture Notes in Computer Science*, pages 19–24, 2004.

[17] C. Chen and F. K. Hwang. Equivalent l-shapes of double-loop networks for the degenerate case. *Journal of Interconnection Networks*, 1(1):47–60, 2000.

[18] C. Chen and F. K. Hwang. The minimum distance diagram of double-loop networks. *IEEE Transactions on Computers*, 49(9):977–979, 2000.

[19] C. Chen, J. K. Lan, and W. S. Tang. An efficient algorithm to find a double-loop network that realizes a given l-shape. *Theoretical Computer Science*, 359(1-3):69–76, 2006.

[20] S. K. Chen, F. K. Hwang, and Y. C. Liu. Some combinatorial properties of mixed chordal rings. *Journal of Interconnection Networks*, 1:3–16, 2003.

[21] Y. Cheng and F. K. Hwang. Diameters of weighted double loop networks. *Journal of Algorithms*, 9:401–410, 1988.

[22] Y. Cheng, F. K. Hwang, I. F. Akyildiz, and D. F. Hsu. Routing algorithms for double loop networks. *International Journal of Foundations of Computer Science*, 3(3):323–331, 1992.

[23] C. Y. Chou, D. J. Guan, and K. L. Wang. A dynamic fault-tolerant message routing algorithm for double-loop networks. *Information Processing Letters*, 70(6):259–264, 1999.

[24] F. Comellas and P. Hell. Broadcasting in generalized chordal rings. *Networks*, 42(3):123–134, 2003.

[25] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. The MIT Press, second edition, 2001.

[26] W. Dally and B. Towles. *Principles and practices of interconnection networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[27] H. P. Dharmasena and X. Yan. An optimal fault-tolerant routing algorithm for weighted bidirectional double-loop networks. *IEEE Transactions on Parallel and Distributed Systems*, 16(9):841–852, 2005.

[28] T. Dobravec and B. Robic. Restricted shortest paths in 2-circulant graphs. *Computer Communications*, 32(4):685–690, 2009.

[29] E. Dobson. On isomorphisms of circulant digraphs of bounded degree. *Discrete Mathematics*, 308(24):6047–6055, DEC 28 2008.

[30] D. Z. Du, D. F. Hsu, Q. Li, and J. Xu. A combinatorial problem related to distributed loop networks. *Networks*, 20(2):173–180, 1990.

[31] J. Duato, S. Yalamanchili, and N. Lionel. *Interconnection networks: an engineering approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[32] P. Esqué, F. Aguiló, and M. A. Fiol. Double commutative-step digraphs with minimum diameters. *Discrete Mathematics*, 114:147–157, 1993.

[33] M. A. Fiol, J. L. A. Yebra, I. Alegre, and M. Valero. A discrete optimization problem in local networks and data alignment. *IEEE Transactions on Computers*, C-36(6):702–713, 1987.

[34] D. Gomez, J. Gutierrez, and A. Ibeas. Cayley digraphs of finite abelian groups and monomial ideals. *SIAM Journal on Discrete Mathematics*, 21(3):763–784, 2007.

[35] D. Gómez, J. Gutierrez, and Á. Ibeas. Optimal routing in double loop networks. *Theoretical Computer Science*, 381(1-3):68–85, 2007.

[36] D. J. Guan. An optimal message routing algorithm for double-loop networks. *Information Processing Letters*, 65(5):255–260, 1998.

[37] L.-H. Hsu and C.-K. Lin. *Graph theory and interconnection networks*. CRC Press, Inc., Boca Raton, FL, USA, 2008.

[38] F. K. Hwang. A complementary survey on double-loop networks. *Theoretical Computer Science*, 263(1-2):211–229, 2001.

[39] F. K. Hwang. A survey on multi-loop networks. *Theoretical Computer Science*, 299(1-3):107–121, 2003.

[40] F. K. Hwang, T. S. Lin, and R.-H. Jan. A permutation routing algorithm for double loop networks. *Parallel Processing Letters*, 7(3):259–265, 1997.

[41] F. K. Hwang and P. E. Wright. Survival reliability of some double-loop networks and chordal rings. *IEEE Transactions on Computers*, 44(12):1468–1471, 1995.

[42] F. K. Hwang and Y. H. Xu. Double loop networks with minimum delay. *Discrete Mathematics*, 66(1-2):109–118, 1987.

[43] J. K. Lan and C. Chen. On distance-related problems of mixed chordal ring networks. *Submit to Discrete Applied Mathematics*.

[44] J. K. Lan, V. W. Liu, and C. Chen. Improved upper and lower bounds on the optimization of mixed chordal ring networks. *Information Processing Letters*, 109(13):757–762, 2009.

[45] J. S. Lee, J. K. Lan, and C. Chen. On degenerate double-loop l-shapes. *Journal of Interconnection Networks*, 7:195–215, 2006.

[46] F. T. Leighton. *Introduction to parallel algorithms and architectures: array, trees, hypercubes*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.

[47] M. T. Liu. Distributed loop computer networks. *Advances in Computers*, 17:163–221, 1978.

[48] V. W. Liu. On the diameter of a mixed chordal ring network. Master's thesis, National Chiao Tung University, Taiwan, 2004.

[49] Y. L. Liu, Y. L. Wang, and D. J. Guan. An optimal fault-tolerant routing algorithm for double-loop networks. *IEEE Transactions on Computers*, 50(5):500–505, 2001.

[50] P. Morillo, F. Comellas, and M. A. Fiol. The optimization of chordal ring networks. In *Proc. International Conference on Communication Technology, ICCT*, pages 295–299, 1987.

[51] M. Muzychuk. A solution of the isomorphism problem for circulant graphs. *Proceedings of The London Mathematical Society*, 88(Part 1):1–41, JAN 2004.

[52] A. Nayak, V. Accia, and P. Gissi. A note on isomorphic chordal rings. *Information Processing Letters*, 55(6):339–341, SEP 29 1995.

[53] N. Obradovic, J. Peters, and G. Ruzic. Reliable broadcasting in double loop networks. *Networks*, 46(2):88–97, 2005.

[54] B. Parhami. *Introduction to parallel processing: algorithms and architectures.* Kluwer Academic Publishers, Norwell, MA, USA, 1999.

[55] B. Parhami and D. M. Kwai. Challenges in interconnection network design in the era of multiprocessor and massively parallel microchips. In *Proc. Int'l Conf. Comm. in Computing*, pages 241–246, June 2000.

[56] B. Parhami and M. A. Rakov. Performance, algorithmic, and robustness attributes of perfect difference networks. *IEEE Trans. Parallel Distrib. Syst.*, 16(8):725–736, 2005.

[57] Ö. J. Rödseth. Weighted multi-connected loop networks. *Discrete Mathematics*, 148:161–173, 1996.

[58] P. Sabariego and F. Santos. Triple-loop networks with arbitrarily many minimum distance diagrams. *Discrete Mathematics*, 309(6):1672–1684, 2009.

[59] D. Tzvieli. Minimal diameter Double-Loop Networks. I. Large infinite optimal families. *Networks*, 21(4):387–415, 1991.

[60] E. A. Van Doorn. Connectivity of circulant digraphs. *Journal of Graph Theory*, 10(1):9–14, 1986.

[61] A. Varma and C. S. Raghavendra. *Interconnection networks for multiprocessors and multicomputers: theory and practice*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1993.

[62] J. Žerovnik and T. Pisanski. Computing the diameter in multiple-loop networks. *J. Algorithms*, 14(2):226–243, 1993.

[63] D. B. West. *Introduction to graph theory*. Prentice Hall, Upper Saddle River, NJ, second edition, 2001.

[64] C. K. Wong and D. Coppersmith. A combinatorial problem related to multimodule memory organizations. *Journal of ACM*, 21(3):392–402, 1974.

[65] J.-S. Yang, J.-M. Chang, S.-M. Tang, and Y.-L. Wang. Reducing the height of independent spanning trees in chordal rings. *IEEE Trans. Parallel Distrib. Syst.*, 18(5):644–657, 2007.

[66] J. Zerovnik and T. Pisanski. Computing the diameter in multiple-loop networks. *Journal of Algorithms*, 14(2):226–243, 1993.