Permutation Routing in All-Optical Networks

Weifa Liang[†‡] George Havas[†] Xiaojun Shen[§]

† Computer Science and Electrical Engineering
‡ Department of Mathematics
The University of Queensland
Queensland 4072, Australia

§ Computer Science Telecommunications Program
University of Missouri - Kansas City
Kansas City, MO 64110, USA

Abstract

We study permutation routing techniques for all-optical networks. In these networks, messages travel in optical form and switching is performed directly on the optical signal. By using different wavelengths, several messages may use the same fiber-optic link concurrently. However, messages assigned the same wavelength must use disjoint paths, or else be routed in separate rounds.

First we present some lower bounds on the number of wavelengths needed for implementing any permutation on an all-optical network in terms of bisection and edge-connectivity of the network. We propose an algorithm for implementing in one round any permutation in a directed symmetric network with $O(\frac{n}{\lambda})$ wavelengths on the wavelength non-conversion model, where n is the number of nodes and λ is the edge connectivity of the network.

Then we study permutation routing on product networks which are defined as a direct product of two networks. Product networks are important because many well known commercial networks such as hypercubes, meshes, tori, etc, are product networks. For product networks, we obtain a lower bound on the number of wavelengths needed for implementing any permutation, and present permutation routing algorithms for the wavelength non-conversion and conversion models, respectively.

Finally we investigate permutation routing on cube-connected-cycles networks. For such networks with constant degree three we show that the number of wavelengths needed for implementing any permutation in one round is $\lfloor 2\log n \rfloor$, which improves on a previously known general result for bounded degree graphs by a factor of $O(\log^3 n)$ for this special case.

Keywords: *Optical networks, permutation routing, product networks, wavelength assignment, algorithm design and analysis, graph theory.*

1 Introduction

In this paper we address designing routing algorithms for an emerging new generation of networks known as *all-optical networks* [1, 10, 16, 24]. This kind of network offers the possibility of interconnecting hundreds to thousands of users, covering local to wide area, and providing capacities exceeding substantially those a conventional network can provide. The network promises data transmission rates several orders of magnitudes higher than current electronic networks. The key to high speed in the network is to maintain the signal in optical form rather than traditional electric form. The high bandwidth of the fiber-optic links is utilized through *Wavelength-Division Multiplexing* (WDM) technology which supports the propagation of multiple laser beams through a single fiber-optic link provided that each laser beam uses a distinct optical wavelength. Intuitively we may think of light rays of different colors. Each laser beam, viewed as a carrier signal, provides transmission rates of 2.5 to 10 Gbps, thus the significance of WDM in very high speed networking is paramount. The major applications of the network are in video conferencing, scientific visualization and real-time medical imaging, supercomputing and distributed computing [24, 26]. A comprehensive overview of the physical theory and applications of this technology can be found in the books by Green [10] and McAulay [16].

Some studies [1, 18, 23] model the all-optical network as an *undirected* graph, and thus the routing paths for implementing requests on this network are also undirected. However, it has since become apparent that optical amplifiers placed on fiber-optic links are directed devices. Following recent studies [5, 9, 17], we model the all-optical network as a *directed symmetric* graph, and each routing path on it is a directed path.

The Model. An all-optical network consists of vertices (nodes, stations, processors, etc), interconnected by point-to-point fiber-optic links. Each fiber-optic link supports a given number of wavelengths. The vertex may be occupied either by terminals, switches, or both. *Terminals* send and receive signals. *Switches* direct the input signals to one or more of the output links. Several types of optical switches exist. The *elementary switch* is capable of directing the signals coming along each of its input links to one or more of the output ones. The elementary switch cannot, however, differentiate between different wavelengths coming along the same link. Rather, the entire signal is directed to the same output(s) [6, 1, 23]. The *generalized switch*, on the other hand, is capable of switching incoming signals based on their wavelengths [1, 23]. Using acousto-optic filters, the switch splits the incoming signals to different streams associated with various wavelengths, and may direct them to separate outputs. In both cases, on any given link different messages must use different wavelengths. Unless otherwise specified, we adopt generalized switches for our routing algorithms. Besides there being a difference in the use of switches, there is a difference regarding the wavelength assignment for routing paths. In most cases, we only allow the assignment of a unique wavelength to each routing path. We call this WDM model the *Wavelength Non-Conversion Model.* If we allow each routing path to be assigned different wavelengths for its different segments, we call it a *Wavelength Conversion Model* (called the λ*-routing model* in [7]).

Previous Related Work. Optical routing in an arbitrary undirected network G was considered by Raghavan and Upfal [23]. They proved an $\Omega(1/\beta^2)$ lower bound on the number of wavelengths needed to implement any permutation in one round, where β is the edge expansion of G (which is defined later). For an upper bound, they presented an algorithm which implements any permutation on bounded degree graphs in $O(\log^2 n/\log^2\lambda)$ wavelengths within one round with high probability, where n is the number of nodes in G and λ is the second largest eigenvalue (in absolute value) of the transition matrix of the standard random walk on G. For degree d arrays, they presented an algorithm with an $O(dn^{1/d}/\log n)$ worst case performance. Aumann and Rabani [5] presented a near optimal implementation algorithm for bounded degree networks in one round. Their algorithm needs $O(\log^2 n/\beta^2)$ wavelengths. For any bounded dimension array, any given number of wavelengths and an instance I, Aumann and Rabani [5] suggested an algorithm which realizes I using at most $O(\log n \log |I| T_{opt}(I))$ rounds, where $T_{opt}(I)$ is the minimum number of rounds necessary to implement I. Later, Kleinberg and Tardos [13] obtained an improved bound of $O(\log n)$ on the approximation of the number of rounds required to implement I. Recently, Rabani [22] further improved the result in [13] to $O(poly\ \log\log n T_{opt}(I))$. Pankaj [18] considered the permutation routing issue in hypercube based networks. For hypercube, shuffle-exchange, and De Bruijn networks, he showed that routing can be achieved with $O(\log^2 n)$ wavelengths. Aggrawal et al [1] showed that $O(\log n)$ wavelengths are sufficient for the routing in this case. Aumannn and Rabani [5] demonstrated that routing on hypercubes can be finished with a constant number of wavelengths. Gu and Tamaki [11] further showed that two wavelengths for directed symmetric hypercubes and eight wavelengths for undirected hypercubes are sufficient.

Pankaj [18, 19] proved an $\Omega(\log n)$ worst case lower bound on the number of wavelengths needed for routing a permutation in one round on a bounded degree network. Barry and Humblet [6, 7] gave bounds for routing in passive (switchless) and λ- networks. An almost matching upper bound is presented later in [1]. Peiris and Sasaki [20] considered bounds for elementary switches. The connection between packet routing and optical routing is discussed in [1]. The integral multicommodity flow problem related to optical routing has been discussed in [5, 3, 4]. A comprehensive survey for optical routing is in Beauquier et al [9].

Our Results. We first present some lower bounds on the number of wavelengths needed for implementing any permutation on all-optical networks in terms of bisection and edge-connectivity of the networks We propose an algorithm for implementing any permutation in a directed symmetric network in one round with $\lceil \frac{n+3}{\lambda} \rceil$ wavelengths on the wavelength non-conversion model, where λ is the edge connectivity value of the network. We then deal with product networks, which can be decomposed into a *direct product* of two networks. Many well known networks such as hypercubes, meshes, tori, etc, are product networks. Therefore, studying the wavelength routing on product networks is important. In Section 4 we first show a lower bound on the number of wavelengths needed for implementing any permutation on product networks, then present permutation routing algorithms for product networks, based on the wavelength non-conversion and conversion models respectively. Finally we discuss the permutation issue on cube-connected-cycles networks. The number of wavelengths needed for implementing any permutation on such networks with constant degree three in one round is $\lfloor 2 \log n \rfloor$. This improves on a general result for bounded degree networks in [5] by a factor of $O(\log^3 n)$ for this special case.

2 Preliminaries

We now define some basic concepts involved in this paper.

Definition 1 *An all-optical network can be modeled as a* directed symmetric *graph $G(V, E)$ with $|V| = n$ and $|E| = m$, where for each pair of vertices u and v, if a directed edge $\langle u, v\rangle \in E$, then $\langle v, u\rangle \in E$ too.*

Definition 2 *A* request *is an ordered pair of vertices (u, v) in G which corresponds to a message to be sent from u to v. u is the* source *of v and v is the* destination *of u.*

Definition 3 *An* instance *I is a collection of requests. If $I = \{(i, \pi(i)) \mid i \in V\}$, where π is a permutation of vertices in G, each vertex appears in I as the source and destination exactly once.*

Definition 4 *Let $P(x, y)$ denote a directed path in G from x to y. A* routing *for an instance I is a set of directed paths $\mathcal{R} = \{P(x, y) \mid (x, y) \in I\}$. An instance I is* implemented *by assigning wavelengths to the routing paths and setting the switches accordingly.*

Definition 5 *The* conflict graph*, associated with a permutation routing*

$$\mathcal{R} = \{P(i, \pi(i)) \mid i \in V,\ i \text{ is a source and } \pi(i) \text{ is the destination of } i\}$$

on a directed or undirected graph $G(V, E)$, is an undirected graph $G_{\mathcal{R},\pi} = (V', E')$, where each directed (undirected) routing path in $\mathcal{R}$ is a vertex of V' and there is an edge in E' if the two corresponding paths in $\mathcal{R}$ share at least a common directed (or undirected) edge of G.

Definition 6 *The* edge-expansion *$\beta(G)$ of $G(V, E)$ is the minimum, over all subsets S of vertices, $|S| \le n/2$, of the ratio of the number of edges leaving S to the size of S $(\subset V)$.*

Definition 7 *A* bisection *of a graph is defined as follows. Let $G(V, E)$ be an undirected graph and partition the vertex set V into two disjoint subsets V_1 and V_2 such* *that* $|V_1| = \lceil |V|/2 \rceil$ *and* $|V_2| = \lfloor |V|/2 \rfloor$. *The* bisection problem *is to find a partition* (V_1, V_2) *of* V *such that* $|C|$ *is minimized, where*

$$C = \{(i,j) \mid i \in V_2,\ j \in V_1,\ \text{and}\ \ (i,j) \in E\}.$$

Let $c(G) = |C|$. *The bisection concept for undirected graphs can be extended to directed graphs. For the directed version, define*

$$C = \{\langle i,j\rangle \mid i \in V_2,\ j \in V_1,\ \text{and}\ \ \langle i,j\rangle \in E\}.$$

Definition 8 *The* congestion *of a permutation* π *on* $G(V,E)$ *is defined as follows. Let* $\mathcal{R}$ *be a set of routing paths for* π *on* G. *Define*

$$C(e,\mathcal{R}) = \{P(i,\pi(i)) \mid i \in V,\ e \in P(i,\pi(i)),\ \text{and}\ \ P(i,\pi(i)) \in \mathcal{R}\}.$$

Then, the congestion problem for π *on* G *is to find a routing* $\mathcal{R}$ *such that* $\max_{e\in E}\{|C(e,\mathcal{R})|\}$ *is minimized. Denote by*

$$congest(G,\pi) = \min_{\mathcal{R}} \max_{e\in E}\{|C(e,\mathcal{R})|\}.$$

Let Π *be the set of all permutations on* G. *Then, the congestion of* G, $congest(G)$, *is defined as*

$$congest(G) = \max_{\pi\in\Pi}\{congest(G,\pi)\}.$$

In this paper, *an algorithm* means that the algorithm runs in polynomial time.

3 Lower Bounds

To show a certain number of wavelengths, say $w_{\min}$, is the minimum needed to solve the permutation problem on an all-optical network G, we first give a lower bound on the number of wavelengths for routing on G in terms of a bisection (V_1, V_2) of G.

Theorem 1 *For any all-optical network* $G(V,E)$, *let* $c(G)$ *be the number of edges in a bisection of* G, $congest(G)$ *be the congestion of* G, *and* $w_{\min}$ *be the number of wavelengths needed to implement any permutation on* G *in one round. Then*

$$w_{\min} \geq congest(G) \ \geq\ \frac{n-1}{2c(G)}.$$

Proof It is easy to see that $w_{\min} \geq congest(G)$ by the congestion definition and the optical routing rule that different signals through a single fiber-optic link must be assigned different wavelengths.

Let (V_1, V_2) be a bisection of G and $|V_1| \geq |V_2|$. Assume that there is a permutation π which permutes all vertices in V_2 to V_1. Then the congestion of G for π is $congest(G,\pi) \geq \lfloor n/2 \rfloor / c(G) \geq \frac{n-1}{2c(G)}$, Since $congest(G) \geq congest(G,\pi)$, the theorem then follows. □

Remark Theorem 1 always holds no matter whether the WDM model is the wavelength conversion model or not. □

From this theorem we have the following corollaries.

Corollary 1 *In a directed symmetric chain L_n of n vertices, the number of wavelengths for implementing any permutation on it in one round is at least $\lfloor n/2 \rfloor$.*

Proof The proof is straightforward, omitted. □

Corollary 2 *In a directed symmetric ring R_n of n vertices, the number of wavelengths for implementing any permutation on it in one round is at least $\frac{n-1}{4}$.*

Proof Let $w_{\min}$ be the minimum number of wavelengths needed on R_n for implementing any permutation in one round. Then $w_{\min} \geq \lceil \lfloor n/2 \rfloor /2 \rceil \geq \lceil \frac{n-1}{4} \rceil$ because $c(R_n) = 2$, by Theorem 1. □

Lemma 1 *In a directed symmetric ring R_n of n vertices, there is an efficient algorithm for implementing any permutation on it in one round by using at most $\lceil n/4 \rceil$ wavelengths, based on the wavelength non-conversion model.*

Proof By the algorithm due to Shen et al [25], any permutation can be implemented in $\lceil n/4 \rceil$ rounds on a circuit-switched R_n. They showed that this bound is tight. So, if $\lceil n/4 \rceil$ wavelengths are available, then any permutation on R_n can be implemented in one round. □

Clearly, Lemma 1 delivers a better result than that of Corollary 2.

Corollary 3 *Let M be a directed symmetric $l \times h$ mesh with $n = hl$ vertices. Suppose $l \leq h$. The number of wavelengths for implementing any permutation on M in one round is at least $\frac{\sqrt{n}}{2} - 1$.*

Proof Since $l \leq h$ and $n = hl$, then $l + 1 \leq \sqrt{n} + 1$. Furthermore, it is well known that $l \leq c(M) \leq l + 1$. Thus, the number of wavelengths needed is at least $\frac{\lfloor n/2 \rfloor}{c(M)} \geq \frac{n}{2(l+1)} - \frac{1}{(l+1)} \geq \frac{(\sqrt{n}-1)(\sqrt{n}+1)+1}{2(l+1)} - \frac{1}{(l+1)} \geq \frac{\sqrt{n}-1}{2} - \frac{1}{2(l+1)} \geq \frac{\sqrt{n}}{2} - 1$. □

From Corollary 3, we know that if we want to implement any permutation in a 2-D mesh in one round, the number of wavelengths needed is $\Omega(\sqrt{n})$.

From Theorem 1, we see that *congest*(G) is a lower bound on the number of wavelengths needed for implementing any permutation routing on G. However, the problem of computing *congest*(G) itself is NP-complete [12]. So, in practice, for this problem people devise approximation algorithms for finding an approximate solution for it in polynomial time. Those approximate solutions for the congestion problem lead to a permutation routing algorithm for the network with $f(\mathit{congest}(G, \mathcal{A}, \pi))$ wavelengths, where $\mathit{congest}(G, \mathcal{A}, \pi)$ is the congestion obtained by the approximation algorithm $\mathcal{A}$ for permutation π, and f is a function of $\mathit{congest}(G, \mathcal{A}, \pi)$. We state this in the following lemma.

Lemma 2 *Assume that for an arbitrary all-optical network $G(V, E)$ with a given permutation π, there is a permutation algorithm $\mathcal{A}$ which routes all sources to their destinations for π with congestion $\mathit{congest}(G, \mathcal{A}, \pi)$. Then implementing π on G can be done either with $\mathit{congest}(G, \mathcal{A}, \pi)$ wavelengths if we use the wavelength conversion model; or with $l_{\max}\mathit{congest}(G, \mathcal{A}, \pi) + 1$ wavelengths if we use the wavelength non-conversion model, where $l_{\max}$ is the length of the longest path for π given by algorithm $\mathcal{A}$.*

Proof Let $\mathcal{R} = \{P(x, y) \mid P(x, y)$ is the routing path from x to y given by $\mathcal{A}\}$. We construct a bipartite graph $G_B(X, Y, E_B)$ as follows. Let $X = \mathcal{R}$ and Y be the set of all directed edges in G. If $P(x, y)$ uses an edge $\langle u, v\rangle \in E$, then there is an edge in E_B between these two vertices. Obviously G_B is a bipartite multigraph, and the maximum degree of vertices in Y is equal to $\mathit{congest}(G, \mathcal{A}, \pi)$. G_B can be edge-colored with $\mathit{congest}(G, \mathcal{A}, \pi)$ colors such that all edges incident to a vertex in Y are colored with different colors. Note that this coloring is different from the traditional edge-coloring because we allow some edges incident to a vertex in X to be colored with the same color. As results, each color corresponds to a wavelength. The coloring is done as follows. First sort the vertices in X, and assign each of them a rank according to the sorting. Then, for each vertex $y \in Y$ we sort each edge incident to y by the rank of its other endpoint as the key. Finally we assign the ith edge incident to y with color i. Thus, the edges in each routing path are assigned with colors (wavelengths). So, implementing π on the wavelength conversion model can be done in one round with $\mathit{congest}(G, \mathcal{A}, \pi)$ wavelengths.

If we only allow the assignment of a unique wavelength to each routing path, then we proceed as follows. We construct a conflict graph $G_{\mathcal{R},\pi}$. The chromatic number $\chi(G_{\mathcal{R},\pi})$ is the minimum number of wavelengths needed for implementing π on G in one round. However, finding the chromatic number of a graph is NP-complete. We instead find an approximation solution for this classic vertex coloring problem. Since the maximum degree of $G_{\mathcal{R},\pi}$ is no more than $l_{\max}\mathit{congest}(G, \mathcal{A}, \pi)$, we can use $l_{\max}\mathit{congest}(G, \mathcal{A}, \pi) + 1$ colors to color all vertices of G such that any two adjacent vertices are colored with different colors. Therefore, the corresponding path of a vertex in $G_{\mathcal{R},\pi}$ is assigned a wavelength according to the color of the vertex. □

We next give a lower bound on the number of wavelengths needed to implement any permutation on some network G in one round in terms of the *edge-connectivity* $\lambda(G)$ (λ for short) of G. We start with the following theorem.

Theorem 2 *There exists a directed symmetric all-optical network G such that, to implement any permutation on it in one round, the number of wavelengths needed is $\Omega(\frac{n}{2\lambda})$, where λ is the edge connectivity of G.*

Proof Consider a graph G whose vertex set is partitioned into two subsets V_1 and V_2 such that $|V_1| = \lceil n/2 \rceil$ and $|V_2| = \lfloor n/2 \rfloor$, the induced subgraph by V_i is a complete graph, $i = 1, 2$, and there are λ directed edges connecting the vertices from V_2 to V_1. Obviously the edge connectivity of this graph is λ. Suppose that there is a permutation which permutes all vertices in V_2 to V_1. Then, the number of wavelengths for implementing this permutation in one round is at least $\lfloor n/2 \rfloor / \lambda \geq \frac{n}{2\lambda} - 1$. $\square$

One direct corollary of Theorem 2 is expressed as follows.

Corollary 4 *Assume that G is a directed graph with edge connectivity λ. Let d_{in} and d_{out} be the minimum in-degree and out-degree of G. Then the number of wavelengths for implementing any permutation on G in one round is $\Omega(\frac{n}{2d_{\min}})$, where $d_{\min} = \min\{d_{in}, d_{out}\}$.*

Proof Consider the graph G defined in the proof of Theorem 2. Because $\lambda \leq d_{\min}$, by Theorem 2 the corollary follows. $\square$

Theorem 3 *Let G be a directed symmetric all-optical network with the edge connectivity λ. Then, there is a permutation algorithm which can implement any permutation on G in one round with $\lceil \frac{n+3}{\lambda} \rceil$ wavelengths on the wavelength non-conversion model.*

Proof We prove this theorem by giving a permutation routing algorithm for finding the routing paths and assigning a wavelength to each routing path. The algorithm is as follows.

First, partition the n sources into g groups such that the destinations of the source vertices in the same group are not in the group. Then, establish edge-disjoint routing paths for the vertices in a group, and assign the same wavelength to all routing paths in the group.

To do so, we first construct an undirected graph $G_\pi(V, E_\pi)$ where $(v, \pi(v)) \in E_\pi$ and π is the permutation function. It is clear that G_π consists of simple cycles only. We then partition the vertex set V of G_π into three disjoint subsets U_0, U_1, and U_2 such that each U_i is an independent set of G_π. Clearly there exists such a vertex partition because G_π is a special graph. Now we describe the partition. For each simple cycle with an even number of vertices in G_π, we choose any one vertex on it as the starting point (and first vertex), and traverse the cycle by labeling the ith vertex with label l, where $l = (i \bmod 2)$. For a simple cycle with an odd number of vertices in G_π, we choose any vertex on it as the starting point, and traverse the cycle from it by labeling the ith vertex with label $l = (i \bmod 2)$, except that the last vertex (which is also adjacent to the starting vertex) is labeled 2. Then $U_l = \{v \mid v \in V \text{ and } v \text{ is labeled by } l\}$. It is easy to verify that U_l is an independent set of G_π. Let $n_l = |U_l|$. The vertices in U_l are then further partitioned into several groups such that each group except the last group consists of λ vertices and the last group consists of $n_l - (\lceil n_l/\lambda \rceil - 1)\lambda$ vertices, $l = 0, 1, 2$. As a result, suppose that there are g groups, then $g = \sum_{l=0}^{2} \lceil n_l/\lambda \rceil \leq \lceil \frac{n+3}{\lambda} \rceil$. With this partition, the destinations of the vertices in a group are not in the group. Denote by $\mathcal{P}_j$ the jth group, $1 \leq j \leq g$.

Finally, we route all sources in $\mathcal{P}_j$ to their destinations with the same wavelength, i.e, $|\mathcal{P}_j|$ edge disjoint directed routing paths for $\mathcal{P}_j$ in G must be established, $1 \leq j \leq g$. It is obvious that there exist $|\mathcal{P}_j|$ edge disjoint directed routing paths in G because G is λ edge-connected and $|\mathcal{P}_j| \leq \lambda$. The following is used for this purpose. For each group $\mathcal{P}_j$, we construct a directed graph

$$G'_j = (V \cup \{s, t\}, E \cup \{\langle s, v \rangle \mid v \in \mathcal{P}_j\} \cup \{\langle \pi(v), t \rangle \mid v \in \mathcal{P}_j\}).$$

Assign unit capacity to each directed edge in G'_j. Then find the maximum flow on G'_j from s to t. Clearly the maximum flow of G'_j from s to t is $|\mathcal{P}_j|$. Thus, we establish $|\mathcal{P}_j|$ edge disjoint directed paths for $\mathcal{P}_j$, and assign the same wavelength to each such path. In order to implement any permutation on G in one round, $g \leq \lceil \frac{n+3}{\lambda} \rceil$ wavelengths suffices. $\Box$

Note that, if the edge connectivity value λ of G is not given in advance, it can be found in $O(mn)$ time by the algorithm due to Mansour and Schieber [15].

4 Product Networks

Define the *direct product* of two graphs $G = (V_1, E_1)$ and $H = (V_2, E_2)$, denoted by $G \times H$, as follows. The vertex set of $G \times H$ is the Cartesian product $V_1 \times V_2$. There is an edge between (v_1, v_2) and (v'_1, v'_2) when $v_1 = v'_1$ and $(v_2, v'_2) \in E_2$, or $v_2 = v'_2$ and $(v_1, v'_1) \in E_1$. G and H are called the *factors* of $G \times H$. Notice that $G \times H$ consists of $|V_2|$ copies of G connected by $|V_1|$ copies of H, arranged in a grid-like fashion. Each copy of H is called a *row* and each copy of G is called a *column*. An edge in $G \times H$ between (v_1, u) and (v_2, u) is called a *G-edge* if $(v_1, v_2) \in E_1$, and an edge of $G \times H$ between (v, u_1) and (v, u_2) is called an *H-edge* if $(u_1, u_2) \in E_2$. Now let $W = G \times H$ and π be a permutation on W. Assume the factors G and H of W are networks that each come equipped with a routing algorithm. We are interested in designing a routing algorithm for the new network W by using the routing algorithms for the factors as *subroutines*.

4.1 Lower bounds on the number of wavelengths

Following the same idea as in Section 3, we present a lower bound on the number of wavelengths for implementing any permutation on W in one round, in terms of bisections of its factors. We start with the following lemma.

Lemma 3 *Let $G \times H$ be a directed symmetric product network. Let $c(X)$ be the number of edges in a bisection of X and $V(X)$ be the vertex set of X. Suppose that*

d_G and d_H are the maximum in-degrees (out-degrees) of G and H. Then we have the following lower bounds for $c(G \times H)$:

1. $\min\{|V(H)|c(G), |V(G)|c(H)\}$
   if both $|V(G)|$ and $|V(H)|$ are even;

2. $\min\{|V(G)|c(H) + \lfloor |V(G)|/2 \rfloor d_H + c(G), |V(H)|c(G)\}$
   if $|V(G)|$ is even and $|V(H)|$ is odd;

3. $\min\{|V(H)|c(G) + \lfloor |V(H)|/2 \rfloor d_G + c(H), |V(G)|c(H)\}$
   if $|V(G)|$ is odd and $|V(H)|$ is even;

4. $\min\{|V(G)|c(H) + \lfloor |V(G)|/2 \rfloor d_H + c(G), |V(G)|c(G) + \lfloor |V(G)|/2 \rfloor d_G + c(H)\}$
   otherwise (both $|V(G)|$ and $|V(H)|$ are odd).

Proof We first consider the case where $p = |V(G)|$ and $q = |V(H)|$ are even. Let $V(G) = v_1, v_2, \ldots, v_p$ be the vertex set of G. Suppose that a bisection of G is $(V_1(G), V_2(G))$, where $V_1(G) = \{v_1, v_2, \ldots, v_{p/2}\}$ and $V_2(G) = \{v_{p/2+1}, \ldots, v_p\}$. We replace each vertex of G by graph H, and make the corresponding connection between two vertices in different copies of H. This leads to the graph W. Now we partition the vertex set of W into two disjoint subsets of equal size according to the bisection of G. Then the number of edges of W in this partition is $|V(H)|c(G)$. Since W is a symmetric network, there is another partition which satisfies the above partition condition. The number of edges in this latter partition is $|V(G)|c(H)$. Since $c(G \times H)$ is a partition that has the minimum number of edges, $c(G \times H) \leq \min\{|V(H)|c(G), |V(G)|c(H)\}$.

We next consider the case where $p = |V(G)|$ is even and $q = |V(H)|$ is odd. We use only G-edges as the edges of W for the partition. Then, the number of edges of W in this partition is $|V(H)|c(G)$ by the above discussion. In the rest we consider using both H- and G-edges for the bisection of W. Let $V(H) = \{u_1, u_2, \ldots u_q\}$. Suppose that a bisection of H is $(V_1(H), V_2(H))$, where $V_1(H) = \{u_1, u_2, \ldots, u_{\lceil q/2 \rceil}\}$ and $V_2(H) = \{u_{\lceil q/2 \rceil+1}, \ldots, u_q\}$. It is clear that $|V_1(H)| - |V_2(H)| = 1$ because q is odd. Now we give another partition $(V_1'(H), V_2'(H))$ of H, where $V_1'(H) = V_2(H) \cup \{v\}$ and $V_2'(H) = V_1(H) - \{v\}$. The number of H-edges in this new partition is no more than $c(H) + d_H(v) \leq c(H) + d_H$. We replace each vertex of H by graph G, and make the corresponding connection between two vertices in different copies of G. This leads to the graph W. Partition the vertex set of W into two disjoint subsets such that: the size difference is at most one according to a bisection of H; and the vertices in the copy of G corresponding to v of H are partitioned into two disjoint subsets by a bisection of G. Thus, the number of edges of W in this partition is no more than $\lceil |V(G)|/2 \rceil c(H) + \lfloor |V(G)|/2 \rfloor (c(H) + d_H(v)) + c(G) \leq |V(G)|c(H) + \lfloor |V(G)|/2 \rfloor d_H + c(G)$. Therefore $c(G \times H) \leq \min\{|V(H)|c(G), |V(G)|c(H) + \lfloor |V(G)|/2 \rfloor d_H + c(G)\}$. The other two cases can be shown similarly, omitted. $\square$

Having Lemma 3, we derive the following lower bounds on the number of wavelengths required.

Theorem 4 *Let $W = G \times H$ be a directed symmetric product network. Let $c(X)$ be the number of edges in a bisection of X. Suppose that d_G and d_H are the maximum in-degrees (out-degrees) of G and H. Then, a lower bound for the minimum number of wavelengths $w_{\min}(W)$ for implementing any permutation on W in one round is:*

1. $\max\{\frac{|V(G)|}{2c(G)} - 1, \frac{|V(H)|}{2c(H)} - 1\}$
   if both $|V(G)|$ and $|V(H)|$ are even;

2. $\max\{\frac{|V(H)|}{2c(H)+d_H+2c(G)/|V(G)} - 1, \frac{|V(G)|}{2c(G)} - 1\}$
   if $|V(G)|$ is even and $|V(H)|$ is odd;

3. $\max\{\frac{|V(G)|}{2c(G)+d_G+2c(H)/|V(H)} - 1, \frac{|V(H)|}{2c(H)} - 1\}$
   if $|V(G)|$ is odd and $|V(H)|$ is even;

4. $\max\{\frac{|V(G)|}{2c(G)+d_G+2c(H)/|V(H)|} - 1, \frac{|V(H)|}{2c(H)+d_H+2c(G)/|V(G)} - 1\}$
   otherwise (both $|V(G)|$ and $|V(H)|$ are odd).

Proof From Theorem 1, it is clear that $w_{\min}(W) \geq \frac{|V(G)||V(H)|}{2c(G \times H)} - 1$. Use of the lower bounds for $c(G \times H)$ from Lemma 3 in this inequality gives the theorem. $\Box$

Remark. Theorem 4 always holds no matter which model is used, the wavelength non-conversion model or conversion model.

4.2 A routing algorithm on the packet-passing model

The permutation routing issue on a direct product network W has been addressed for the packet-passing model [2, 8]. Baumslag and Annextein [8] presented an efficient algorithm for permutation routing on that model. Basically, their algorithm consists of the following three phases.
1. Route some set of permutations on the copies of G;
2. Route some set of permutations on the copies of H;
3. Route some set of permutations on the copies of G.
Since the product network is a symmetric network, the above three phases can be applied alternatively, i.e, by first routing on H, followed by G, and followed by H.

Before we proceed, consider the following naive routing method. First route each source in a column to its destination row in the column. Then route each source in a row to its destination column in the row. This fails to be an edge congestion-free algorithm because there may be several sources in the same column that have their destinations in a single row (causing congestion at the intersection of the column and the row). By using an initial extra phase, the above congestion problem can be solved. That is, we first "rearrange" each column so that, after the rearrangement, each row consists of a set of sources whose destinations are all in distinct columns. After this, a permutation of each row is required to get each source to its correct column after the rearrangement. Once all of the sources are in their correct columns, a final permutation of each column suffices to get each source to its correct destination. This final phase is indeed a permutation since all destinations of sources are distinct. Thus, the aim of the first phase is to find a set of sources P_R, once per column, such that every source in P_R has its destination in a distinct column for each row R.

To this end, a bipartite graph $G_B(X, Y, E_B)$ is constructed as follows. Let X and Y represent the set of columns of W. There is an edge between $x_i \in X$ and $y_j \in Y$ for each source in column i whose destination is in column j. Since π is a permutation, it follows that G_B is a regular, bipartite multigraph. Thus, G_B can be decomposed into a set of edge disjoint perfect matchings. Note that the sources involved in a single perfect matching all have their destinations in distinct columns. Therefore, for each row R, use all of the sources that correspond to a single perfect matching for the set P_R. Each of these sets P_R, is "lifted" to row R during the first phase of the algorithm. Since each source is involved in precisely one perfect matching, the mapping of sources in a column during the first phase is indeed a permutation of the column.

4.3 Routings on the wavelength conversion model

The algorithm in Section 4.2 can be expressed in a different way. That is, π is decomposed into three permutations σ_i, $i = 1, 2, 3$, where σ_1 and σ_3 are permutations in columns of W and σ_2 is a permutation in rows of W. For example, let v be a source in W at the position of row i_1 and column j_1, and $\pi(v) = u$ be the destination of v at the position row i_2 and column j_2. Suppose v is "lifted" to the position of row i'_1 and column j_1 in the first phase. Then, $\sigma_1(i_1, j_1) = (i'_1, j_1)$, followed by the second phase of $\sigma_2\sigma_1(i_1, j_1) = \sigma_2(i'_1, j_1) = (i'_1, j_2)$, and followed by the third phase of $\sigma_3\sigma_2\sigma_1(i_1, j_1) = \sigma(i'_1, j_2) = (i_2, j_2)$.

In the following we present a permutation routing algorithm on W for π on the wavelength conversion model, based on the above observation. We start with the following major theorem.

Theorem 5 *Given permutation routing algorithms for networks G and H, there is a routing algorithm for the product network $G \times H$. The number of wavelengths for any permutation in one round is at most* $\max\{2w(G), w(H)\}$ *if $w(G) \leq w(H)$; or* $\max\{2w(H), w(G)\}$ *otherwise, where $w(X)$ represents the number of wavelengths needed to implement any permutation on network X in one round.*

Proof The routing algorithm by Baumslag and Annexstein (Section 4.2) is for the packet-passing model, which is a totally different model from ours - the WDM model. Hence, some modifications to their algorithm are necessary. As we can see, implementing the permutation routing on W for π can be decomposed into three permutations σ_i, $1 \leq i \leq 3$.

Without loss of generality, assume $w(G) \leq w(H)$. Following the three phases of the algorithm of Baumslag and Annexstein, the permutation σ_1 can be implemented with $w(G)$ wavelengths (in other words, all routing paths can be colored with $w(G)$

colors); the permutation σ_2 can be implemented with $w(H)$ wavelengths (all routing paths can be colored with $w(H)$ colors, and the colors for σ_1 can be re-used here); the permutation σ_3 can be implemented with $w(G)$ wavelengths (all routing paths can be colored with $w(G)$ colors, and the colors for σ_1 cannot be used here). Now, for a request $(i, \pi(i))$, let (u_1, v_1) be the position of i in W and (u_2, v_2) be the position of $\pi(i)$ in W. Then, the routing path L_i for request $(i, \pi(i))$ consists of three routing segments $L_{i,1}$, $L_{i,2}$ and $L_{i,3}$ which correspond to σ_1, σ_2, and σ_3, where $L_{i,1}$ only contains the vertices in column v_1 of W and consists of G-edges; $L_{i,2}$ only contains the vertices in row u_1' of W and consists of H-edges, assuming that $\sigma_1(i)$ is at row u_1' and column v_1 in W; and $L_{i,3}$ only contains the vertices in column v_2 of W, and G-edges.

Any path L_i can further be made a simple path L_i' by deleting the cycles it contains. Let $L_{i,j}'$ be the corresponding segment of $L_{i,j}$, $1 \le j \le 3$. L_i' then is assigned three different wavelengths for each of the segments of L_i', which are the wavelengths for $L_{i,j}$ originally, $j = 1, 2, 3$. That is, each L_i' for a request $(i, \pi(i))$ can be implemented with at most three wavelengths. But, the wavelengths used for $L_{i,1}$ cannot be used for $L_{i,3}$. So, at least $2w(G)$ wavelengths are needed. In summary, implementing any permutation π on W in one round can be done with $\max\{2w(G), w(H)\}$ wavelengths. □

For the wavelength conversion model, the following corollaries can be derived from Theorem 5 directly.

Corollary 5 *There is an algorithm for implementing any permutation in a directed symmetric hypercube H_q of $n = 2^q$ vertices with 2 wavelengths.*

Proof Since $H_q = K_2 \times H_{q-1}$ and $w(K_2) = 1$. By Theorem 5, $w(H_q) = \max\{2w(K_2), w(H_{q-1})\}$. Furthermore, $H_{q-1} = K_2 \times H_{q-2}$, and it is easy to show that $w(H_q) = 2$ by induction on q. □

Corollary 6 *For any directed symmetric $l \times h$ mesh M with $l \le h$ and $n = lh$, there is an algorithm for implementing any permutation on M in one round. The number of wavelengths used by this algorithm is at most* $\max\{l, \lfloor h/2 \rfloor\}$.

Proof Assume $l \le h$ and $M = L_l \times L_h$, where L_i represents a chain of i vertices. Obviously $w(L_i) = \lfloor i/2 \rfloor$. By Theorem 5, $w(M) = \max\{2w(L_l), w(L_h)\} \le \max\{l, \lfloor h/2 \rfloor\}$. □

Corollary 7 *There is an algorithm for implementing any permutation in a directed symmetric $\sqrt{n/2} \times \sqrt{2n}$ mesh with n vertices in one round. The number of wavelengths used by this algorithm is at most $\sqrt{n/2}$ which is almost optimal.*

Proof Let M be the $\sqrt{n/2} \times \sqrt{2n}$ mesh. By Corollary 6, $w(M) = \max\{2w(L_{\sqrt{n/2}}), w(L_{\sqrt{2n}})\} = \sqrt{n/2}$. By Corollary 3, the lower bound of the number of wavelengths for any permutation on M is at least $\Omega(\sqrt{n})$, so, this bound is almost tight. □

Finally, we discuss design issues of all-optical networks based on product networks, provided that the two factor networks are given. Suppose that $w(X, n_i)$ represents the number of wavelengths used for implementing any permutation on network X of n_i vertices in one round. Theorem 5 implies some principles to the design of a class of all-optical networks which can be decomposed into two factors.

(i) G and H have the same topology (for example, a chain topology). If the number of wavelengths used for implementing any permutation within one round in G or H is independent of the number of vertices in the network (for example, on hypercubes), then the number of wavelengths needed for implementing any permutation on W in one round is the same as the number for one of its factors. Otherwise, the number of wavelengths is a function $w(X, n_i)$ of the number of vertices n_i in X. Hence, the number of wavelengths used for implementing any permutation on W in one round, $w(W)$, is determined by the following equations.

$$\begin{aligned} w(W) &= \max\{2w(G, n'), w(G, n/n')\} \\ w(G, n') &= w(G, n/n')/2 \\ n'^2 &\leq n \end{aligned}$$

(ii) G and H have different topologies. If $w(G, k) \leq w(H, k)$, it is possible to construct a W for implementing any permutation in one round with $w(W)$ wavelengths such that G contains n_1 vertices and H contains n_2 vertices, where $w(W)$ and n_i, $i = 1, 2$, can be obtained by the following equations.

$$\begin{aligned} w(W) &= \max\{2w(G, n_1), w(H, n_2)\} \\ w(G, n_1) &= w(H, n_2)/2 \\ n &= n_1 \times n_2 \\ n_i &\geq 2, \ i = 1, 2 \end{aligned}$$

4.4 Routings on the wavelength non-conversion model

Let $W = G \times H$ be a symmetric network. In this subsection we study permutation routings on W for the wavelength non-conversion model in which each routing path is assigned a unique wavelength. We only consider the case of $w(G) \leq w(H)$. The case where $w(H) \leq w(G)$ is analogous and is omitted.

Following the proof of Theorem 5, we obtain a routing path L'_i for every $i \in V(W)$. We know that L'_i can further be divided into three segments $L'_{i,k}$, $k = 1, 2, 3$, and each of the segments has been assigned a wavelength (a color). Let γ_j be the color (wavelength) of $L'_{i,k}$. Then $(\gamma_1, \gamma_2, \gamma_3)$ is the ordered color tuple of

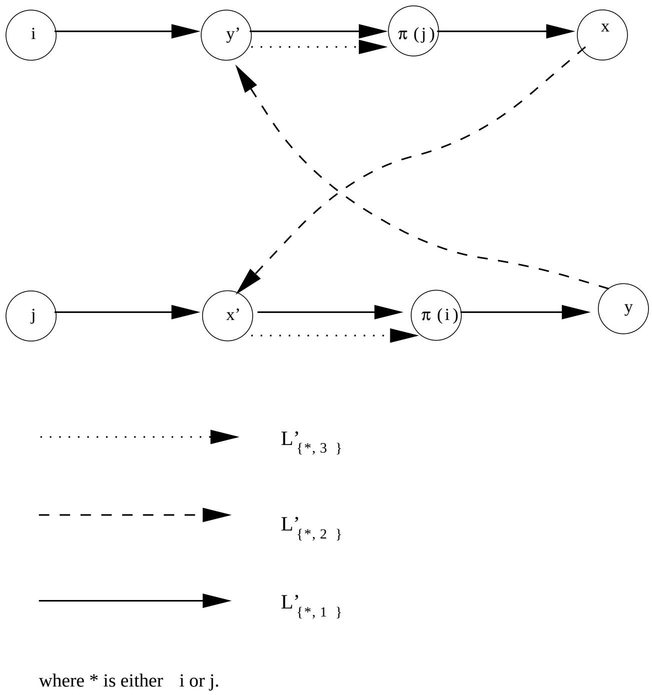

Figure 1: An example.

L'_i. We further treat $(\gamma_1, \gamma_2, \gamma_3)$ as a coordinate point in a 3-dimensional Cartesian coordinate system. Assume that each coordinate point in the system has been assigned a unique label, i.e, L'_i is assigned the wavelength numbered by the label of $(\gamma_1, \gamma_2, \gamma_3)$ in the system. Then, the total number of the corresponding coordinate points for all routing paths on W for any permutation is $w(G) \times w(H) \times w(G) = w(G)^2 w(H)$.

Now we consider two routing paths L'_i and L'_j, $i \neq j$. Let L'_i be colored with $(\gamma_1, \gamma_2, \gamma_3)$ and L'_j be colored with $(\beta_1, \beta_2, \beta_3)$. If there exists a k such that $\gamma_k \neq \beta_k$, $1 \leq k \leq 3$, then it is clear that the wavelengths assigned for L'_i and L'_j are different. Otherwise, if $\gamma_k = \beta_k$ for all k, $1 \leq k \leq 3$, then L'_i and L'_j are assigned the same wavelength. However, this wavelength assignment is incorrect because there still exist some routing paths sharing common edges which are assigned the same wavelength. We illustrate this situation by an example (see Figure 1). Assume that the given permutation is π. Consider two routing paths L'_i and L'_j, where L'_i starts from i, goes through y', $\pi(j)$, x, x', and ends at $\pi(i)$, L'_j starts from j, goes through x', $\pi(i)$, y, y', and ends at $\pi(j)$. Clearly L'_i and L'_j share two common segments which are from y' to $\pi(j)$ and from x to $\pi(i)$ even though they have the same color tuple. So, they cannot be assigned the same wavelength on the WDM model.

To cope with this situation, we use the following approach. Let $l_{\max}(G)$ be the number of edges in the longest routing path on G for any permutation. Define $\mathcal{R}_l = \{L'_i \mid L'_i \text{ is labeled by } l \text{ in the coordinate system}\}$. Obviously the set of entire routing paths for π is $\mathcal{R} = \bigcup_{l=1}^{w(G)^2 w(H)} \mathcal{R}_l$.

For each $\mathcal{R}_l$, we construct an auxiliary graph $G_l = (V_l, E_l)$, which is a subgraph of the conflict graph on W defined in Section 2, as follows. Every vertex in V_l corresponds to an element in $\mathcal{R}_l$. There is an edge between two vertices if and only if the two routing paths share at least one common edge. Then we have

Lemma 4 *Let $G_l = (V_l, E_l)$ be defined as above, then the maximum degree of G_l is $l_{\max}(G)$.*

Proof Let L'_i and L'_j be the corresponding routing paths of two vertices in G_l. We know that L'_k consists of three segments $L'_{k,1}$, $L'_{k,2}$, and $L'_{k,3}$, $k = i$ or $k = j$. By the definition, $L'_{i,p}$ and $L'_{j,p}$ are edge disjoint for all p, $1 \le p \le 3$. But $L'_{i,1}$ and $L'_{j,3}$ (similarly $L'_{j,1}$ and $L'_{i,3}$) may be not edge disjoint, see Fig. 1. Since the number of edges in any routing path is no greater than $l_{\max}(G)$, there are at most $l_{\max}(G)$ other routing paths sharing common edges with L'_i. Therefore, the maximum degree of G_l is $l_{\max}(G)$. $\square$

By Lemma 4, all vertices in G_l can be colored with $l_{\max}(G) + 1$ colors such that the adjacent vertices are colored with different colors. This coloring can be done in polynomial time by a greedy approach. This means, for each $\mathcal{R}_l$, we can assign $l_{\max}(G) + 1$ wavelengths for the routing paths in it such that those routing paths sharing common edges are assigned different wavelengths. There are $w(G)^2 w(H)$ different $\mathcal{R}_l$. Therefore, the total number of wavelengths required for any permutation is $(l_{\max}(G) + 1)w(G)^2 w(H)$, which is formally described as follows.

Theorem 6 *Given permutation routing algorithms for networks G and H, there is a permutation routing algorithm for the product network $W = G \times H$. The number of wavelengths for any permutation on W in one round is $(l_{\max}(G)+1)w(G)^2 w(H)$ if $w(G) \le w(H)$; or $(l_{\max}(H)+1)w(H)^2 w(G)$ otherwise, where $w(X)$ represents the number of wavelengths needed to implement any permutation on network X in one round, and $l_{\max}(X)$ is the number of edges of the longest routing path on X for any permutation.*

Theorem 6 implies the following facts. If $w(G)$, $l_{\max}(G)$, and $w(H)$ are constants, then any permutation can be implemented on $G \times H$ in one round, using a constant number of wavelengths. Otherwise, when $w(G)$, $l_{\max}(G)$, and $w(H)$ are functions of their vertex sizes, Theorem 6 is meaningful only if either $w(H)$, or $w(G)$, or $l_{\max}(G)$ is a sublinear function of the vertex size of G or H. Otherwise, the number of wavelengths needed by this routing is beyond what we might expect, which can be explained as follows.

Let $|V(G)| = p$, $|V(H)| = q$, and $n = pq$. Assume that both p and q are functions of n. If both $w(H)$ and $w(G)$ are linear functions of the vertex sizes of G and H, for example, $w(G) = ap$ and $w(H) = bq$ where a and b are constants with $0 < a, b < 1$. Without loss of generality, we further assume that $w(G) \le w(H)$. Then, it needs

$(l_{\max}(G)+1)w(G)^2w(H) = (l_{\max}(G)+1)(a^2b)pn = cn^{1+\alpha} > n$ wavelengths where $p = n^{\alpha}$. As a matter of the fact, any permutation can be implemented on an arbitrary all-optical network in one round if n wavelengths are available. To cope with this case, we present another permutation routing algorithm. We start with the following perfect matching lemma.

Lemma 5 *[14] Let $G_B(X, Y, E)$ be a bipartite graph such that for every subset S of X, we have $|N(S)| \geq |S|$, where $N(S)$ is the subset of Y that are adjacent to vertices in S. Then G_B has a perfect matching of size* $\min\{|X|, |Y|\}$.

Suppose $p \leq q$. Our idea comes directly from Youssef [27]. That is, at each time, we select p sources and their destinations such that these sources belong to distinct rows and their destinations belong to distinct columns, i.e, assign the routing paths for these source-destination pairs the same wavelength because there are no routing congestions between them. Such sources can be found using perfect matching in a bipartite graph $G_B = (X, Y, E)$ where X is the set of rows and Y is the set of columns. There is an edge connecting $x \in X$ and $y \in Y$ if there is a source in row x whose destination is in column y. Clearly G_B is a bipartite multigraph, the degree of every vertex of G_B in X is q, and the degree of every vertex of G_B in Y is p. Since any subset $S \subseteq X$, $|N(S)| \geq S$. Thus, there is a perfect matching in G_B by Lemma 5. By deleting this matching, we can find the next perfect matching in the remaining graph, and so on. As a result, G_B is decomposed into q edge disjoint perfect matchings. So, we have

Lemma 6 *Given a directed symmetric network $G \times H$ with $|V(G)| = p$, $|V(H)| = q$ and $p \leq q$, there is an algorithm for implementing any permutation on $G \times H$ in one round if q wavelengths are available.*

Proof By the algorithm above, the routing paths from the sources to their destinations in a perfect matching are edge disjoint. So, all of them can be assigned the same wavelength. Therefore the lemma follows. □

A direct corollary of Lemma 6 is expressed as follows.

Corollary 8 *Let M be a directed symmetric $l \times h$ mesh with $n = lh$ vertices and $l \leq h$. Then there is an algorithm for implementing any permutation on M in one round. The number of wavelengths needed is $h = n/l$. In particular, when $l = h$, the number of wavelengths needed is $\sqrt{n}$.*

If we allow $\max\{w(G), w(H)\}$ wavelengths to assign every fiber-optic link of $G \times H$, we have

Theorem 7 *Let $G \times H$ be a directed symmetric network. On the wavelength non-conversion model, if there are permutation algorithms for implementing any permutation on G and H in one round with $w(G)$ and $w(H)$ wavelengths respectively, then there is a permutation algorithm for implementing any permutation on*

$G \times H$ in $\frac{|V(H)|}{\max\{w(H),w(G)\}}$ ($\leq 2c(H)+1$) rounds with $\max\{w(H), w(G)\}$ wavelengths if $|V(G)| \leq |V(H)|$, or in $\frac{|V(G)|}{\max\{w(H),w(G)\}}$ ($\leq 2c(G)+1$) rounds with $\max\{w(H), w(G)\}$ wavelengths, where $w(G)$ and $w(H)$ are the linear functions of their sizes and $c(X)$ is the number of edges in a bisection of X.

Proof We only consider the case $|V(G)| \leq |V(H)|$. The analogous case $|V(H)| < |V(G)|$ is omitted. By Theorem 1, $w(H) \geq \frac{|V(H)|-1}{2c(H)}$. So, $|V(H)| \leq 2w(H)c(H) + 1$. According to Lemma 6, in order to implement any permutation on $G \times H$ with $\max\{w(H), w(G)\}$ wavelengths, the number of rounds needed is at most

$$\frac{|V(H)|}{\max\{w(H), w(G)\}} \leq \frac{2w(H)c(H) + 1}{\max\{w(H), w(G)\}}.$$

That is, the number of rounds is at most

1. $|V(H)|/w(H) \leq \frac{2w(H)c(H)+1}{w(H)} \leq 2c(H) + 1$ if $w(G) \leq w(H)$;

2. or $|V(H)|/w(G) \leq \frac{2w(H)c(H)+1}{w(G)} \leq 2c(H) + 1$ otherwise.

□

5 Permutation Routings for Specific All-Optical Networks

In this section we study permutation routings on some specific all-optical networks. Before we proceed, we introduce a result on a hypercube H_q by Gu and Tamaki [11].

Lemma 7 *[11] Let H_q be a directed symmetric hypercube. Then any permutation on it can be implemented in one round with two wavelengths.*

We now consider the well known *cube-connected-cycles* (CCC) network [21]. The network $G(V, E)$ has n vertices, with $n = 2^q \times 2^r$ and $q = 2^r$ (so $q + r = \log n$ and $r = \log q$). $G(V, E)$ can be thought of as a supergraph where each supervertex is a graph. This supergraph is a hypercube of 2^q supervertices, and each supervertex is a ring of 2^r vertices. Each vertex in the CCC may be labeled by a number comprising $q + r$ bits, where the first q bits are for hypercube-connection, and the last r bits are for ring-connection. Let v and u be two distinct vertices in the CCC, v labeled (in bits) $x = x_{q+r-1}x_{q+r-2} \ldots x_r x_{r-1} \ldots x_0$, and u labeled (in bits) $y = y_{q+r-1}y_{q+r-2} \cdots y_r y_{r-1} \cdots y_0$. Then, there is an edge in the CCC connecting u and v if and only if one of the following conditions holds: (i) $x_i = y_i$ and $x_{j_0} = 1-y_{j_0}$ for all $0 \leq i \leq q+r-1$, $i \neq j_0$, and $r \leq j_0 \leq q+r-1$; (ii) $x_i = y_i$ for all i, $r \leq i \leq q+r-1$, and either $|x \bmod 2^r - y \bmod 2^r| = 1$ or $|x \bmod 2^r - y \bmod 2^r| = 2^r - 1$. Obviously the degree of each vertex in the CCC is three. For the CCC, we have the following theorem.

Theorem 8 *Let $G(V, E)$ be a directed symmetric CCC network defined as above. Then any permutation on G can be implemented either in $\lfloor \log n \rfloor$ rounds using 2 wavelengths, or in one round using $\lfloor 2 \log n \rfloor$ wavelengths on the wavelength non-conversion model.*

Proof The basic idea is to choose a vertex from each ring as a source such that no two sources have their destinations in the same ring. Route these chosen sources to their destinations on the supergraph, which can be implemented in one round using 2 wavelengths by Lemma 7. Thus, any permutation on the CCC can be implemented in 2^r rounds. Since $2^r = q = \log n - r \leq \lfloor \log n \rfloor$, $\lfloor \log n \rfloor$ rounds suffices.

It remains to show how to choose the sources for each round such that none of the destinations of two sources are in the same ring. The approach proceeds as follows. First we construct a bipartite graph $G(X, Y, E_{XY})$ where X and Y represent the sets of 2^q supervertices of G. Let $v \in X$ and $u \in Y$, if the source v will route its message to the destination u, then an undirected edge (u, v) is added to E_{XY}. Clearly, $G(X, Y, E_{XY})$ is a regular bipartite multigraph with degree q $(= 2^r)$. By Hall's theorem, $G(X, Y, E_{XY})$ can be decomposed into q $(= 2^r)$ edge disjoint perfect matchings, and each perfect matching can be found in polynomial time. The corresponding routing paths for each perfect matching can be implemented using two wavelengths by Lemma 7. Therefore, any permutation in the CCC can be implemented with either $\lfloor \log n \rfloor$ rounds using 2 wavelengths, or one round using $2 \times 2^r \leq \lfloor 2 \log n \rfloor$ wavelengths. $\Box$

Note that the CCC network is a regular network with out-degree (in-degree) 3. Aumann and Rabani [5] showed that $O(\log^2 n / \beta^2(G))$ wavelengths suffice for implementing any permutation on a network G of constant degree in one round, where $\beta(G)$ is the edge-expansion of G. The CCC network is a network of degree three, and $\beta(\mathrm{CCC}) \leq \frac{2^{q-1}}{2^{q-1} 2^r} = 1/2^r$. For this special network, our permutation routing algorithm implements any permutation in one round with $2 \log n$ wavelengths, which improves the number of wavelengths used in [5] by a factor of $O(\log^3 n)$.

Next we generalize the CCC network further to the following network $G(V, E)$ with $n = 2^q \times 2^s$ vertices satisfying $q + s = \log n$ and $2^s \geq q$. This network is a hypercube supergraph of 2^q supervertices, and each supervertex is a ring of 2^s vertices. Clearly the CCC network is a special case of this generalized network with $s = \log q$. Now we see that the number of wavelengths needed for implementing any permutation on G depends on the number of vertices of degree three. For example, if $2^q = \sqrt{n}$, then $2^s = \sqrt{n}$. The network G contains $2^q \times q = \frac{\sqrt{n} \log n}{2}$ vertices of degree three, and $n - \frac{\sqrt{n} \log n}{2}$ vertices of degree two. Thus, implementing any permutation on G using the above approach requires either $\sqrt{n}$ rounds if there are 2 wavelengths available, or one round if there are $2\sqrt{n}$ wavelengths available. In general, we have

Lemma 8 *Let $G(V, E)$ be a directed symmetric, generalized CCC network defined as above, with $q + s = \log n$ and $2^s \geq q$. Then any permutation on G can be implemented either in 2^s rounds using 2 wavelengths, or in one round using 2^{s+1}*

wavelengths on the wavelength non-conversion model. The number of vertices of degree three in G is $q2^q$, and the number of vertices of degree two is $n - q2^q$.

6 Conclusions

In this paper we have shown some lower bounds on the number of wavelengths needed for implementing any permutation on all-optical networks in terms of bisection and edge-connectivity, and proposed an algorithm for implementing any permutation in a directed symmetric network with $\lceil \frac{n+3}{\lambda} \rceil$ wavelengths on the wavelength non-conversion model where λ is the edge connectivity value of the network. We have shown a lower bound on the number of wavelengths required for implementing any permutation on the product networks, and presented permutation routing algorithms for the network, based on the wavelength non-conversion and conversion models respectively. We have considered the permutation issue on cube-connected-cycles networks with constant degree three. The number of wavelengths needed for implementing any permutation on this network in one round is $\lfloor 2 \log n \rfloor$, which improves on a general result for bounded degree networks in [5] by a factor of $O(\log^3 n)$ for this special case.

Acknowledgment. We thank Xin-Gui Fang for stimulating discussions and constructive comments on this paper. The first two authors were partially supported by the Australian Research Council.

References

[1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber and M. Sudan. Efficient routing in optical networks. *J. of the ACM* **46:**973–1001, 1996.

[2] N. Alon, F. R. K. Chung, and R. L. Graham. Routing permutations on graphs via matchings. *SIAM J. Discrete Math.* **7:**516–530, 1994.

[3] B. Awerbuch, Y. Azar, and S. Plotkin. Throughput competitive on-line routing. *Proc. of 34th Annual IEEE Symposium on Foundations of Computer Science*, 1993, 32–40.

[4] B. Awerbuch, Y. Bartal, A. Fiat and A. Rosén. Competitive non-preemptive call control. *Proc. of 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1994, 312–320.

[5] Y. Aumann and Y. Rabani. Improved bounds for all optical routing. *Proc. of 6th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1995, 567–576.

[6] R. A. Barry and P. A. Humblet. Bounds on the number of wavelengths needed in WDM networks. *In LEOS'92 Summer Topical Mtg. Digest* 1992, 114–127.

[7] R. A. Barry and P. A. Humblet. On the number of wavelengths and switches in all-optical networks. *IEEE Trans. Commun.* **42:**583–591, 1994.

[8] M. Baumslag and F. Annexstein. A unified framework for off-line permutation routing in parallel networks. *Math. Systems Theory* **24:**233–251, 1991.

[9] B. Beauquier, J-C Bermond, L. Cargano, P. Hell, S. Pérnces, and U. Vaccaro. Graph problems arising from wavelength-routing in all-optical networks. *Proc. of 2nd Workshop on Optics and Computer Science*, IPPS'97, April, 1997.

[10] P. E. Green. *Fiber- Optic Communication Networks.* Prentice Hall, 1992.

[11] Q -P Gu and H. Tamaki. Routing a permutation in the hypercube by two sets of edge-disjoint paths. *Proc. 10th International Parallel Processing Symp.*, IEEE Computer Society Press, 561–576, 1996.

[12] J. M. Kleinberg. *Approximation Algorithms for Disjoint Paths Problems* . Ph.D dissertation, Dept. of EECS, MIT, Cambridge, Mass., 1996.

[13] J. Kleinberg and E. Tardos. Disjoint paths in densely embedded graphs. *Proc. of 36th Annual IEEE Symposium on Foundations of Computer Science*, 1995, 52–61.

[14] C. L. Liu. *Introduction to Combinatorial Mathematics.* McGraw-Hill, New York, 1968.

[15] Y. Mansour and B. Schieber. Finding the edge connectivity of directed graphs. *J. of Algorithms* **10:**76–85, 1989.

[16] A. D. McAulay. *Optical Computer Architectures.* John Wiley, 1991.

[17] M. Mihail, C. Kaklamanis, and S. Rao. Efficient access to optical bandwidth. *Proc. of 36th Annual IEEE Symposium on Foundations of Computer Science*, 1995, 548–557.

[18] R. K. Pankaj. *Architectures for Linear Lightwave Networks.* Ph.D dissertation. Dept. of EECS, MIT, Cambridge, Mass., 1992.

[19] R. K. Pankaj and R. G. Gallager. Wavelength requirements of all-optical networks. *IEEE/ACM Trans. Networking* **3:**269–280, 1995.

[20] G. R. Pieris and G. H. Sasaki. A linear lightwave Beneš network. *IEEE/ACM Trans. Networking* **1:**441–445, 1993.

[21] F. Preparata and J. Vuillemin. The cube-connected cycles: A versatile network for parallel computation. *Comm. ACM* **24:**300–309, 1981.

[22] Y Rabani. Path coloring on the mesh. *Proc. of 37th Annual IEEE Symposium on Foundations of Computer Science*, Oct., 1996, 400–409.

[23] P. Raghavan and E. Upfal. Efficient routing in all-optical networks. *Proc. of 26th Annual ACM Symposium on Theory of Computing*, May, 1994, 134–143.

[24] R. Ramaswami. Multi-wavelength lightwave networks for computer communication. *IEEE Communications Magazine* **31:**78–88, 1993.

[25] X. Shen, Q. Hu, H. Dai, and X. Wang. Optimal routing of permutations on rings. *ISAAC'94*, Lecture Notes in Computer Science 834, 360–368, 1994.

[26] R. J. Vitter and D. H. C. Du. Distributed computing with high-speed optical networks. *IEEE Computer* **26:**8–18, 1993.

[27] A. Youssef. Off-line permutation routing on circuit-switched fixed-routing networks. *Networks* **23:**441–448: 1993.