

Embedding Meshes in the Star-Connected Cycles Interconnection Network

Marcelo Moraes de Azevedo and Nader Bagherzadeh

Department of Electrical and Computer Engineering
University of California, Irvine – Irvine, CA 92717

Shahram Latifi

Department of Electrical and Computer Engineering
University of Nevada, Las Vegas – Las Vegas, NV 89154-4026

Technical Report ECE 95-10-01 – October 1995

Embedding Meshes in the Star-Connected Cycles Interconnection Network

Marcelo Moraes de Azevedo* and Nader Bagherzadeh†

Department of Electrical and Computer Engineering – University of California – Irvine, CA 92717 – USA
mazevedo, nader@ece.uci.edu Phone: (714) 824-8720 FAX: (714) 824-2321

Shahram Latifi

Department of Electrical and Computer Engineering – University of Nevada – Las Vegas, NV 89154 – USA
latifi@jb.ee.unlv.edu Phone: (702) 895-4016 FAX: (702) 895-4075

Abstract — *The star-connected cycles (SCC) graph was recently proposed as an attractive interconnection network for parallel processing, using a star graph to connect cycles of nodes. This paper presents embeddings of p -dimensional meshes, $1 \leq p \leq n$, into an n -dimensional SCC graph. It is shown that such embeddings can be accomplished with load 1, expansion 1, and dilation ranging from 1 to $3+2\lfloor(n-1)/2\rfloor+2\lfloor n/4\rfloor$ (the dilation varies among mesh dimensions). A comparison between SCC and star graphs is presented, addressing the issue of communication slowdown resulting from embedding meshes in these graphs. We show that, despite the fact that the dilation of our embeddings increases with n , a small communication slowdown can be obtained in the case of the SCC graph, thanks to the feasibility of employing high bandwidth links within the cycles.*

Key words — *Interconnection networks, mesh embedding, parallel processing, star-connected cycles graph.*

1 Introduction

Let G_k be a k -dimensional graph with hierarchical structure [1], such that a $(k+1)$ -dimensional graph G_{k+1} can be obtained recursively from c_k copies of G_k . Several graphs belonging to the class of *Cayley graphs* have this recursive decomposition structure, such as the hypercube [2] and the star graph [3] (which we denote by Q_k and S_n , respectively). Q_{k+1} , for example, can be obtained by connecting two Q_k 's, while S_{n+1} can be obtained by connecting $(n+1)$ S_n 's. The links connecting the c_k copies of G_k that exist within G_{k+1} are referred to as $(k+1)^{th}$ -dimension links.

A graph G_k modeling a particular interconnection network can be described by a set of nodes $V(G_k)$ and a set of links $E(G_k)$, such that each node in $V(G_k)$ corresponds to a processing element (PE) and each link in $E(G_k)$ corresponds to a communication channel connecting two of these PEs. One possible technique for simulating G_k in a second graph H_n is referred to as *embedding* G_k into H_n . The embedding of $G_k = \{V(G_k), E(G_k)\}$ into $H_n = \{V(H_n), E(H_n)\}$ consists of a mapping of $V(G_k)$ into $V(H_n)$ and of $E(G_k)$ into paths of H_n . The graph that is being embedded (G_k) is referred to as the *guest* graph, while the graph that receives the embedding (H_n) is referred to as the *host* graph [4]. The *load* of an embedding is the maximum number of nodes of the guest graph that are embedded into any single node of the host graph. We assume in this paper that each node of the guest graph is mapped to a distinct node of the host graph, i.e. the load of the embedding is 1. Clearly, in order to achieve such an embedding we must have $|V(H_n)| \geq |V(G_k)|$, where $|V(H_n)|$ and $|V(G_k)|$ are, respectively, the number of nodes of H_n and the number

*This research was supported in part by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq - Brazil), under the grant No. 200392/92-1.

†All correspondence should be addressed to N. Bagherzadeh.

of nodes of G_k . The ratio $|V(H_n)|/|V(G_k)|$ is referred to as the *expansion* of the embedding. The *dilation* of the embedding is the longest path in H_n used to map any link $(u, v) \in E(G_k)$, where $u, v \in V(G_k)$.

A common data structure found in many parallel processing applications is the *p-dimensional array*, which we denote by A_p . A_p consists of a *set of positions* $V(A_p)$. The number of positions of A_p is $|V(A_p)| = \ell_1 \times \ell_2 \times \dots \times \ell_p$. Each position in A_p usually holds a particular type of data (such as a real number), and is uniquely identified by a label $\underline{a} = a_1 a_2 \dots a_p$. The j^{th} symbol in \underline{a} (i.e., a_j) is referred to as the j^{th} -*dimension coordinate* of position \underline{a} . The *range* of the j^{th} -dimension coordinate of any position $\underline{a} \in V(A_p)$ is the set of integer values that a_j may take, and follows the convention $0 \leq a_j < \ell_j$ in this paper. Two positions $\underline{a} = a_1 a_2 \dots a_p$ and $\underline{a}' = a'_1 a'_2 \dots a'_p \in V(A_p)$ are *adjacent along dimension j* iff $a_k = a'_k, \forall k \neq j$, and either $a_j = a'_j + 1$ or $a_j = a'_j - 1$. A set of ℓ_j consecutive positions that are adjacent along dimension j in A_p is an j^{th} -*dimension row* of A_p . We refer to ℓ_j as the *size of A_p along the j^{th} dimension*, i.e. there are ℓ_j positions in any j^{th} -dimension row of A_p .

An interconnection network being considered for applications such as numerical analysis, image processing, sorting, and matrix algorithms, to name a few, must provide an efficient support for p -dimensional arrays. One interconnection network that can handle such arrays in a very straightforward manner is the *p-dimensional mesh*, which we denote by M_p . M_p can be defined similarly as A_p , noting however that each label $\underline{m} \in V(M_p)$ identifies a *node* of M_p . Hence, under this approach each node $\underline{m} \in V(M_p)$ holds the data assigned to a corresponding position $\underline{a} \in V(A_p)$. It can be shown trivially that, if the number of nodes of M_p is $|V(M_p)| = s_1 \times s_2 \times \dots \times s_p$, and $\ell_j \leq s_j$, for $1 \leq j \leq p$, then any p -dimensional array A_p containing $|A_p| = \ell_1 \times \ell_2 \times \dots \times \ell_p$ positions can be mapped onto M_p via a one-to-one mapping function $\Gamma : V(A_p) \mapsto V(M_p)$, defined by $\underline{m} = \underline{a}$. Other mapping functions can be derived when these conditions do not hold, such as in the case of algorithms requiring a p' -dimensional array $A_{p'}$ to be mapped onto M_p , where $p' \neq p$.

As a general approach to running array-based algorithms in an interconnection network I_n , one can embed M_p into I_n . This paper considers the embedding of M_p , $1 \leq p \leq n$, into an n -dimensional star-connected cycles graph (SCC_n). The SCC graph was recently proposed [5, 6] as an attractive extension of the star graph [3, 1]. It exhibits a fixed-degree topology, requiring only three I/O ports per node, and presents a less complex lay-out than does a star graph of similar size. Broadcasting can also be executed more efficiently in the SCC graph than it is in the star graph [6], which is advantageous to many parallel algorithms.

This paper is organized as follows. Section 2 briefly reviews the star and SCC graphs. Section 3 presents the embedding of an n -dimensional mesh M_n , containing $|V(M_n)| = 2 \times 3 \times \dots \times (n-2) \times (n-1) \times n \times (n-1) = n! \cdot (n-1)$ nodes, into SCC_n . The proposed embedding has expansion 1, load 1, and dilation ranging from 1 to $3 + 2\lfloor(n-1)/2\rfloor + 2\lfloor n/4\rfloor$. Section 4 discusses how the technique presented in Section 3 can be readily extended to embed p -dimensional meshes, $1 \leq p < n$, into SCC_n . Particularly, an example considering the embedding of an $(n-1)$ -dimensional mesh into an SCC_n graph is given. Section 5 compares SCC and star graphs as possible hosts for mesh embeddings, addressing the issue of the communication slowdown resulting from such embeddings. Concluding remarks are presented in Section 6.

2 Background

2.1 The star graph (S_n)

An n -star graph (S_n) contains $|V(S_n)| = n!$ nodes that are labeled with the $n!$ possible permutations of n distinct symbols. In this paper, we use the digits $\{1, 2, \dots, n\}$ to label the nodes of S_n . A node $\pi = p_1 p_2 \dots p_i \dots p_n$ is connected to $(n-1)$ distinct nodes, respectively labeled with permutations $\pi_i =$

$p_i p_2 \dots p_{i-1} p_{i+1} \dots p_n$, $2 \leq i \leq n$ (i.e., π_i is the permutation resulting from exchanging the digits occupying the first and the i^{th} position in π) [3, 1]. Each of these $(n - 1)$ possible exchange operations is referred to as a *generator* of S_n . Two nodes π and π_i of an n -star graph are connected by a link if there is a generator g_i such that $\pi \cdot g_i = \pi_i$. The link connecting π and π_i is referred to as an i^{th} -dimension link and is labeled i . Figure 1 shows S_4 .

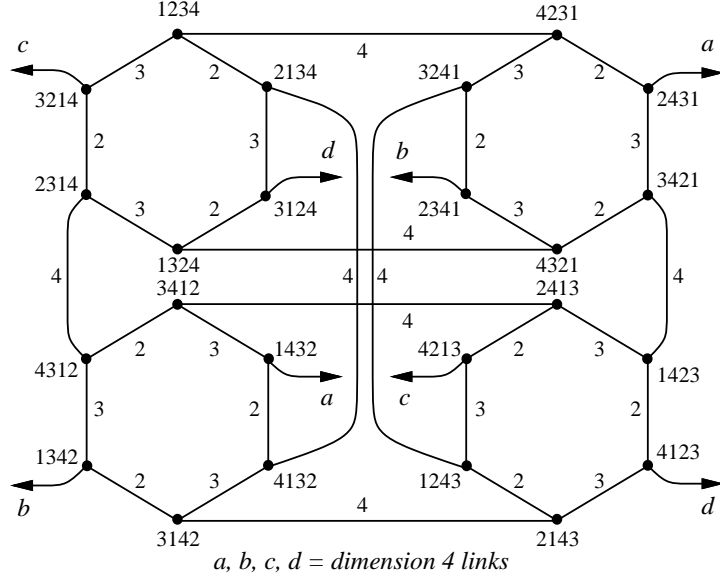


Fig. 1: The S_4 graph

S_n is a regular, vertex- and edge-symmetric graph with degree $\delta(S_n) = n - 1$ and diameter $\phi(S_n) = \lceil 3(n - 1)/2 \rceil$. Its degree and diameter are sublogarithmic on the size of the graph [3], which makes the star graph compare favorably with the hypercube.

2.2 The star-connected cycles graph (SCC_n)

An n -SCC graph (SCC_n) is obtained by replacing each node of S_n with a ring of $(n - 1)$ nodes, namely a *supernode*. The connections between nodes inside the same supernode are referred to as *local links*. Each supernode is connected to $(n - 1)$ adjacent supernodes, using *lateral links* according to the topology of S_n . Figure 2 shows SCC_4 . Although not shown in Figure 2, we assume that an i^{th} -dimension lateral link is labeled i . The nodes in each ring are identified by a label $\langle i, \pi \rangle$, where $2 \leq i \leq n$ and π is a permutation of n digits. Then, two nodes $\langle i, \pi \rangle$ and $\langle i', \pi' \rangle$ are connected by a link $(\langle i, \pi \rangle, \langle i', \pi' \rangle)$ in SCC_n iff either

1. $(\langle i, \pi \rangle, \langle i', \pi' \rangle)$ is a local link, i.e. $\pi = \pi'$ and $\min(|i - i'|, n - 1 - |i - i'|) = 1$, or
2. $(\langle i, \pi \rangle, \langle i', \pi' \rangle)$ is a lateral link, i.e. $i = i'$ and π differs from π' only in the first and i^{th} digits such that $\pi(1) = \pi'(i)$ and $\pi(i) = \pi'(1)$.

SCC_n is a regular, vertex-symmetric graph containing $|V(SCC_n)| = (n - 1) \cdot n!$ nodes. The degree of SCC_n is $\delta(SCC_n) = n - 1$ (for $n \leq 3$), and $\delta(SCC_n) = 3$ (for $n \geq 4$). The diameter of SCC_n is [6, 5]:

$$\phi(SCC_n) = \begin{cases} 6, & \text{for } n = 3 \\ \frac{1}{2}(n^2 + n - 4), & \text{for even } n \\ \frac{1}{2}(n^2 + 3n - 8), & \text{for odd } n \geq 5 \end{cases} \quad (1)$$

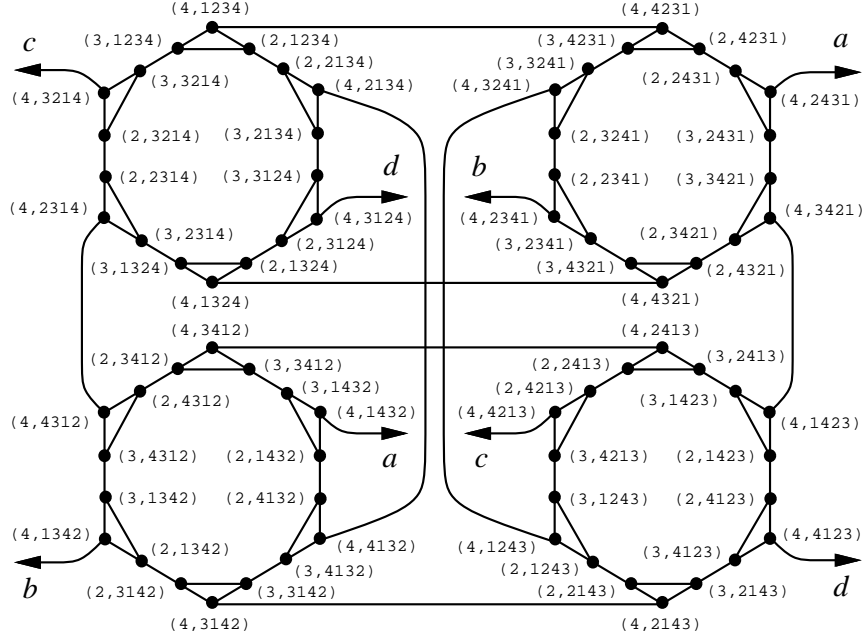


Fig. 2: The SCC_4 graph

2.3 Basic considerations about routing in SCC_n

Routing between two nodes $\langle i_s, \pi_s \rangle$ and $\langle i_d, \pi_d \rangle$ in an SCC graph is equivalent to routing from $\langle i_s, \pi_{ds} \rangle$ to $\langle i_d, \pi_1 \rangle$, where $\pi_{ds} = \pi_d^{-1} \cdot \pi_s$, $\pi_1 = 123 \dots n$, and π_d^{-1} is the *inverse or reciprocal* of permutation π_d , such that $\pi_d \cdot \pi_d^{-1} = \pi_d^{-1} \cdot \pi_d = \pi_1$ [1, 5, 7]. In particular, we note that routing from supernode π_s to supernode π_d (or, equivalently, from supernode π_{ds} to supernode π_1) can be accomplished with the same routing techniques that have been proposed for the star graph [3].

We may organize the digits of permutation π_{ds} as a set of cycles – i.e., cyclically ordered sets of digits with the property that each digit's desired position is that occupied by the next digit in the set. For example, a permutation $\pi_{ds} = 23154$ labeling a supernode of a 5-SCC graph can be represented in cyclic format by $(1\ 2\ 3)(4\ 5)$. This cyclic representation has the particularity that the cycles can be listed in any order. In addition, any cyclic shift of the digits within a cycle can be used without affecting the unique permutation label of π_{ds} . For example, $(1\ 2\ 3)(4\ 5)$, $(1\ 2\ 3)(5\ 4)$, $(2\ 3\ 1)(4\ 5)$, $(2\ 3\ 1)(5\ 4)$, $(3\ 1\ 2)(4\ 5)$ and $(3\ 1\ 2)(5\ 4)$ are all valid cyclic representations of permutation $\pi_{ds} = 23154$. In this paper, all cycles are written in canonical form (i.e. the smallest digit appears first in each cycle). However, the cycles can be listed in any order.

A cycle containing r digits is referred to as an r -*cycle*. Note that any digit already in its correct position in π_{ds} appears as a 1-cycle. Let $C_i \in \pi_{ds}$ be an r -cycle of the form $C_i = (a_1\ a_2\ \dots\ a_r)$, $2 \leq r \leq n$. The *execution of C_i* corresponds to a path R_i in the quotient n -star embedded in the n -SCC graph and can be expressed as a sequence of lateral links as follows [5, 8]:

- If $a_1 = 1$:

$$R_i = (a_2, a_3, \dots, a_r) \quad (2)$$

- If $a_1 \neq 1$:

$$R_i = (a_{1+k \bmod r}, a_{1+(k+1) \bmod r}, \dots, a_{1+(k+r-1) \bmod r}, a_{1+k \bmod r}) \quad , \text{ for } 0 \leq k \leq r-1 \quad (3)$$

The execution of each cycle $C_i \in \pi_{ds}$ is part of a route containing a minimum number of lateral links between the source and the destination supernodes. For example, one possible path R_{ds} from supernode $\pi_{ds} = 23154 = (1\ 2\ 3)(4\ 5)$ to supernode $\pi_1 = 12345 = (1)(2)(3)(4)(5)$ in a 5-SCC graph is the sequence of lateral links (2, 3, 4, 5, 4):

$$23154 \xrightarrow{2} 32154 \xrightarrow{3} 12354 \xrightarrow{4} 52314 \xrightarrow{5} 42315 \xrightarrow{4} 12345$$

Let c be the number of cycles of length at least 2 in π_{ds} and m the total number of digits in these cycles. The minimum number of lateral links required in the routing between supernodes π_{ds} and π_1 is [3]:

$$|R_{ds}| = \begin{cases} c + m, & \text{if the first digit in } \pi_{ds} \text{ is 1} \\ c + m - 2, & \text{if the first digit in } \pi_{ds} \text{ is not 1} \end{cases} \quad (4)$$

The number of local links that is required to traverse a supernode between its i^{th} and j^{th} nodes is:

$$d(i, j) = \min(|i - j|, n - 1 - |i - j|) \quad (5)$$

The number of local links that is required to execute an r -cycle $C_i = (a_1\ a_2\ \dots\ a_r) \in \pi_{ds}$, $r \geq 2$, is:

$$loc(C_i) = \begin{cases} \sum_{j=2}^r d(a_{j-1}, a_j) + d(a_r, a_1), & \text{if } a_1 \neq 1 \\ \sum_{j=3}^r d(a_{j-1}, a_j), & \text{if } a_1 = 1 \end{cases} \quad (6)$$

Note that if $a_1 \neq 1$ Equation 6 holds for any of the r different sequences R_i that may be used to execute C_i (Equation 3). This observation immediately follows from the cyclic nature of these sequences.

The order chosen to execute the cycles in π_{ds} affects the number of local links that have to be traversed in the corresponding path in the SCC graph. A routing algorithm that finds the shortest path between a pair of nodes $\langle i_s, \pi_s \rangle$ to $\langle i_d, \pi_d \rangle$ in an n -SCC graph is described in [6]. In general, the total cost of a path $P(\ell_1 \mapsto \ell_f)$ from $\langle i_s, \pi_s \rangle$ to $\langle i_d, \pi_d \rangle$ along a sequence of f lateral links $R(\ell_1 \mapsto \ell_f) = (\ell_1, \ell_2, \dots, \ell_f)$ is:

$$|P(\ell_1 \mapsto \ell_f)| = f + d(i_s, \ell_1) + \sum_{j=1}^{f-1} d(\ell_j, \ell_{j+1}) + d(\ell_f, i_d) \quad (7)$$

To illustrate the cost of local links in the routing, assume routing between nodes $\langle 3, 34125 \rangle$ and $\langle 2, 12345 \rangle$ in SCC_5 . A path along the sequence $R(2 \mapsto 3) = (2, 4, 2, 3)$ contains 4 lateral links and 7 local links ($|P(2 \mapsto 3)| = 11$). However, if the sequence of lateral links $R(3 \mapsto 2) = (3, 2, 4, 2)$ is used, a path with 4 lateral links and 5 local links results ($|P(3 \mapsto 2)| = 9$).

3 Embedding an n -dimensional mesh into SCC_n

In this section, we consider the embedding of an n -dimensional mesh (M_n) into SCC_n . As discussed in Section 4, such an embedding can be easily transformed into other embeddings to support the more general case of p -dimensional meshes, $1 \leq p < n$.

3.1 Mapping technique

Let M_n be an n -dimensional mesh of size $|V(M_n)| = 2 \times 3 \times 4 \times \dots \times (n-1) \times n \times (n-1)$. The nodes of M_n are labeled with an n -digit vector $\underline{m} = m_1 m_2 \dots m_n$, $0 \leq m_j < s_j$, where s_j is the size of the mesh along its j^{th} dimension:

$$s_j = \begin{cases} j + 1, & \text{for } 1 \leq j \leq n - 1 \\ j - 1, & \text{for } j = n \end{cases} \quad (8)$$

Two nodes $\underline{m} = m_1 m_2 \dots m_n$ and $\underline{m}' = m'_1 m'_2 \dots m'_n \in V(M_n)$ are adjacent along the j^{th} dimension of M_n iff $m_k = m'_k, \forall k \neq j$, and either $m_j = m'_j + 1$ or $m_j = m'_j - 1$. We assume that M_n contains no wrap arounds (i.e., two nodes whose labels are $m_1 m_2 \dots 0 \dots m_n$ and $m_1 m_2 \dots (s_j - 1) \dots m_n$ are *not* adjacent along the j^{th} dimension of M_n). It can be easily verified that the degree of any node in M_n is at most $2n - 1$. This results in the following lemma:

Lemma 1 *There is no load 1, dilation 1 embedding of M_n into SCC_n , $n > 2$.*

Proof: For $1 \leq n \leq 2$, M_n and SCC_n are isomorphic graphs and, therefore, a straightforward embedding with load 1, dilation 1 exists. We now consider the case $n > 2$. A load 1, dilation 1 embedding of a graph G into a graph H requires that the degree of a node in G must be less than or equal to the degree of a node in H . The degree of a node $\underline{m} \in V(M_n)$ is bounded by $\delta(\underline{m}) \leq 2n - 1$. As described in Subsection 2.2, SCC_n is a regular graph with $\delta(SCC_n) = 2$ (for $n = 3$), and $\delta(SCC_n) = 3$ (for $n \geq 4$). A load 1, dilation 1 embedding of M_n into SCC_n would require that $2n - 1 \leq \delta(SCC_n)$. For $n = 3$, this condition becomes $2n - 1 \leq 2$, or $n \leq 1$. For $n \geq 4$, this condition becomes $2n - 1 \leq 3$, or $n \leq 2$. As a contradiction is produced in either case, there is no load 1, dilation 1 embedding of M_n into SCC_n , unless for $n = 1$ and $n = 2$. \square

The goal of our work consisted of obtaining a load 1, expansion 1 embedding of M_n into SCC_n . These characteristics ensure that the amount of processing load in the nodes of M_n is the same as in SCC_n , and that there are no unused nodes left in the host graph. Considering the limitations posed by Lemma 1 and the topology of the SCC_n graph, we obtained an embedding that produces a small communication slowdown, despite the fact that its dilation grows with n (the issue of communication slowdown is discussed in Section 5). We use a mapping function $\Gamma : V(M_n) \mapsto V(SCC_n)$, inspired by the technique proposed by Ranka et al. for embedding an $(n - 1)$ -dimensional mesh into an n -star graph [9]. Such an embedding has load 1, expansion 1, and dilation 3.

The mapping of a node $\underline{m} = m_1 m_2 \dots m_n, \underline{m} \in V(M_n)$, onto a node $\langle i, \pi \rangle = \langle i, p_1 p_2 \dots p_n \rangle, \langle i, \pi \rangle \in V(SCC_n)$, is accomplished by the algorithm shown below. We assume that the identity node of M_n (namely, $00 \dots 0$) is mapped onto the identity node of SCC_n (namely, $\langle 2, 12 \dots n \rangle$).

Algorithm 1 (Mapping $\underline{m} \in V(M_n)$ onto $\langle i, \pi \rangle \in V(SCC_n)$):

```

mesh_to_SCC (int m[ ], int n, int i, int p[ ])
{
  int j, k, temp;
  for (j = 1; j ≤ n; j++) p[j] = j;
  for (j = 1; j < n; j++)
    for (k = 0; k < m[j]; k++) {
      temp = p[j - k];
      p[j - k] = p[j - k + 1];
      p[j - k + 1] = temp;
    }
  i = m[n] + 2;
}

```

In Algorithm 1, the label of the mesh node (\underline{m}) is stored as a vector ($m[\]$). Accordingly, the two portions of the label of the SCC_n node (i.e., i and π) are stored in an integer variable (i) and a vector ($p[\]$), respectively. Algorithm 1 initially sets permutation π to $12 \dots n$. The next step of the algorithm consists of an iterative inspection of the first $(n - 1)$ coordinates of the mesh node (\underline{m}). Assuming that the coordinate of \underline{m} along the j^{th} dimension of M_n is m_j , the j^{th} iteration of the external *for* loop results in the following sequence of transpositions:

$$(p[j] \leftrightarrow p[j + 1]), (p[j - 1] \leftrightarrow p[j]), \dots, (p[j - m_j + 1] \leftrightarrow p[j - m_j + 2])$$

The transposition of the digits occupying the r^{th} and the s^{th} positions in permutation π (i.e., $(p[r] \leftrightarrow p[s])$) can be represented in cyclic format by $(r\ s)$. Table 1 lists the sequences of transpositions used by the mapping algorithm along the first $(n - 1)$ dimensions of the mesh. Note that if the coordinate of the mesh node along dimension j is m_j , then only the first m_j transpositions of the sequence corresponding to dimension j are used. Finally, the coordinate of the mesh node along the n^{th} dimension (m_n) is mapped directly onto i via the assignment $i = m_n + 2$. This completes the mapping of $\underline{m} \in V(M_n)$ onto $\langle i, \pi \rangle \in V(SCC_n)$.

Dimension (j)	Sequence of transpositions
1	(1 2)
2	(2 3) (1 2)
\vdots	\vdots
$n - 1$	$(n - 1\ n) (n - 2\ n - 1) \dots (1\ 2)$

Table 1: Sequences of transpositions used by Algorithm 1

As an example, assume the mapping of node $\underline{m} = 1032 \in V(M_4)$ onto a node $\langle i, \pi \rangle \in V(SCC_4)$. Initially, Algorithm 1 sets $\pi = 1234$. Since $m_1 = 1$, a (1 2) transposition is performed on 1234, giving 2134. Next, the algorithm examines the coordinate of \underline{m} along the 2nd dimension (m_2). Since $m_2 = 0$, no transpositions are performed at this step. Next, m_3 is examined, resulting in a sequence of transpositions (3 4) (2 3) (1 2). Such a sequence affects π as shown below:

$$2134 \xrightarrow{(34)} 2143 \xrightarrow{(23)} 2413 \xrightarrow{(12)} 4213$$

Next, m_4 is mapped onto $i = 4$. Therefore, $\underline{m} = 1032$ is mapped onto $\langle i, \pi \rangle = \langle 4, 4213 \rangle$. Figure 3 shows the embedding of M_4 into SCC_4 , resulting from the mapping function implemented by Algorithm 1.

The mapping algorithm described in this paper differs slightly from that proposed by Ranka et al. [9] in respect to the definition of a transposition $(r\ s)$. In Algorithm 1, $(r\ s)$ corresponds to an exchange of the digits occupying the r^{th} and the s^{th} positions in permutation π . In the corresponding algorithm described in [9], $(r\ s)$ corresponds to an exchange of *digits* r and s in π . As shown later, the approach used by Algorithm 1 is useful to minimize the dilation of the embedding of M_n into SCC_n along the $(n - 1)^{th}$ dimension of M_n , for some values of n . However, before we can derive an expression for the dilation of the embedding of M_n into SCC_n , a few lemmas and definitions are necessary.

As far as lateral links are concerned, Algorithm 1 exhibits some similarities with the technique of Ranka et al. Let $\underline{m}, \underline{m}' \in V(M_n)$ be two nodes which are adjacent along the j^{th} dimension of M_n , and assume that Algorithm 1 maps \underline{m} and \underline{m}' onto nodes $\langle i, \pi \rangle, \langle i', \pi' \rangle \in V(SCC_n)$, respectively. The following lemma holds in the case $j < n$:

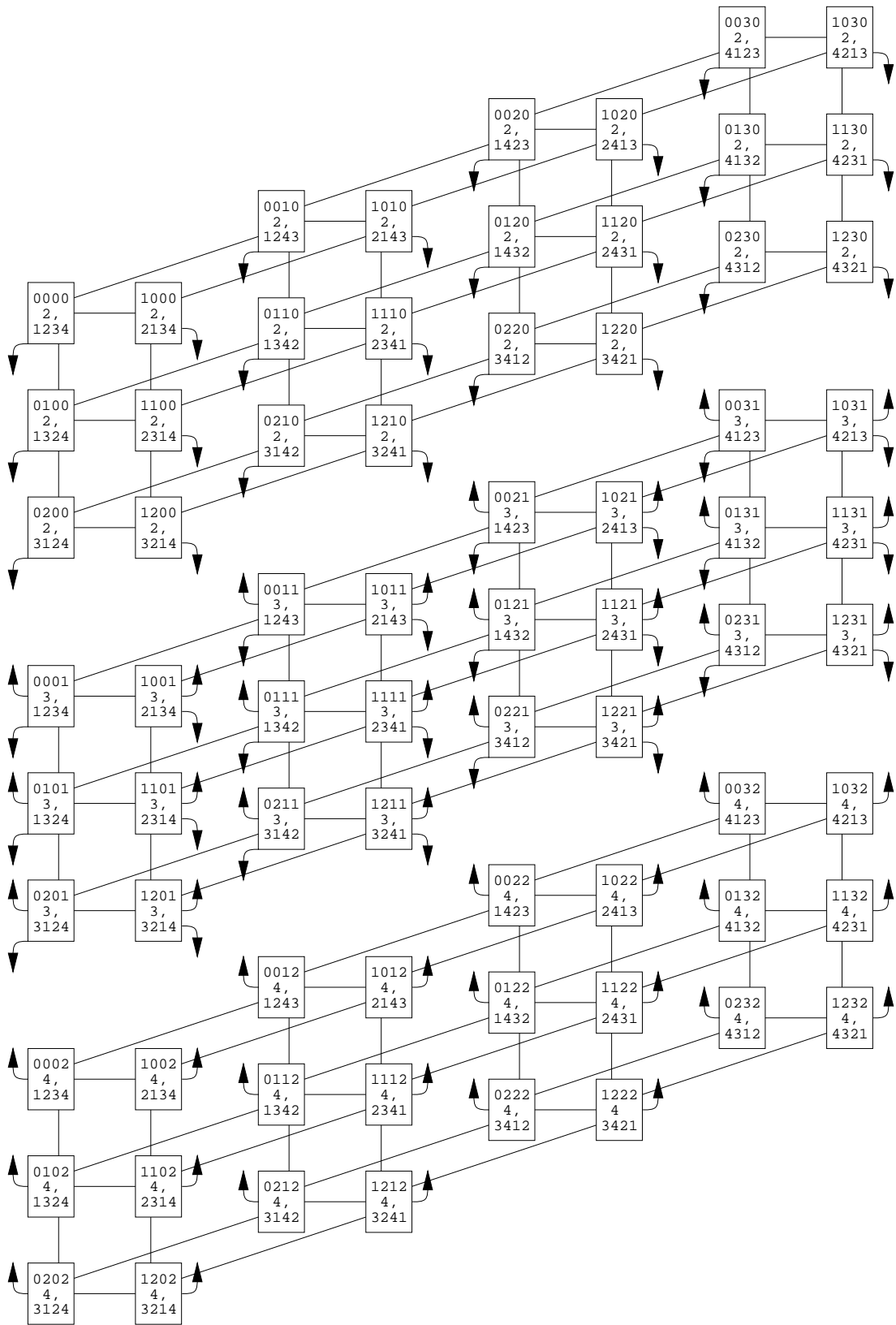


Fig. 3: Embedding of M_4 into SCC_4

Lemma 2 Any two nodes $\underline{m}, \underline{m}' \in M_n$ which are adjacent along the j^{th} dimension of M_n , $j < n$, are connected by a path containing either 1 or 3 lateral links in the corresponding embedding into SCC_n .

Proof: Without loss of generality, assume that the j^{th} coordinates of \underline{m} and \underline{m}' (respectively, m_j and m'_j) are such that $m'_j = m_j + 1$. By inspection of Algorithm 1 and Table 1, we note that the mapping of $\underline{m} \in V(M_n)$ onto $\langle i, \pi \rangle \in V(SCC_n)$ uses a sequence of transpositions of the form:

$$\sigma = (a\ b)(c\ d) \dots (o\ p)(s\ t) \dots (y\ z)$$

Accordingly, \underline{m}' is mapped onto $\langle i', \pi' \rangle$ via a sequence of transpositions of the form:

$$\sigma' = (a\ b)(c\ d) \dots (o\ p)(q\ r)(s\ t) \dots (y\ z)$$

Note that identical transpositions are used in σ and σ' , except for the fact that σ' has one transposition more than does σ (namely, $(q\ r) = ((j - m'_j + 1)(j - m'_j + 2))$). Hence, the mapping of \underline{m} and \underline{m}' is initially carried out identically, with transpositions $(a\ b)(c\ d) \dots (o\ p)$ being used both in σ and in σ' . At the j^{th} iteration of the externally nested *for* loop of Algorithm 1, however, the extra transposition $(q\ r)$ is used only in the mapping sequence of \underline{m}' .

Let the digits occupying the q^{th} and the r^{th} positions of the permutation resulting from applying transpositions $(a\ b)(c\ d) \dots (o\ p)$ to the identity $123 \dots n$ be respectively v_1 and v_2 . Therefore, $(q\ r)$ moves v_1 into v_2 's position, and v_2 into v_1 's position. Since the remaining transpositions in σ, σ' (i.e., $(s\ t) \dots (y\ z)$) are also identical, π and π' will differ only in the final positions occupied by digits v_1 and v_2 . Let these positions be α and β , such that if in π we have $p_\alpha = v_1$ and $p_\beta = v_2$, in π' we have $p_\alpha = v_2$ and $p_\beta = v_1$. Hence, routing from supernode π to supernode π' in SCC_n requires a single transposition $(\alpha\ \beta)$. From Equations 2 and 3, it results that transposition $(\alpha\ \beta)$ uses one lateral link if $\alpha = 1$ and three lateral links otherwise. \square

We now consider the case where \underline{m} and \underline{m}' are adjacent along the n^{th} dimension of M_n :

Lemma 3 Any two nodes $\underline{m}, \underline{m}' \in M_n$ which are adjacent along the n^{th} dimension of M_n are connected by a single local link in the corresponding embedding into SCC_n .

Proof: Let $\underline{m} = m_1 m_2 \dots m_n$. Then, either $\underline{m}' = m_1 m_2 \dots (m_n - 1)$ or $\underline{m}' = m_1 m_2 \dots (m_n + 1)$. Since the first $(n - 1)$ coordinates of \underline{m} and \underline{m}' are equal, Algorithm 1 generates the same sequence of transpositions for both \underline{m} and \underline{m}' . Therefore, if \underline{m} is mapped onto $\langle i, \pi \rangle$ and \underline{m}' is mapped onto $\langle i', \pi' \rangle$, then either $\langle i', \pi' \rangle = \langle i - 1, \pi \rangle$ or $\langle i', \pi' \rangle = \langle i + 1, \pi \rangle$. In any case, routing from $\langle i, \pi \rangle$ to $\langle i', \pi' \rangle$ can be carried out inside supernode $\pi = \pi'$ via a single local link (Equation 5). \square

Definition 1 A sequence of lateral links $\widehat{R}(\ell_1 \mapsto \ell_f) = (\ell_1, \ell_2, \dots, \ell_f)$ is said to be normal if and only if $d(\ell_k, \ell_{k+1}) = 1, \forall k, 1 \leq k < f$.

Note that all transpositions used by Algorithm 1 (Table 1) are actually normal sequences containing either 1 or 3 lateral links. The utilization of normal sequences is particularly advantageous in the case of the SCC graph, since such sequences can often minimize the number of local links in the routing. Suppose that there exists a path $\widehat{P}(\ell_1 \mapsto \ell_f)$ from $\langle \widehat{i}_s, \widehat{\pi}_s \rangle$ to $\langle \widehat{i}_d, \widehat{\pi}_d \rangle$ in an SCC graph, in which the lateral links form a normal sequence $\widehat{R}(\ell_1 \mapsto \ell_f)$. Let the number of lateral links in $\widehat{R}(\ell_1 \mapsto \ell_f)$ be f . By simple substitution into Equation 7, one can verify that the total cost of $\widehat{P}(\ell_1 \mapsto \ell_f)$ is:

$$|\widehat{P}(\ell_1 \mapsto \ell_f)| = 2f - 1 + d(\widehat{i}_s, \ell_1) + d(\ell_f, \widehat{i}_d) \quad (9)$$

In general, there is no guarantee that a normal sequence containing a minimum number of lateral links in the routing between two arbitrary nodes $\langle i_s, \pi_s \rangle, \langle i_d, \pi_d \rangle$ exists. However, as shown in the proof of the following lemma, normal sequences exist between any two nodes $\underline{m}, \underline{m}'$ that are adjacent along the $(n-1)^{th}$ dimension of M_n , in the corresponding embedding into SCC_n .

Lemma 4 *Let \underline{m} and \underline{m}' be two nodes that are adjacent along the $(n-1)^{th}$ dimension of M_n , and assume that in the corresponding embedding into SCC_n , \underline{m} and \underline{m}' are mapped onto $\langle i, \pi \rangle$ and $\langle i', \pi' \rangle$, respectively. Hence, $\langle i, \pi \rangle$ and $\langle i', \pi' \rangle$ are connected by a path containing a normal sequence of either 1 or 3 lateral links. Moreover, the cost of such a path is at most $3 + 2\lfloor n/2 \rfloor$.*

Proof: Let $\underline{m} = m_1 m_2 \dots m_{n-1} m_n$ and $\underline{m}' = m'_1 m'_2 \dots m'_{n-1} m'_n$. Without loss of generality, assume that $\underline{m}' = m_1 m_2 \dots (m_{n-1} + 1) m_n$. By inspection of Algorithm 1 and Table 1, we note that the mapping of $\underline{m} \in V(M_n)$ onto $\langle i, \pi \rangle \in V(SCC_n)$ uses a sequence of transpositions of the form:

$$\sigma = (a\ b)(c\ d) \dots (u\ v)(w\ x)$$

Accordingly, \underline{m}' is mapped onto $\langle i', \pi' \rangle$ via a sequence of transpositions of the form:

$$\sigma = (a\ b)(c\ d) \dots (u\ v)(w\ x)(y\ z)$$

Note that identical transpositions are used in σ and σ' , except for the fact that σ' has one transposition more than does σ (namely, $(y\ z) = ((n - m_{n-1} + 1)\ (n - m_{n-1} + 2))$). Hence, the mapping of \underline{m} and \underline{m}' is initially carried out identically, with transpositions $(a\ b)(c\ d) \dots (w\ x)$ being used both in σ and in σ' . At the $(n-1)^{th}$ iteration of the externally nested *for* loop of Algorithm 1, however, the extra transposition $(y\ z)$ is used only in the mapping sequence of \underline{m}' .

By the same reasoning used in the proof of Lemma 2, we note that routing from supernode π to supernode π' in SCC_n can be accomplished with a single transposition $(\alpha\ \beta)$. However, in this case $(\alpha\ \beta)$ is precisely $(y\ z)$, which by inspection of Table 1 is a normal sequence. Moreover, such a normal sequence contains either one or three lateral links (Equations 2 and 3). Due to our assumption that cycles are represented in canonical form, we have $z = y + 1$. Hence, the sequence of lateral links used to execute $(y\ z)$ is of the form:

$$(y\ z) \equiv \begin{cases} z, & \text{if } y = 1 \\ z, y, z \equiv y, z, y, & \text{if } y \neq 1 \end{cases} \quad (10)$$

We now compute the maximum length of a path between $\langle i, \pi \rangle$ and $\langle i', \pi' \rangle$ in the SCC graph. Since \underline{m} and \underline{m}' have the same n^{th} coordinate (i.e., $m_n = m'_n$), it follows that $i = i' = m_n + 2$. From Equation 9, we note that a path from $\langle i, \pi \rangle$ to $\langle i', \pi' \rangle$ along the possible sequences of lateral links given by Equation 10 has a maximum length of:

$$\begin{aligned} \text{If } y = 1: & \quad 1 + 2 \max(d(i, z)), & \quad \text{for } 2 \leq i, z \leq n \\ \text{If } y \neq 1: & \quad 5 + 2 \max[\min(d(i, z), d(i, y))], & \quad \text{for } 2 \leq i \leq n, 2 \leq y < n, \text{ and } z = y + 1 \end{aligned} \quad (11)$$

A solution to Equation 11 can be obtained by fixing y (and, consequently, z) and varying i from 2 to n :

$$\begin{aligned} \text{If } y = 1: & \quad 1 + 2 \left\lfloor \frac{n-1}{2} \right\rfloor \\ \text{If } y \neq 1: & \quad 3 + 2 \left\lfloor \frac{n}{2} \right\rfloor \end{aligned} \quad (12)$$

Hence, in the worst case the path from $\langle i, \pi \rangle$ to $\langle i', \pi' \rangle$ has a length of $3 + 2\lfloor n/2 \rfloor$. \square

We now consider the case of nodes \underline{m} , \underline{m}' that are adjacent along the j^{th} dimension of M_n , $j < (n - 1)$. As shown in the proof of the following lemma, the shortest path connecting \underline{m} and \underline{m}' in the corresponding embedding into SCC_n does not necessarily contain a normal sequence of lateral links. Consequently, a worst-case analysis reveals paths whose length may exceed that given by Lemma 4.

Lemma 5 *Let \underline{m} and \underline{m}' be two nodes that are adjacent along the j^{th} dimension of M_n , $j < (n - 1)$, and assume that in the corresponding embedding into SCC_n , \underline{m} and \underline{m}' are mapped onto $\langle i, \pi \rangle$ and $\langle i', \pi' \rangle$, respectively. The shortest path connecting $\langle i, \pi \rangle$ and $\langle i', \pi' \rangle$ contains a sequence of either 1 or 3 lateral links, which is not necessarily normal. Moreover, the cost of such a path is at most $3 + 2\lfloor(n - 1)/2\rfloor + 2\lfloor n/4\rfloor$.*

Proof: Let $\underline{m} = m_1 m_2 \dots m_n$ and $\underline{m}' = m'_1 m'_2 \dots m'_n$. Since \underline{m} and \underline{m}' are adjacent along the j^{th} dimension of M_n , $j < (n - 1)$, then either $\underline{m}' = m_1 m_2 \dots (m_j - 1) \dots m_n$ or $\underline{m}' = m_1 m_2 \dots (m_j + 1) \dots m_n$. As in the proof of Lemma 4, we have $m_n = m'_n$ and, consequently, $i = i' = m_n + 2$.

The fact that the path connecting $\langle i, \pi \rangle$ and $\langle i', \pi' \rangle$ contains a sequence of either 1 or 3 lateral links follows directly from Lemma 2. In fact, by the same reasoning used in the proof of Lemma 2, we note that $\pi' = (\alpha \beta) \cdot \pi$ (i.e., supernode π' can be reached by applying a transposition $(\alpha \beta)$ to π , and vice-versa).

Note that although the transpositions used by Algorithm 1 are normal, the resulting transposition $(\alpha \beta)$ needed to route from supernode π to supernode π' is not necessarily normal. As an example, consider the mapping of the following nodes $\underline{m}, \underline{m}' \in M_n$ onto nodes of SCC_n :

- $\underline{m} = 000000 \dots 0$, $\underline{m}' = 010000 \dots 0 \Rightarrow \langle i, \pi \rangle = \langle 2, 1234567 \dots n \rangle$, $\langle i', \pi' \rangle = \langle 2, 1324567 \dots n \rangle$ ($(\alpha \beta) = (2 \ 3)$)
 - $\underline{m} = 001000 \dots 0$, $\underline{m}' = 011000 \dots 0 \Rightarrow \langle i, \pi \rangle = \langle 2, 1243567 \dots n \rangle$, $\langle i', \pi' \rangle = \langle 2, 1342567 \dots n \rangle$ ($(\alpha \beta) = (2 \ 4)$)
 - $\underline{m} = 001100 \dots 0$, $\underline{m}' = 011100 \dots 0 \Rightarrow \langle i, \pi \rangle = \langle 2, 1245367 \dots n \rangle$, $\langle i', \pi' \rangle = \langle 2, 1345267 \dots n \rangle$ ($(\alpha \beta) = (2 \ 5)$)
 - $\underline{m} = 001110 \dots 0$, $\underline{m}' = 011110 \dots 0 \Rightarrow \langle i, \pi \rangle = \langle 2, 1245637 \dots n \rangle$, $\langle i', \pi' \rangle = \langle 2, 1345627 \dots n \rangle$ ($(\alpha \beta) = (2 \ 6)$)
- ⋮

As in the proof of Lemma 4, longer paths occur when $\alpha, \beta \neq 1$, since a total of three lateral links is required to execute transposition $(\alpha \beta)$ (Equation 3). Figure 4 shows the possible paths connecting $\langle i, \pi \rangle$ to $\langle i', \pi' \rangle$, along a sequence of three lateral links.

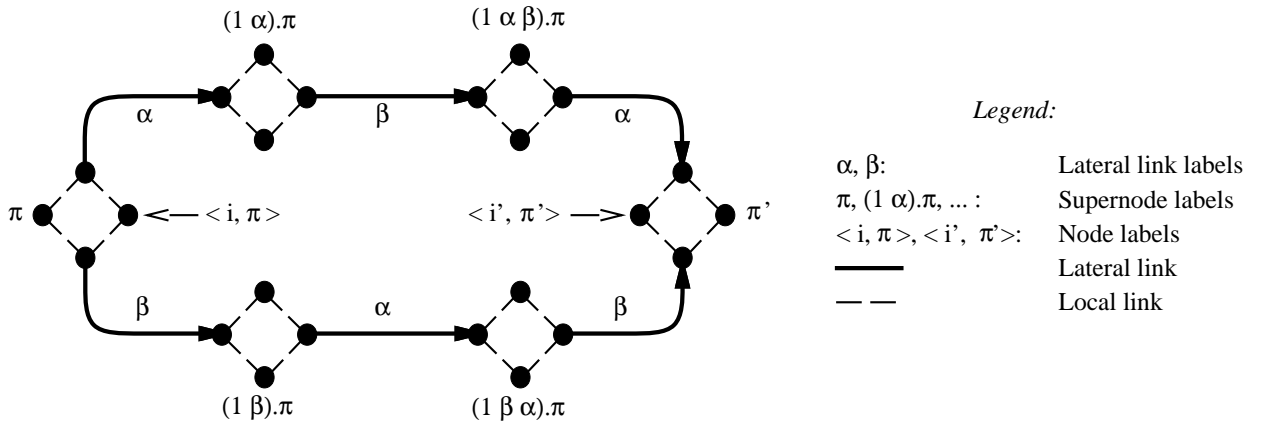


Fig. 4: Paths between nodes that are adjacent along the j^{th} dimension of M_n , $j < (n - 1)$

When $\alpha, \beta \neq 1$, the routing from $\langle i, \pi \rangle$ to $\langle i', \pi' \rangle$ can use two possible sequences of lateral links (namely, (α, β, α) or (β, α, β)) (Figure 4). As the previous example indicates, $(\alpha \beta)$ is not necessarily normal. In fact,

for $\alpha, \beta \neq 1$, the number of local links that must be traversed between two adjacent lateral links (α, β) is at most $\max(d(\alpha, \beta)) = \lfloor (n-1)/2 \rfloor$. In the worst-case, the length of the shortest path connecting $\langle i, \pi \rangle$ to $\langle i', \pi' \rangle$ is:

$$3 + 2 \left\lfloor \frac{n-1}{2} \right\rfloor + 2 \max[\min(d(i, \alpha), d(i, \beta))], \quad \text{for } 2 \leq i \leq n, \text{ and } d(\alpha, \beta) = \left\lfloor \frac{n-1}{2} \right\rfloor \quad (13)$$

A solution to Equation 13 can be obtained by fixing α (and, consequently, β , since a worst-case scenario where $d(\alpha, \beta) = \lfloor (n-1)/2 \rfloor$ is assumed) and varying i from 2 to n . Under these constraints, an inspection of a supernode of SCC_n reveals that the maximum value of $\min(d(i, \alpha), d(i, \beta))$ is $\lfloor n/4 \rfloor$. Therefore, $\langle i, \pi \rangle$ and $\langle i', \pi' \rangle$ are connected by a path whose length is at most $3 + 2\lfloor (n-1)/2 \rfloor + 2\lfloor n/4 \rfloor$. \square

It becomes apparent from Lemmas 3, 4, and 5 that the dilation of the embedding of M_n into SCC_n varies according to the dimension along which two adjacent nodes of M_n are connected. Due to this reason, we refer to the embedding of M_n into SCC_n presented in this paper as a *variable-dilation embedding*.

Definition 2 Let d_j be the cost of the longest path of H_n onto which a j^{th} -dimension link of G_k is mapped in an embedding of G_k into H_n . We refer to such an embedding as having variable dilation if and only if for at least two different dimensions j, j' of G_k , $d_j \neq d_{j'}$.

Definition 3 The dilation vector of a variable-dilation embedding of G_k into H_n , denoted by $\bar{d} = [d_1, d_2, \dots, d_j, \dots, d_k]$, is a vector in which the j^{th} element, d_j , specifies the dilation corresponding to the j^{th} -dimension links of G_k .

Definition 4 Let $\bar{d} = [d_1, d_2, \dots, d_j, \dots, d_k]$ be the dilation vector characterizing a variable-dilation embedding of G_k into H_n . The average dilation of such an embedding, denoted by d_{avr} , is the arithmetic mean of the values in \bar{d} , i.e.:

$$d_{avr} = \frac{1}{k} \sum_{j=1}^k d_j \quad (14)$$

Given the definitions above, we state the following theorem:

Theorem 1 Algorithm 1 accomplishes a load 1, expansion 1, variable-dilation embedding of M_n into SCC_n . The dilation vector of such an embedding is:

$$d_j = \begin{cases} 3 + 2 \left\lfloor \frac{n-1}{2} \right\rfloor + 2 \left\lfloor \frac{n}{4} \right\rfloor, & \text{if } 1 \leq j < (n-1) \\ 3 + 2 \left\lfloor \frac{n}{2} \right\rfloor, & \text{if } j = n-1 \\ 1, & \text{if } j = n \end{cases} \quad (15)$$

Proof: Follows directly from Lemmas 3, 4, and 5. \square

Table 2 lists the dilation vector and the average dilation of the embedding of M_n into SCC_n , for $3 \leq n \leq 7$. Note that, among the embeddings depicted in Table 2, a smaller dilation is obtained along the $(n-1)^{\text{th}}$ -dimension of M_n than along dimensions 1 through $(n-2)$, in the cases $n=5$ and $n=7$. This is justified by the utilization of normal sequences of lateral links, as explained in the proof of Lemma 4.

n	M_n	$ V(M_n) $	Dilation vector (\bar{d})	Average dilation (d_{avr})
3	$2 \times 3 \times 2$	12	$[5, 5, 1]$	3.67
4	$2 \times 3 \times 4 \times 3$	72	$[7, 7, 7, 1]$	5.50
5	$2 \times 3 \times 4 \times 5 \times 4$	480	$[9, 9, 9, 7, 1]$	7.00
6	$2 \times 3 \times 4 \times 5 \times 6 \times 5$	3600	$[9, 9, 9, 9, 9, 1]$	7.67
7	$2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 6$	30240	$[11, 11, 11, 11, 11, 9, 1]$	9.29

Table 2: Dilation vector and average dilation of embeddings of M_n into SCC_n

4 Embedding p -dimensional meshes into SCC_n ($1 \leq p < n$)

A simple extension of the technique described in the previous section can be used to embed a p -dimensional mesh into SCC_n , where $1 \leq p < n$. To illustrate the extended technique, assume the embedding of a 3-dimensional mesh containing $(2 \times 3) \times 4 \times 3 = 6 \times 4 \times 3$ nodes into a 4-SCC graph (Figure 5). The embedding shown in Figure 5 is obtained from that of Figure 4 by unfolding M_4 along its first and second dimensions. Hence, dimensions 1 and 2 of M_4 are collapsed into a single dimension, which results in a $6 \times 4 \times 3$ mesh.

In general, one can apply this technique along any two dimensions j and k of M_n . To form a single dimension jk from j and k , the nodes of M_n must be rearranged in snakelike order along dimensions j and k . Let the size of M_n along dimensions j and k be s_j and s_k , respectively, and let the mesh obtained by unfolding M_n along dimensions j and k be denoted by $M_{n-1}^{j,k}$. $M_{n-1}^{j,k}$ is $(n-1)$ -dimensional, and its size along dimension jk is $s_j \times s_k$. Both M_n and $M_{n-1}^{j,k}$ have $(n-1) \cdot n!$ nodes, and can be embedded with load 1, expansion 1 into SCC_n . The dilation along dimension jk of $M_{n-1}^{j,k}$ is $\max(d_j, d_k)$, where d_j, d_k are the dilations along dimensions j and k of M_n in its corresponding embedding into SCC_n , respectively. This is justified by the fact that $M_{n-1}^{j,k}$ is obtained by unfolding M_n in snakelike order and, hence, nodes that are adjacent in $M_{n-1}^{j,k}$ along dimension jk were originally adjacent in M_n along either dimension j or dimension k . This reasoning can be further extended to meshes with an even smaller dimensionality. An $(n-2)$ -dimensional mesh, for example, can be obtained by unfolding $M_{n-1}^{j,k}$ along any two dimensions, and so on. Moreover, due to the several possible ways in which M_n can be unfolded, one can select the size of a p -dimensional mesh along each of its dimensions by choosing a proper combination of the dimensions originally available in M_n .

The embedding of p -dimensional meshes, $1 \leq p < n$, into SCC_n can be accomplished as follows. Let M_p be a p -dimensional mesh, and let \underline{m} be a node of M_p . Recalling that \underline{m} is a label containing p coordinates, we initially transform \underline{m} into a label with n coordinates, which we denote by \underline{q} . This transformation is carried out according to the order chosen to unfold M_n . Once \underline{q} has been obtained, a straightforward mapping of $\underline{q} \in V(M_n)$ onto $\langle i, \pi \rangle \in V(SCC_n)$ is accomplished via a call to Algorithm 1. To illustrate this two-step method, a mapping algorithm that can be used for $(n-1)$ -dimensional meshes of the form $M_{n-1}^{1,2}$ follows. Such an algorithm was employed in the embedding of $M_3^{1,2}$ into SCC_4 shown in Figure 5.

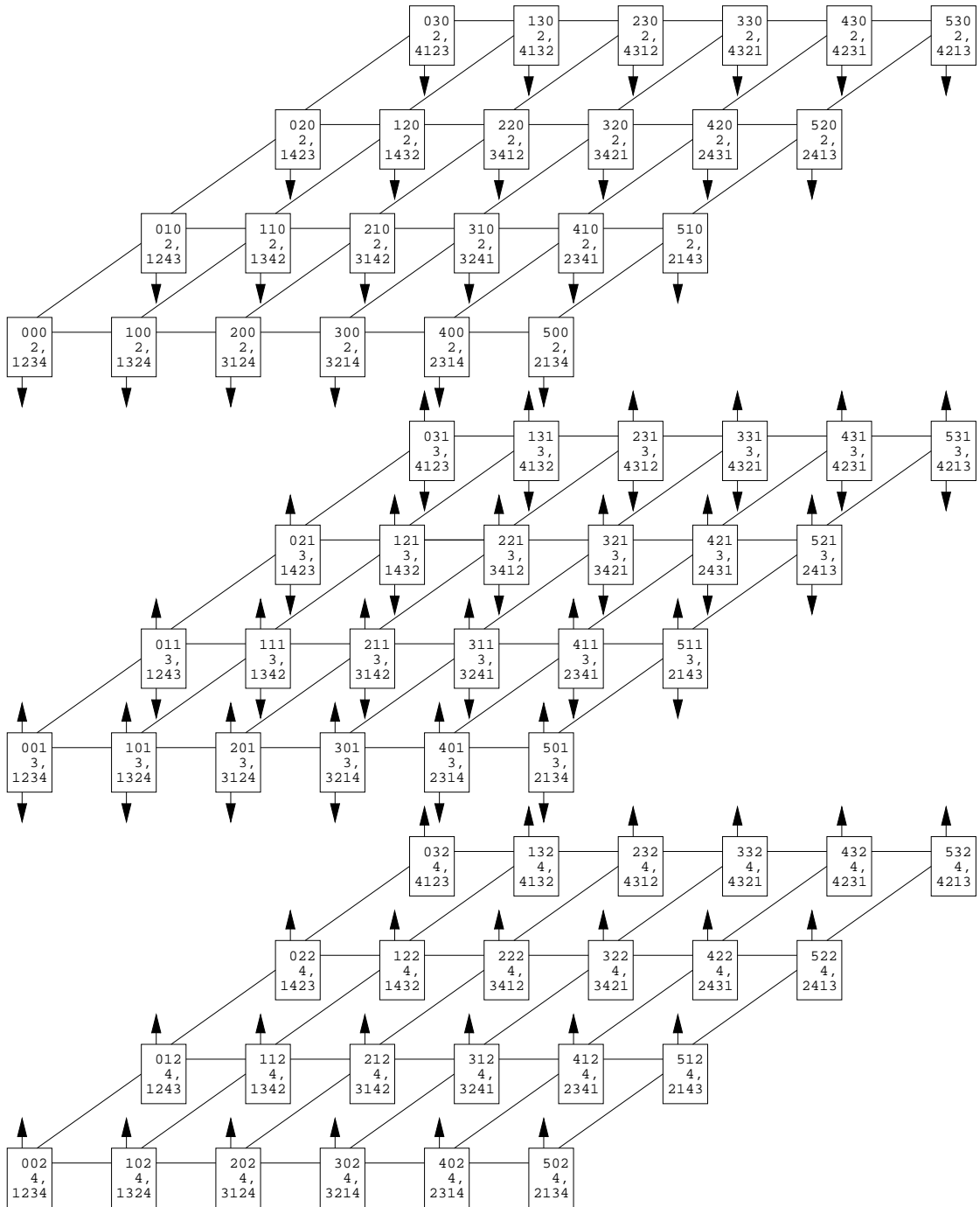


Fig. 5: Embedding of a $6 \times 4 \times 3$ mesh ($M_3^{1,2}$) into SCC_4

Algorithm 2 (Mapping $\underline{m} \in V(M_{n-1}^{1,2})$ onto $\langle i, \pi \rangle \in V(SCC_n)$):

```

N-1mesh_to_SCC (int m[ ], int n, int i, int p[ ])
{
  int j, q[ ];
  for (j = 3; j ≤ n; j++) q[j] = m[j - 1];
  if (m[1] < 3) {
    q[1] = 0;
    q[2] = m[1];
  }
  else {
    q[1] = 1;
    q[2] = 5 - m[1];
  }
  mesh_to_SCC (q[ ], n, i, p[ ])
}

```

5 Communication slowdown

In this section, we compare SCC and star graphs as possible candidates for hosting the embedding of an n -dimensional mesh. Before we do so, we shed some light on the issue of the communication slowdown produced by an embedding, which is the main criteria used in our comparison.

An embedding of a guest graph G_k into a host graph H_n is a useful technique by which algorithms originally designed to run in G_k can run in H_n . However, one usually expects some reduction in performance when H_n is used to simulate G_k , mainly because of the communication slowdown introduced by the embedding, which we denote by η . Frequently, the dilation of the embedding (d) is used to estimate η (i.e., $\eta \approx d$).

In the case of variable-dilation embeddings, the communication slowdown for a particular algorithm A will depend on how the dimensions of G_k are used during A 's execution. Let $\bar{d} = [d_1, d_2, \dots, d_k]$ be the dilation vector characterizing a variable-dilation embedding of G_k into H_n , and let the maximum dilation of the embedding be $d_{max} = \max(d_i)$, for $1 \leq i \leq k$. A conservative estimate for the communication slowdown of such an embedding is d_{max} . Many algorithms, however, use a limited number of dimensions at any given step of their execution, resulting in a smaller slowdown. Notably, algorithms following the SIMD computational model require that all nodes in the interconnection network operate in a lockstep fashion, using a unique dimension at every step of the algorithm. Note that the slowdown for an algorithm devised according to the MIMD computational model may also be smaller than d_{max} , depending on how the dimensions of G_k are used and on the nature of code dependencies arising during the execution of the algorithm.

To simplify the following discussion, assume that A is an algorithm devised to run under the SIMD computational model in G_k . Let τ_j be the number of steps using j^{th} -dimension links in A , $1 \leq j \leq k$. The communication slowdown during A 's execution, resulting from a variable-dilation embedding of G_k into H_n , can be estimated by:

$$\eta \approx \left(\sum_{j=1}^k \tau_j d_j \right) \left(\sum_{j=1}^k \tau_j \right)^{-1} \quad (16)$$

If we make the simplifying assumption that A is such that $\tau_j = \tau, \forall j$, Equation 16 reduces to

$$\eta \approx \frac{1}{k} \sum_{j=1}^k d_j = d_{avr}, \quad (17)$$

where d_{avr} is the *average dilation* of the variable-dilation embedding of G_k into H_n (Definition 4).

Estimates such as those given by Equations 16 and 17 can provide a good indication about the quality of an embedding. Assuming that two different mapping functions Γ and Φ can be used to embed G_k into H_n , one will usually choose Γ if Φ results in an embedding with larger dilation, and consequently, larger communication slowdown. Of course, other criteria are also important when comparing different embeddings, such as load and expansion, but the discussion here will focus on the issue of communication slowdown only.

A possible pitfall exists when one is not only evaluating different mapping functions, but also different host graphs. In this case, estimates based purely on the dilation of the embedding may overlook some important characteristics of the host graphs, which possibly leads to inaccurate comparisons. For example, assume that we want to compare an embedding of G_k into H_n with an embedding of G_k into J_n . Hence, not only the dilation of each embedding is a point of concern for an analysis of the communication slowdown, but also the bandwidth of the links of H_n and J_n becomes an issue.

To illustrate this reasoning, we take as an example the embedding of meshes into S_n and into SCC_n . Such a comparison seems particularly adequate for this paper, since a first glance at the dilation figures presented in Table 2 seems to indicate that the SCC graph is not a good candidate for hosting a mesh. The star graph, for example, seems to be a superior choice under this aspect, since it provides embeddings with a fixed and small dilation $d = 3$ [9].

From the proofs of Lemmas 4 and 5, it becomes apparent that a major contribution to the larger dilation posed by the SCC graph comes from the local links. In fact, paths between nodes that are adjacent in M_n always contain at most three lateral links in the corresponding embedding into SCC_n . The number of local links in such paths, however, increases with n , except in the case of nodes that are adjacent along the n^{th} dimension of M_n .

As discussed in [6], due to the cluster-like nature of the supernodes, the SCC can be built with high bandwidth parallel buses in the local links, at the expense of a minimum increase in the lay-out complexity. The lay-out of the lateral links, however, is characterized by similar wiring constraints found in high-dimensional networks such as the star graph and the hypercube [10], which often have to be implemented with narrower bandwidth communication links.

Let BW_{loc} and BW_{lat} be, respectively, the bandwidth of the local and lateral links of an SCC graph, and let $\nu = BW_{loc}/BW_{lat}$. Since the lateral links of SCC_n and S_n exhibit the same topology and, consequently, similar wiring and lay-out constraints, it seems reasonable to assume in our comparison that the bandwidth of the links of S_n is also BW_{lat} .

Assume that the embedding described in [9] is used as a reference for comparison with the results presented in this paper. Let $\eta(S_n)$ be the communication slowdown resulting from embedding an $(n - 1)$ -dimensional mesh into S_n via the technique of Ranka et al. [9]. If we use Equation 17 to evaluate $\eta(S_n)$, then $\eta(S_n) \approx 3$. This follows from the observation that the embedding presented in [9] has a fixed dilation $d = 3$. Other significant characteristics of such an embedding are: load 1, expansion 1.

The embedding of M_n into SCC_n also has load 1 and expansion 1. Hence, the communication slowdown is indeed the main criteria in a comparison between the embedding of M_n into SCC_n and the reference embedding described above. From the previous discussion regarding the bandwidth of the links of S_n and SCC_n , and due to the utilization of the embedding proposed by Ranka et al. as a reference for comparison,

we estimate the communication slowdown of the embedding of M_n into SCC_n , along the j^{th} dimension of M_n , by:

$$\eta(SCC_n)_j \approx \begin{cases} 3 + \frac{1}{\nu} \left(2 \left\lfloor \frac{n-1}{2} \right\rfloor + 2 \left\lfloor \frac{n}{4} \right\rfloor \right), & \text{if } 1 \leq j < (n-1) \\ 3 + \frac{1}{\nu} \left(2 \left\lfloor \frac{n}{2} \right\rfloor \right), & \text{if } j = n-1 \\ \frac{1}{\nu}, & \text{if } j = n \end{cases} \quad (18)$$

Equation 18 is obtained from Equation 15 by scaling the contributions due to the local links of SCC_n with a factor of $1/\nu$. Accordingly, the average communication slowdown of the embedding of M_n into SCC_n is estimated by:

$$\eta(SCC_n)_{avr} \approx \frac{1}{n} \sum_{j=1}^n \eta(SCC_n)_j \quad (19)$$

Table 3 compares estimates for the communication slowdown resulting from the embedding of meshes into S_n and SCC_n . The values for $\eta(SCC_n)_{avr}$ are computed from Equations 18 and 19, assuming that $\nu = n$. Such an assumption tries to reflect the fact that, as n grows, the size of both S_n and SCC_n increases rapidly, augmenting the complexity in the wiring and lay-out of lateral links. As a consequence, the ratio $\nu = BW_{loc}/BW_{lat}$ is expected to increase as well. A more precise investigation on the dependence of ν with n lies outside the scope of this paper, and is object of future research combining the VLSI lay-out and channel bandwidths of the SCC graph. As a matter of fact, we suspect that ν actually grows at an even faster rate than the assumption $\nu = n$ adopted in this paper.

n	Mesh embedded into S_n	Mesh embedded into SCC_n	Size of S_n ($ V(S_n) $)	Size of SCC_n ($ V(SCC_n) $)	$\eta(S_n)$	$\eta(SCC_n)_{avr}$
3	2×3	$2 \times 3 \times 2$	6	12	3	2.56
4	$2 \times 3 \times 4$	$2 \times 3 \times 4 \times 3$	24	72	3	3.06
5	$2 \times 3 \times 4 \times 5$	$2 \times 3 \times 4 \times 5 \times 4$	120	480	3	3.32
6	$2 \times 3 \times 4 \times 5 \times 6$	$2 \times 3 \times 4 \times 5 \times 6 \times 5$	720	3600	3	3.36
7	$2 \times 3 \times 4 \times 5 \times 6 \times 7$	$2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 6$	5040	30240	3	3.53

Table 3: Communication slowdown of embeddings of meshes into S_n and SCC_n

It becomes apparent from Table 3 that the SCC graph is indeed a good candidate for embedding meshes, which is not evident by a rough analysis of the dilation figures shown in Table 2. From the previous discussion and the results presented in Table 3, we note that the communication slowdown of embeddings into SCC_n may actually be similar to those achieved with embeddings into S_n .

Moreover, the embeddings presented in this paper also exhibit some of the advantages of those proposed in [9] for S_n . Specifically, since the embeddings have load 1 and expansion 1, the processing load resulting from mapping a mesh node onto a node of SCC_n (S_n) is maintained, and the host graph is used in its totality.

Finally, we note that an SCC_n graph has about the same number of nodes of an S_{n+1} graph, since $|V(SCC_n)| = (n-1) \cdot n!$ and $|V(S_{n+1})| = (n+1)! = (n+1) \cdot n!$. Hence, when selecting a host graph to embed an n -dimensional mesh, one may consider either SCC_n or S_{n+1} . This fact contributes favorably to SCC_n , since the number of lateral links in SCC_n (i.e., $(n-1) \cdot n!/2$) is significantly smaller than the number of links in S_{n+1} (i.e., $n \cdot (n+1)!/2 = n \cdot (n+1) \cdot n!/2$). As a consequence, it is expected that wiring and

lay-out constraints may actually cause the bandwidth of the lateral links of SCC_n to be larger than that of the links of S_{n+1} . In other words, SCC_n may actually outperform S_{n+1} as far as embedding n -dimensional meshes is concerned. Other advantages of SCC graphs in comparison to star graphs are discussed in [6, 5].

6 Conclusion

This paper presented a technique for embedding an n -dimensional mesh (M_n) into an SCC_n graph. The proposed embedding has load 1, expansion 1, and presents a variable-dilation nature: the dilation varies from 1 to $3 + 2\lfloor(n-1)/2\rfloor + 2\lfloor n/4\rfloor$, according to the dimensions of the mesh. Our results indicate that the maximum and average dilation of the embedding grow with n . For $3 \leq n \leq 7$, for example, the maximum dilation of the proposed embedding grows from 5 to 11, while the average dilation grows from 3.67 to 9.29. The dependency of the dilation on n is caused by the fixed-degree nature of the SCC graph and its routing characteristics. In contrast, $(n-1)$ -dimensional meshes can be embedded into an n -dimensional star graph (S_n) with a constant dilation of 3 [9].

The maximum dilation of the embedding of M_n into SCC_n results from a worst-case analysis that reveals paths containing up to 3 lateral links and $2\lfloor(n-1)/2\rfloor + 2\lfloor n/4\rfloor$ local links. Hence, lateral links have a fixed contribution to the maximum dilation of the embedding, making our approach similar to that of [9] in that respect. Noting that the topology of the SCC graph favors implementations using high bandwidth buses in the local links [6], we showed that the communication slowdown resulting from embedding meshes into SCC graphs may in fact be similar to or smaller than that obtained with embeddings into star graphs.

A simple extension of our embedding of M_n into SCC_n was also presented in this paper, addressing the more general case of embeddings of p -dimensional meshes into SCC_n , for $1 \leq p < n$.

References

- [1] S. B. Akers and B. Krishnamurthy, "A Group-Theoretic Model for Symmetric Interconnection Networks," *IEEE Transactions on Computers*, Vol. 38, No. 4, April 1989, pp. 555-566.
- [2] Y. Saad and M. H. Schultz, "Topological Properties of Hypercubes," *IEEE Transactions on Computers*, Vol. 37, No. 7, July 1988, pp. 867-872.
- [3] S. B. Akers, D. Harel and B. Krishnamurthy, "The Star Graph: An Attractive Alternative to the n -Cube," *Proc. Int'l Conf. on Parallel Processing*, 1987, pp. 393-400.
- [4] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays · Trees · Hypercubes*, Morgan Kaufmann Publishers, San Mateo, California, 1992, pp. 466-469.
- [5] S. Latifi, M.M. Azevedo and N. Bagherzadeh, "The Star-Connected Cycles: a Fixed-Degree Interconnection Network for Parallel Processing," *Proc. Int'l. Conf. Parallel Processing*, 1993, Vol. 1, pp. 91-95.
- [6] S. Latifi, M. M. Azevedo and N. Bagherzadeh, "A Star-Based I/O-Bounded Network for Massively Parallel Systems," *IEE Proceedings - Computers and Digital Techniques*, Vol. 142, No. 1, January 1995, pp. 5-14.
- [7] W. Ledermann, *Introduction to the Theory of Finite Groups*, Oliver and Boyd, London, 1964, pp. 34-64.

- [8] S. Latifi, "Parallel Dimension Permutations on Star Graph," *IFIP Transactions A: Computer Science and Technology*, 1993, A23, pp. 191-201.
- [9] S. Ranka, J.-C. Wang and N. Yeh, "Embedding Meshes on the Star Graph," *Journal of Parallel and Distributed Computing* 19, 1993, pp. 131-135.
- [10] W. J. Dally, "Performance Analysis of k -ary n -cube Interconnection Networks," *IEEE Transactions on Computers*, Vol. 39, No. 6, June 1990, pp. 775-785.