Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Information Sciences 151 (2003) 51–70

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

On the fault-tolerant embeddings of complete binary trees in the mesh interconnection networks

Wei-Chen Fang [a], Chiun-Chieh Hsu [a,*], Chien-Ming Wang [b]

[a] *Department of Information Management, National Taiwan University of Science and Technology, 43, Section 4, Kee-Lung Road, Taipei 10772, Taiwan, ROC*
[b] *Institute of Information Science Academia Sinica, China*

Received 30 December 1999; received in revised form 8 May 2002; accepted 6 June 2002

Abstract

In this article, several schemes are proposed for embedding complete binary trees (CBT) into meshes. All of the proposed methods outperform those in the previous studies. First, a link congestion 1 embedding is achieved. Its expansion ratio is at the lowest level as we know now. Except for this superiority, it also provides another capability for fault tolerance to resist abnormal system faults, thus the embedding structure can be more guaranteed and the node utilization is raised further. Second, a link congestion 1 embedding with no-bending constraint is obtained. This scheme provides efficient CBT embedding for both the optical mesh and general mesh at the same time while keeping those good properties as the previous scheme. The last one is an optimal embedding which is applied to a 3D cubic mesh, where the node is almost fully utilized and its link congestion is 2.

1. Introduction

Four criteria can be used to evaluate embedding results. The first criterion, dilation, which is the maximum distance between any two nodes on the

* Corresponding author. Tel.: +886-2-737-6766; fax: +886-2-737-6777.
E-mail address: cchsu@cs.ntust.edu.tw (C.-C. Hsu).

0020-0255/02/$ - see front matter
doi:10.1016/S0020-0255(02)00389-4

newly derived structure measures the system communication time. The second is congestion, which is the key factor affecting the system delay and packet loss during data transmission. The third is expansion, which is the efficiency of node utilization for an embedding scheme. The last is the capability of fault tolerance, with which the derived structure will be more reliable and robust.

There are many parallel architectures proposed in the literature. The mesh is one of the most popular structure because it has a simple structure and an efficient layout. For a long time, larger dilation of meshes has been criticized for causing longer communication time. With the advancement in technique, wormhole routing [1–4] and the optical model [5–8] have offered solutions to the communication problem of mesh. Therefore, the dilation issue for embedding problems on meshes is not so critical because the wormhole routing is regarded as a dilation insensitive scheme and the optical model can provide high speed transmission. Both alternatives can significantly decrease the communication cost. When wormhole routing and optical technique are adopted as the communication scheme and media, link congestion for an embedding scheme will become more important than dilation. However, for the optical model, in order to take the advantage of optics, the no-bending constraint will be introduced and considered in addition.

For parallel processing, the complete binary tree (CBT) is one of the most important structure and many well-known algorithms are developed on it. However, if a fault occurs, the tree usually suffers serious reconfiguration or size degradation. Particularly, if this faulty node is an internal tree node and locates in the middle layer of the tree, it will make a large CBT shrink into a smaller one.

In the previous studies, Gibbons and Patterson [9] showed that T_h can be embedded with link congestion 2 into $M\left(\sqrt{2^{h-1}} \times \sqrt{2^{h-1}}\right)$ and $M\left(\sqrt{2^h} \times \sqrt{2^h}\right)$ meshes, where h is odd and even respectively. Lee and Choi [10] improved this result in that, with link congestion 1, T_h can be embedded into $M\left(\sqrt{\frac{9}{8}2^h} \times \sqrt{\frac{9}{8}2^h}\right)$ and $M\left(\frac{9}{8}\sqrt{2^h} \times \frac{9}{8}\sqrt{2^h}\right)$ where h is odd and even respectively. Nevertheless, all these embeddings ignore the possibility of system faults, and thus the derived results cannot be guaranteed.

In this paper, we develop schemes with smaller expansions, which implies that our schemes have higher node utilization. In addition to this advantage, we also emphasize the necessity of fault tolerance mechanism. As a result, the size of the derived CBT can be better guaranteed. Let T_h denote the CBT with height h. Three main results derived are as follows:

(1) For the general mesh, T_h is embedded with link congestion 1 into $M\left(\sqrt{\frac{9}{8}2^h} \times \sqrt{\frac{9}{8}2^h}\right)$ if h is odd and $h \geqslant 7$ or $M\left(\sqrt{2^h} \times \frac{9}{8}\sqrt{2^h}\right)$ if h is even and $h \geqslant 8$.

The maximum faulty node f that (1) can tolerate is

$$f = 2^{h-\rho} - 1, \qquad \rho = \begin{cases} 7 & \text{if } h \text{ is odd,} \\ 8 & \text{if } h \text{ is even,} \end{cases} \quad \text{where } h \geqslant 9.$$

(2) For both general and optical meshes, T_h is embedded with link congestion 1 into $M\left(\sqrt{\frac{9}{8}2^h} \times \sqrt{\frac{9}{8}2^h}\right)$ and $M\left(\frac{9}{8}\sqrt{2^h} \times \frac{9}{8}\sqrt{2^h}\right)$ under no-bending constraint, when h is odd and even respectively.

The maximum faulty node f that (2) can tolerate is

$$f = 2^{h-\rho} - 1, \qquad \rho = \begin{cases} 7 & \text{if } h \text{ is odd,} \\ 6 & \text{if } h \text{ is even,} \end{cases} \quad \text{where } h \geqslant 8.$$

(3) For the 3D cubic mesh, T_h is embedded with link congestion 2 into $M(2^p \times 2^p \times 2^p)$, $M(2^{p+1} \times 2^p \times 2^p)$, and $M(2^{p+1} \times 2^{p+1} \times 2^p)$ when $h = 3p$, $3p + 1$, and $3p + 2$ respectively.

The rest of this paper is organized as follows: In Section 2, some basic terms related to our embedding scheme are defined. In Section 3, we outline the principle and concept on which the CBT is constructed in our scheme. In Section 4, an optical model is introduced and related constraint and solution are also presented. In Section 5, an optimal embedding for 3D mesh is given. Finally, we make a complete comparison and conclusion.

2. Preliminaries

In this section, we give some basic definitions used through the paper. Different communication models are also stated.

To start, we first establish an embedding with T_h onto a basic mesh and such an embedding is called *basic embedding*. We then use the basic embedding as the base to construct a larger CBT for a larger mesh. As an example, we show a basic embedding for a T_7 on a 12×12 mesh in Fig. 1(a). In this figure, all those gray nodes denote the tree nodes while the white nodes denote the unused node. The black node in the middle of the mesh is the root node of the tree. In addition, the arrow lines illustrate the tree edges from tree nodes toward their children. At the first glance, we can easily catch that such an embedding follows a specific pattern as Fig. 1(b) shown. As we can see in the figure, the entire square represents an $m \times n$ mesh in which circles labeled R and u denote the root node of the embedded T_h and the unused node respectively. The dash lines denote the free links which are not used. Any embedding with the arrangement as Fig. 1(b) is called a *common pattern* embedding. As our algorithm proceeds, we keep this structure all the time to maintain the property for recursive construction.

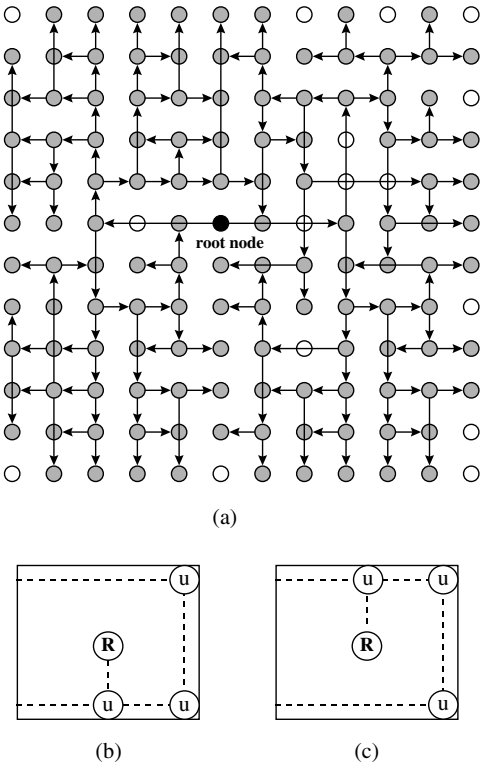

Fig. 1. (a) Basic embedding of T_7 (α_7^c) into a 12×12 mesh. (b) Common embedding pattern. (c) Fig. 1(b) flipped across x-axis.

Definition 1 (*Flipping function*). A flipping function f_d is a mechanism which is used to map the target embedding across different directions, where the subscript d denotes the flipping dimension.

Example 1. Fig. 1(c) is the common pattern embedding with flipping function f_x, which means that the derived pattern is a flipped version of common pattern in Fig. 1(b) across x-axis.

As we described, two communication models, the wormhole model and optical model, are used for the mesh routing.

In the wormhole model, each packet consists of a sequence of elementary units called *flits*. During each step, each flit of a packet can advance across a wire, but only one flit can advance across a single wire in a single step. Although the packet can be distributed across many nodes and wires at any time, it is always tied together as a single entity. In other words, contiguous flits in a

packet are always contained in the same or adjacent nodes of the network. Thus an effective data movement is achieved in a general mesh by using this model.

Nevertheless, the optical model has recently received a lot of attention. It is because this model offers the advantages of high bandwidth and multiple channels, which can provide high speed and congestion free transmission for the system communication mechanism.

Consider a mesh-based computing architecture with optical medium interconnecting the nodes in each row and column. Due to the characteristic of optical transmission, the optical medium must not be bending. Thus, for the reason of communication efficiency, the no-bending property should be kept in the embedding process under the optical model. To satisfy this condition, all the tree nodes of the CBT must be embedded into the mesh such that all neighbors are in either the same row or column. In this case, physical distance is no longer a critical issue. However, if one tree node has to reach its neighbor across different dimension, this will result in a serious transmission delay. Hence, under such a criterion, a tree edge in the mesh which is across different dimension is not acceptable.

Since the dilation problem can be ignored by applying the two schemes above, we will then focus on how to improve the node utilization and fault tolerance for both the general mesh and optical mesh.

3. Congestion 1 CBT embedding

In this section, we explain how we deploy a recursive algorithm to construct a complete binary tree.

In our scheme, four smaller meshes following common pattern embedding constitute a larger mesh with a larger CBT on it and keep common pattern at the end of the construction phase. Without loss of generality, let $\alpha_{\hat{h}}$ be the embedding with a $T_{\hat{h}}$ onto a mesh. If a $T_{\hat{h}}$ is embedded following the common pattern, we then denote the embedding as $\alpha^{c}_{\hat{h}}$. Generally speaking, we use four $\alpha^{c}_{\hat{h}}$'s to construct $\alpha^{c}_{\hat{h}+2}$. The physical allocation for the four $\alpha^{c}_{\hat{h}}$'s and the construction are illustrated in Fig. 2. In the figure, the circle labeled R means the root of the new tree $T_{\hat{h}+2}$ and the circles labeled S and T are its two children which are used to connect the roots of the four subtrees $r1$, $r2$, $r3$ and $r4$ for each $\alpha^{c}_{\hat{h}}$'s. Note that common embedding pattern is still remained for $\alpha_{\hat{h}+2}$.

In addition to the general construction process mentioned above, the basic embedding we choose to initialize the construction depends on the height of destinated CBT, h. It can be divided into two cases. If h is odd, we use α^{c}_{7} as the basic embedding, where α^{c}_{7} is a 12×12 mesh into which T_7 following common pattern is embedded. If h is even, we then use α^{c}_{8} as the basic embedding and α^{c}_{8} is an embedding onto a 16×18 mesh. Since we cannot obtain α^{c}_{8} straightforward,

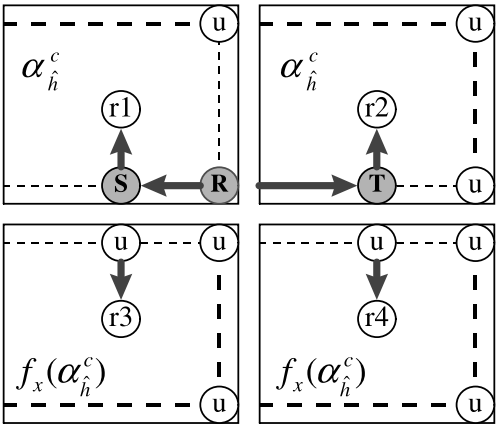

Fig. 2. Construction of $\alpha^c_{\hat{h}+2}$ by using four $\alpha^c_{\hat{h}}$'s.

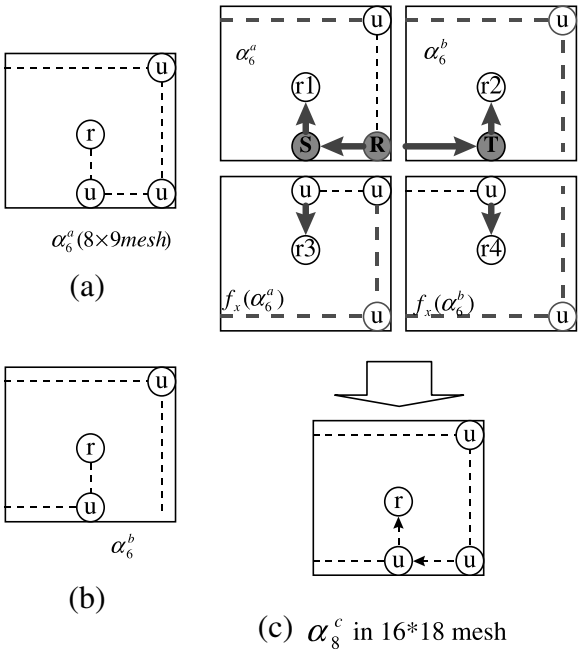

Fig. 3. (a) α^a. (b) α^b. (c) Construction of α^c_8 by using α^a_6's and α^b_6's.

we then utilize another two types of embedding pattern, α^a_6 and α^b_6, to create it.

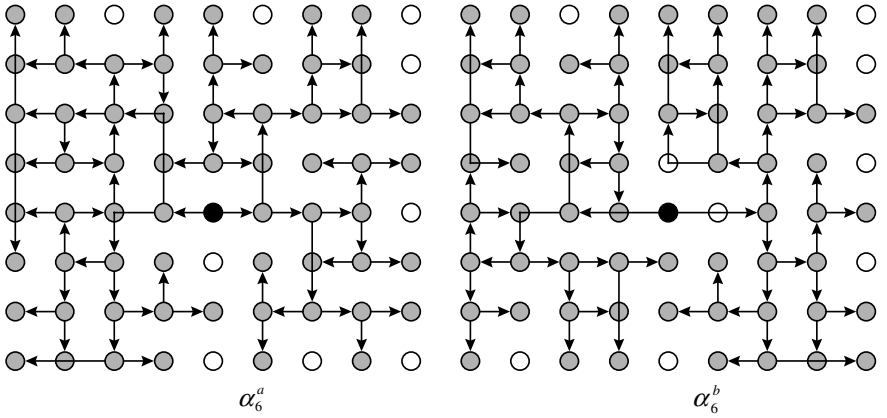

Fig. 4. Real embedding of α_6^a and α_6^b on a 8×9 mesh.

In the following, we use Fig. 3 to explain the entire construction process of α_8^c. The patterns, α_6^a and α_6^b, are shown in Fig. 3(a) and (b), respectively. There are four parts:, α_6^a, $f_x(\alpha_6^a)$, α_6^b, and $f_x(\alpha_6^b)$ in the construction space and their arrangements are shown in Fig. 3(c). By applying this construction, α_8^c can be obtained.

Overall, common pattern is the basic structure for the recursive construction of a larger CBT. As we mentioned, CBTs of both odd and even heights will be built in the same way when α_7^c and α_8^c are prepared. Every time the construction proceeds, the height of the CBT will be increased by 2. We then reexamine the larger CBT embedding. It still maintains the common pattern embedding just as the smaller CBT embedding does. The real embedding for α_7^c, α_6^a, and α_6^b are shown in Figs. 1(a) and 4. Since the meshes for α_6^a and α_6^b embedding are of size 8×9, therefore, the basic embeddings α_7^c and α_8^c are of mesh size 12×12 and 16×18, respectively.

The algorithm of CBT construction is as follows.

Embedding Algorithm Φ_1:
T_h: a destinated CBT with height h
DO CASE
CASE h is odd:
  Step 1. Embed a α_7^c into a 12×12 mesh
  Step 2. (Initialization) $\hat{h} = 7$
CASE h is even:
  Step 1. Embed α_6^a and α_6^b into two 8×9 mesh, separately
  Step 2. Use α_6^a, $f_x(\alpha_6^a)$, α_6^b and $f_x(\alpha_6^b)$ to construct the basic α_8^c as shown in Fig. 3(c). (Initialization) $\hat{h} = 8$
END CASE

Step 3. Do While $\hat{h} < h$
Call RECURSIVE_CBT($\alpha^c_{\hat{h}}$)
END Φ_1
Procedure RECURSIVE_CBT($\alpha^c_{\hat{h}}$)
Use $\alpha^c_{\hat{h}}$ as basis. Apply recursive construction (see Fig. 2)
$\hat{h} = \hat{h} + 2$
Return $\alpha^c_{\hat{h}}$

According to the algorithm stated, we then have the following theorems.

Theorem 1 (for general mesh). *The embedding algorithm Φ_1 can embed a CBT with height h and link congestion* 1 *into a mesh M whose size is*
(1) $M\left(\sqrt{\frac{9}{8}2^h} \times \sqrt{\frac{9}{8}2^h}\right)$ *if h is odd and $h \geqslant 7$;*
(2) $M\left(\sqrt{2^h} \times \frac{9}{8}\sqrt{2^h}\right)$ *if h is even and $h \geqslant 8$.*

Proof. It is clear from the construction that there will be no mesh link used as the tree link for twice. Eventually, the achieved link congestion is 1. As to size of the destinated mesh, the proof is by induction. Let h be the height of the embedded CBT. To complete the proof, we must consider the two cases when h is odd or even.

(a) In the case that h is odd:
Induction base. Recall the construction described above, the base case α^c_7 is an embedding of a T_7 into a 12×12 mesh $M\left(\sqrt{\frac{9}{8}2^7} \times \sqrt{\frac{9}{8}2^7}\right)$.
Induction hypothesis. Assume that α^c_{h-2} embeds T_{h-2} into $M\left(\sqrt{\frac{9}{8}2^{h-2}} \times \sqrt{\frac{9}{8}2^{h-2}}\right)$
Induction step. To complete T_h, Fig. 2 shows the construction of α^c_h from α^c_{h-2}'s into $M\left(\sqrt{\frac{9}{8}2^h} \times \sqrt{\frac{9}{8}2^h}\right)$.

(b) In the case that h is even:
α^c_8 is another base case which embeds a T_8 into a 16×18 mesh $M\left(\sqrt{2^8} \times \frac{9}{8}\sqrt{2^8}\right)$. Again, by inducting in the same way, an embedding of a T_h into an $M\left(\sqrt{2^h} \times \frac{9}{8}\sqrt{2^h}\right)$ is obtained. □

As mentioned before, fault tolerance is a critical factor for evaluating embedding result. Following the design of our embedding pattern, we scatter those free nodes on their appropriate sites. Thus, they can contribute to system reconfiguration when some nodes become faulty. Therefore, the following theorem holds.

Theorem 2. *Using fault-tolerant embedding Φ_1, the maximum number of faulty nodes that can be recovered for an embedding α_h is*

$$f = 2^{h-\rho} - 1, \qquad \rho = \begin{cases} 7 & \text{if } h \text{ is odd}, \\ 8 & \text{if } h \text{ is even}, \end{cases} \quad \text{where } h \geqslant 9.$$

Proof. As we know from the algorithm, all the ascendants of the basic CBT at the top $h - \rho$ levels have a spare node for the replacement when any of these ascendants become faulty. As mentioned before, the height of basic tree ρ is 7 when our destination tree is of odd height. In another case when h is even, the height is 8. As a result, the maximum faulty nodes that we can tolerate is $2^{h-\rho} - 1$. □

4. Congestion one CBT embedding without bending

By proceeding in a recursive manner as Section 3, a congestion 1 CBT embedding algorithm Φ_2 without bending is accomplished in this section. The result obtained in this section can also be applied to the general mesh. Thus, we can improve the fault tolerance capability compared with that proposed by Szymanki [4]. Besides, it can apply to different models.

The basic cases for Φ_2 are α_6^c and α_7^c when the height of the destinated CBT is odd and even, respectively. Their instances are shown in Figs. 5 and 1(a). Note that all the tree edges in α_6^c and α_7^c are bendless.

Embedding Algorithm Φ_2:
T_h: a destinated CBT with height h
DO CASE

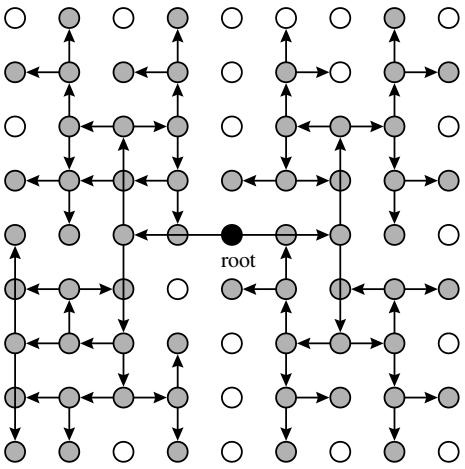

Fig. 5. Bendless α_6^c on 9×9 mesh.

CASE h is odd:
  Step 1. Embed a α_7^c into a 12×12 mesh
  Step 2. (Initialization) $\hat{h} = 7$
CASE h is even:
  Step 1. Embed a α_6^c into a 9×9 mesh
  Step 2. (Initialization) $\hat{h} = 6$
END CASE
Step 3. Do While $\hat{h} < h$
  Call RECURSIVE_CBT($\alpha_{\hat{h}}^c$)
END Φ_2
Procedure RECURSIVE_CBT($\alpha_{\hat{h}}^c$)
  Use $\alpha_{\hat{h}}^c$ as basis. Apply recursive construction as shown in Fig. 2
  $\hat{h} = \hat{h} + 2$
  Return $\alpha_{\hat{h}}^c$

Theorem 3 (for both optical mesh and general mesh). *Using the basic embedding α_6^c and α_7^c in Fig. 5, Φ_2 can embed a T_h into a mesh M with link congestion* 1 *and hold no bending constraint where M is of size*
(1) $M\left(\frac{9}{8}\sqrt{2^h} \times \frac{9}{8}\sqrt{2^h}\right)$ *if h is even and $h \geqslant 6$;*
(2) $M\left(\sqrt{\frac{9}{8}2^h} \times \sqrt{\frac{9}{8}2^h}\right)$ *if h is odd.*

Proof. The proof is similar to that in Theorem 1. □

Theorem 4. *Using fault-tolerant embedding Φ_1, the maximum number of faulty nodes that can be recovered for an embedding α_h is*

$$f = 2^{h-\rho} - 1, \qquad \rho = \begin{cases} 7 & \text{if } h \text{ is odd}, \\ 6 & \text{if } h \text{ is even}, \end{cases} \quad \text{where } h \geqslant 8.$$

Proof. The proof is similar to that in Theorem 2. □

5. Optimal CBT embedding onto 3D mesh

The embedding of a CBT into a 2D mesh has been described in Sections 3 and 4. In the following, we extend the same idea to 3D meshes.

Fig. 6(a) shows the basic embedding $\hat{\beta}$ for 3D mesh, which maps a basic tree (T_3) into a $2 \times 2 \times 2$ 3D mesh as shown in Fig. 6(b). We can see that there exists a link congestion 1 embedding for T_3 in $\hat{\beta}$. In addition, there is an empty node e' on the corner of the $2 \times 2 \times 2$ mesh and its location is diagonal to the root node A. Moreover, three links which are denoted by dotted lines are unused. At a closer look, these free links constitute a *free*

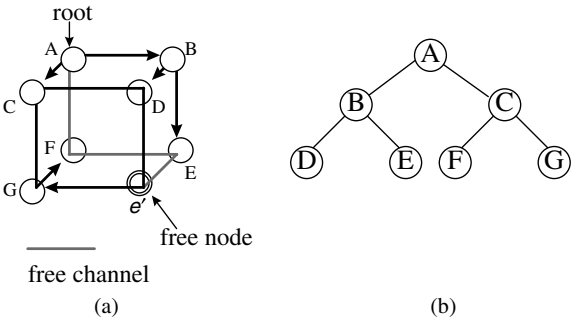

Fig. 6. The basic embedding $\hat{\beta}$ and the basic tree.

channel connecting root A and empty node e'. In order to apply our recursive construction to larger CBTs, we must use the basic embedding or its derived embedding as the *elementary embedding* to proceed further construction in our algorithm. An elementary embedding is a composition version of the basic embeddings ($\hat{\beta}$'s) or the other smaller elementary embeddings. For the design philosophy, an elementary embedding must follow a specific embedding pattern. Therefore, without loss of generality, we define β_h^p as the elementary embedding which has a T_h embedded on it and follows the required embedding pattern, *type-p*.

As describe previously, embedding pattern is the most critical design for recursive construction scheme to proceed further. Two required embedding patterns, *Type-A* and *Type-B*, are shown in Fig. 7(a) and (b). As we can see in these two figures, there is a cube represents a cubic 3D mesh while there are another three nodes labeled e', r, and c'. The double circle labeled e' is the node which is not used in the embedding, while the circle labeled c' represents the corner node of e' which is located diagonally to e'. Apart from these two nodes c' and e', there is another node labeled $\boldsymbol{r}$ which denotes the root node of the complete binary tree T_h in the 3D mesh. Furthermore, there exist two channels ψ_1 and ψ_2 which connect c' and r as well as r and e' respectively. ψ_1 is illustrated by double line which represents the channel having more than two link congestion 1 disjoint paths and ψ_2 with irregular single bold line is the channel with a link congestion 1 path. In addition to the channels and nodes, those border links illustrated by single bold lines mean congestion 1 links while the single thin lines denote do not care links. All these links are also part of the embedding patterns.

As mentioned, we use elementary embeddings applying different rotation to constructing larger 3D meshes. Therefore, for the ease of description, we next define the rotation function $\Re$.

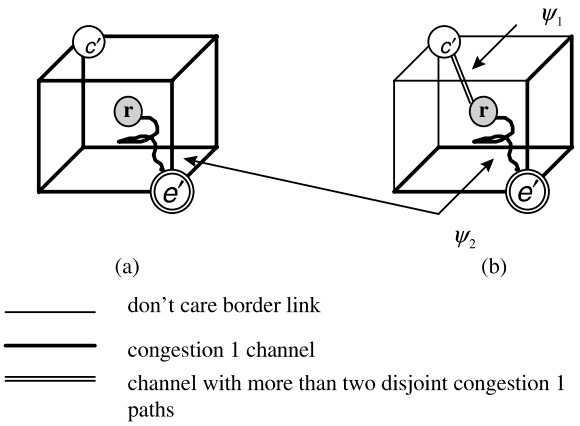

Fig. 7. The embedding patterns for an optimal CBT embedding onto 3D mesh: (a) *type-A* embedding pattern; (b) *type-B* embedding pattern.

Like the hypercube, each 3D mesh has $N = 8$ nodes on the corner. Let the position p_j of each corner node in the 3D mesh corresponds to a $\log N$-bit binary string, and two linked nodes differ in exactly one bit.

Definition 2. Let $\Re_{\langle\omega,\lambda\rangle}$ be a rotation function where $\omega = (n_0, n_1, \ldots, n_i, \ldots, n_7, n_0)$ is the Eulerian tour sequence corresponds to the position sequence $(p_0, p_1, \ldots, p_j, \ldots, p_7, p_0)$ and let λ be the right cyclic shift vector. Applying the rotation we defined, the derived sequence will be node $n_{(8-\lambda+j) \bmod 8}$ at p_j.

Example. Let $\omega = (A \to B \to C \to D \to E \to F \to G \to H \to A)$ be a Eulerian tour sequence of the original cube in Fig. 8(a) initially and $(000 \to 001 \to 011 \to 010 \to 110 \to 111 \to 101 \to 100 \to 000)$ be the corresponding position sequence. Fig. 8(b) shows the derived cube with the rotation

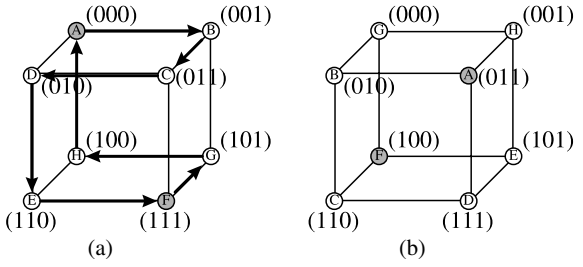

Fig. 8. (a) A Eulerian tour sequence on a 3D mesh. (b) A 3D mesh with a rotation $\Re_{\langle\omega,2\rangle}$.

$R_{\langle\omega,2\rangle}$, and as we see, the derived cube is now with nodes G at (000), H at (001), A at (011), B at (010), C at (110), D at (111), E at (101), and F at (000).

Our optimal 3D mesh embedding follows a bottom-up strategy. That is, we use two, four, and eight $\beta_{\hat{h}}$'s to build trees of size $\hat{h}+1$, $\hat{h}+2$, and $\hat{h}+3$ respectively where $\hat{h}$ is multiple of 3. As we mentioned previously, $\beta_{\hat{h}}$ is the elementary embedding and composed of eight $\beta_{\hat{h}-3}$'s. Due to the different allocation of these elementary embeddings while construction proceeds, we then use $B_{x,y,z}$ to identify their distinct location on the construction space. Let us partition each dimension of the three-dimensional space into two blocks, then eight quadrants are obtained. As we define, the subscriptions x, y, and z indicate the number of block of distinct dimension and their values are 0 or 1.

As we construct, the $\beta_{\hat{h}}$'s can be regarded as conforming to either the embedding *pattern-A* or *B*. We will prove this property later on. In general, for constructing $\beta_{\hat{h}+3}$, eight $\beta_{\hat{h}}$'s following specific embedding pattern and their locations and relative rotations are as follows:

$$
\begin{aligned}
B_{0,0,0} &= \beta_h^{\mathrm{A}},\\
B_{1,0,0} &= \mathfrak{R}_{\langle\omega,1\rangle}(\beta_h^{\mathrm{A}}),\\
B_{0,1,0} &= \mathfrak{R}_{\langle\omega,3\rangle}(\beta_h^{\mathrm{A}}),\\
B_{1,1,0} &= \mathfrak{R}_{\langle\omega,2\rangle}(\beta_h^{\mathrm{A}}),\\
B_{0,0,1} &= \mathfrak{R}_{\langle\omega,7\rangle}(\beta_h^{\mathrm{A}}),\\
B_{1,0,1} &= \mathfrak{R}_{\langle\omega,6\rangle}(\beta_h^{\mathrm{A}}),\\
B_{0,1,1} &= \mathfrak{R}_{\langle\omega,4\rangle}(\beta_h^{\mathrm{A}}),\\
B_{1,1,1} &= \beta_h^{\mathrm{B}}.
\end{aligned}
$$

Let the free node in each block, except the one in $B_{1,1,1}$, act as the *internal nodes* of the new 3D meshes and constitute a new $2\times 2\times 2$ 3D *internal mesh* β'. As we understand, there will be a lot of new links generated while joining individual elementary embeddings to form a larger one. Furthermore, let E be the new link set and E' be the subset of E connecting the internal nodes of the individual elementary embeddings. Thus we can utilize β' to construct a new T_3', and then use its leaves to link the root of T_3 in each block, where each leave of T_3' is responsible for connecting two root nodes of the eight T_3's. Thus the height of the new CBT is increased by 3. The allocation of T_3' and its leaves connection are shown in Fig. 9.

In order to help the understanding of our algorithm, we next give an informal description. Due to the difference of cases, we bring up two illustrations, β_6 and $\beta_{\hat{h}}$, where $\hat{h}\geqslant 9$, for explanation.

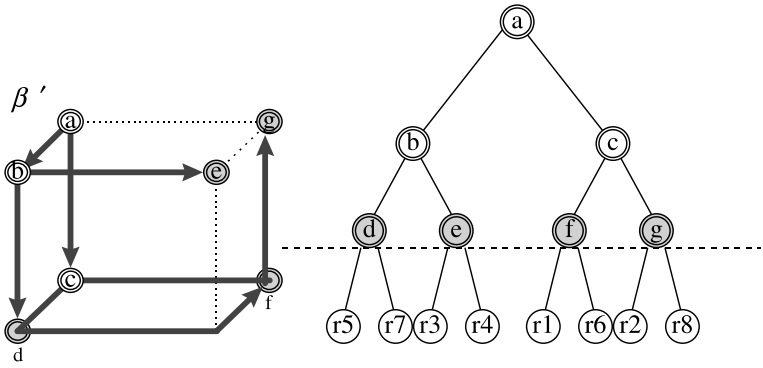

Fig. 9. T_3' construction by using β'.

In the construction of α_6, see Fig. 10(a), there is a free channel left for the connection between the root and free node for each $\hat{\beta}$. By using T_3', we connect the leaves with tree roots in the blocks, thus an optimal embedding of T_6 in a $4 \times 4 \times 4$ mesh is derived. The links in each block are still congestion-free. The real construction of β_6 is shown in Fig. 10(a). Note that the embedding in this example also generates the elementary embedding β_6, and as we observe, it conforms to either embedding *pattern-A* or *B*. We will prove this later on. As a consequence, it can be used as the construction base for T_h, where $h > 6$.

Apply the same construction process by employing eight β_6's, we obtain β_9, as shown in Fig. 10(b). In addition to the construction of T_h where $h \bmod 3 = 0$, the other two cases with $h+1$ and $h+2$ are shown in Fig. 10(c) and (d).

The exact optimal embedding algorithm Φ_3 is as follows.

Embedding Algorithm Φ_3:
T_h: a CBT with height h
$\beta_{\hat{h}}$: an embedding with $T_{\hat{h}}$ onto $M(2^{\hat{h}/3} \times 2^{\hat{h}/3} \times 2^{\hat{h}/3})$ where $\hat{h} \bmod 3 = 0$
Step 1. Obtain β_3 (or $\hat{\beta}$) as Fig. 6
Step 2. (Initialization) $\Gamma = \beta_3$
Step 3. For $m = 1$ to $\lfloor h/3 \rfloor - 1$
  Call RECURSIVE_CBT(Γ)
Step 4. Select case $h \bmod 3$
  Case 1. Apply recursive construction as Fig. 10(c)
  Case 2. Apply recursive construction as Fig. 10(d)
  End Case
Procedure RECURSIVE_CBT(β_i)
Select case i
Case 3.
  Apply recursive construction, see Fig. 10(a)
Otherwise:

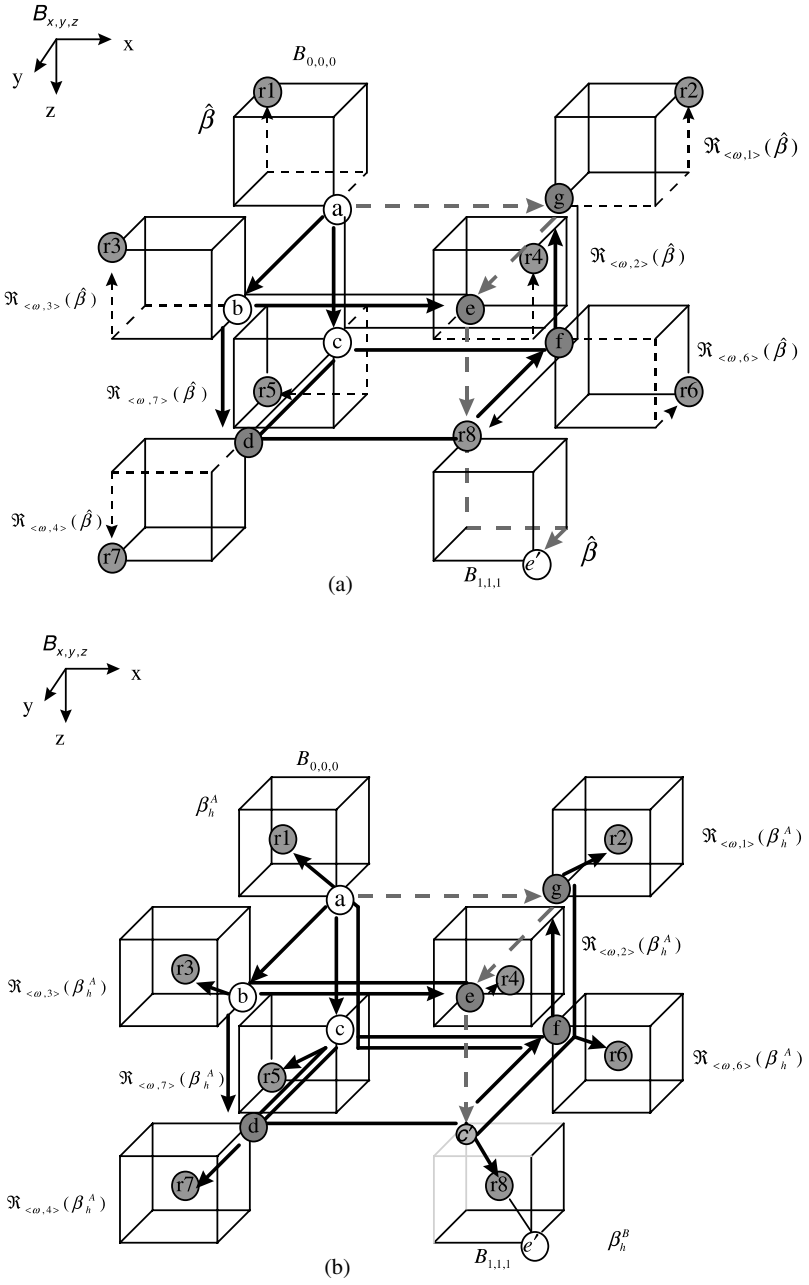

Fig. 10. (a) Construction of β_6. (b) Construction of β_h where $h \bmod 3 = 0$ and $h \geqslant 9$. (c) Construction of T_h, where $h \bmod 3 = 1$. (d) Construction of T_h, where $h \bmod 3 = 2$.

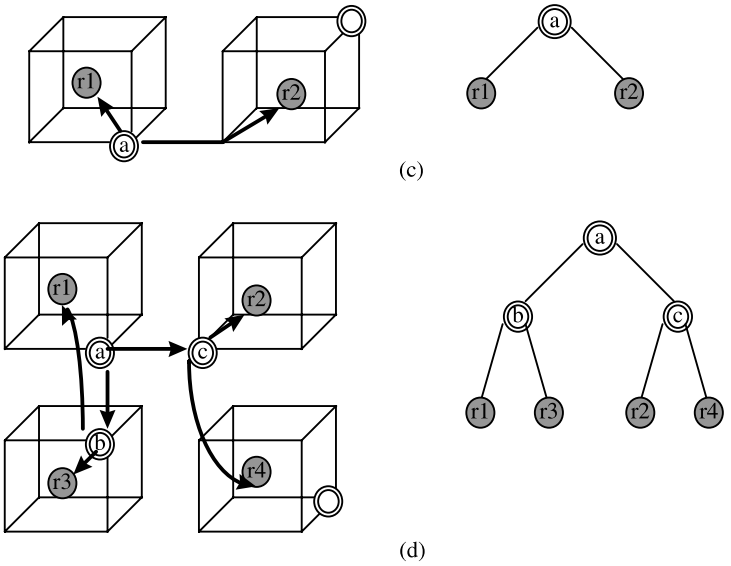

Fig. 10 (*continued*)

Apply recursive construction, see Fig. 10(b)
$\Gamma = \beta_{i+3}$
Return Γ

According to the above description, the following lemmas and theorems can be obtained.

Lemma 1. *By the construction of* Φ_3, *for any cubic elementary embedding* $\beta_{\hat{h}}$ *where* $\hat{h} \bmod 3 = 0$, *either the embedding pattern-A or B will be satisfied.*

Proof. Three cases complete the proof.

CASE β_3 (or $\hat{\beta}$). See Fig. 6(a).
Let the node r map into node c', as shown in Fig. 7. Thus, the channel ψ_1 of $\hat{\beta}$ will be null or inside r, while the congestion 1 channel ψ_2 between r and e' will be replaced by the free channel. Combining with all the border links, therefore, the two embedding patterns *type-A* or *B* can be satisfied by $\hat{\beta}$.

CASE β_6. See Fig. 10(a).

(1.1) As we can see, channel ψ_2 is composed of path φ_1 and φ_2. $\varphi_1(a - g - e - r8)$ is a free channel where the links (a, g), (g, e), and $(e, r8)$ belong to the new edge set E' of E, and φ_2 between $r8$ and the empty node

of $\hat{\beta}$ in $B_{1,1,1}$ is another free channel, thus $\varphi_1 + \varphi_2$ constitute the channel ψ_2 for the newly derived embedding β_6.

(1.2) Since all the links of each $\hat{\beta}$ and those edges $E - E'$ are with congestion 1 or congestion-free, the border links of the newly derived $M(4 \times 4 \times 4)$ are also link congestion 1.

From (1.1) and (1.2), we know that β_6 in $M(4 \times 4 \times 4)$ as illustrated in Fig. 10(a) conforms to *pattern-A* embedding.

(1.3) As the figure shown, ψ_1 of β_6 comes directly from the border links of $\hat{\beta}$ in block $B_{0,0,0}$. And as we known, $\hat{\beta}$ in $B_{0,0,0}$ is a link congestion 1 cube, thus there will always exists three link disjoint paths.

(1.4) Directly from (1.1), there is a link congestion-free channel between the new root node a and the only free node e', thus it directly provide us with the channel ψ_2.

(1.5) Since all the links of $\hat{\beta}$ in each block are with congestion 1 and all the new edges except E' are congestion-free, the border links as embedding pattern *type-B* can be obtained.

From (1.3) and (1.5), we know that β_6 in $M(4 \times 4 \times 4)$ as shown in Fig. 10(a) conforms to *pattern-B*. As a consequence, we can establish that either *pattern-A* or *B* can be satisfied by the embedding β_6.

CASE $\beta_{\hat{h}}$, where $\hat{h} \geqslant 9$ and $\hat{h} \bmod 3 = 0$. See Fig. 10(b),

(2.1) As shown in the figure, channel ψ_2 is composed of two paths, $\varphi_1(a - g - e - c')$ and $\varphi_2(c' - r8 - e')$. However, φ_1 is a free channel of internal cube β'. Besides, as we know that the cube $\beta_{\hat{h}-3}$ in $B_{1,1,1}$ is a *type-B* elementary embedding, which means that there exist three link congestion 1 paths. As a result, there are more than one link congestion 1 paths left after the tree edge between node g and $r8$ has been constructed. Combing φ_1 with φ_2, thus, a link congestion 1 channel ψ_2 always exists.

(2.2) Similar to (1.2). Because all the border links of $M(2^{\hat{h}/3-1} \times 2^{\hat{h}/3-1} \times 2^{\hat{h}/3-1})$ in each block except the ones in $B_{1,1,1}$ are with link congestion 1, we then combine these links with those link congestion 1 border links of $\beta^B_{\hat{h}-3}$ in $B_{1,1,1}$ and the new edges $E - E'$. Thus all the border links of newly derived $M(2^{\hat{h}/3} \times 2^{\ddot{h}/3} \times 2^{\dot{h}/3})$ are achieved and remain to be link congestion 1.

From (2.1) and (2.2), we know that $\beta_{\hat{h}}$ in $M(2^{\dot{h}/3} \times 2^{\ddot{h}/3} \times 2^{\dot{h}/3})$ as illustrated in Fig. 10(b) can be a *pattern-A* embedding.

(2.3) Like (1.3), channel ψ_1 of $\beta_{\hat{h}}$ comes directly from border links of $\beta_{\hat{\ddot{h}}-3}$ in block $B_{0,0,0}$.

(2.4) Directly from (2.1), there will exist a link congestion 1 channel between the new root node a and the only free node e', and it constitutes path ψ_2.

(2.5) Similar to (1.5). Since all the border links of $M(2^{\hat{h}/3-1} \times 2^{\hat{h}/3-1} \times 2^{\hat{h}/3-1})$ in each block except the one in $B_{1,1,1}$ are with link congestion 1, we combine these link congestion 1 border links with those ones of $\beta^B_{\hat{h}-3}$ in $B_{1,1,1}$. Thus the link congestion 1 border links which belong to $\beta^B_{\hat{h}}$ are derived.

From (2.3), (2.4), and (2.5), we know that $\beta_{\hat{h}}$ in $M(2^{\hat{h}/3} \times 2^{\hat{h}/3} \times 2^{\hat{h}/3})$ as illustrated in Fig. 10(b) can be a *pattern-B* embedding.

According to the above proof, we can obtain that either embedding *pattern-A* or *B* can be satisfied for any $\beta_{\hat{h}}$ where $\hat{h} \bmod 3 = 0$. □

Lemma 2. *For any tree edge of the destinated CBT, the link congestion is at most* 2.

Proof. It can be directly derived from Lemma 1 and Φ_3. The new edges for connecting roots of subtrees of each block in each construction phase are with congestion 2 and there is always a link congestion 1 path between the free node and the root of subtree for each cubic submesh. Thus the embedding Φ_3 is of link congestion 2.

Theorem 5. *For any N-node 3D cubic mesh, the embedding algorithm* Φ_3 *will embed a CBT of optimal size* $N-1$ *into it with load* 1 *and link congestion* 2.

Proof. Directly comes from the construction scheme and Lemma 2. □

Theorem 6. *The optimal embedding algorithm* Φ_3 *can embed a CBT with height h onto a 3D mesh M, where* $p = \lfloor h/3 \rfloor$ *and* $q = h \bmod 3$. *Besides, M is of size*
(1) $M(2^p \times 2^p \times 2^p)$ *if* $q = 0$;
(2) $M(2^{p+1} \times 2^p \times 2^p)$ *if* $q = 1$;
(3) $M(2^{p+1} \times 2^{p+1} \times 2^p)$ *if* $q = 2$.

Proof. Our scheme uses two, four, and eight basic embedding $\beta_{\hat{h}}$'s to construct a larger CBT with height $\hat{h}+1$, $\hat{h}+2$, and $\hat{h}+3$, respectively. The construction cases are illustrated in Fig. 10(b)–(d). It is quite straightforward that two, four, and eight $M(2^p \times 2^p \times 2^p)$'s can be merged into a $M(2^{p+1} \times 2^p \times 2^p)$, $M(2^{p+1} \times 2^{p+1} \times 2^p)$, and $M(2^{p+1} \times 2^{p+1} \times 2^{p+1})$ respectively. The relationship of p and h is $p = \lfloor h/3 \rfloor$, and as we know, q is the reminder of h modulus by 3. □

6. Comparison and conclusion

In this section, the performance of the proposed algorithms are summarized and discussed. Table 1 shows the comparison between Φ_1, Φ_2 and the scheme of Lee and Choi [4]. As we can see, the scheme in [4] has two major drawbacks: lower node utilization and lack of fault tolerance capability. As a result, the availability is limited.

Table 1
Performance comparison for congestion 1 CBT Embedding

	Lee and Choi	Our scheme Φ_1	Our scheme Φ_2
Expansion	$81/64 \approx 1.2656$	$9/8 \approx 1.125$	$81/64 \approx 1.2656$
Fault tolerance	No	Yes Maximum recovery: f $f = 2^{h-\rho} - 1$, $\rho = \begin{cases} 7 & \text{if } h \text{ is odd} \\ 8 & \text{if } h \text{ is even} \end{cases}$ where $h \geqslant 9$.	Yes Maximum recovery: f $f = 2^{h-\rho} - 1$, $\rho = \begin{cases} 7 & \text{if } h \text{ is odd} \\ 6 & \text{if } h \text{ is even} \end{cases}$ where $h \geqslant 8$.
Congestion	1	1	1
Bend	Yes	Yes	No

Table 2
Comparison of optimal embedding

	Optimal embedding
Existing result	2D square mesh with link congestion 2
Our result	3D cubic mesh with link congestion 2

Consider the performance of the proposed Φ_1. A CBT has the same height as that in [4] but with lower expansion. Besides, most of the remaining free nodes are used for faulty node recovery. Thus we can avoid the situation that CBT degrade quickly when some faults occur in the internal node of the CBT. Therefore, both the node utilization and application reliability can be improved at the same time. Since the scheme of Lee and Choi has higher node expansion and has no fault tolerance capability, we achieve a great deal of enhancement. Our maximum node recovery capability is as Theorem 2 holds.

As shown in Table 2, no-bending constraint is included in Φ_2. In addition to this excellent property, the embedding scheme also provides the same fault tolerance capability as Φ_1. Besides, the expansion rate remains the same as that in Lee and Choi. However, it is noteworthy that this scheme is for both the general mesh and optical mesh.

For the optimal embedding, the existing scheme is for 2D mesh only. Our algorithm Φ_3 extends this result into 3D mesh.

Throughout this paper, we have focused on various embedding requirements, such as high node utilization, fault tolerance, and transmission efficiency. Previous studies have not specifically dealt with the problem of fault tolerance. Besides, utilization can still be more efficient. Consequently, the embedding results we achieve in this paper make a tremendous improvement compared with those of the previous studies. Apart from providing additional better features, the performance level has been the best so far.

References

[1] D. Kim, S.-H. Kim, O($\log n$) numerical algorithms on a mesh with wormhole routing, Inform. Process. Lett. 50 (3) (1994) 129–136.
[2] M. Barnett, D.G. Payne, R.A. van de Geijn, J. Watts, Broadcasting on meshes with wormhole routing, J. Parallel Distrib. Comput. 35 (2) (1996) 111–122.
[3] L. Schweibert, D.N. Jayasimha, A necessary and sufficient condition for deadlock-free wormhole routing, J. Parallel Distrib. Comput. 32 (1) (1996) 103–117.
[4] T.H. Szymanki, Hypermeshes: optical interconnection networks for parallel computing, J. Parallel Distrib. Comput. 26 (1) (1995) 1–23.
[5] S.T. Obenaus, T.H. Szymanski, Embeddings of star graphs into optical meshes without bends, J. Parallel Distrib. Comput. 44 (1997) 97–106.
[6] Z. Guo, R. Melhem, D. Chiarulli, S. Levitan, Pipelined communications in optically interconnected arrays, J. Parallel Distrib. Comput. 12 (3) (1991) 269–282.
[7] S.T. Obenaus, T.H. Szymanski, Embeddings of star graphs into optical meshes without bends, J. Parallel Distrib. Comput. 44 (1997) 97–106.
[8] T.H. Szymanski, H.S. Hinton, Reconfiguration intelligent optical backplane for parallel computing and communications, Appl. Opt. 35 (8) (1996) 1253–1268.
[9] A. Gibbons, M. Patterson, Dense edge-disjoint embedding of binary trees in the mesh, in: Proc. Fourth Ann. ACM Symp. Parallel Algorithms and Architecture, 1992, pp. 257–263.
[10] S.-K. Lee, H.-A. Choi, Embedding of complete binary trees into meshes with row-column routing, IEEE Trans. Parallel Distrib. System 7 (5) (1996) 493–497.