Embedding complete binary trees in product graphs

Kemal Efe[a], Adrienne L. Broadwater[b] and Antonio Fernandez[c]

[a] *Department of Computer Engineering, Bilkent University, Ankara 06533, Turkey*
E-mail: efe@cs.bilkent.edu.tr
[b] *Computer Science Department, University of Southwestern Louisiana, Lafayette, LA 70504, USA*
E-mail: axb0650@usl.edu
[c] *Dipartimento de Arquitectura y Tecnogia de Computadores, Escuela Universitaria de Informatica, Ctra. Valencia Km. 7, 28031 Madrid, Spain*
E-mail: anto@eui.upm.es.

This paper shows how to embed complete binary trees in products of complete binary trees, products of shuffle-exchange graphs, and products of de Bruijn graphs with small dilation and congestion. In the embedding results presented here the size of the host graph can be fixed to an arbitrary size, while we define no bound on the size of the guest graph. This is motivated by the fact that the host architecture has a fixed number of processors due to its physical design, while the guest graph can grow arbitrarily large depending on the application. The results of this paper widen the class of computations that can be performed on these product graphs which are often cited as being low-cost alternatives for hypercubes.

1. Introduction

Let $G^r(N)$ denote the r-dimensional product graph obtained from the N-node graph $G(N)$. Note that $G^r(N)$ contains N^r nodes. (As a special case, every graph $G(N)$ is a one-dimensional product of itself, and we omit r when $r = 1$.) Let $T(N)$ be the N-node complete binary tree, where $N = 2^h - 1$. We prove the following results:

1. $T(2^{rh-\lceil r/2\rceil+l} - 1)$, where $l > 1$, can be embedded in the r-dimensional product of complete binary trees, $T^r(2^h-1)$, with dilation 2, congestion 2, and load 2^l-1.

2. Given the r-dimensional product of shuffle-exchange graphs, $S^r(N)$,

   (a) $T(N^r2^{l-1} - 1)$ can be embedded in it with dilation 3, congestion 2, and load $2^l - 1$,

   (b) $T((N2^l)^r - 1)$ can be embedded in it with dilation 4, congestion 4, and load 2^{rl}.

3. $T((N2^l)^r - 1)$ can be embedded in the r-dimensional product of de Bruijn graphs, $D^r(N)$, with dilation 2, congestion 2, and load 2^{rl}.

© J.C. Baltzer AG, Science Publishers

The significance of the first result above is not only due to its own merit, but also due to the transitive nature of graph embedding results. If a graph can be efficiently embedded in G, it can also be embedded, with the same efficiency, in any graph that contains G as a subgraph. The first result implies that complete binary trees can be efficiently embedded in any graph that contains the product of complete binary trees as a subgraph. In practice, product networks are likely to be built from those factor networks that either contain the complete binary tree or that can emulate the complete binary tree with low dilation and congestion. For any such network, the first result above gives an upper bound for the dilation and congestion of embedding complete binary trees. On the other hand, it is possible that a better dilation and congestion can be obtained if we find a direct embedding rather than a transitive method. The second and third results are examples of such direct methods.

The first problem above, for unit load, was originally addressed in [2], where it was shown that $T(2^{r(h-1)+1} - 1)$ is a subgraph of $T^r(2^h - 1)$. When $r = 2$ this method embeds the largest possible tree for the number of nodes in $T^r(2^h - 1)$, but when $r > 2$ the size of the tree is smaller by a factor of 2^{r-1}. Thus, as r grows the method of [2] becomes less and less interesting. To utilize more nodes of the host, a unit-load embedding was presented in [3] with dilation 3 and congestion 3. Our emphasis here is how to embed arbitrarily-large complete binary trees in the fixed size host graph. It turns out that the dilation and congestion values can be reduced from 3 to 2 when the load is increased.

The second and third problems above were addressed in [8] for unit load, but the methods presented there only apply for two dimensions and use only about half of the nodes of the product graph. Our methods utilize all (but one) of the nodes of the product graph and are applicable for any number of dimensions. Also, our methods yield perfectly balanced loads for the nodes of the host graphs.

Since a parallel architecture has a fixed size due to its physical design, these results have significant practical importance as they show a way for solving arbitrarily large tree computations on fixed-size parallel computers. These important practical concerns appear to have been omitted in most of the papers in the literature except by a few researchers [1,6,7].

2. Definitions and notation

The nodes of the N-node *complete binary tree* are assigned the labels $1, \ldots, N$. Each node u, $u < N/2$, is connected to nodes $2u$ and $2u + 1$. This labeling will be referred to as the *level-order* labeling of $T(N)$ (see figure 1). The graph $T(2^h - 1)$ will often be also called the h-level complete binary tree.

The N-node *shuffle-exchange graph*, denoted $S(N)$, contains $N = 2^n$ nodes, labeled $0, \ldots, N - 1$, and $3 \times 2^{n-1}$ edges connected as follows:

(a) (u, v) is an "exchange" edge if $v = u + 1$, where u is even, or $v = u - 1$, where u is odd, or

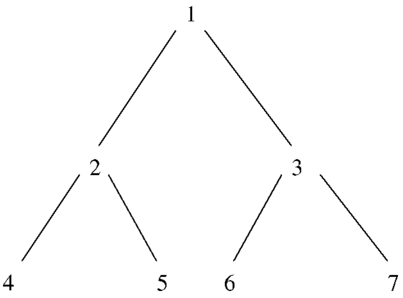

Figure 1. Level-order labeling of the complete binary tree.

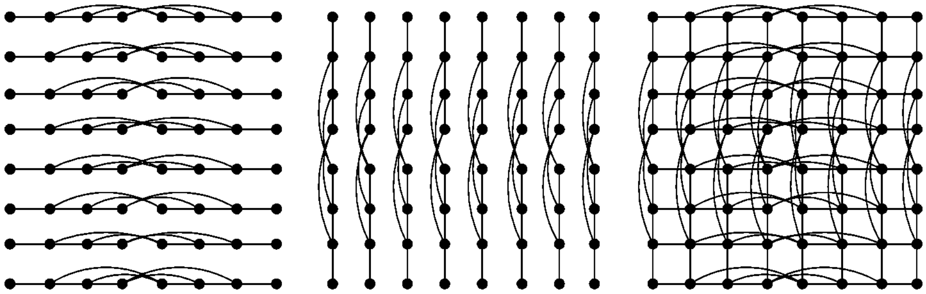

Figure 2. Construction of the 2-dimensional product of the shuffle-exchange graph $S(8)$. Both rows and columns are connected in the pattern of the basic shuffle-exchange graph.

(b) (u, v) is a "shuffle" edge if $v = 2u$, where $u < N/2$, or $v = (2u \text{ mod} N) + 1$, where $u \geqslant N/2$.

The N-node *de Bruijn graph*, denoted $D(N)$, contains $N = 2^n$ nodes, labeled $0, \ldots, N - 1$, and 2^{n+1} edges connected as follows: (u, v) is an edge of $D(N)$ if $v = 2u \text{ mod} N$ or $v = (2u \text{ mod} N) + 1$.

Let $G = (V_G, E_G)$ and $H = (V_H, E_H)$ be two arbitrary graphs. Their *Cartesian product* is the graph $P = G \otimes H$ whose vertex set is $V_G \times V_H$ and whose edge set contains all edges of the form (x_1x_0, y_1y_0) such that either $x_1 = y_1$ and $(x_0, y_0) \in E_G$, or $x_0 = y_0$ and $(x_1, y_1) \in E_H$.

The *r-dimensional homogeneous product of an N-node graph* $G(N)$, denoted $G^r(N)$, is:

(1) a single vertex with no labels and no edges if $r = 0$,

(2) $G(N) \otimes G^{r-1}(N)$ when $r > 0$.

Figure 2 illustrates this definition by presenting the construction of the 2-dimensional product $S^2(8)$.

An *embedding* of a "guest" graph G in a "host" graph H is a mapping of the vertices of G into the vertices of H and the edges of G into paths in H. The main cost measures used in embedding efficiency are (see [2]):

- *Load* of an embedding is the maximum number of vertices of G mapped to any vertex of H.
- *Dilation* of an embedding is the maximum path length in H representing an edge of G.
- *Congestion* of an embedding is the maximum number of paths (that correspond to the edges of G) that share any edge of H.

The level-order labeling of a complete binary tree as in figure 1 defines an embedding of $T(N-1)$ in $S(N)$ with dilation 2, congestion 2, and load 1 [5]. This labeling also shows that $T(N-1)$ is a subgraph of $D(N)$ [8].

3. Embedding in the product of complete binary trees

In this paper we use the embedding method of [2] as part of the improved embedding method presented here. For easy reference this result is included here.

Theorem 1. $T(2^{r(h-1)+1}-1)$ is a subgraph of $T^r(2^h-1)$.

As an example, figure 3 shows the embedding for $r=2$.

The main result of this section is the following:

Theorem 2. $T(2^{rh-\lceil r/2\rceil+l}-1)$, where $l>1$, can be embedded in $T^r(2^h-1)$ with dilation 2, congestion 2, and load 2^l-1.

Before proving the theorem, we will first distinguish a particular node in the $T^r(N)$ graph as follows:

- *Root of* $T^r(N)$. The node $v=v_{r-1}\ldots v_1v_0$ is the root of $T^r(N)$ if and only if $v_i=1$ (that is, v_i is the root of $T(N)$), for all $0\leqslant i\leqslant r-1$.

First we show that a 63-node complete binary tree can be embedded in $T^2(7)$ with dilation 2, congestion 2, and load 3. A simple modification of this gives an

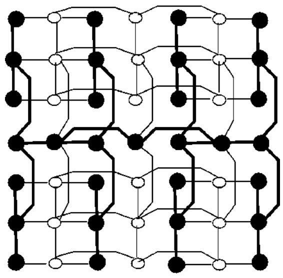

Figure 3. Embedding the complete binary tree $T(31)$ in $T^2(7)$ by theorem 1. The complete binary tree subgraph is highlighted by heavy dark lines.

embedding for $T(2^{l+5} - 1)$ in $T^2(7)$ with the same dilation and congestion, but the load is increased to $2^l - 1$, where $l > 1$. Next, we use induction on r to show that $T(2^{\lfloor 5r/2 \rfloor + 1} - 1)$ can be embedded in $T^r(7)$ with dilation 2, congestion 2, and load 3. Finally, by combining these results and theorem 1 the claim of the theorem is obtained.

Lemma 1. $T(63)$ can be embedded in $T^2(7)$ with dilation 2 and congestion 2, such that 10 nodes have load 3 and 33 nodes have load 1. The remaining 6 nodes of $T^2(7)$ are unused. In this embedding the root of the embedded tree coincides with the root of $T^2(7)$.

Proof. Figure 4(a) presents a subgraph of $T^2(7)$ extended with some new nodes (the small empty nodes). We emphasize that the small empty nodes in figure 4(a) do not exist in $T^2(7)$ itself; we just added these nodes for convenience in the presentation of proof (we will eventually erase these nodes). Figure 4(b) presents a 63-node complete binary tree drawn in a form suitable for the following discussion.

Consider embedding the graph of figure 4(b) in the graph of figure 4(a) by superimposing the nodes of the two graphs on top of each other. It can be easily checked that any edge in figure 4(b) corresponds to a path of length no more than 3 in figure 4(a). Dilation-3 edges are those that connect the large dark nodes to small empty nodes in figure 4(b). It can be also easily seen that the maximum congestion of 3 is found in some of the edges connecting large empty nodes with small empty nodes in figure 4(a). (The reader can trace the connections sharing the edge from the large empty node to the small empty node in the rightmost column of figure 4(a).)

Finally, by contracting the edges between the large empty nodes and small empty nodes in figure 4(a) we obtain a real subgraph of $T^2(7)$, while we increase the load in the large empty nodes to 3. This process also reduces both the dilation and congestion values to 2. Since the tree of figure 4(b) has 6 levels we have obtained an embedding of $T(63)$ in $T^2(7)$ with dilation and congestion values of 2, and load 3. From the figure it is easily verified that the root of the embedded tree coincides with the root of $T^2(7)$. □

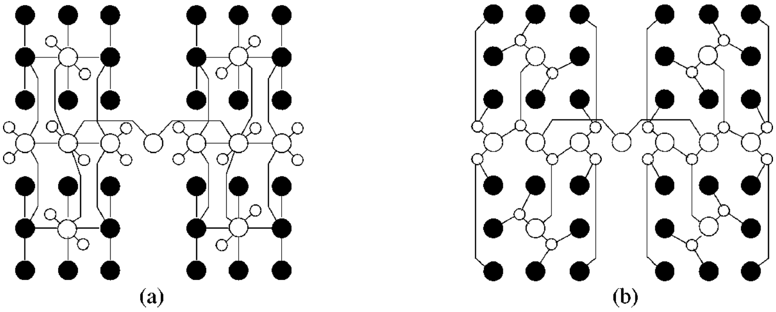

Figure 4. Embedding the $(l + 5)$-level complete binary tree in a subgraph of $T^2(7)$.

Corollary 1. $T(2^{l+5}-1)$, where $l > 1$, can be embedded in $T^2(7)$, such that 32 nodes of $T^2(7)$ have load $2^l - 1$, 10 nodes have load 3, and the root has load 1.

This is obtained by simply replacing the dark nodes of figure 4(b) (the leaves of the embedded tree) by l-level complete binary trees, and then using the embedding method above.

The properties of the embedding highlighted in the statement of lemma 1 are needed in lemma 2 below. This lemma uses induction on r to increase the number of dimensions.

Lemma 2. $T(2^{\lfloor 5r/2 \rfloor+1} - 1)$ can be embedded in $T^r(7)$ with dilation 2, congestion 2, and load 3. In this embedding the root of the embedded tree is the root of $T^r(7)$ and the leaves are in unit-load nodes.

Proof. We prove the claim by induction on the number of dimensions, r. We will have two initial base cases (cases of $r = 1$ and $r = 2$) and an induction step that increases the number of dimensions by two. This allows to prove the claim for any number of dimensions, since depending on whether r is odd or even, we can use either $r = 1$ or $r = 2$ as the base case, respectively.

The base cases are trivially verified. For $r = 1$, $T^1(7)$ is isomorphic to $T(2^{\lfloor 5/2 \rfloor+1} - 1)$. For $r = 2$, lemma 1 above shows the embedding.

In the induction step, given an embedding of $T(2^{\lfloor 5k/2 \rfloor+1} - 1)$ in $T^k(7)$ with dilation 2, congestion 2, and load 3, we show that it is possible to embed $T(2^{\lfloor 5(k+2)/2 \rfloor+1} - 1)$ in $T^{k+2}(7)$ with the same dilation, congestion, and load. In this embedding the root of the embedded tree is the root of $T^{k+2}(7)$.

By removing all the edges along dimensions k and $k+1$ from $T^{k+2}(7)$ we obtain 49 disjoint copies of $T^k(7)$. From the induction hypothesis, we can embed a disjoint copy of $T(2^{\lfloor 5k/2 \rfloor+1} - 1)$ in each of these copies.

Now consider only the roots of the embedded trees and reconnect them along dimensions k and $k+1$. Considering only the dimensions k and $k+1$, we have a graph isomorphic to $T^2(7)$. From lemma 1, we know that a 6-level complete binary tree can be embedded in this graph. The leaves of this tree (the dark nodes of figure 4(a)) correspond to the roots of embedded $T(2^{\lfloor 5k/2 \rfloor+1} - 1)$ graphs. (The trees whose roots fall in the large empty nodes are not considered.)

By this procedure, we have obtained an embedding of the $(2^{\lfloor 5k/2 \rfloor+1+5} - 1) = (2^{\lfloor 5(k+2)/2 \rfloor+1} - 1)$-node complete binary tree in $T^{k+2}(7)$ with dilation 2, congestion 2, and load 3, as claimed. □

Proof of theorem 2. If we remove the 2 lowest levels from every tree along each dimension in $T^r(2^h - 1)$ we obtain a graph isomorphic to $T^r(2^{h-2} - 1)$. From theorem 1 we can embed an $(r(h-3)+1)$-level tree in this subgraph of $T^r(2^h - 1)$ such that the leaves of the tree are mapped to the leaves of $T^r(2^{h-2} - 1)$.

Similarly, if we remove the $h-3$ top levels from every tree along each dimension we obtain a disconnected graph formed by $2^{r(h-3)}$ disjoint copies of $T^r(7)$. Then, by using lemma 2, we embed a $(\lfloor 5r/2 \rfloor + 1)$-level tree in each copy of $T^r(7)$, where the roots of the embedded trees coincide with the roots of $T^r(7)$ graphs. The combination of both embeddings in $T^r(2^h-1)$ yields an embedding of the $(\lfloor 5r/2 \rfloor + 1 + r(h-3)) = (rh - \lceil r/2 \rceil + 1)$-level complete binary tree in $T^r(2^h-1)$ with dilation 2, congestion 2, and load 3. Note that in this tree the leaves are embedded with unit load.

Finally, by replacing the leaves of the embedded tree with l-level trees (as in corollary 1) we obtain a dilation 2 and congestion 2 embedding where the load is $2^l - 1$. □

This proves the first result claimed in the introduction and completes this section.

4. Embedding in the product of shuffle-exchange graphs

In this section we focus our attention on embeddings of complete binary trees of arbitrary size in $S^r(N)$. We start by presenting a method to embed $T(N^r-1)$ in $S^r(N)$ with dilation 3, congestion 2, and unit load. We continue by showing how to extend this method for arbitrarily large trees with the same dilation and congestion values, thus, proving the result 2(a) claimed in section 1.

However, in this embedding half of the nodes (minus one) of $S^r(N)$ have unit load, while the other half are collectively mapped most of the nodes of the embedded tree. In the next section we comment on a method to embed arbitrarily large trees with perfectly-balanced load distribution (result 2(b)).

Theorem 3. $T(N^r-1)$ can be embedded in $S^r(N)$ with dilation 3, congestion 2, and unit load.

Proof. We prove the theorem by induction on the number of dimensions. We already mentioned that $T(N-1)$ can be embedded in $S(N)$ with dilation 2 and congestion 2, which proves the base case $r=1$. We now illustrate the induction step by presenting the construction of the embedding of $T(N^2-1)$ in $S^2(N)$. The generalization of this process for an arbitrary number of dimensions is similar and will be briefly described.

We begin by embedding $T(N-1)$ in each of the subgraphs isomorphic to $S(N)$ that form the dimension-1 connections in $S^2(N)$. Since each node has a label of the form v_1v_0, we can do this by using the level-order embedding of $T(N-1)$ in $S(N)$ using the v_0 part of the label. Note that the roots of these N trees all have the form $v_1 1$ and that the nodes $v_1 0$ are all unused. See figure 5 (looking at row connections only). We can now embed another $N-1$ node complete binary tree using the level-order labeling in the nodes of the form $v_1 0$ using dimension-2 connections. This tree forms the "top" of the N^2-1 node complete binary tree. The root of this tree is at 10. The leaves of this tree are found in the nodes $k0$, where $N/2 \leqslant k \leqslant N-1$. Each of these leaves now becomes the root of two subtrees as described next.

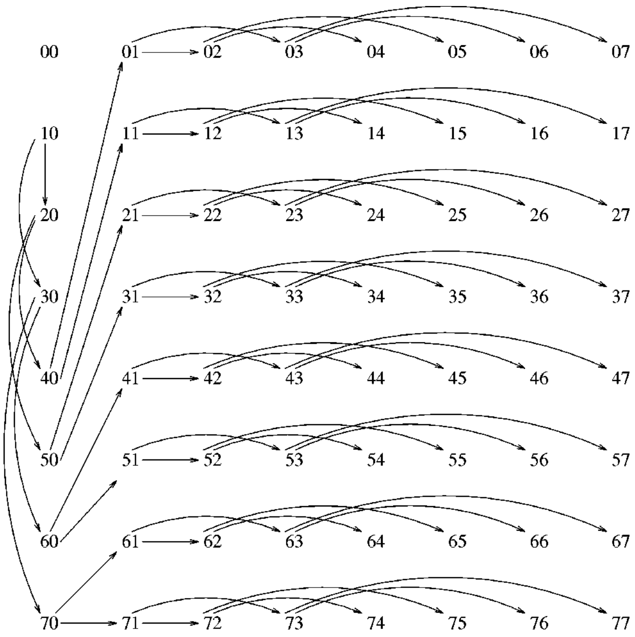

Figure 5. Embedding of 63-node complete binary tree in the 2-dimensional product of shuffle-exchange graphs.

Let $k^l = 2k - N$ and $k^r = 2k - N + 1$. The left child of $k0$ is k^l1 and the right child of $k0$ is k^r1 (see figure 5). The connection between $k0$ and k^r1 is realized by a path of length 2 in $S^2(N)$. The path from $k0$ to k^r1 is formed by the following edges:

1. $k0$ is connected to $k1$ by an exchange edge in dimension 1.
2. $k1$ is connected to k^r1 by a shuffle edge in dimension 2. Since the binary form of k has a '1' in the most significant position, the shuffle of k results in the label value $2k - N + 1$.

The connection between $k0$ and k^l1 is realized by a path of length 3. That path is formed by the following edges:

1. Traverse the two edges as described above, $k0$ to $k1$ to k^r1.
2. k^r1 is connected to k^l1 by an exchange edge in dimension 2.

The dilation of this embedding is clearly 3. The congestion is 2 because the paths to the left and right child of $k0$ coincide with each other but do not coincide with any other path between adjacent nodes in the tree. This completes the case for $r = 2$.

Given that there is an embedding of an $(N^{r-1} - 1)$-node complete binary tree in $S^{r-1}(N)$, with the root at node $10\ldots0$ and with congestion 2, and dilation 3, we can construct an embedding of the $N^r - 1$ node complete binary tree in $S^r(N)$ with these same properties. We do this by first embedding the $(N^{r-1} - 1)$-node complete binary tree in the N subgraphs isomorphic to $S^{r-1}(N)$ formed if the highest dimension connections are not considered. All nodes within each subgraph have the same value v_{r-1} in their labels. We now embed an $(N-1)$-node complete binary tree in the new dimension in the subgraph isomorphic to $S(N)$ formed by the nodes of the form $v_{r-1}0\ldots0$. The root of this tree is at $10\ldots0$. We form the connections between the $N/2$ leaves of this tree and the roots of the N subtrees in the same manner as in the 2-dimensional case. This time only v_{r-1} and v_{r-2} will be considered when connecting $k0\ldots0$ to its descendents. □

Corollary 2. $T(N^r 2^{l-1} - 1)$ can be embedded in $S^r(N)$ with dilation 3, congestion 2, and load $2^l - 1$.

This embedding is obtained by simply replacing the leaves of the embedded tree by an l-level complete binary tree, as in corollary 1. This proves the result 2(a) claimed in section 1.

Note that if $l > 1$, the load of the embedding described in the above corollary is not fully balanced. Half the nodes of $S^r(N)$ will have load $2^l - 1$, while the other half (except one unused) has unit load. It is possible to obtain a better load balance by increasing the dilation and congestion slightly. It will be easier to explain how to do this once we see the embedding method in products of de Bruijn graphs.

5. Embedding in the product of de Bruijn graphs

All the results presented in the previous sections are also applicable to products of de Bruijn graphs. The reason is that $T^r(N-1)$ is a subgraph of $D^r(N)$ [2, theorem 13] and that $S^r(N)$ is a subgraph of $D^r(N)$ (combining [4, theorem 2] and [2, theorem 3]). However, we are able to obtain better embeddings in $D^r(N)$ if we consider this network directly.

Again here we initially focus our attention on embeddings with unit load. Then we comment on how to extend this method for embedding arbitrarily large trees with perfectly balanced load distribution, thereby proving the result 3 claimed in section 1.

Theorem 4. $T(N^r - 1)$ can be embedded in $D^r(N)$ with dilation 2, congestion 2, and unit load.

Proof. This proof is similar to that of theorem 3. In the interest of brevity, we only sketch the basic idea pointing out the differences from the above case.

It was shown in [8] that $D(N)$ contains the $(N-1)$-node tree as a subgraph. This result can be used for the first dimension connections of figure 5. The connections in the second dimension require congestion 2, just as for $S^r(N)$, but a dilation of 2 instead of 3. This is because the connection between $k0$ and k^r1 is realized by a path of length 2 in $D^r(N)$. This path is formed by the following edges:

1. $k0$ is connected to $k1$ by an edge in dimension 1.
2. $k1$ is connected to k^r1 by an edge in dimension 2. Since the binary form of k has a '1' in the most significant position, the shuffle of k results in the value $2k - N + 1$.

The connection between $k0$ and k^l1 is realized by a path also of length 2. That path is formed by the following edges:

1. $k0$ is connected to $k1$ by an edge in dimension 1.
2. $k1$ is connected to k^l1 by the edge connecting k to label value $2k - N$ in dimension 2.

This completes the proof for the case of $r = 2$. For $r > 2$, similar arguments as in theorem 3 apply. □

We could use now this result to embed larger trees using the same technique as used in corollaries 1 and 2. Like in these results, the embedding obtained would not fully balance the load among the nodes of the host graph.

However, it is possible to map arbitrarily large complete binary trees to a fixed-size product $D^r(N)$ with perfectly-uniform load distribution. That is, if the product graph contains N^r nodes, we can embed $T((N2^l)^r - 1)$ in it with a uniform load of 2^{rl} for all nodes of the product graph, with the exception of one node that will be mapped $2^{rl} - 1$ nodes.

The new embedding can be done in two steps. In the first step, we embed $T((N2^l)^r - 1)$ in $D^r(N2^l)$ with dilation 2, congestion 2, and load 1 by the method of theorem 4. In the second step, we embed $D^r(N2^l)$ in $D^r(N)$ with dilation 1, congestion 1, and load 2^{rl} by the method given in [2, corollary 8]. This induces an embedding for $T((N2^l)^r - 1)$ in $D^r(N)$ with dilation 2, congestion 2, and load 2^{rl}, as claimed in section 1 (result 3).

This result can also be used to obtain an embedding of $T((N2^l)^r - 1)$ in $S^r(N)$ with a perfectly-balanced load of 2^{rl} (result 2(b)). To do so, we simply combine it with an embedding of $D^r(N)$ in $S^r(N)$ with dilation 2, congestion 2, and unit load [2,5]. This leads to the dilation and congestion values of 4.

6. Remarks

The embedding methods in this paper can also be applied, without any change, to product graphs made from graphs containing different numbers of nodes for different dimensions.

It is shown in [2] that if G contains the complete binary tree as a subgraph, then $G^r(N)$ contains the r-dimensional product of complete binary trees as a subgraph. Combined with this result, theorem 2 implies that for any graph G, if G contains the complete binary tree as a subgraph, then its r-dimensional product can embed the complete binary tree with dilation 2 and congestion 2. This observation makes theorem 2 very powerful and widely applicable since the factor graph G used in practice is likely to contain the complete binary tree as a subgraph. For example, we easily conclude from theorem 2 that arbitrarily large complete binary trees can be embedded in products of de Bruijn graphs with the dilation 2 and congestion 2.

Besides the generality of this result, we inspected two specific cases to see if the cost measures can be improved by considering the specific connections in these graphs. The methods in sections 4 and 6 did not improve the dilation or congestion of embedding, but they improved the load distribution by mapping exactly the same number of tree nodes to each node of the host graph.

Acknowledgements

This research has been supported by Louisiana Board of Regents, grant No. LEQSF (1998-01)-RD-A-36; the U.S. Department of Energy, grant No. DE-FG02-97ER1220; Spanish Ministry of Education, grant No. PF94-04166960; and the Turkish Scientific and Engineering Research Council, grant No. EEEAG-199E013.

References

[1] K. Efe, Embedding large complete binary trees in hypercubes with load balancing, Journal of Parallel and Distributed Computing 35(1) (1996) 104–109.

[2] K. Efe and A. Fernández, Products of networks with logarithmic diameter and fixed degree, IEEE Transactions on Parallel and Distributed Systems 6 (September 1995) 963–975.

[3] K. Efe and A. Fernández, Mesh connected trees: A bridge between grids and meshes of trees, IEEE Transactions on Parallel and Distributed Systems, to appear (1996).

[4] R. Feldmann and W. Unger, The cube-connected cycles network is a subgraph of the butterfly network, Parallel Processing Letters 2(1) (1992) 13–19.

[5] A. Fernández, Homogeneous product networks for processor interconnection, Ph.D. thesis, University of Southwestern Louisiana, Lafayette, LA (1994).

[6] J.P. Fishburn and R.A. Finkel, Quotient networks, IEEE Transactions on Computers 31 (April 1982) 288–295.

[7] R. Koch, T. Leighton, B. Maggs, S. Rao and A.L. Rosenberg, Work-preserving emulations of fixed-connection networks, in: *Proc. of the 21st Annual ACM Symposium on Theory of Computing*, Seattle (May 1989) pp. 227–240.

[8] A.L. Rosenberg, Product-shuffle networks: Toward reconciling shuffles and butterflies, Discrete Applied Mathematics 37/38 (July 1992) 465–488.