Embedding of Hypercubes into Complete Binary Trees

Paul Manuel[1], Indra Rajasingh[2], Jasintha Quadras[3]

[1] Department of Information and Computer Science,
King Fahd University of Petroleum and Minerals, Dhahran,
Saudi Arabia 31261. manuel@ccse.kfupm.edu.sa
[2] Department of Mathematics, Loyola College, Chennai
India 600 034. indrarajasingh@yahoo.com
[3] Department of Mathematics, Stella Maris College, Chennai
India 600 086. jquadras@yahoo.com

Abstract

We estimate and characterize the edge congestion-sum measure for embeddings of hypercubes into complete binary trees. Our algorithms produce optimal values of sum of edge-congestions in linear time.

Keywords: hypercube, tree, embedding, bandwidth, dilation, congestion.

1 Introduction and Terminology

Let G and H be finite graphs with n vertices. $V(G)$ and $V(H)$ denote the vertex sets of G and H respectively. $E(G)$ and $E(H)$ denote the edge sets of G and H respectively. A $1-1$ mapping $f : V(G) \to V(H)$ is called an embedding of G into H. H is normally called a host graph. The graph G that is being embedded is sometimes called a virtual graph or a guest graph. Some authors use the name labeling instead of embedding. We may use both terminologies here.

[1]This research is supported by King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.

The Dilation Problem: The dilation $\widehat{D}_f(G,H)$ of an embedding f of G into H is defined as

$$\widehat{D}_f(G,H) = \max_{(u,v)\in E(G)} d_H(f(u), f(v)) \tag{1}$$

where $d_H(x,y)$ denotes the distance between x and y in H. Then, the dilation of G into H is defined as

$$\widehat{D}(G,H) = \min_{f} \widehat{D}_f(G,H) \tag{2}$$

where the minimum is taken over all embeddings f of G into H. The dilation is also known as *bandwidth* when the host graph is a path [14, 17]. The *dilation problem* [1, 3, 18] for a graph G into H is that of finding an embedding of G into H that induces the dilation $\widehat{D}(G,H)$.

The Dilation-sum Problem: For an embedding f of G into H, the dilation-sum of f is given by

$$\widetilde{D}_f(G,H) = \sum_{(u,v)\in E(G)} d_H(f(u), f(v)) \tag{3}$$

Then, the *minimum dilation-sum* of G into H is defined as

$$\widetilde{D}(G,H) = \min_{f} \widetilde{D}_f(G,H) \tag{4}$$

where the minimum is taken over all embeddings f of G into H. The *dilation-sum problem* for a graph G into H is that of finding an embedding of G into H that induces the minimum dilation-sum $\widetilde{D}(G,H)$.

The Congestion-sum Problem: The congestion of an embedding f of G into H is the maximum number of edges of the guest graph that are embedded on any single edge of the host graph. Normally an embedding f of G into H defines a mapping of $V(G)$ into $V(H)$ and does not map the edges of $E(G)$ into $E(H)$. In the congestion problem, we need the information as to how the edges of $E(G)$ are embedded into $E(H)$. For every edge (u,v) of G, there are several paths between its images $f(u)$ and $f(v)$ in H. Let us assume that the embedding f of G into H defines a unique path between $f(u)$ and $f(v)$ in H for every edge (u,v) of G. Let us say $P_f(f(u), f(v))$ denotes the path between $f(u)$ and $f(v)$ in H for the edge (u,v) of G defined by f. This may be expressed in a mathematical format. Let $C_f(G, H(e))$ denote the number of edges (u,v) of G such that e is in the path $P_f(f(u), f(v))$ between $f(u)$ and $f(v)$ in H. In other words,

$$C_f(G, H(e)) = |\{(u,v) \in E(G) | e \in P_f(f(u), f(v))\}|$$

Top

The *congestion-sum of an embedding* f of G into H is given by

$$\widetilde{C}_f(G,H) = \sum_{e \in E(H)} C_f(G, H(e)) \tag{5}$$

Then, the *minimum congestion-sum* of G into H is defined as

$$\widetilde{C}(G,H) = \min_f \widetilde{C}_f(G,H) \tag{6}$$

where the minimum is taken over all embeddings f of G into H. The *congestion-sum problem* [5, 6] of a graph G into H is that of finding an embedding of G into H that induces the congestion-sum $\widetilde{C}(G,H)$.

We are also interested in the *minimum congestion of an edge* which is defined as

$$C_{\min}(G, H(e)) = \min_f C_f(G, H(e)) \tag{7}$$

where the minimum is taken over all embeddings f of G into H.

Remark 1 *Subsequently, for the sake of simplicity $C_f(G, H(e))$ will be represented by $C_f(e)$ and $C_{\min}(G, H(e))$ will be represented by $C_{\min}(e)$. Notice that the sum of $C_{\min}(e)$ over all the edges e of H constitutes a lower bound for the minimum congestion-sum $\widetilde{C}(G,H)$. To solve the congestion-sum problem of a graph G into H, we simply identify the embedding that yields $C_{\min}(e)$ for every edge e of H.*

Remark 2 *Notice that the congestion-sum problem and the dilation-sum problem are the same.*

The following lemma provides a key technique to estimate the congestion-sum of an embedding. It will be used throughout this paper.

Lemma 1 *[21] Let f be an embedding of a graph G into an arbitrary tree T_n. Let $e \in E(T_n)$ and T_1 be a component of $T_n - e$. Then the congestion $C_f(e)$ on e is given by*

$$C_f(e) = \sum_{v \in G_1} d_G(v) - 2|E(G_1)| \tag{8}$$

where G_1 is the subgraph of G induced by the vertices $\{f^{-1}(u) | u \in T_1\}$ and $d_G(v)$ denotes the degree of v in G.

The dilation-sum of a graph embedding arises from VLSI designs, data structures and data representations, networks for parallel computer systems, biological models that deal with cloning and visual stimuli, parallel architecture, structural engineering and so on [17]. The dilation-sum problem of an arbitrary graph on a path is called linear layout or the linear arrangement problem in VLSI literature [17]. The concept of embedding is widely studied in the literature of fixed interconnection parallel architectures [18].

Top

The dilation problem is NP-complete for two classes of 'almost' caterpillars on a path [15] and trees of maximum degree 3 on paths [17, 24]. The dilation-sum problem is expected to be harder than the dilation problem [24]. From the above NP-complete results, it is easy to understand that these problems are in general hard. That is why, even though there are numerous results and discussions on the dilation-sum problem and the congestion-sum problem, most of them are dealing only approximate results.

The dilation-sum problem is studied for binary trees into paths [8, 12], hypercubes into grids [5], complete graphs into hypercubes [16]. The bounded cost of dilation and congestion has been estimated for the embedding on binary trees [24]. Most of the works on the dilation-sum problem and the dilation problem are on a particular case in which the host graph is a path, or a cycle [17]. There are also other general results on embeddings [2].

The concept of congestion is similar to cutwidth in graph theory [23, 26, 11]. There are several results on the congestion problem of various architectures such as trees into stars [25], trees into hypercubes [19], hypercubes into grids [5, 6], complete binary trees into grids [20], ladders and caterpillars into hypercubes [7, 10].

The embeddings discussed in this paper produce optimal congestion-sum. We demonstrate that the congestion-sum problem of hypercubes into complete binary trees is polynomially solvable. The nucleus part of this paper is the estimation of the congestion-sum of hypercubes into complete binary trees in linear time.

Here are a few definitions and notations we use in this paper.

Definition 1 *For $n \geq 1$ let Q^n denote the graph of the n-dimensional hypercube. The vertex set of Q^n is formed by the collection of all n-dimensional vectors with binary entries. Two vertices $x, y \in V(Q^n)$ are adjacent if and only if the corresponding vectors differ exactly in one entry [4, 18]. For convenience, the labels $\{0, 1, 2, \ldots, 2^n - 1\}$ of Q^n are represented by $\{1, 2, \ldots, 2^n\}$ respectively. See Figure 2.*

Notation 1 *Let S_α denote a set of α vertices of a guest graph G and $G[S_\alpha]$ denote the subgraph of G induced by S_α. Let $\overline{S}_\alpha$ represent some S_α for which the number of edges $|E(G[S_\alpha])|$ is maximum.*

Notation 2 *T^n denotes a complete binary tree. Let e_α denote an edge of T^n such that $|T_{e_\alpha}| = \alpha$ where T_{e_α} is a component of $T^n - e_\alpha$ and T_{e_α} is rooted at an end of e_α. Let $\Psi(e_\alpha)$ denote the number of e_α's in T^n. In other words, $\Psi(e_\alpha)$ denotes the number of subtrees of T^n with α vertices. See Figure 1 (a).*

Top

2 Embedding of Hypercubes into Complete Binary Trees

Throughout this section, we assume that an embedding f maps Q^n into T^n where T^n is a complete binary tree with 2^n nodes.

2.1 Properties of Hypercubes and Complete Binary Trees

Here are a few properties of hypercubes and complete binary trees.

Property 1 [4, 18] The maximum subgraph of Q^n induced by α vertices is given by

$$\left|E(G[\,\overline{S}_\alpha])\right| = \begin{cases} m2^{m-1} & \text{if} \quad \alpha = 2^m \\ m(2^{m-1}-1) & \text{if} \quad \alpha = 2^m - 1 \end{cases}$$

Property 2 [4, 9, 20, 25, 26] The number of subtrees of T^n with α vertices is given by

$$\Psi(e_\alpha) = \begin{cases} 1 & \text{if} \quad \alpha = 2^m, \quad m = 1, 2, \ldots, n-1 \\ 2^{n-m} - 1 & \text{if} \quad \alpha = 2^m - 1, \quad m = 2, 3, \ldots, n-1 \\ 2^{n-1} & \text{if} \quad \alpha = 1 \end{cases}$$

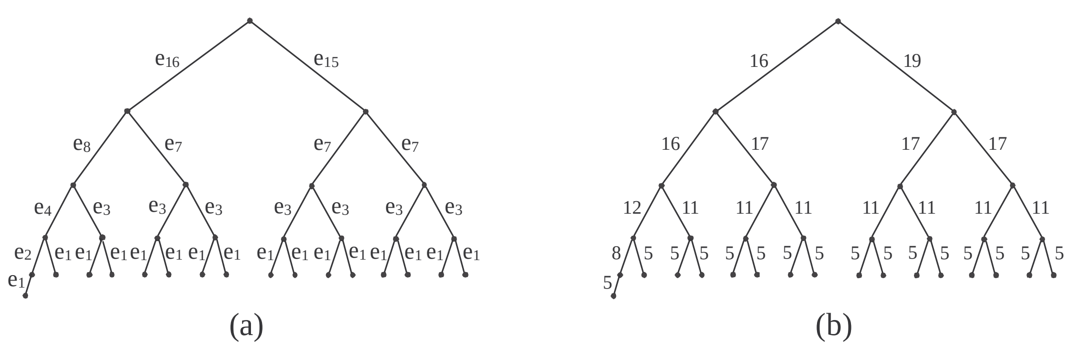

Figure 1: (a) Number of e_α's. (b) For each edge e, the value of $C_{\min}(e)$ is marked.

2.2 An Estimation of the Congestion-sum of a Hypercube on a Complete Binary Tree

We now estimate the congestion-sum of a hypercube on a complete binary tree. Here we estimate $C_{\min}(Q^n, T^n(e))$ for each edge e of T^n (see Figure 1 (b)).

Top

Lemma 2 *For the guest graph Q^n and the host graph T^n, the minimum edge-congestion is estimated as*

$$C_{\min}(e_\alpha) \geq \begin{cases} (n-m)\,2^m & \text{if } \ \alpha = 2^m \\ (n-m)\,2^m + (2m-n) & \text{if } \ \alpha = 2^m - 1 \end{cases}$$

where e_α is an edge of T^n such that $|T_{e_\alpha}| = \alpha$ and T_{e_α} is a component of $T^n - e_\alpha$.

Proof. Let $\overline{S}_\alpha$ be a maximal subgraph of Q^n induced by α vertices as defined in Property 1. Let f be an arbitrary embedding of Q^n into T^n. By Lemma 1, Property 1 and 2, if $\alpha = 2^m$, then

$$\begin{aligned} C_f(e_\alpha) &\geq n2^m - 2(m2^{m-1}) \\ &= (n-m)2^m \end{aligned}$$

Therefore

$$C_{\min}(e_\alpha) \geq (n-m)2^m \tag{9}$$

If $\alpha = 2^m - 1$, then

$$\begin{aligned} C_f(e_\alpha) &\geq n(2^m - 1) - 2(m(2^{m-1} - 1)) \\ &= (n-m)2^m + (2m-n) \end{aligned}$$

Therefore

$$C_{\min}(e_\alpha) \geq (n-m)2^m + (2m-n) \tag{10}$$

The lemma follows from equalities (9) and (10).

Now we estimate the congestion-sum of Q^n into T^n.

Lemma 3 *For a hypercube Q^n and a complete binary tree T^n, the minimum congestion-sum is at least $n2^{n-1} + \sum_{m=2}^{n-1} (2^{n-m} - 1)\,((n-m)\,2^m + (2m-n)) + \sum_{m=1}^{n-1} 2^m(n-m)$.*

Proof. By Property 2 and Lemma 2, we have

$$\begin{aligned} \widetilde{C}(Q^n, T^n) &= \Psi(e_1)C_{\min}(e_1) + \sum_{m=2}^{n-1} \Psi(e_{2^m-1})C_{\min}(e_{2^m-1}) \\ &\quad + \sum_{m=1}^{n-1} \Psi(e_{2^m})C_{\min}(e_{2^m}) \\ &\geqslant n2^{n-1} + \sum_{m=2}^{n-1} \left(2^{n-m} - 1\right)\left((n-m)\,2^m + (2m-n)\right) \\ &\quad + \sum_{m=1}^{n-1} 2^m(n-m) \end{aligned}$$

Top

Hence the lemma.

Example 1 *In Figure 2, the congestion-sum* $\widetilde{C}(Q^5, T^5)$ *of a hypercube* Q^5 *on a complete binary tree* T^5 *is* 279*. That is,*

$$\begin{aligned}
\widetilde{C}(Q^n, T^n) &= n2^{n-1} + \sum_{m=2}^{n-1} \left(2^{n-m} - 1\right) \left((n-m)\, 2^m + (2m-n)\right) \\
&\quad + \sum_{m=1}^{n-1} 2^m (n-m) \\
&= 80 + 77 + 51 + 19 + 8 + 12 + 16 + 16 \\
&= 279
\end{aligned}$$

2.3 An Embedding Algorithm of Hypercubes into Complete Binary Trees

Now we apply the well-known inorder traversal to construct an optimal embedding of Q^n into T^n. Inorder traversal on a tree is a widely known technique [13]. This traversal is used to read the labels of the tree and output the inorder listing of the labels. Here we use this technique to assign labels $\{1, 2, \ldots, 2^{n-1}\}$ to the internal nodes of the tree T^n. The remaining labels $\{2^{n-1}+1, \ldots, 2^n\}$ are assigned to the leaves of T^n.

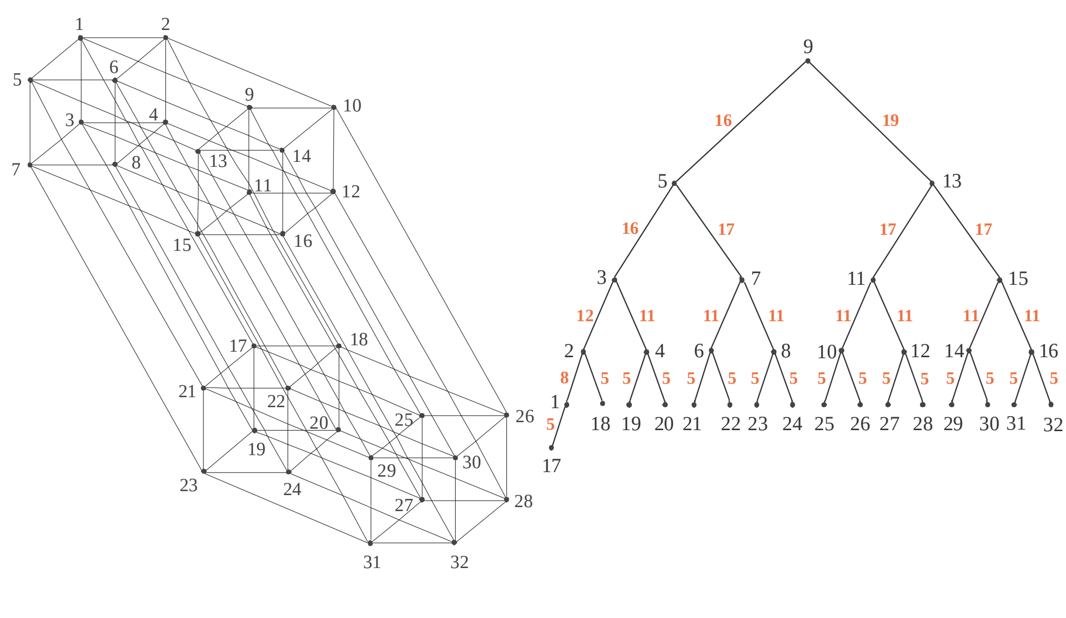

Figure 2: Embedding F of hypercube Q^5 into complete binary tree with minimum congestion-sum.

Top

2.3.1 Embedding Algorithm

Input: A hypercube Q^n with vertex set $\{1, 2, \ldots, 2^n\}$ and a complete binary tree T^n.

Algorithm: The leaves of the complete binary tree T^n are assigned labels $\{2^{n-1}+1, 2^{n-1}+2, \ldots, 2^n\}$ from left to right. The internal nodes of $T^n$are assigned labels $\{1, 2, \ldots, 2^{n-1}\}$ by inorder traversal from left to right (see Figure 2).

Output: An embedding F of Q^n into the complete binary tree T^n with minimum congestion-sum $\widetilde{C}(Q^n, T^n)$.

Proof of Correctness: The subtree T_{e_α} is a component of $T - e_\alpha$ such that $|T_{e_\alpha}| = \alpha$ and T_{e_α} is rooted at e_α. Let Π_α denote the number of edges of Q^n being embedded into T_{e_α} by the algorithm. In other words, $\Pi_\alpha = |E(G_\alpha)|$ where G_α is a subgraph of Q^n induced by the vertices $\{F^{-1}(u)\,|u \in T_{e_\alpha}\}$. It is enough to show that

$$\Pi_\alpha = \begin{cases} m2^{m-1} & \text{if} \quad \alpha = 2^m \\ m(2^{m-1} - 1) & \text{if} \quad \alpha = 2^m - 1 \end{cases} \tag{11}$$

Let e_i^k, $i = 1, 2, \ldots, 2^{n-k-1}$ and $0 \leq k \leq n-2$, denote the edges of T^n which are at a distance of $n-k-2$ from the root of T^n (see Figure 3). Even though the notations e_i^k and e_α carry different meanings, they are related to each other as given below:

$$e_i^k = \begin{cases} e_{2^{k+1}-1} & \text{if} \quad i \neq 1 \\ e_{2^{k+1}} & \text{if} \quad i = 1 \end{cases} \tag{12}$$

This is true because $T_{e_i^k}$, the subtree of T^n rooted at e_i^k has $2^{k+1}-1$ vertices if $i \neq 1$ and $T_{e_i^k}$ has 2^{k+1} vertices if $i = 1$.

Case $(i = 1)$**:** First we consider the case $i = 1$. The internal nodes of the subtree $T_{e_1^k}$ are assigned the labels of consecutive integers from 1 to 2^k. The leaves of $T_{e_1^k}$ are assigned the labels of consecutive integers from $1 + 2^{n-1}$ to $2^k + 2^{n-1}$. In other words, the internal nodes of $T_{e_1^k}$ are assigned the labels $\left\{1, 2, \ldots, 2^k\right\}$ and the leaves of $T_{e_1^k}$ are assigned the labels $\left\{1 + 2^{n-1}, \ldots, 2^k + 2^{n-1}\right\}$. The labels of $T_{e_1^k}$ are $\{1, 2, \ldots, 2^k\} \cup \{1+2^{n-1}, \ldots, 2^k+2^{n-1}\}$. Now the subgraph of Q^n induced by $\{1, 2, \ldots, 2^k\} \cup \{1 + 2^{n-1}, \ldots, 2^k + 2^{n-1}\}$ is a sub-hypercube on 2^{k+1} vertices. The subgraph of Q^n induced by $\left\{1, 2, \ldots, 2^k\right\} \cup \left\{1 + 2^{n-1}, \ldots, 2^k + 2^{n-1}\right\}$ has $(k+1)2^k$ edges. Moreover, $T_{e_1^k}$ has 2^{k+1} vertices. Thus $\Pi_{2^{k+1}} = (k+1)2^k$. In other words,

$$\Pi_\alpha = m2^{m-1} \text{ if } \quad \alpha = 2^m \tag{13}$$

Case $(i \neq 1)$**:** The internal nodes of $T_{e_i^k}$ are assigned the labels of consecutive integers from $x + 1$ to $x + 2^k - 1$ where $x = 2^k(i-1) + 1$, $k = 0, 1, \ldots, n-2$.

Top

The leaves of $T_{e_i^k}$ are assigned the labels of consecutive integers from $x + 2^{n-1}$ to $(x + 2^k - 1) + 2^{n-1}$. In other words, the internal nodes of $T_{e_i^k}$ are assigned the labels $\{x+1, x+2, \ldots, x+2^k-1\}$ where $x = 2^k(i-1)+1$. The leaves of $T_{e_i^k}$ are assigned the labels $\{x + 2^{n-1}, (x+1) + 2^{n-1}, \ldots, (x + 2^k - 1) + 2^{n-1}\}$. Let Λ denote the labels of $T_{e_i^k}$which is the set union of $\big\{x+1, x+2, \ldots, x+2^k-1\big\}$ and $\big\{x + 2^{n-1}, (x+1) + 2^{n-1}, \ldots, (x + 2^k - 1) + 2^{n-1}\big\}$. It is true that the subgraph of Q^n induced by $\Lambda \cup \{x\}$ is a sub-hypercube of 2^{k+1} vertices in Q^n. Thus the subgraph of Q^n induced by Λ has $(k+1)2^k - (k+1)$ edges. Moreover, $T_{e_i^k}$ has $2^{k+1} - 1$ vertices. Thus $\Pi_{2^{k+1}-1} = (k+1)2^k - (k+1) = (k+1)(2^k - 1)$. In other words,

$$\Pi_\alpha = m(2^{m-1} - 1) \text{ if } \quad \alpha = 2^m - 1 \tag{14}$$

The equation (11) follows from equations (13) and (14). Hence the proof.

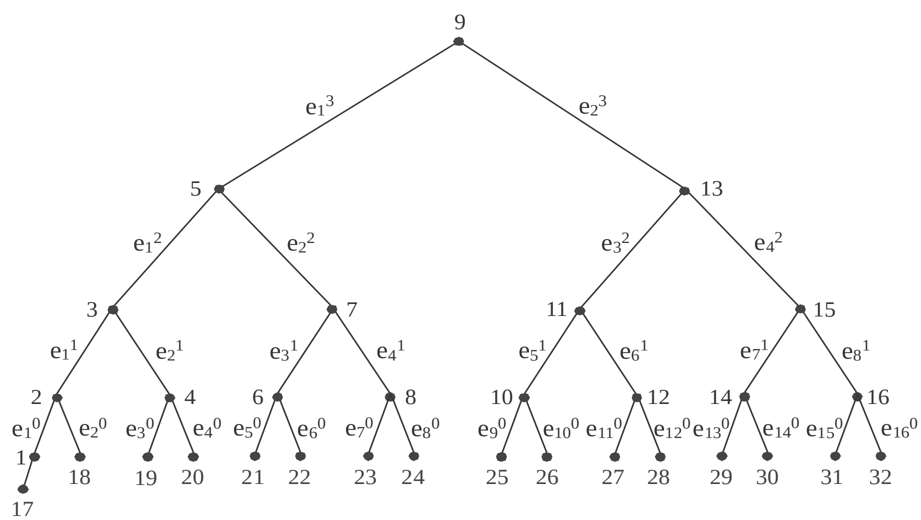

Figure 3: Definition of e_i^k.

Theorem 3 *The above embedding algorithm gives minimum congestion-sum in linear time.*

Since the algorithm performs inorder traversal, with O(1) time per node, the overall running time is linear.

3 Conclusion

We have solved the congestion-sum problem for an embedding of hypercubes into complete binary trees. The embeddings we constructed in this paper produce

Top

minimum congestion-sum in linear time. This technique can be extended to any guest graph G and any tree T. For application of the method one should estimate the minimum edge-cut for a set of vertices S_α of G for only certain values of α. These values correspond to the sizes of the components of $T - \{e\}$ for e in $E(T)$.

References

[1] N. Bagherzadeh, M. Dowds, and N. Nassif, Embedding an arbitrary tree into the star graph, IEEE Transactions on Computers, vol. 45, pp. 475-481, 1996.

[2] D. Barth, P. Fragopoulou, and M. C. Heydemann, Uniform emulations of Cartesian-product and Cayley graphs, Discrete Appl. Math., vol. 116, pp. 37–54, 2002.

[3] S. Bettayeb, B. Cong, M. Girou, and I. H. Sudborough, Embedding of star networks into hypercubes, IEEE Transactions on Computers, vol. 45, pp. 186–194, 1996.

[4] S. L. Bezrukov, Embedding complete trees into the hypercube, Discrete Appl. Math., vol. 110, pp. 101–119, 2001.

[5] S. L. Bezrukov, J. D. Chavez, L. H. Harper, M. Röttger, and U. P. Schroeder, Embedding of hypercubes into grids, MFCS 1998: Brno, Czech Republic, 1998, pp. 693–701, Electronic Edition Springer, Lecture Notes in Computer Science 1450.

[6] S. L. Bezrukov, J. D. Chavez, L. H. Harper, M. Röttger, and U. P. Schroeder, The congestion of n-cube layout on a rectangular grid, Discrete Mathematics, vol. 213, pp. 13–19, 2000.

[7] S. Bezrukov, B. Monien, W. Unger, and G. Wechsung, Embedding ladders and caterpillars into the hypercube, Discrete Appl. Math., vol. 83, pp. 21–29, 1998.

[8] D. Bienstock, On embedding graphs in trees, J. Combin. Theory Ser. B, vol. 49, pp. 103–136, 1990.

[9] A. Bouabdallah, M. C. Heydemann, J. Opatrny, and D. Sotteau, Embedding complete binary trees into star and pancake graphs, Theory Comput. Syst. 31 (1998), no. 3, 279 - 305.

[10] R. Caha and V. Koubek, Optimal embeddings of generalized ladders into hypercubes, Discrete Mathematics, vol. 233, pp. 65–83, 2001.

[11] J. D. Chavez and R. Trapp, The cyclic cutwidth of trees, Discrete Appl. Math., vol. 87, pp. 25–32, 1998.

Top

[12] M. Chrobak and W. Rytter, Two results on linear embeddings of complete binary trees, Theoretical Computer Science, vol. 136, pp. 507–526, 1994.

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to algorithms, MIT Press and McGraw-Hill, New York, 2001.

[14] J. Diaz, J. Petit and M. Serna, A survey of graph layout problems, Computing Surveys, vol. 34, pp. 313–353, 2002.

[15] L. T. Q. Hung, M. M. Syslo, M. L. Weaver and D. B. West, Bandwidth and density for block graphs, Discrete Math., vol. 189, pp. 163 - 176, 1998.

[16] M. Klugerman, A. Russell and R. Sundaram, On embedding complete graphs into hypercubes, Discrete Math., vol. 186, pp. 289–293, 1998.

[17] Y. L. Lai and K. Williams, A survey of solved problems and applications on bandwidth, edgesum, and profile of graphs, J. Graph Theory, vol. 31, pp. 75–94, 1999.

[18] T. F. Leighton, Introduction to parallel algorithms and architecture: arrays, trees, hypercubes, Morgan Kaufmann Publishers, San Mateo, Calif., 1992.

[19] A. Matsubayashi and S. Ueno, Small congestion embedding of graphs into hypercubes, Networks, vol. 33, pp. 71–77, 1999.

[20] J. Opatrny and D. Sotteau, Embeddings of complete binary trees into grids and extended grids with total vertex-congestion 1, Discrete Appl. Math., vol. 98, pp. 237–254, 2000

[21] I. Rajasingh, J. Quadras, P. Manuel, and A. William, Embedding of cycles and wheels into arbitrary trees, accepted for publication in Networks.

[22] M. Rottger and U. P. Schroeder, Efficient embeddings of grids into grids, Discrete Appl. Math., vol. 108, pp. 143–173, 2001.

[23] H. Schröder, O. Sykora, and I. Vrto, Cyclic cutwidth of the mesh, SOFSEM'99: theory and practice of informatics, Milovy, Czech Republic, 1999, pp. 443–452, Lecture Notes in Comput. Sci., 1725, Springer, Berlin, 1999.

[24] S. Simonson and I. H. Sudborough, On the complexity of tree embedding problems, Information Processing Letters, vol. 44, pp. 323–328, 1992.

[25] Y. C. Tseng, Y. S. Chen, T. Y. Juang, and C. J. Chang, Congestion-free, dilation-2 embedding of complete binary trees into star graphs, Networks, vol. 33, pp. 221–231, 1999.

[26] I. Vrto, Cutwidth of the r-dimensional mesh of d-ary trees, Theor. Inform. Appl., vol. 34, pp. 515–519, 2000.

Top