In *Mathematical and Computational Modelling* **11** (1988), pp. 222-227.

Embeddings in Hypercubes

Marilynn Livingston[1] Quentin F. Stout[2]

Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI 48109-2122

Abstract: One important aspect of efficient use of a hypercube computer to solve a given problem is the assignment of subtasks to processors in such a way that the communication overhead is low. The subtasks and their inter-communication requirements can be modeled by a graph, and the assignment of subtasks to processors viewed as an embedding of the task graph into the graph of the hypercube network. We survey the known results concerning such embeddings, including expansion/dilation tradeoffs for general graphs, embeddings of meshes and trees, packings of multiple copies of a graph, the complexity of finding good embeddings, and critical graphs which are minimal with respect to some property. In addition, we describe several open problems.

Keywords: hypercube computer, n-cube, embedding, dilation, expansion, cubical, packing, random graphs, critical graphs.

1 Introduction

Let Q_n denote an n-dimensional binary cube where the nodes of Q_n are all the binary n-tuples and two nodes are adjacent if and only if their corresponding n-tuples differ in exactly one position. (See Figure 1.) An n-dimensional hypercube computer, or n-cube, is a parallel computer with 2^n processors and network topology that of Q_n. That is, each node of Q_n represents a processor — a fairly powerful computer with its own local memory — and each edge of Q_n represents a direct communication link between the corresponding processors. A detailed account of one of the commercially available hypercubes, the N-CUBE, is given by Hayes et al. [17].

Many of the properties of the hypercube that make it a desirable general purpose parallel machine are a direct consequence of the graph-theoretic properties of Q_n. For example, the fact that Q_n can be defined recursively as the graph product

$$Q_n = Q_{n-1} \times K_2$$

[1] On leave from Southern Illinois University, Edwardsville, IL 62026

[2] Research supported by National Science Foundation grant DCR-8507851 and an Incentives for Excellence award from Digital Equipment Corporation.

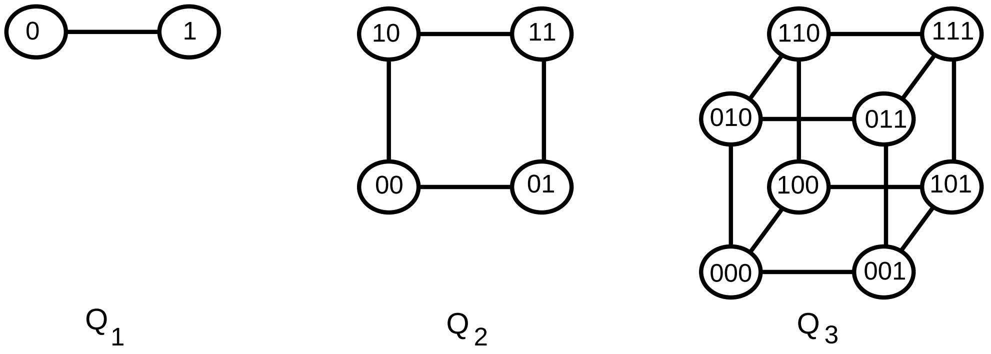

Figure 1: Some Hypercubes

suggests that a hypercube can support divide-and-conquer strategies very well. Certain algorithms such as Bitonic Sort and the FFT can be implemented efficiently on a hypercube network, with all communication occuring between pairs of adjacent nodes. The recursive definition also shows that a hypercube can be partitioned among multiple users, each receiving a subcube. The fact that Q_n is homogeneous (given any two nodes p and q there exists an automorphism σ of Q_n for which $\sigma(p) = q$), allows algorithms to be written assuming some node has a distinguished role and then rotated so that any other desired node assumes that role. The fact that Q_n has diameter n, a relatively small diameter for the number of nodes it possesses, implies that no single message between arbitrary processors need travel very many communication links. Broadcasting (one node sending the same information to all others) is a crucial operation for many applications such as gaussian elimination, and can be implemented in only n parallel communication steps via simple "recursive doubling". In addition to the above properties, the fact that Q_n is n-connected suggests that the network enjoys a high degree of fault-tolerance.

Members of several important classes of graphs including rectangular grid graphs and trees appear as subgraphs of Q_n. This suggests that the hypercube can simulate these networks with little or no overhead and that algorithms designed for these networks can be easily adapted to the hypercube. When the hypercube is used to simulate a network with graph G the nodes of G must be mapped to the nodes of Q_n, and, in order to keep communication overhead down, adjacent nodes of G should map to adjacent nodes of Q_n insofar as possible. In designing (or adapting) an algorithm that performs a task T on the hypercube network, T is modeled with a "task graph", G_T, in which the nodes represent subtasks and the edges represent communication requirements between the corresponding subtasks. Once again, the efficiency of the implementation depends strongly on the nature of the mapping into Q_n. To keep communication overhead low, the nodes of G_T must be mapped to the nodes of Q_n so that pairs of adjacent nodes of G_T map to pairs of adjacent nodes of Q_n.

An *embedding* of a graph $G = <V, E>$ into a graph $G' = <V', E'>$ is a one-to-one map ϕ of V into V' such that if $(u, v) \in E$ then $(\phi(u), \phi(v)) \in E'$ for all $(u, v) \in E$. A graph G is called *cubical* if, for some n, there is an embedding of G into Q_n. Cubical graphs form the central topic of Section 2.

Many graphs that emerge as task graphs are not cubical. Thus, we are motivated to study mappings of graphs with no adjacency requirement. We will call ϕ a *weak embedding* of G into G' provided only that ϕ is one-to-one. Now it is the case that every graph G with at most 2^n nodes can be weakly embedded into Q_n, but pairs of adjacent nodes of G may map to pairs of nodes which are connected but not adjacent. The practical considerations associated with this type of embedding give rise to the following notions. The *dilation* of ϕ, denoted $dil(\phi)$, is defined as

$$dil(\phi) = \max\{dist(\phi(u), \phi(v)) \,|\, (u, v) \in E\}$$

where $dist(a, b)$ denotes the Hamming distance between the binary n-tuples a and b. Dilation is a measure of the communication overhead induced by the map ϕ. The *expansion* of ϕ, denoted $ex(\phi)$, measures processor utilization and is defined as

$$ex(\phi) = |V'|/|V|.$$

Other measures of efficiency, such as the maximum number of edges of V mapped onto a single edge of V', will not be considered here.

In practice, one would like to have embeddings with dilation and expansion near one, but for many graphs this is an impossible situation. Often one can make trade-offs between the two, increasing one at the expense of the other. Section 3 contains a discussion of the relevant results concerning weak embeddings and the trade-offs that exist between dilation and expansion. In Section 4 some extensions are discussed, and several open problems are described.

Throughout, lg denotes $\log_2$.

2 Cubical Graphs

As defined in Section 1, a graph G is cubical if there is an embedding of G into Q_n for some n. If G is cubical, then the least positive integer n for which G can be embedded into Q_n is called the *cubical dimension* of G, denoted $cd(G)$. The star graph with $m + 1$ nodes, $K_{1,m}$, is clearly cubical and $cd(K_{1,m}) = m$. It is equally straightforward to see that a simple path with m nodes, P_m, is cubical, and the fact that $cd(P_m) = \lceil \lg m \rceil$ follows from the existence of a Hamiltonian path in Q_n for any n. On the other hand, note that neither the complete graph K_3 nor the complete bipartite graph $K_{2,3}$ is cubical.

In an early paper on isometric embeddings into Q_n, Firsov [5] showed that all trees are cubical, and also noted that all cubical graphs are bipartite. Later, Havel and Morávek [16] discovered necessary and sufficient conditions that a graph be cubical. (These conditions are given below.) Using this, Havel and Liebl [14, 15] deduced that trees, rectangular meshes, and hexagonal meshes are cubical, and they gave embeddings of these. They also proved that a cycle is cubical if and only if it is even. These results have been rediscovered numerous times.

The embeddings of rectangular meshes are quite simple and illustrative, as well as being useful in many applications. A d-dimensional mesh M of size $n_1 \times n_2 \times \ldots \times n_d$ has nodes $\{(a_1, \ldots, a_d) \,|\, 0 \le a_i < n_i \text{ for } 1 \le i \le d\}$, where an edge exists between two nodes if and only if their labels differ by one in one component, and are identical in all other components. To embed M into a hypercube, one utilizes binary Gray codes [7]. The most common Gray codes, the reflected binary ones, are recursively defined as follows: $\mathcal{G}_n$ is a bijection from $\{0, 1, \ldots, 2^n - 1\}$ onto $\{0, 1\}^n$, given by $\mathcal{G}_1(0) = 0$, $\mathcal{G}_1(1) = 1$, and

$$\mathcal{G}_n(x) \quad = \quad \begin{cases} 0\mathcal{G}_{n-1}(x) & 0 \le x \le 2^{n-1} - 1 \\ 1\mathcal{G}_{n-1}(2^n - 1 - x) & 2^{n-1} \le x \le 2^n - 1 \end{cases}$$

for $n \geq 2$. Let ϕ denote the mapping that associates the d-tuple $(a_1, \ldots, a_d)$ with the concatenation $\mathcal{G}_{r_1}(a_1) \cdot \ldots \cdot \mathcal{G}_{r_d}(a_d)$, where $r_i = \lceil \lg n_i \rceil$. It is straightforward to show that indeed ϕ is an embedding of M into Q_r, where $r = \sum_i r_i$. Therefore $cd(M) \leq r$, and by using the labeling conditions given below one can show that $cd(M) = r$.

2.1 Deciding if a Graph is Cubical

Note that a graph is cubical if and only if all of its connected components are cubical. The work of Havel and Morávek [16] can be rephrased slightly to show that a connected graph G can be embedded into Q_n if and only if it is possible to label the edges of G with the integers $\{1, \ldots, n\}$ such that

1. Edges incident with a common node are of different labels;

2. In each path of G there is some label that appears an odd number of times; and

3. In each cycle of G no label appears an odd number of times.

Each such labeling gives rise to a (not necessarily unique) embedding in which the label of an edge is the dimension along which its endpoints differ.

Using these embedding conditions, it is straightforward to show that embeddings of crossproducts of nonempty connected graphs must be crossproducts (concatenations) of embeddings of the factors. This implies that connected nonempty graphs G_1 and G_2 are cubical if and only if $G_1 \times G_2$ is cubical, and that

$$cd(G_1 \times G_2) = cd(G_1) + cd(G_2).$$

The conditions also show that a tree of n nodes can be embedded into Q_{n-1} by using a labeling which is a bijection between the edges and $\{1, \ldots, n-1\}$. Using this, it is easy to show that a graph is cubical if and only if its biconnected components are, since the biconnected components form a forest.

Graham [9] has given a condition which can be used to prove that a graph is not cubical. A graph G is *decomposable* if it has a minimal cutset C of edges such that no two edges in C have a common node, and the removal of C disconnects G. G is *completely decomposable* if every subgraph of more than one node is decomposable. Graham proved that if $w(k)$ denotes the number of 1's in the binary expansion of k, and $W(k) = \sum_{j=1}^{k} w(j)$, then every completely decomposable graph with n nodes and e edges satisfies $e \leq W(n-1)$. Cycles of odd length greater than 3 show that not all completely decomposable graphs are cubical, but all hypercubes are completely decomposable since, for any connected subgraph, once can pick any edge in the subgraph and choose as a cutset that edge and all edges parallel to it. Therefore a cubical graph G with n nodes is completely decomposable and has no more than $W(n-1)$ edges. Further, this bound is best possible, as is shown by the induced subgraph of Q_m consisting of nodes $\{0, \ldots, n-1\}$, $n \leq 2^m$, where we have made the natural identification between binary m-tuples and integers. A direct divide-and-conquer approach can also be used to prove the slightly weaker fact that G has at most $0.5 \cdot n \lg n$ edges.

Garey and Graham [6], and earlier Havel and Morávek [16], considered the problem of finding *cube-critical* graphs, that is, graphs that are not cubical though all of their proper subgraphs are cubical. Cycles of odd length are cube-critical, as is a diamond with one pair of opposite nodes connected by an extra path of length two. Garey and Graham, and Gorbatov and Kazansky [8], have given procedures for constucting arbitrarily large cube-critical graphs from other cube-critical graphs, and Havel [12] has constructed arbitrarily large cube-critical graphs using meshes with extra edges.

Deciding if a graph is cubical, or deciding if a noncubical graph is cube-critical, can be quite difficult. Afrati, Papidimitriou, and Papageorgiou [1] and Krumme, Venkataraman, and Cybenko [20], showed that the problem of deciding whether an arbitrary graph is cubical is NP-complete. The proof of NP-completeness given in [1] shows that the problem remains NP-complete even if the problem is restricted so that the maximum degree of any node is 4. If the maximal degree of any node is 2 then the problem can be decided in linear time, and it is apparently an open question whether the problem is NP-complete when the maximum degree is 3. However, the authors have shown that deciding if a graph is cubical, or deciding if a noncubical graph is cube-critical, can be completed in constant expected time for random graphs in which edges between nodes are included with a fixed probability.

2.2 Cubical Dimension

If G is a cubical graph with connected components $C_1, \ldots, C_k$, then simple packings of subcubes, coupled with an obvious lower bound, shows that

$$\max_i cd(C_i) \;\leq\; cd(G) \;\leq\; \left\lceil \lg \sum_i 2^{cd(C_i)} \right\rceil .$$

Unfortunately, upper and lower bounds for the cubical dimension of an arbitrary connected cubical graph are quite far apart, with the best bounds for a cubical graph G with n nodes being

$$\lceil \lg n \rceil \;\leq\; cd(G) \;\leq\; n-1 .$$

The lower bound is obtained from a simple count of the nodes, and is attained by a path P_n. The upper bound was first proven in Garey and Graham [6], and is attained by the star graph $K_{1,n-1}$.

For more restricted cubical graphs slightly better bounds have been determined. In [1], Afrati, Papadimitriou, and Papageorgiou proved that if G is a biconnected cubical graph of n nodes, then $cd(G) \leq n/2$, and they observed that the graph shown in Figure 2 shows that this bound is the best possible. Using this result, Stout has shown that if G is a cubical graph of n nodes where each node has degree at least 2, then $cd(G) \leq 2(n-1)/3$. The graph in Figure 3 shows that this bound is the best possible. There should be analogues of these results for k-connected graphs, and for graphs in which each node has degree at least d, for arbitrary integers $k, d \geq 3$.

While trees are easily shown to be cubical, determining their cubical dimension has proven to be a difficult problem. Havel and Liebl [15] showed that the cubical dimension of the complete binary tree T_n with height n and $2^{n+1}-1$ nodes is a most $n+2$ for $n \geq 2$, and Nebesky [21] later proved that $cd(T_n) = n+2$ when $n \geq 2$. Other bounds on cubical dimensions of specific trees appear in Havel and Liebl [14, 15] and Wagner [24]. Afrati, Papadimitriou, and Papageorgiou [1] gave a polynomial time algorithm which embeds a tree into a cube with dimension at most the square of the cubical dimension of the tree, and they conjectured that the problem of calculating the cubical dimension of a tree is NP-complete.

For general cubical graphs, the problem of calculating the cubical dimension is NP-complete. In fact, Krumme found a family of graphs where each member with 2^d nodes embeds into Q_{d+1}, but it is NP-complete to decide if it embeds into Q_d.

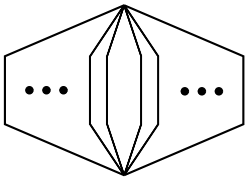

Figure 2: A bipartite graph with large cubical dimension

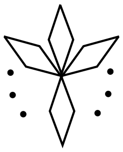

Figure 3: A graph with minimal degree 2 and large cubical dimension

Call a cubical graph G *dimension-critical* if the deletion of any edge reduces the cubical dimension. Stars with more than 3 nodes are dimension-critical, and each cubical graph G of n nodes with $cd(G) > \lceil \lg n \rceil$ must contain a dimension-critical subgraph H with n nodes and $cd(G) = cd(H)$. Although this is a natural analogue of cube-critical graphs, we know of no prior work on this class of graphs.

2.3 Packings

In some applications it is desirable to embed multiple copies of a given cubical graph G into Q_n so that the embeddings are edge-disjoint or node-disjoint, with the goal of using as many copies as possible. For example, on some machines processors have little memory, and some programs need more memory at each node. In such a setting it may be necessary to have $p+1$ processors working together, with one master and p slaves supplying information. To minimize communication time, the processors should be arranged as the star $K_{1,p}$, with the master in the middle. Further, to utilize as many processors as possible, one wants to pack in as many stars as possible. For some communication problems edge-disjoint embeddings are desired instead, where different information travels in different copies of a graph and the object is to simultaneously use as many communication links as possible [18, 23].

We will use $pac_0(G, Q_n)$ to denote the maximum number of node-disjoint copies of G that can be embedded in Q_n, and $pac_1(G, Q_n)$ to denote the maximum number of edge-disjoint copies of G that can be embedded in Q_n. It is easy to show that

$$pac_0(G, Q_n) \geq 2^{n-m} pac_0(G, Q_m)$$

for $n \geq m$, and hence if node-disjoint copies of G cover Q_m then node-disjoint copies of G cover Q_n for all $n \geq m$. It is also easy to show that

$$pac_1(G, Q_n) \geq 2^{n-m} pac_1(G, Q_m)$$

for $n \geq m$, but since the number of edges in Q_n is $n2^{n-1}$ it is not necessarily true that if edge-disjoint copies of G cover Q_m then edge-disjoint copies of G cover Q_{m+1}. For example, $pac_1(Q_m, Q_n) = \lfloor n/m \rfloor 2^{n-m}$, so edge-disjoint copies of Q_m cover Q_n if and only if n is an integral multiple of m. One can show that

$$pac_1(G, Q_{an+bm}) \geq a2^{(a-1)n+bm} pac_1(G, Q_n) + b2^{an+(b-1)m} pac_1(G, Q_m)$$

for arbitrary nonnegative integers a and b. This shows that if edge-disjoint copies of G cover Q_n and Q_m for relatively prime n and m then edge-disjoint copies of G cover all cubes of sufficiently high dimension.

The theory of error-correcting codes shows that node-disjoint copies of the star $K_{1,n}$ cover Q_n if and only if $n+1$ is an integral power of 2. Using Hamiltonian paths and a node counting argument, is easy to see that node-disjoint copies of the path P_n cover Q_m if and only if n divides 2^m, and Eulerian paths show that that edge-disjoint copies of P_n cover Q_m whenever m is even and $n-1$ divides $m2^{m-1}$. Fink [4] and Stout [22] independently showed that if T is a tree with n edges, then edge-disjoint copies of T cover Q_n. Stout also showed that if G is a cubical graph with n nodes, then $\lim_{m\to\infty} pac_0(G, Q_m)n/2^m = 1$.

3 Weak Embeddings

This section is concerned with *weak embeddings* into hypercubes, that is, maps which are not required to preserve adjacency. Throughout this section we will use *embedding* to mean weak embedding. As was mentioned in the introduction, considerations of communications overhead and processor utilization lead one to consider the dilation and expansion of embeddings. Any graph has an embedding with expansion less than 2, and an embedding with dilation no greater than 2, though simultaneously minimizing dilation and expansion is usually quite difficult. An embedding of minimal expansion can be attained by mapping a graph with n nodes into $Q_{\lceil \lg n \rceil}$ via any one-to-one mapping, though the dilation may be as large as $\lceil \lg n \rceil$. To achieve dilation no greater than 2, note that K_n can be mapped into Q_{n-1} by mapping one node of K_n onto node 0 of Q_{n-1} and mapping all other nodes onto neighbors of node 0. This shows that K_n, and hence any graph of n nodes, can be mapped into Q_{n-1} with dilation 2, though the expansion is $2^{n-1}/n$.

Some properties of cubical embeddings carry over to weak embeddings with minor modifications. For example, if ϕ_1 is an embedding of G_1 into Q_{n_1} and ϕ_2 is an embedding of G_2 into Q_{n_2}, then $\phi_1 \times \phi_2$ maps $G_1 \times G_2$ into $Q_{n_1+n_2}$, and

$$\begin{aligned} dil(\phi_1 \times \phi_2) &= \max\{dil(\phi_1), dil(\phi_2)\} \\ ex(\phi_1 \times \phi_2) &= ex(\phi_1) * ex(\phi_2)\ . \end{aligned}$$

However, not all embeddings of crossproduct graphs are crossproducts of embeddings of the factors, and as is discussed below, Greenberg [10] has used this to show that all 2-dimensional meshes can be embedded with dilation no greater than 3, while keeping the expansion less than 2.

Suppose G is a graph with connected components $C_1, \ldots, C_k$, and ϕ_i is an embedding of C_i into Q_{n_i} for $1 \le i \le k$. By using packings of subcubes, there is an embedding ϕ of G into Q_n, where $n = \lceil \lg \sum_i 2^{n_i} \rceil$, with $dil(\phi) = \max_i dil(\phi_i)$. Notice that

$$\min_i ex(\phi_i) \le ex(\phi) < 2 \max_i ex(\phi_i).$$

If G has biconnected components $C_1, \ldots, C_k$, and ϕ_i is an embedding of C_i for $1 \le i \le k$, then there is an embedding ϕ of G such that $dil(\phi) = \max_i dil(\phi_i)$.

3.1 Dilation/Expansion Trade-offs

Among all possible embeddings of a graph, those with small dilation and expansion are the most desirable. Unfortunately, it is often impossible to simultaneously minimize both of these, in which case there is often a range of trade-offs possible. To help measure this, let G be a graph of n nodes, and define the closely related functions ϵ and δ by:

$$\begin{aligned}\epsilon(G,d) &= \min\{ex(\phi)\,|\,\phi \text{ embeds } G, dil(\phi)=d\},\\ \delta(G,m) &= \min\{dil(\phi)\,|\,\phi \text{ embeds } G \text{ into } Q_m\}.\end{aligned}$$

For example, the star graph $K_{1,n}$ has $\delta(K_{1,n},m) = d$, where $2^m \geq n+1$ and d is the least positive integer for which

$$n \leq \sum_{i=1}^{d} \binom{m}{i}.$$

An embedding which achieves this maps the center of the star to node 0, and all other nodes into nodes of Q_m within distance d of node 0. If $n+1$ is a power of 2, then by varying m one observes that

$$\begin{aligned}\epsilon(K_{1,n},1) &= 2^n/(n+1)\\ \epsilon(K_{1,n},2) &= 2^{\Theta(\sqrt{n})}/(n+1)\\ \epsilon(K_{1,n},3) &= 2^{\Theta(\sqrt[3]{n})}/(n+1)\\ &\vdots\\ \epsilon(K_{1,n},\lg(n+1)) &= 1\,.\end{aligned}$$

As was noted in Section 2, the full binary tree T_n of height n with $2^{n+1}-1$ nodes has cubical dimension $n+2$ for $n \geq 2$, and therefore any embedding with dilation 1 must have expansion greater than 2. However, as was noted by Nebesky [21], and rediscovered in [2], T_n can be embedded into Q_{n+1} with only one edge undergoing dilation 2. For such an embedding the expansion is as small as possible for a graph with $2^{n+1}-1$ nodes.

Several other authors have considered embeddings of binary trees with small dilation or small expansion. Bhatt and Ipsen [3] showed that for an arbitrary binary tree T with n nodes, $\epsilon(T,1) \leq \mathrm{O}(n^{0.71})$. They also showed that $\delta(T, 1+\lceil\lg n\rceil) \leq \lg\lg n + 9$. This result was superceded by that of Bhatt et al. [2], who gave a polynomial time algorithm which shows that $\delta(T,\lceil\lg n\rceil) \leq 10$. Their methods extend to graphs with O(1)-separators, such as trees of bounded degree and outerplanar graphs of bounded degree. For binary trees, the best uniform bound on $\delta(T,\lceil\lg n\rceil)$ is unknown, as is the best bound on $\epsilon(T,1)$, as well as all of the trade-offs inbetween.

As was shown in Section 2, a mesh of size $n_1 \times \ldots \times n_d$ can be embedded into Q_r with dilation 1, where $r = \sum_i \lceil\lg n_i\rceil$, and the embedding conditions of Havel and Morávek [16] can be used to show that this is the smallest cube for which a dilation 1 embedding is possible. If each factor is a power of 2 then the expansion is also 1, but otherwise the expansion can increase by almost a factor of 2 for each dimension that is not a power of 2. Thus, for example, a 2-dimensional mesh may require expansion arbitrarily close to 4 in order to achieve dilation 1. Greenberg [10] has shown that any 2-dimensional mesh can be embedded with dilation no greater than 2 and expansion no greater than 3, and

Ho and Johnson [19] have shown that many 2-dimensional meshes can be embedded with dilation and expansion no greater than 2. It is an open question whether all 2-dimensional meshes can be embedded with dilation and expansion both no greater than 2, and the best bounds for meshes of higher dimensions are also unknown.

3.2 Dilation with Minimal Expansion

Suppose G has n nodes. Since processor utilization is often the most critical parameter of performance, in many applications only embeddings into $Q_{\lceil \lg n \rceil}$ are possible, and hence determining $\delta(G, \lceil \lg n \rceil)$ is particularly important. Unfortunately it is also quite difficult, and the results of Krumme mentioned in Section 2 show that it is NP-complete to decide if $\delta(G, \lceil \lg n \rceil) = 1$.

It is easy to show that any embedding of G into $Q_{\lceil \lg n \rceil}$ must map onto a pair of antipodal nodes of $Q_{\lceil \lg n \rceil}$. Suppose an embedding maps nodes p and q of G onto antipodal nodes. A path of minimal length in G between p and q has at most diameter(G) edges, and is mapped onto a path in the hypercube with length at least $\lceil \lg n \rceil$. Therefore

$$\delta(G, \lceil \lg n \rceil) \geq \lceil \lg n \rceil / \mathrm{diameter}(G).$$

One natural problem is to determine when $\delta(G, \lceil \lg n \rceil)$ equals $\lceil \lg n \rceil$. This can be answered in terms of the complement graph $\bar{G}$. If n is a power of 2, then $\delta(G, \lg n) < \lg n$ if and only if it is possible to partition the nodes into $n/2$ pairs, where no pair is adjacent in G. Mapping each pair to antipodal nodes in the hypercube shows that $\delta(G, \lg n) < \lg n$. Moreover, G having such a partition is equivalent to $\bar{G}$ having a maximal matching. When n is not a power of 2 some nodes can be mapped to hypercube nodes where no other node is mapped to the antipodal hypercube node, and it is then easily seen that $\delta(G, \lceil \lg n \rceil) < \lceil \lg n \rceil$ if and only if $\bar{G}$ has a matching with at least $n - 2^{\lceil \lg n \rceil - 1}$ pairs.

4 Final Remarks

We have surveyed some results concerning embeddings and weak embeddings of graphs into hypercubes. Several open questions were listed in previous sections, and many more immediately suggest themselves. For example, one could extend the notions of cube-critical and dimension-critical to say a graph G is $\epsilon(d)$-*critical* if $\epsilon(G, d) > \epsilon(H, d)$ for every proper subgraph H, and G is $\delta(m)$-*critical* if $\delta(G, m) > \delta(H, m)$ for every proper subgraph H. Each cube-critical graph is $\delta(m)$-critical for some m, and each dimension-critical graph is $\epsilon(1)$-critical. If $\delta(G, m) \geq 2$ then G contains a $\delta(m)$-critical subgraph, and if $\epsilon(G, d) \geq 2$ then G contains a $\epsilon(d)$-critical subgraph, but very little is known about ϵ- or δ-critical graphs.

For any class of graphs, one could attempt to analyze the expansion/dilation trade-offs, as well as consider the complexity of determining optimal or nearly optimal embeddings. A particularly interesting class of graphs are random graphs where edges between nodes are included with some probability p, where p may be a nonincreasing function of the number of nodes. The authors have begun an investigation of expected expansion/dilation trade-offs for random graphs, and of algorithms to find nearly optimal embeddings in small expected time. Other interesting classes include planar graphs of bounded degree, arbitrary graphs of bounded degree, and Δ-neighbor graphs consisting of n points in the plane (or 3-space), where there is an edge between two points if and only if they are within Δ of each other.

Only one-to-one mappings have been considered here. Other useful possibilities include one-to-many mappings in which nodes are mapped to subcubes, and many-to-one mappings. The latter are needed in the common situation where the task graph has more nodes than the target hypercube computer. In such a setting one is still concerned with dilation, as well as various measures of load-balancing for the hypercube nodes and edges, perhaps starting with a task graph having weighted nodes and edges. The use of multiple objective functions makes such problems intractible, so there has tended to be an emphasis on mapping heuristics, or on algorithms applicable to a very narrow range of graphs. This is the most important class of embeddings from a practical standpoint, but so far it has had the least exact analysis.

References

[1] F. Afrati, C. Papadimitriou, and G. Papageorgiou. The complexity of cubical graphs. *Inform. and Control* 66 (1985) 53-60.

[2] S. Bhatt, F. Chung, T. Leighton, and A. Rosenberg. Optimal simulations of tree machines. *Proc. Found. of Comp. Sc.* (1986) 274-282.

[3] S. Bhatt and I. Ipsen. How to embed trees in hypercubes. Tech. Rept. YALEU/DC/RR-443 Yale Univ. (1985).

[4] J. Fink. On the isomorphic decomposition of n-cubes into trees. Presented at *11th Midwest Graph Theory Conf.* (1987) Eastern Mich. U.

[5] V. Firsov. On isometric embeddings of a graph into a boolean cube. *Cybernetics* 1 (1965) 112-113.

[6] M. Garey and R. Graham. On cubical graphs. *J. Comb. Th.*(B) 18 (1975) 84-95.

[7] E.N. Gilbert. Gray codes and paths on the n-cube. *Bell System Tech. J.* May (1958) 815-826.

[8] V. Gorbatov and A. Kazanskiy. Characterization of graphs embedded in n-cubes. *Engin. Cybernetics, IEEE* 20 No. 2 (1983) 96-102.

[9] R.L. Graham. On primitive graphs and optimal vertex assignments. *Ann. N. Y. Acad. Sci.* 175 (1970) 170-186.

[10] D.S. Greenberg. Optimal embeddings of meshes in hypercubes. Tech. Rept. YALEU/CSD/RR-535, Yale U. (1987), in preparation.

[11] F. Harary, J.P. Hayes, and H. Wu. A survey of the theory of hypercube graphs. *Comput. Math. Appl.* (to appear)

[12] I. Havel. Embedding grids with diagonals into cubes. *Proc. Sympos. on Combin. Anal.* Ed. by M. Borowiecki, Z. Skupien, and L. Szamkolowicz. Zielona Gora. (1976) 89-103.

[13] I. Havel. Embedding graphs in undirected and directed cubes. *Spr. Verl. Lect. Notes* 1018 (1983) 60-68.

[14] I. Havel and P. Liebl. Embedding the dichotomic tree into the n-cube. *Casopis Pest. Mat.* 97 (1972) 201-205.

[15] I. Havel and P. Liebl. Embedding the polytomic tree into the n-cube. *Casopis Pest. Mat.* 98 (1973) 307-314.

[16] I. Havel and J. Morávek. B-valuations of graphs. *Czech. Math. J.* 22 (1972) 338-351.

[17] J.P. Hayes, T.N. Mudge, Q.F. Stout, S. Colley, and J. Palmer. A microprocessor-based hypercube supercomputer. *IEEE Micro* 6 (1986) 6-17.

[18] C. Ho and S.L. Johnsson. Spanning graphs for optimal broadcasting and personalized communication in hypercubes. Tech. Rept. YALEU/CSD/RR-500, Yale U. (1986).

[19] C. Ho and S.L. Johnsson. On the embedding of arbitrary meshes in boolean cubes with expansion two dilation two. *Proc. Int'l. Conf. Parallel Proc.* (1987) 188-191.

[20] D. Krumme, K. Venkataraman, and G. Cybenko. Hypercube embedding is NP-complete. *Hypercube Multiprocessors 1987.* Ed. by M. Heath. 148-157.

[21] L. Nebesky. On cubes and dichotomic trees. *Casopis Pest. Mat.* 99 (1974) 164-167.

[22] Q.F. Stout. Packings into hypercubes. In preparation.

[23] Q.F. Stout and B. Wagar. Passing messages in link-bound hypercubes. *Hypercube Multiprocessors 1987*, Ed. by M. Heath. 251-257.

[24] A. Wagner. Embedding binary trees in a hypercube. *Univ. of Toronto Tech. Rept.* CSRI-175 (1985)