Yale University
Department of Computer Science

Embedding Meshes in Boolean Cubes by Graph Decomposition

Ching-Tien Ho and S. Lennart Johnsson

YALEU/DCS/TR-689
March 1989

This work was supported by the Office of Naval Research under contract N00014-86-K-0310. Approved for public release: distribution is unlimited.

Embedding Meshes in Boolean Cubes by Graph Decomposition

Ching-Tien Ho and S. Lennart Johnsson[1]
Department of Computer Science
Yale University
New Haven, CT 06520
Ho@cs.yale.edu, Johnsson@cs.yale.edu, Johnsson@think.com

Abstract. This paper explores the embeddings of multidimensional meshes into minimal Boolean cubes by graph decomposition. The dilation and the congestion of the product graph $(G_1 \times G_2) \rightarrow (H_1 \times H_2)$ is maximum of the two embeddings $G_1 \rightarrow H_1$ and $G_2 \rightarrow H_2$. The graph decomposition technique can be used to improve the average dilation and average congestion for existing mappings. One property used frequently in mesh embedding by graph decomposition is that a $\ell_1 \times \ell_2 \times \cdots \times \ell_k$ mesh is a subgraph of the product graph of the two meshes $\ell'_1 \times \ell'_2 \times \cdots \times \ell'_k$ and $\ell''_1 \times \ell''_2 \times \cdots \times \ell''_k$, if $\ell_i \leq \ell'_i \ell''_i$ for all $1 \leq i \leq k$. The graph decomposition technique combined with some particular two-dimensional embeddings allows for minimal expansion, dilation two, congestion two embeddings of about 87% of all two-dimensional meshes, asymptotically. With the recent result in [4], the graph decomposition technique, and some three-dimensional mappings presented in this paper, more than 96% of all three-dimensional meshes contained in a $512 \times 512 \times 512$ mesh can be embedded in a minimal Boolean cube with dilation two. The graph decomposition technique is also used to generalize the mesh embeddings to include wrap-around with an increase in the dilation by at most 1, compared to a mesh without wrap-around. The expansion is preserved for the majority of meshes, if a wrap-around feature is added to the mesh.

1 Introduction

Many linear algebra computations can be performed effectively on processor networks configured as two-dimensional meshes, with or without wrap-around. Processor networks configured as two- or higher dimensional meshes are also effective for the solution of partial differential equations whenever regular grids are appropriate. Four-dimensional regular grids are used in quantum electrodynamics and quantum chromodynamics calculations. A multiprocessor architecture has to satisfy many needs. An interconnection network that can emulate many communication patterns, or that provide multiple paths between arbitrary pairs of processors, may be chosen as the physical interconnection network. Boolean cube networks are versatile networks in that they can emulate many other networks with little or no slowdown. The emulations are often not unique, and in some cases several emulations can be performed concurrently without conflict. The use of the available bandwidth is thereby maximized. For instance, multiple spanning tree embeddings are described in [13, 16].

In this paper we focus on the embedding of meshes of an arbitrary number of dimensions, and shapes, in Boolean cubes. Encoding the indices of each axis in a Gray code, is effective

[1]Also with Dept. of Electrical Engineering, Yale Univ., and Thinking Machines Corp., Cambridge, MA 02142.

if the number of nodes along each axis is a power of two [17]. The most often used Gray codes are *binary-reflected* Gray codes [23]. However, if the length of the axis is not a power of two the Gray code embedding forces the number of processors to be allocated to an axis to be a power of two, which for meshes of high dimension may yield a very poor processor utilization. The *expansion* is defined as the ratio between the nodes required for the embedding, and the nodes of the mesh being embedded. The inverse of the expansion is a measure of the processor utilization. The expansion may be as high as $\approx 2^k$ for a k-dimensional mesh [12]. The advantage of the Gray code embedding is that nodes that are nearest neighbors in the mesh, are also nearest neighbors after the embedding in the Boolean cube. If this requirement is relaxed, then the expansion may be reduced. The *dilation* of an edge is the length of the path into which the edge is mapped when embedded in the Boolean cube. Recently, Chan has shown that any three-dimensional mesh can be embedded in a Boolean cube with dilation at most seven and minimal expansion [6], and that any k-dimensional mesh for $k > 3$ can be embedded with a dilation of at most $4k + 1$ [5]. For two-dimensional meshes dilation two, minimal expansion embeddings have been known for some time for most meshes [7, 14]. Dilation two, minimal expansion embeddings for all two-dimensional meshes were recently found by Chan [4].

In [14] we used a graph decomposition, or factoring technique to obtain dilation-two minimal-expansion embeddings for most two-dimensional meshes. Here we generalize this technique to meshes of arbitrary dimensions, and show that this technique for minimal-expansion embeddings yields a lower dilation than the bounds given by Chan for many meshes. Dilation and expansion are two measures of the characteristics of an embedding. The dilation is a measure of the minimum distance (time) for a message to move from one node to an adjacent node in the mesh, when embedded in the Boolean cube. However, several mesh edges may be mapped to paths that use the same Boolean cube edge. The *congestion* is a measure of the number of mesh edges using the same Boolean cube edge. Other important characteristics with respect to the efficiency of the embedding are the number of communication channels, the *active-degree*, and the number of messages, the *node-congestion*, a processor may need to service concurrently to fully support the mesh emulation. We consider these characteristics of the embedding, and show that embedding meshes by applying the graph decomposition technique may improve upon the dilation, except in the two-dimensional case, the average dilation, the congestion, the node-congestion, or the active-degree, compared to embedding the undecomposed mesh.

Most of the results for the embedding of meshes of arbitrary dimensions into Boolean cubes apply to meshes without wrap-around. However, in many linear algebra computations cyclic shifts are important operations, and in the solution of partial differential equations periodic boundary conditions play an important role. We extend the results for meshes without wrap-around to meshes with wrap-around.

The graph decomposition technique is based upon the idea that the original mesh can be decomposed into two or more graphs for which good embeddings are known. The decomposition is not necessarily unique. Moreover, for the embedding, we also consider the embedding of meshes obtained by extending the number of nodes of the original mesh along any axis, as long as the expansion is minimal with respect to the original mesh. These extended meshes may allow a decomposition into smaller meshes for which effective embeddings are known. The main theorem in this paper is, if an $\ell_1 \times \ell_2 \times \cdots \ell_k$ mesh M can be embedded into an n_1-cube with dilation d_1 and an $\ell'_1 \times \ell'_2 \times \cdots \times \ell'_k$ mesh M' can be embedded into an n_2-cube with dilation d_2, then the $\ell_1\ell'_1 \times \ell_2\ell'_2 \times \cdots \times \ell_k\ell'_k$ mesh $\tilde{M}$ can be

if the number of nodes along each axis is a power of two [17]. The most often used Gray codes are *binary-reflected* Gray codes [23]. However, if the length of the axis is not a power of two the Gray code embedding forces the number of processors to be allocated to an axis to be a power of two, which for meshes of high dimension may yield a very poor processor utilization. The *expansion* is defined as the ratio between the nodes required for the embedding, and the nodes of the mesh being embedded. The inverse of the expansion is a measure of the processor utilization. The expansion may be as high as $\approx 2^k$ for a k-dimensional mesh [12]. The advantage of the Gray code embedding is that nodes that are nearest neighbors in the mesh, are also nearest neighbors after the embedding in the Boolean cube. If this requirement is relaxed, then the expansion may be reduced. The *dilation* of an edge is the length of the path into which the edge is mapped when embedded in the Boolean cube. Recently, Chan has shown that any three-dimensional mesh can be embedded in a Boolean cube with dilation at most seven and minimal expansion [6], and that any k-dimensional mesh for $k > 3$ can be embedded with a dilation of at most $4k + 1$ [5]. For two-dimensional meshes dilation two, minimal expansion embeddings have been known for some time for most meshes [7, 14]. Dilation two, minimal expansion embeddings for all two-dimensional meshes were recently found by Chan [4].

In [14] we used a graph decomposition, or factoring technique to obtain dilation-two minimal-expansion embeddings for most two-dimensional meshes. Here we generalize this technique to meshes of arbitrary dimensions, and show that this technique for minimal-expansion embeddings yields a lower dilation than the bounds given by Chan for many meshes. Dilation and expansion are two measures of the characteristics of an embedding. The dilation is a measure of the minimum distance (time) for a message to move from one node to an adjacent node in the mesh, when embedded in the Boolean cube. However, several mesh edges may be mapped to paths that use the same Boolean cube edge. The *congestion* is a measure of the number of mesh edges using the same Boolean cube edge. Other important characteristics with respect to the efficiency of the embedding are the number of communication channels, the *active-degree*, and the number of messages, the *node-congestion*, a processor may need to service concurrently to fully support the mesh emulation. We consider these characteristics of the embedding, and show that embedding meshes by applying the graph decomposition technique may improve upon the dilation, except in the two-dimensional case, the average dilation, the congestion, the node-congestion, or the active-degree, compared to embedding the undecomposed mesh.

Most of the results for the embedding of meshes of arbitrary dimensions into Boolean cubes apply to meshes without wrap-around. However, in many linear algebra computations cyclic shifts are important operations, and in the solution of partial differential equations periodic boundary conditions play an important role. We extend the results for meshes without wrap-around to meshes with wrap-around.

The graph decomposition technique is based upon the idea that the original mesh can be decomposed into two or more graphs for which good embeddings are known. The decomposition is not necessarily unique. Moreover, for the embedding, we also consider the embedding of meshes obtained by extending the number of nodes of the original mesh along any axis, as long as the expansion is minimal with respect to the original mesh. These extended meshes may allow a decomposition into smaller meshes for which effective embeddings are known. The main theorem in this paper is, if an $\ell_1 \times \ell_2 \times \cdots \ell_k$ mesh M can be embedded into an n_1-cube with dilation d_1 and an $\ell'_1 \times \ell'_2 \times \cdots \times \ell'_k$ mesh M' can be embedded into an n_2-cube with dilation d_2, then the $\ell_1\ell'_1 \times \ell_2\ell'_2 \times \cdots \times \ell_k\ell'_k$ mesh $\tilde{M}$ can be

embedded into an $(n_1 + n_2)$-cube with dilation $\max(d_1, d_2)$. The mesh $\tilde{M}$ is decomposable into the two meshes M and M' by being a subgraph of the product graph $M \times M'$ (shown later). In decomposing a graph it is represented as the product of graphs each of which has fewer nodes. The number of the axes increases, and the number of nodes along an axis decreases. If the number of nodes along an axis is a power of two, then the binary-reflected Gray code is used for the embedding of the nodes along that axis. If the number of nodes along an axis is not a power of two, but is decomposable into some power of two times an integer, then a graph decomposition may be made, increasing the number of the axes by one. For instance, a $12 \times 16 \times 20$ mesh can be decomposed into a $3 \times 4 \times 16 \times 5 \times 4$ mesh, which can be embedded as product of a 3×5 mesh and a $4 \times 16 \times 4$ mesh. Another alternative is to extend the mesh slightly, either to make it decomposable, or to a decomposition with a better embedding. For instance, a $3 \times 24 \times 3$ mesh can be extended and decomposed into a 3×5 mesh and a 5×3 mesh. The graph decomposition technique may also make use of the reshaping of subgraphs. Reshaping preserves the number of axes, but changes the number of nodes along an axis. The reshaped mesh has at least as many nodes as the original mesh.

The outline of the paper is as follows. In the next section we define the graph embedding concepts used for the characterization of the embeddings, and give some basic results. Then, we review some techniques used for reshaping two-dimensional meshes, such as the break-and-fold technique [20], step embedding [1], modified step-embedding, and line compression [1]. In Section 4, we give a few direct embeddings for two- and three-dimensional meshes with dilation two and minimal expansion. Section 5 presents some results for graph decomposition in combination with the reshaping and direct embedding techniques. In applying the graph decomposition technique together with the reshaping and direct embedding techniques, we have attained dilation two and minimal expansion embeddings of 96% of all three-dimensional meshes contained in, or equal in size to a $512 \times 512 \times 512$ mesh. Section 6 extends the results to meshes with wrap-around. We conclude by a few remarks.

2 Preliminaries

For a graph G let $\mathcal{V}_G$ be its set of vertices, and $\mathcal{E}_G$ its set of edges. Let $|\mathcal{S}|$ denote the cardinality of a set $\mathcal{S}$, and $\lceil x \rceil_2$ and $\lfloor x \rfloor_2$ denote $2^{\lceil \log_2 x \rceil}$ and $2^{\lfloor \log_2 x \rfloor}$, respectively.

The *embedding function* φ maps each vertex in the *guest graph* G into a unique vertex in the *host graph* H. The *expansion* ε of the mapping is $|\mathcal{V}_H|/|\mathcal{V}_G|$. The *relative expansion* for embedding a graph G into a hypercube H is $|\mathcal{V}_H|/\lceil |\mathcal{V}_G| \rceil_2$. Under the mapping function $\varphi : G \rightarrow H$, node $i \in \mathcal{V}_G$ is mapped to node $\varphi(i) \in \mathcal{V}_H$, and edge $e = (i, j) \in \mathcal{E}_G$ is mapped to a path $\varphi(e)$ consisting of the set of edges $\mathcal{E}_{\varphi(e)} = \{(\varphi(i), v_1), (v_1, v_2), \ldots, (v_{p-1}, \varphi(j))\} \subseteq \mathcal{E}_H$. The path $\varphi(e)$ has the node set $\mathcal{V}_{\varphi(e)} = \{\varphi(i), v_1, v_2, \ldots, v_{p-1}, \varphi(j)\}$. Let $dist(i, j)$ be the shortest path between nodes i and j in the considered graph. The *dilation* of the mapping φ is $\max(dist(\varphi(i), \varphi(j))$, for all $(i, j) \in \mathcal{E}_G$. The *dilation of an edge* $(i, j) \in \mathcal{E}_G$ is $dist(\varphi(i), \varphi(j))$, and the *average dilation* of the mapping φ is

$$\frac{1}{|\mathcal{E}_G|} \sum_{(i,j) \in \mathcal{E}_G} \{dist(\varphi(i), \varphi(j))\}.$$

The *congestion of an edge* $e' \in \mathcal{E}_H$ is $\sum_{e \in \mathcal{E}_G} |\{e'\} \cap \mathcal{E}_{\varphi(e)}|$, and the *congestion* of the embed-

ding is

$$\max_{e' \in \mathcal{E}_H} \left\{ \sum_{e \in \mathcal{E}_G} \left| \{e'\} \cap \mathcal{E}_{\varphi(e)} \right| \right\}.$$

The *average congestion* of the embedding is similarly defined. The *node-congestion of a node* $v \in \mathcal{V}_H$ is $\sum_{e \in \mathcal{E}_G} |\{v\} \cap \mathcal{V}_{\varphi(e)}|$, and the *node-congestion* of the embedding is

$$\max_{v \in \mathcal{V}_H} \left\{ \sum_{e \in \mathcal{E}_G} \left| \{v\} \cap \mathcal{V}_{\varphi(e)} \right| \right\}.$$

The *average node-congestion* is defined similarly. The *adjacency node set* of a node $v \in \mathcal{V}_H$ is the set of nodes $\mathcal{V}_v = \{v_j \mid (v, v_j) \in \mathcal{E}_H\}$, and the *edge set of a node* $v \in \mathcal{V}_H$ is $\mathcal{E}_v = \{(v, v_j) \in \mathcal{E}_H\}$. The *active-degree of a node* v is $|\mathcal{E}_v \cap \{\cup_{e \in \mathcal{E}_G} \mathcal{E}_{\varphi(e)}\}|$, and the *active-degree* of the embedding is

$$\max_{v \in \mathcal{V}_H} \left| \mathcal{E}_v \cap \{\cup_{e \in \mathcal{E}_G} \mathcal{E}_{\varphi(e)}\} \right|.$$

The *average active-degree* is defined similarly.

If each node of the guest graph is mapped to a distinct node of the host graph, then the expansion is a measure of processor utilization. The slow down due to nearest-neighbor communication in the original graph being extended to communication along paths is a function of the dilation of the edges on the path and their congestion. With a limited communications bandwidth at the nodes, the time for nearest-neighbor communication in the guest graph may also be influenced by the active-degree and the node-congestion.

A Boolean cube is a graph B with node set $\mathcal{V}_B$ such that $|\mathcal{V}_B| = 2^n$ for some n and edge set $\mathcal{E}_B = \{(i_{n-1}i_{n-2}\ldots i_j \ldots i_0, i_{n-1}i_{n-2}\ldots \bar{i}_j \ldots i_0) \mid j = \{0, 1, \ldots, n-1\}, i_j = \{0, 1\}\}$. $|\mathcal{E}_B| = n2^{n-1}$, and there exist n edge-disjoint paths between any pair of nodes. The number of nodes at distance d is $\binom{n}{d}$, and of the n paths between a pair of nodes at distance d, d paths are of length d, and $n - d$ paths are of length $d + 2$. The distance between nodes $i = (i_{n-1}i_{n-2}\ldots i_0)$ and $j = (j_{n-1}j_{n-2}\ldots j_0)$ in an n-cube is $Hamming(i, j) = \sum_{m=0}^{n-1}(i_m \oplus j_m)$, where $\oplus$ is the exclusive-or function. In the following, subcube 0 denotes the subcube that consists of all the nodes with the most significant bit of its address being 0. Subcube 1 is defined accordingly.

2.1 Graph decomposition

In this section we state and prove a few properties of product graphs, and the embedding characteristics of the product graph as a function of the embedding characteristics of the graphs forming the product graph.

Definition 1 Let G_1 and G_2 be two graphs. Then, the (Cartesian) *product* graph $G_1 \times G_2$ is defined as

$$\begin{aligned} \mathcal{V}_{G_1 \times G_2} &= \{[v_i, v_j] \mid \forall v_i \in \mathcal{V}_{G_1}, v_j \in \mathcal{V}_{G_2}\}, \text{ and} \\ \mathcal{E}_{G_1 \times G_2} &= \{([v_i, v_j], [v_i, v_k]) \mid \forall v_i \in \mathcal{V}_{G_1}, (v_j, v_k) \in \mathcal{E}_{G_2}\} \\ &\quad \bigcup \{([v_j, v_i], [v_k, v_i]) \mid \forall v_i \in \mathcal{V}_{G_2}, (v_j, v_k) \in \mathcal{E}_{G_1}\}. \end{aligned}$$

$G_1 \times G_2$ can be derived by replacing each vertex of G_1 by G_2 and replacing each edge of G_1 by a set of edges connecting corresponding vertices of G_2. Note that $G_1 \times G_2 = G_2 \times G_1$, $|\mathcal{V}_{G_1 \times G_2}| = |\mathcal{V}_{G_1}| * |\mathcal{V}_{G_2}|$, and $|\mathcal{E}_{G_1 \times G_2}| = |\mathcal{V}_{G_1}| * |\mathcal{E}_{G_2}| + |\mathcal{V}_{G_2}| * |\mathcal{E}_{G_1}|$.

Theorem 1 *Let φ_1 be an embedding function which maps a graph G_1 into a graph H_1 with expansion ε_1, dilation d_1 and congestion c_1. Also, let φ_2 be an embedding function which maps a graph G_2 into a graph H_2 with expansion ε_2, dilation d_2 and congestion c_2. Then, there exists an embedding function φ which maps the graph $G_1 \times G_2$ into the graph $H_1 \times H_2$ with expansion $\varepsilon_1\varepsilon_2$, dilation $\max(d_1, d_2)$ and congestion $\max(c_1, c_2)$.*

Proof: We prove the theorem by constructing an embedding function $\varphi : (G_1 \times G_2) \rightarrow (H_1 \times H_2)$. Let $\mathcal{S}_1^{v_i} = \{([u_j, v_i], [u_k, v_i]) | \forall (u_j, u_k) \in \mathcal{E}_{G_1}\}$ and $\mathcal{S}_2^{u_i} = \{([u_i, v_j], [u_i, v_k]) | \forall (v_j, v_k) \in \mathcal{E}_{G_2}\}$. Clearly,

$$\mathcal{E}_{G_1 \times G_2} = (\cup_{v_i \in \mathcal{V}_{G_2}} \mathcal{S}_1^{v_i}) \bigcup (\cup_{u_i \in \mathcal{V}_{G_1}} \mathcal{S}_2^{u_i}).$$

Intuitively, the edges of $G_1 \times G_2$ are partitioned into "G_1-type edges" and "G_2-type edges". G_1-type edges are further partition into $|\mathcal{V}_{G_2}|$ copies, where each node in G_2 identifies a copy. $\mathcal{S}_1^{v_i}$ is a copy of G_1 identified by node v_i in G_2. G_2-type edges are treated similarly. For the host graph $H_1 \times H_2$, we define $\mathcal{T}_1^{v_i}$ and $\mathcal{T}_2^{u_i}$ similarly. Hence,

$$\mathcal{E}_{H_1 \times H_2} = (\cup_{v_i \in \mathcal{V}_{H_2}} \mathcal{T}_1^{v_i}) \bigcup (\cup_{u_i \in \mathcal{V}_{H_1}} \mathcal{T}_2^{u_i}).$$

For each edge $(u_j, u_k) \in \mathcal{E}_{G_1}$ mapped to a path

$$\varphi_1((u_j, u_k)) = \{(\varphi_1(u_j), w_1), (w_1, w_2), \ldots, (w_{p-1}, \varphi_1(u_k))\}$$

in H_1 under φ_1, we let $([u_j, v_i], [u_k, v_i]) \in \mathcal{S}_1^{v_i}$ be mapped to the path

$$\{([\varphi_1(u_j), \varphi_2(v_i)], [w_1, \varphi_2(v_i)]), ([w_1, \varphi_2(v_i)], [w_2, \varphi_2(v_i)]), \ldots,$$
$$([w_{p-1}, \varphi_2(v_i)], [\varphi_1(u_k), \varphi_2(v_i)])\} \subseteq \mathcal{T}_1^{\varphi_2(v_i)}$$

under the new embedding function φ, for all $v_i \in \mathcal{V}_{G_2}$. We define the mapping from "the copy of G_1 identified by v_i" to "the copy of H_1 identified by $\varphi_2(v_i)$" according to the mapping $\varphi_1 : G_1 \times H_1$. The dilation is preserved to be d_1 for G_1-type edges and d_2 for G_2-type edges. Similarly, the congestion for the G_1-type and G_2-type edges are preserved. Note that (1) for any edges $e \in \mathcal{S}_1^{v_i}$, we have $\varphi(e) \subseteq \mathcal{T}_1^{\varphi_2(v_i)}$, (2) copies of H_1 (H_2) identified by different nodes in H_2 (H_1) are disjoint, and (3) edges of H_1-type and edges of H_2-type are disjoint. Therefore, the congestion is also preserved for all edges in $H_1 \times H_2$. ∎

The property that dilation is preserved under graph product was also observed in [19] and [10]. The latter states that graph simulation is preserved under graph product.

In the above theorem, if φ_1 has an average dilation $\bar{d}_1$, average congestion $\bar{c}_1$, node-congestion c_1', average node-congestion $\bar{c}_1'$ and active-degree a_1, and φ_2 has the same set of metrics with subscript 2, and

$$\alpha = \frac{|\mathcal{V}_{G_1}| * |\mathcal{E}_{G_2}|}{|\mathcal{E}_{G_1 \times G_2}|} \text{ and } \beta = \frac{|\mathcal{V}_{H_1}| * |\mathcal{E}_{H_2}|}{|\mathcal{E}_{H_1 \times H_2}|}.$$

Then, the embedding function φ has the average dilation $\alpha\bar{d}_2 + (1 - \alpha)\bar{d}_1$; the average congestion $\beta\bar{c}_2 + (1 - \beta)\bar{c}_1$; the node-congestion $c_1' + c_2'$; the average node-congestion $\bar{c}_1' + \bar{c}_2'$; and the active-degree $a_1 + a_2$.

Corollary 1 *Let φ_1 be an embedding function which maps a graph G_1 into an n_1-cube with expansion ε_1, dilation d_1 and congestion c_1. Also, let φ_2 be an embedding function which maps a graph G_2 into an n_2-cube with expansion ε_2, dilation d_2 and congestion c_2. Then, there exists an embedding function φ which maps a graph $G_1 \times G_2$ into an (n_1+n_2)-cube with expansion $\varepsilon_1\varepsilon_2$, dilation $\max(d_1,d_2)$ and congestion $\max(c_1,c_2)$.*

The property that dilation is preserved for hypercubes stated in this corollary has been observed in [21] and, recently, in [24]. It is used extensively in embedding the two- or higher dimensional meshes into Boolean cubes using Gray code on each axis.

Lemma 1 *The product graph of an $\ell_1 \times \ell_2 \times \cdots \times \ell_k$ mesh and an $\ell'_1 \times \ell'_2 \times \cdots \times \ell'_{k'}$ mesh is an $\ell_1 \times \ell_2 \times \cdots \times \ell_k \times \ell'_1 \times \ell'_2 \times \cdots \times \ell'_{k'}$ mesh.*

This lemma can be proved from the definition of product graphs and meshes.

Lemma 2 [22] *A k-dimensional mesh contains all linear arrays of at most as many nodes as the mesh as a subgraph.*

Lemma 3 [22] *An $\ell_1 \times \ell_2 \times \cdots \times \ell_k$ mesh is a subgraph of the mesh*

$$\ell_{11}\times\ell_{12}\times\cdots\times\ell_{1\alpha_1}\times\ell_{21}\times\ell_{22}\times\cdots\times\ell_{2\alpha_2}\times\cdots\times\ell_{k1}\times\ell_{k2}\times\cdots\times\ell_{k\alpha_k}, \text{ if } \prod_{j=1}^{\alpha_i}\ell_{ij} \geq \ell_i, \forall 1 \leq i \leq k.$$

This lemma can be easily derived from Theorem 1 and Lemma 2.

Theorem 2 *Let φ_1 be an embedding function which maps an $\ell_{11} \times \ell_{12} \times \cdots \times \ell_{1k}$ mesh M_1 into an n_1-cube with expansion ε_1, dilation d_1 and congestion c_1. Let φ_2 be an embedding function which maps an $\ell_{21} \times \ell_{22} \times \cdots \times \ell_{2k}$ mesh M_2 into an n_2-cube with expansion ε_2, dilation d_2 and congestion c_2. Then, there exists an embedding function φ which maps an $\ell_1 \times \ell_2 \times \cdots \times \ell_k$ mesh M into an (n_1+n_2)-cube with expansion $\varepsilon = \varepsilon_1\varepsilon_2$, dilation $d = \max(d_1,d_2)$ and congestion $c = \max(c_1,c_2)$, where $\ell_j = \ell_{1j}\ell_{2j}$ for all $1 \leq j \leq k$. (If $\ell_j \leq \ell_{1j}\ell_{2j}$ for all $1 \leq j \leq k$, then the embedding function φ has an expansion $\varepsilon = \varepsilon_1\varepsilon_2 \prod_{i=1}^{k}(\ell_{1j}\ell_{2j}/\ell_j)$, dilation $d \leq \max(d_1,d_2)$ and congestion $c \leq \max(c_1,c_2)$.)*

Proof: It follows from Corollary 1 that the product graph $M_1 \times M_2$ can be embedded into an (n_1+n_2)-cube with dilation $\max(d_1,d_2)$. Since the mesh M is a subgraph of the product graph $M_1 \times M_2$ by Lemma 3, the proof is complete. ∎

We have given the proofs of the existence of embedding functions with the desired properties. We now define the embedding function φ satisfying the existence theorems for mesh M. The embedding function φ is defined as a function of φ_1 and φ_2. Let $z = (z_1, z_2, \ldots, z_k)$ be a node in the mesh M. Let $z_i = x_i\ell_{1i} + y_i$ and $0 \leq y_i < \ell_{1i}$ for $1 \leq i \leq k$. Define

$$\tilde{\varphi}_1(x_1,x_2,\ldots,x_k,y_1,y_2,\ldots,y_k) = \varphi_1((y'_1,y'_2,\ldots,y'_k)) \text{ where } y'_i = \begin{cases} y_i, & \text{if } x_i \text{ is even,} \\ \ell_{1i}-1-y_i, & \text{otherwise.} \end{cases}$$

The new function $\tilde{\varphi}_1$ differs from the function φ_1 in that a reflection of the mesh is performed for axes for which x_i is odd. The function φ can be defined as follows:

$$\varphi((z_1, z_2, \ldots, z_k)) = \varphi_2((x_1, x_2, \ldots, x_k))||\tilde{\varphi}_1(x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k).$$

Note that if both φ_1 and φ_2 offer minimal expansion embeddings it does not follow that the embedding function φ inherits this property. φ yields minimal expansion only when $\varepsilon_1\varepsilon_2 < 2$.

Corollary 2 *If an $\ell_1 \times \ell_2 \times \cdots \times \ell_k$ mesh M can be embedded in an n-cube with expansion ε, dilation d and congestion c, then an $\ell'_1 \times \ell'_2 \times \cdots \times \ell'_k$ mesh M', $\ell'_i = \ell_i 2^{n_i}$ can be embedded in an $(n + \sum_{i=1}^{k} n_i)$-dimensional cube with expansion ε, dilation d and congestion c. (If $\ell'_i \leq \ell_i 2^{n_i}$ for all $1 \leq i \leq k$, then the mesh M' can be embedded with an expansion $\varepsilon = \prod_{i=1}^{k}(\ell_i 2^{n_i}/\ell'_i)$, dilation $\leq d$ and congestion $\leq c$.)*

Proof: Simply use the binary-reflected Gray code embedding as the embedding function φ_2 in Theorem 2. ∎

The corollary has been used implicitly in [7] and [14]. The new embedding function can be easily derived as follows:

$$\begin{aligned} \tilde{\varphi}((x_1||y_1, x_2||y_2, \ldots, x_k||y_k)) &= \varphi((x_1, x_2, \ldots, x_k))||\tilde{G}(x_1, y_1)||\tilde{G}(x_2, y_2)||\cdots||\tilde{G}(x_k, y_k), \\ \text{where } \tilde{G}(x_i, y_i) &= \begin{cases} G(y_i), & \text{if } x_i \text{ is even,} \\ G(2^{n_i} - 1 - y_i), & \text{otherwise.} \end{cases} \end{aligned} \quad (1)$$

The above embedding function is not the only one that satisfies Corollary 2. The linear array described in Lemma 2 is not unique. Axis i of the mesh M' in Corollary 2 is derived by finding a linear array embedded in an $\ell_i \times 2^{n_i}$ mesh, say M_i, such that most edges are dilation one edges. The edges along the first axis in the mesh M_i have dilation $\leq d$, but the edges along the second axis have dilation one. Traversing edges mostly along the second axis minimizes the average dilation, in general. Let $\bar{d}_i$ be the average dilation of the edges of the ith axis in the mesh M in Corollary 2, then the average dilation of the embedding of the mesh M' can be shown to be

$$1 + \sum_{i=1}^{k} \left\{ (\bar{d}_i - 1) 2^{(\sum_{j=1}^{k} n_j) - n_i} (\ell_i - 1)(\prod_{j=1}^{k} \ell_j)/\ell_i \right\} \Big/ \sum_{i=1}^{k} \left\{ (\ell_i 2^{n_i} - 1)(\prod_{j=1}^{k} \ell_j 2^{n_j})/(\ell_i 2^{n_i}) \right\}$$

$$\approx 1 + \sum_{i=1}^{k} \frac{\bar{d}_i - 1}{k 2^{n_i}}.$$

The approximated term shows that the average dilation decreases as the size of the subcube (that applied with Gray code embedding) increases.

Corollary 3 *Let φ_i be an embedding function which maps an $\ell_{i1} \times \ell_{i2} \times \cdots \times \ell_{ik}$ mesh M_i into an n_i-cube with expansion ε_i, dilation d_i and congestion c_i for $1 \leq i \leq r$. Then, there exists an embedding function φ which maps an $\ell_1 \times \ell_2 \times \cdots \times \ell_k$ mesh M into a $(\sum_{i=1}^{r} n_i)$-cube with expansion $\varepsilon = \prod_{i=1}^{r} \varepsilon_i$, dilation $d = \max_{1 \leq i \leq r}\{d_i\}$ and congestion $c = \max_{1 \leq i \leq r}\{c_i\}$, where $\ell_j = \prod_{1 \leq i \leq r} \ell_{ij}$ for $1 \leq j \leq k$.*

Proof: Simply apply Theorem 2 $r-1$ times. ∎

Clearly, expressions for the dilation and the congestion are also valid for $\ell_j \leq \prod_{1 \leq i \leq r} \ell_{ij}$ and the expansion becomes

$$\varepsilon = (\prod_{i=1}^{r} \varepsilon_i) * (\prod_{i=1}^{r} |\mathcal{V}_{M_i}|)/|\mathcal{V}_M|.$$

We use the results about the embedding of product graphs to combine previously known techniques [15, 7, 14, 4] for the embedding of meshes in Boolean cubes. We also use the decomposition technique in combination with some new results for the embedding of three-dimensional meshes. The graph decomposition technique together with the direct embedding of three two-dimensional meshes allows 87% of all two-dimensional meshes [14], asymptotically, to be embedded with dilation two, congestion two and minimal expansion. Both the average dilation and average congestion are one asymptotically. The average dilation of the dilation-two embedding of all two-dimensional meshes by Chan [4] does not decrease as the size of the mesh increases. Chan does not consider the congestion. The graph decomposition technique in combination with all previously known results yields dilation-two minimum-expansion embeddings of 96% of all three-dimensional meshes $\ell_1 \times \ell_2 \times \ell_3$, such that $1 \leq \ell_1, \ell_2, \ell_3 \leq 512$. By using Gray code embedding, only 29% of the meshes achieve minimal expansion for the considered three-dimensional domain. The best known upper bound for the dilation for the embedding of all three-dimensional meshes with minimal expansion is seven [6].

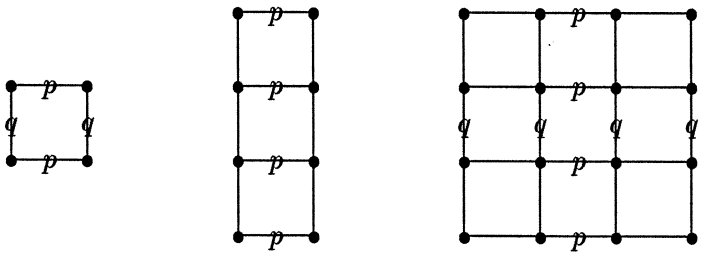
2.2 Lower bounds for the expansion of dilation one embeddings

The binary-reflected Gray code embedding yields dilation one embeddings, but the expansion may be high. In the next section we determine the fraction of all possible meshes for which Gray code embedding yields minimal expansion embeddings. The following theorem due to Havel and Móravek [12] shows that any dilation one embedding has the same expansion as the binary-reflected Gray code embedding. Hence, if the Gray code embedding does not yield minimal expansion, then any minimal expansion embedding must have a dilation of at least two.

Each node of the n-cube is represented by an *n-bit* binary number. Each node of the $\ell_1 \times \ell_2$ mesh is represented by an address (i,j), where $0 \leq i \leq \ell_1 - 1$, $0 \leq j \leq \ell_2 - 1$. There is a one-to-one mapping from edges of the mesh to edges of the cube in a dilation one embedding. By "an edge in a cube is in dimension i" we mean that the addresses of the two end points differ in bit i. The most significant bit (*msb*) is bit $n-1$ and the least significant bit (*lsb*) is bit 0. The label on an edge in the mesh represents the dimension of the corresponding edge in the cube.

Theorem 3 [12] *If an $\ell_1 \times \ell_2 \times \cdots \times \ell_d$ mesh is embedded in an n-cube with dilation one, then $n \geq \sum_{i=1}^{k} \lceil \log_2 \ell_i \rceil$.*

Proof: Consider a two-dimensional mesh first. Label each edge in the mesh with a number that represents the dimension of the corresponding edge in the cube. A legal labeling should satisfy the following two properties:



(a) (b) (c)

Figure 1: (a) Dimensions of a cycle of length 4. (b) Vertical or horizontal cuts. (c) The set of dimensions of vertical edges and the set of dimensions of horizontal edges are disjoint.

1. Any cycle in the mesh must contain any label an even number of times.
2. Any path in the mesh must contain some label an odd number of times.

For any cycle of length 4 in the mesh, the edge labels form a sequence (p, q, p, q), where $p \neq q$, Figure 1-(a). By extending this argument, it follows that any vertical or horizontal cut will only cut edges having the same label, Figure 1-(b). Now consider the labels of any horizontal path connecting all the ℓ_2 nodes of a row. Since the corresponding cube nodes are distinct and the longest node-disjoint path in an n-cube is of length $2^n - 1$, the number of distinct labels on the edges forming a row is at least $\lceil \log_2 \ell_2 \rceil$. The same argument applies to any vertical path. Moreover, the set of labels on the horizontal paths must be disjoint from the set of labels on the vertical paths; otherwise there exist two adjacent edges with the same label, Figure 1-(c). Hence, the minimum number of dimensions required is $\lceil \log_2 \ell_1 \rceil + \lceil \log_2 \ell_2 \rceil$.

Since the set of cube dimensions used as labels for mesh edges in mesh dimension i is disjoint from the set of cube dimensions used as labels for mesh edges in dimension j, $i \neq j$, the proof is complete. ∎

Theorem 3 was independently rediscovered in [3], [8], [14] and [11]. From the theorem it follows that the expansion is in the range of 1 to 2^k.

2.3 Percentage of minimal expansion using Gray code embedding

The percentage of meshes for which Gray code embedding [23], [17], [15], [3], yields minimal expansion embeddings decreases with the number of axes of the mesh. In this section we show that, asymptotically, the percentage is about 61% in two dimensions, but only 27% in three dimensions.

Determining the asymptotic expansion for Gray code embedding is transformed to the following probability problem. Let a_i, $i \geq 1$, be a variable uniformly distributed over an interval $(\frac{1}{2}, 1]$, and a_i and a_j be independent variables for all $i \neq j$. Then, the probability that $\prod_{i=1}^{k} a_i \in (1/2^{\beta+1}, 1/2^{\beta}]$ is the asymptotic fraction of embedding k-dimensional meshes using Gray code embedding with an expansion 2^{β}. For minimal expansion $\beta = 0$.

Let $\alpha \in (\frac{1}{2}, 1]$ and $f_k(\alpha)$ be the probability that $\alpha \leq \prod_{i=1}^{k} a_i \leq 1$. Then,

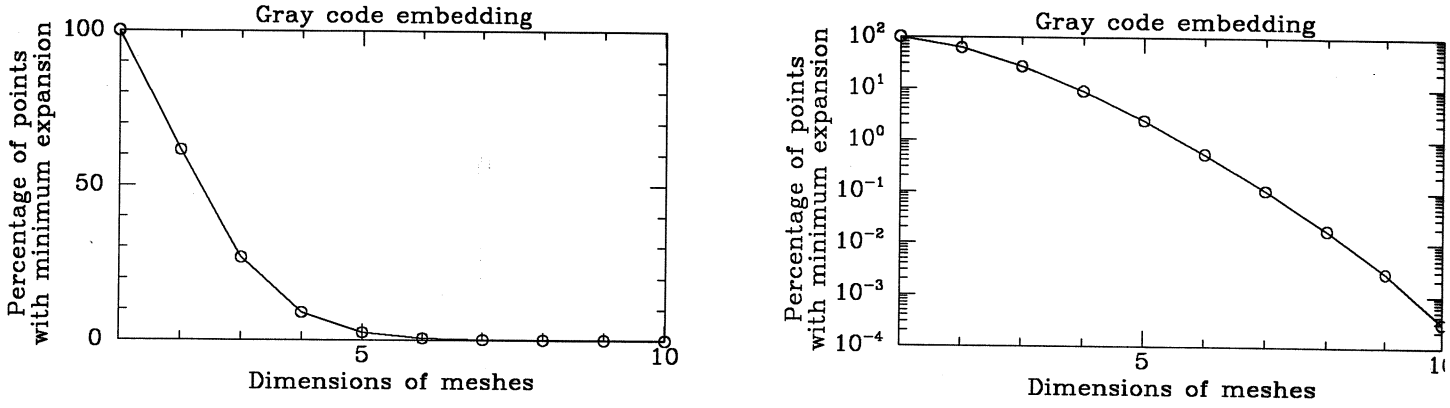


Figure 2: The asymptotic fraction of the domain $(\lceil \ell_i \rceil_2/2) < \ell_i \leq \lceil \ell_i \rceil_2$ for which minimum expansion is attained by Gray code embedding. The right plot has a logarithmic scale for the y-axis.

Lemma 4 *$f_n(\alpha) = 2^n(1 - \alpha \sum_{i=0}^{n-1} \frac{(-1)^i \ln^i \alpha}{i!})$ for $\frac{1}{2} \leq \alpha \leq 1$.*

Proof: By induction on n. $f_1(\alpha) = 2(1-\alpha)$, $\frac{1}{2} \leq \alpha \leq 1$.

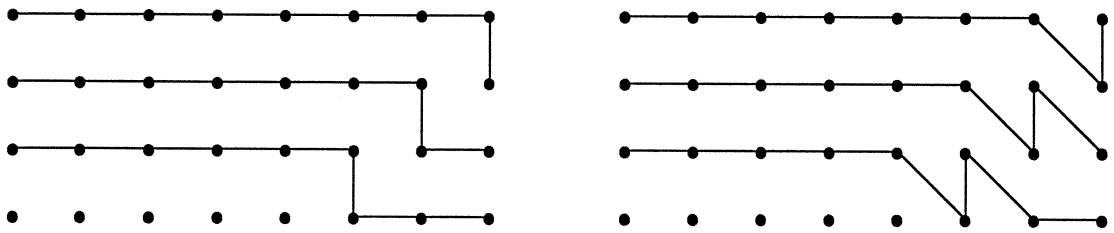
$$\begin{aligned}
f_{n+1}(\alpha) &= 2\int_\alpha^1 f_n\left(\frac{\alpha}{x}\right) dx \\
&= 2\int_\alpha^1 \left(2^n - 2^n \frac{\alpha}{x} \sum_{i=0}^{n-1} \frac{(-1)^i \ln^i \frac{\alpha}{x}}{i!}\right) dx \qquad (2) \\
&= 2^{n+1}\int_\alpha^1 dx - 2^{n+1}\alpha \sum_{i=0}^{n-1} \int_{\ln\alpha}^0 \frac{(-1)^i u^i}{i!} du \qquad (3) \\
&= 2^{n+1} - 2^{n+1}\alpha - 2^{n+1}\alpha \sum_{i=1}^{n} \frac{(-1)^i \ln^i \alpha}{i!} \\
&= 2^{n+1} - 2^{n+1}\alpha \sum_{i=0}^{n} \frac{(-1)^i \ln^i \alpha}{i!}.
\end{aligned}$$

¿From Equation 2 to 3, we let $u = \ln \frac{\alpha}{x}$. Hence, $du = -\frac{dx}{x}$. ∎

Theorem 4 *The fraction of all k-dimensional meshes for which a binary-reflected Gray code embedding yields minimum expansion is $f_k(\frac{1}{2}) = 2^k(1 - \frac{1}{2}\sum_{i=0}^{k-1} \frac{\ln^i 2}{i!})$, asymptotically.*

Proof: By Lemma 4. ∎

Figure 2 shows $f_k(\frac{1}{2})$ as a function of the number of dimensions, k. $f_2(\frac{1}{2}) = 2(1 - \ln 2) \approx 0.61$ and $f_3(\frac{1}{2}) = 4(1 - \ln 2 - \frac{\ln^2 2}{2}) \approx 0.27$.



(a) (b)

Figure 3: Embedding of a 3×9 mesh into a 4×8 mesh by step embedding with, (a) dilation 3, (b) dilation 2.

3 Reshaping techniques

With reshaping techniques we mean techniques for which an $\ell_1 \times \ell_2 \times \ldots \times \ell_k$ mesh is embedded in an $\ell'_1 \times \ell'_2 \times \ldots \times \ell'_k$ mesh. The number of axes is preserved, but the length of the different axes are changed. We only consider two-dimensional meshes reshaped by step embedding [1], modified step embedding, folding [20], and line compression [1]. Embedding rectangular meshes into square meshes is also considered in [9]. Embeddings of a multidimensional mesh into another multidimensional mesh of different shapes and cardinalities is studied by [18] and [22]. By making the reshaped mesh having axes with lengths being powers of two, a Gray code embedding can be applied to the reshaped mesh. This technique was used in [15].

In [1] reshaping techniques were used to square up a rectangular mesh. Given an $\ell_1 \times \ell_2$ mesh, we wish to find an embedding with small dilation into an $N_1 \times N_2$ mesh, where $N_1 = 2^{n_1}$ and $N_2 = 2^{n_2}$, such that $N_1 N_2 = \lceil \ell_1 \ell_2 \rceil_2$. The dilation is 3 for *step embedding* and *modified step embedding*, and 2 for *folding* and *line compression*. For convenience, we assume $\ell_1 \leq \ell_2$ in the following, without loss of generality. The embedding can be represented by joining ℓ_1 lines of length $\ell_2 - 1$ in the $N_1 \times N_2$ mesh, such that any two corresponding nodes on the successive lines are at most a distance 2 (or 3) apart. No node must be traversed by more than one solid line.

3.1 Step embeddings

Figure 3-(a) shows the embedding of a 3×9 mesh into a 4×8 mesh by the step embedding technique. The dilation is three. For $\ell_2 = N_2 + 1$, the dilation can be reduced to 2 as shown in 3-(b). Each row of the guest mesh will "turn" at some point and make a vertical traversal. Different rows traverse different columns, and it follows that $N_2 \geq \ell_1$. Also, row i of the guest mesh will occupy a part of rows i and $i + \ell_2 - N_2$ of the host mesh, and $N_1 \geq \ell_2 - N_2 + \ell_1$. The number of edges with dilation 3 is $(\ell_2 - N_2)(\ell_1 - 1)$. The average dilation is $1 + \frac{2(\ell_2 - N_2)(\ell_1 - 1)}{|\mathcal{E}_G|} \approx 1 + \frac{\ell_2 - N_2}{\ell_2}$, which is in the range 1 - 1.5 if $N_2 = \lfloor \ell_2 \rfloor_2$. Correspondingly, $N_1 = \lceil \ell_1 \rceil_2$, if $N_1 \geq \ell_2 - N_2 + \ell_1$ is satisfied. Figure 16-(b) shows the pairs (ℓ_1, ℓ_2) for which the step embedding method attains minimal expansion, while Gray code embedding cannot. The ratio of the number of these pairs to the total number of pairs is $\approx \frac{1}{8}$.

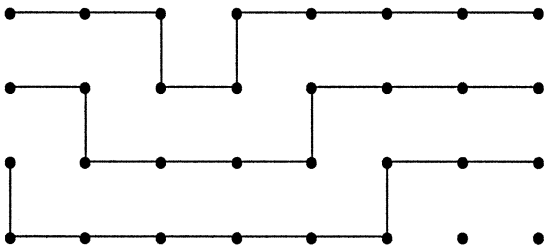


Figure 4: Embedding of a 3 × 10 mesh into a 4 × 8 mesh by modified step embedding.

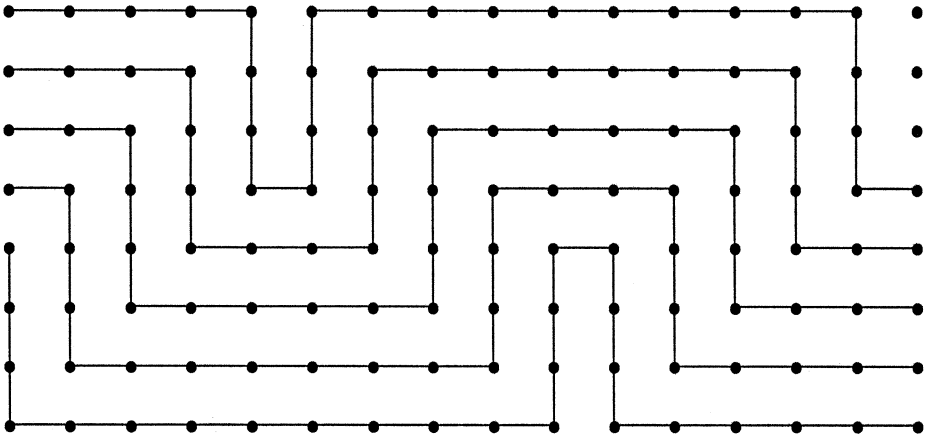


Figure 5: Embedding of a 5 × 25 mesh into a 8 × 16 mesh by modified step embedding.

3.2 Modified step embeddings

In step embedding, a row of the guest mesh only makes one vertical traversal. In modified step embedding, several vertical traversals are allowed, Figures 4 and 5. Each vertical traversal involves ℓ_1 distinct columns, one for each row, and the maximum number of vertical traversals by all rows is $\alpha = \lfloor \frac{N_2}{\ell_1} \rfloor$. Since each vertical traversal can save $N_1 - \ell_1$ nodes from ℓ_2, the condition $N_1 \geq \lceil \frac{\ell_2 - N_2}{\alpha} \rceil + \ell_1$ must hold. Note that the step embedding technique cannot be applied if $\alpha = 0$, i.e., $N_2 < \ell_1$. If $\alpha = 1$ the modified step embedding technique is the same as the step embedding technique.

The modified step embedding allows minimal expansion for more pairs (ℓ_1, ℓ_2) than step embedding, Figure 16-(c). The additional pairs all satisfy the condition $\lceil \ell_2 \rceil_2 / \lceil \ell_1 \rceil_2 \geq 4$ (assuming $\ell_1 \leq \ell_2$). For example, a 3×10 mesh is mapped to a 5×8 mesh by step embedding, but to a 4 × 8 mesh by modified step embedding, Figure 4. A 5 × 25 mesh is mapped to a 14 × 16 mesh by step embedding, but a 8 × 16 mesh by modified embedding, Figure 5. The average dilation is the same as in step embedding, i.e., 1 to 1.5. If $\lfloor \frac{\ell_2 - N_2}{k} \rfloor = 1$ then the dilation can be reduced to 2. However, the average dilation will increase.

3.3 Folding

Folding is based on the break-and-fold technique [20] used in [1] in squaring up an $\ell_1 \times \ell_2$ mesh with $\ell_1 \ll \ell_2$. If there exists an n_1-cube and an n_2-cube such that $N_1 N_2 = \lceil \ell_1 \rceil_2 \lfloor \ell_2 \rfloor_2$

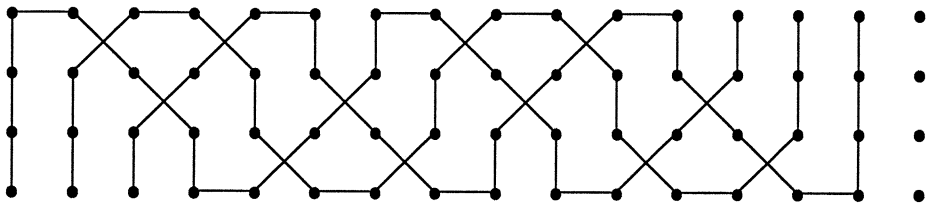


Figure 6: Embedding of a 3 x 20 mesh into a 4 x 16 mesh by folding.

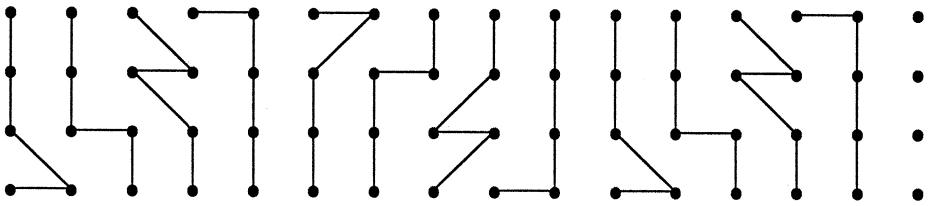


Figure 7: Embedding of a 5 x 12 mesh into a 4 x 16 mesh by the line compression method.

and $\lfloor \frac{N_1}{\ell_1} \rfloor \geq \lceil \frac{\ell_2}{N_2} \rceil$, then the folding technique can yield a minimal expansion embedding. Figure 6 shows the embedding of a 3×20 mesh into a 4×16 mesh. The dilation of the folding embedding is two. Figure 16-(d) shows the pairs (ℓ_1, ℓ_2) for which the folding technique yields minimal expansion, but the Gray code embedding does not. The set of these pairs is mostly disjoint from the set of pairs for which step embedding yields minimal expansion, but the Gray code does not. However, folding and the modified step embedding largely covers the pairs. The average dilation is $\approx 1 + \frac{\ell_1}{N_2}$.

3.4 Line compression

Line compression is also adopted from [1] in which a basic "tile" of size $a \times b$ is compressed into a tile of size $b \times a$. Let $b = a + 1$, then $\lfloor \frac{N_2}{a} \rfloor \geq \ell_2 - N_2$ and $N_1 \geq \ell_1 + \lceil \frac{\ell_1}{a} \rceil$. In order to satisfy these two constraints, $a > 1$ and $\ell_2 \leq \frac{3}{2} N_2$ must be satisfied. One can easily show that any $(2^x - 1) \times (2^x + 1)$ mesh can be reshaped into a $2^x \times 2^x$ mesh by line compression (with $a = 2^x - 1$). The dilation is two and the average dilation is $\approx 1 + \frac{a+2b}{2ab} \approx 1 + \frac{3}{2a}$. Minimizing the average dilation is equivalent to maximizing a. Figure 7 shows the embedding of a 5×12 mesh into a 4×16 mesh by the line compression method. In this figure solid lines represent columns of the guest mesh. Figure 16-(e) shows the pairs (ℓ_1, ℓ_2) for which the line compression method yields minimal expansion, but the Gray code does not. The set of these pairs include mostly the set of pairs for which (a) step embedding, (b) modified step embedding, and (c) folding yields minimal expansion, but Gray code does not.

4 Direct cube embeddings

4.1 Dilation two congestion two embeddings of model meshes

In this section we first give three dilation two, minimal expansion embeddings of two-dimensional meshes in Boolean cubes, then two dilation two and one dilation three, minimal expansion embeddings of three-dimensional meshes. The meshes are of shapes 5×3, 9×7,

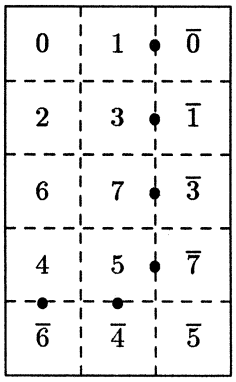


Figure 8: Embedding of a 5 × 3 mesh in a 4-cube.

11 × 11, 3 × 3 × 3, 7 × 3 × 3 and 5 × 5 × 5. Using these direct embeddings together with graph decomposition allows for minimal expansion, dilation two embedding of 70% of all two-dimensional meshes for which Gray code does not yield minimal expansion. For three-dimensional meshes, we use these direct embeddings extended with the two-dimensional result in [4], and the graph decomposition technique. We achieve dilation two minimal expansion embeddings for 96% of the three-dimensional meshes contained within, or equal to a 512 × 512 × 512 mesh. The two-dimensional direct embeddings below appeared in [14].

4.1.1 Two-dimensional direct embeddings

All the three two-dimensional embeddings have the property that the dilation for any edge of the upper-left $\lfloor \ell_1 \rfloor_2 \times \lfloor \ell_2 \rfloor_2$ submesh of the $\ell_1 \times \ell_2$ mesh, is one.

Embedding a 5 × 3 mesh into a 4-cube. To embed a 5 × 3 mesh into a 4-cube, we use the mapping represented by Figure 8, in which the numbers represent the addresses of the cube nodes to which the mesh nodes are mapped. For ease of determining the dilation of edges, we use $\overline{x}$ to represent the node in subcube one that corresponds to node x in subcube zero, i.e., $\overline{x}$ is derived from x by complementing the most significant bit. The '•' sign on the dashed line means that the Hamming distance between the two adjacent mesh nodes is two when embedded in the cube. From Figure 8 it is apparent that the dilation is two.

To determine the congestion of the embedding, we first specify all length-two paths as follows, where the '•' sign above a doubled arrow denotes a cube edge, which is also used as a dilation-one edge.

$\overline{0} \leftrightarrow 0 \overset{\bullet}{\leftrightarrow} 1$, $\quad \overline{1} \leftrightarrow 1 \overset{\bullet}{\leftrightarrow} 3$, $\quad \overline{3} \leftrightarrow 3 \overset{\bullet}{\leftrightarrow} 7$,
$\overline{7} \leftrightarrow 7 \overset{\bullet}{\leftrightarrow} 5$, $\quad \overline{4} \leftrightarrow 4 \overset{\bullet}{\leftrightarrow} 5$, $\quad \overline{6} \leftrightarrow 6 \overset{\bullet}{\leftrightarrow} 4$.

By inspection, one can easily show that the congestion is two; the number of cube edges with congestion two is 6; the active-degree is 4; and the node-congestion is 6. Embedding of a $5 \cdot 2^{n_1} \times 3 \cdot 2^{n_2}$ mesh was also independently found in [7].

Embedding of a 9 × 7 mesh into a 6-cube. Figure 9 shows the embedding of a 9 × 7 mesh in a 6-cube. The length-two paths in the cube are specified as follows:

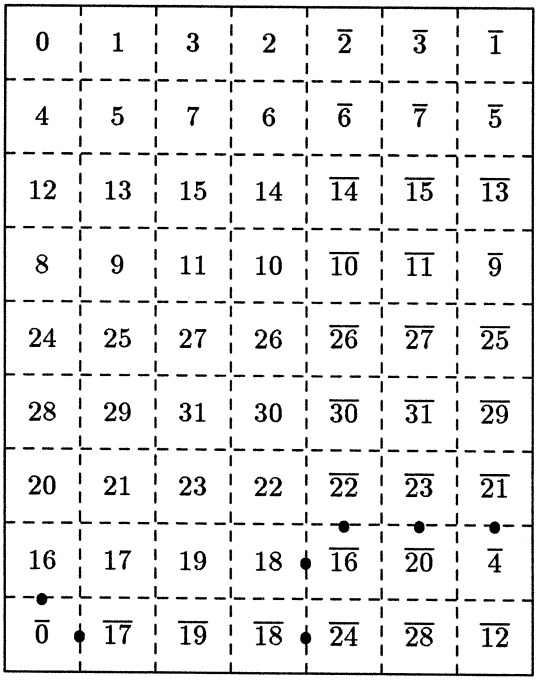


Figure 9: Embedding of a 9 × 7 mesh in a 6-cube.

$\overline{4} \leftrightarrow \overline{5} \leftrightarrow \overline{21}$.
$\overline{16} \leftrightarrow 16 \leftrightarrow 18$.
$\overline{0} \leftrightarrow 0 \leftrightarrow 16$.

$\overline{20} \leftrightarrow \overline{21} \overset{\bullet}{\leftrightarrow} \overline{23}$.
$\overline{24} \leftrightarrow \overline{26} \leftrightarrow \overline{18}$.

$\overline{16} \leftrightarrow \overline{18} \leftrightarrow \overline{22}$.
$\overline{0} \leftrightarrow \overline{16} \leftrightarrow \overline{17}$.

By inspection, the dilation is two; the number of edges with dilation two is 7; the congestion is two; the number of edges with congestion two is one; the active-degree is 6 (for node $\overline{16}$); and the node-congestion is 6 (since all the intermediate nodes of the length-two paths are different).

Embedding of an 11 × 11 mesh into a 7-cube. Figure 10 shows the embedding of an 11 × 11 mesh in a 7-cube. The length-two paths are specified as follows:

$\overline{51} \leftrightarrow \overline{55} \leftrightarrow \overline{39}$,
$\overline{48} \leftrightarrow \overline{16} \leftrightarrow \overline{0}$,
$\overline{16} \leftrightarrow \overline{20} \leftrightarrow \overline{22}$,
$\overline{32} \overset{\bullet}{\leftrightarrow} \overline{33} \leftrightarrow \overline{1}$,
$\overline{35} \leftrightarrow \overline{39} \leftrightarrow 39$,
$\overline{18} \overset{\bullet}{\leftrightarrow} \overline{26} \leftrightarrow \overline{10}$,
$\overline{43} \leftrightarrow \overline{35} \leftrightarrow 35$,

$\overline{49} \leftrightarrow \overline{53} \leftrightarrow \overline{37}$,
$\overline{0} \overset{\bullet}{\leftrightarrow} \overline{1} \leftrightarrow \overline{17}$,
$\overline{22} \leftrightarrow \overline{23} \leftrightarrow \overline{19}$,
$\overline{33} \leftrightarrow \overline{37} \leftrightarrow 37$,
$\overline{35} \overset{\bullet}{\leftrightarrow} \overline{34} \leftrightarrow \overline{2}$,
$\overline{34} \leftrightarrow \overline{38} \leftrightarrow 38$,
$\overline{41} \leftrightarrow \overline{33} \leftrightarrow 33$,

$\overline{48} \leftrightarrow \overline{52} \leftrightarrow \overline{36}$
$\overline{8} \leftrightarrow \overline{24} \leftrightarrow \overline{16}$
$\overline{32} \leftrightarrow 32 \leftrightarrow 36$
$\overline{33} \overset{\bullet}{\leftrightarrow} \overline{35} \leftrightarrow \overline{3}$
$\overline{50} \overset{\bullet}{\leftrightarrow} \overline{18} \leftrightarrow \overline{2}$
$\overline{42} \leftrightarrow 42 \overset{\bullet}{\leftrightarrow} 34$
$\overline{40} \leftrightarrow 40 \overset{\bullet}{\leftrightarrow} 32$

The length-two paths are all edge-disjoint with respect to each other. Hence, the congestion is at most two. By inspection, the dilation is two; the number of edges with dilation two is 21; the congestion is two; the number of edges with congestion two is 8 (marked by '•'); the active-degree is 6 (for instance, nodes $\overline{33}$ and $\overline{35}$); and the node-congestion is 8 (for instance, nodes $\overline{33}$ and $\overline{35}$).

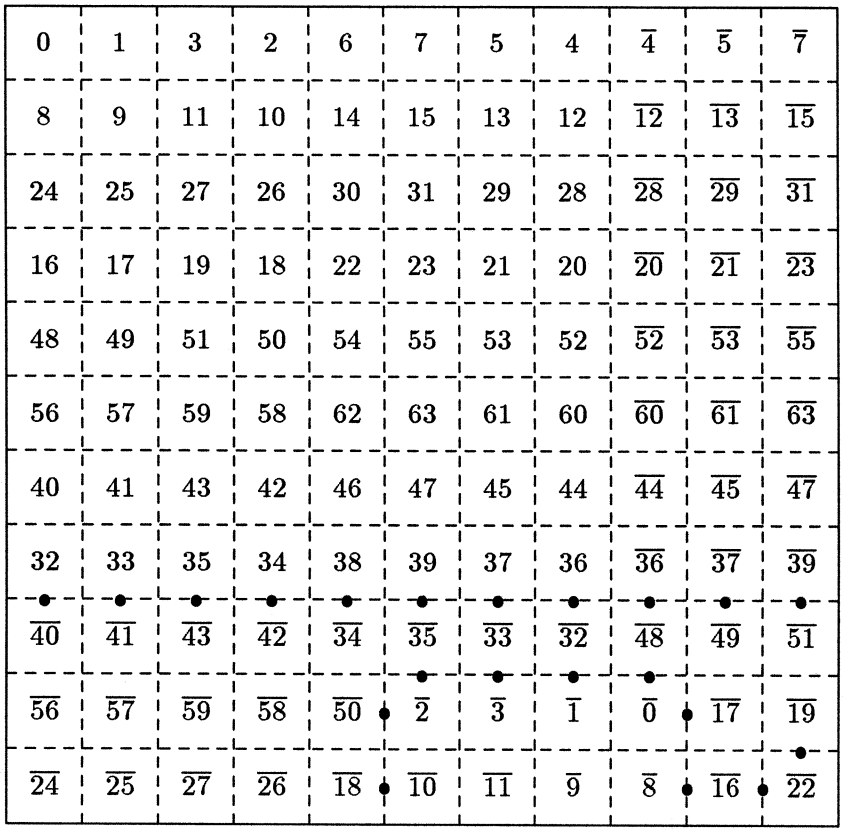


Figure 10: Embedding of an 11 x 11 mesh in a 7-cube.

4.1.2 Three-dimensional direct embeddings

All three embeddings in this section have the property that the Gray code embedding will not yield minimal expansion. The direct embedding of the $3\times3\times3$ and $7\times3\times3$ meshes have a dilation of two, and the embedding of the $5 \times 5 \times 5$ mesh has a dilation of three. All three embeddings have minimal expansion. Applying the reshaping technique, the direct two-dimensional embedding to any pair of the three dimensions, or the decomposition technique does not result in dilation two, minimal expansion embeddings.

Embedding of a $3 \times 3 \times 3$ mesh in a 5-cube. Figure 11 shows the embedding of a $3\times3\times3$ mesh in a 5-cube with dilation two. The number in the figure is the cube address in octal representation to which the mesh node is mapped. The sign "$*$" marks a dilation-two edge between the marked node and the corresponding node in the plane immediately to the left. The number of dilation-two edges is 15.

Embedding of a $7 \times 3 \times 3$ mesh in a 6-cube. Figure 12 shows the embedding of a $7 \times 3 \times 3$ mesh in a 6-cube with dilation two. The number of dilation-two edges is 45.

Embedding of a $5 \times 5 \times 5$ mesh in a 7-cube. Figure 13 shows the embedding of a $5 \times 5 \times 5$ mesh in a 7-cube with dilation three.

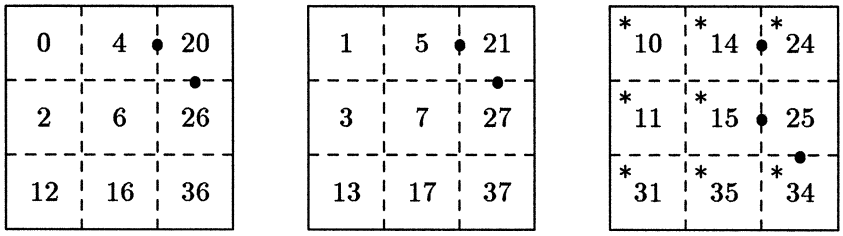


Figure 11: Embedding of a $3 \times 3 \times 3$ mesh in a 5-cube. The number is the cube address in octal representation to which the mesh node is mapped.

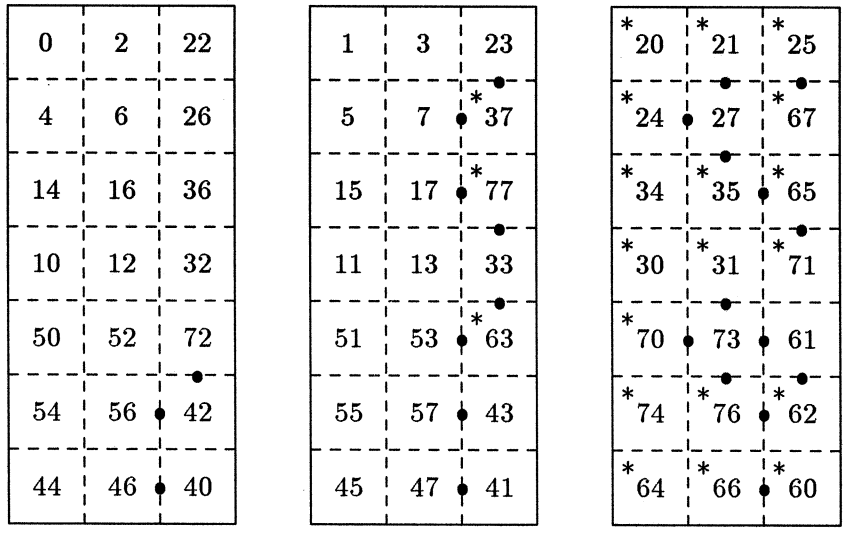


Figure 12: Embedding of a $7 \times 3 \times 3$ mesh in a 6-cube. The number is the cube address in octal representation to which the mesh node is mapped.

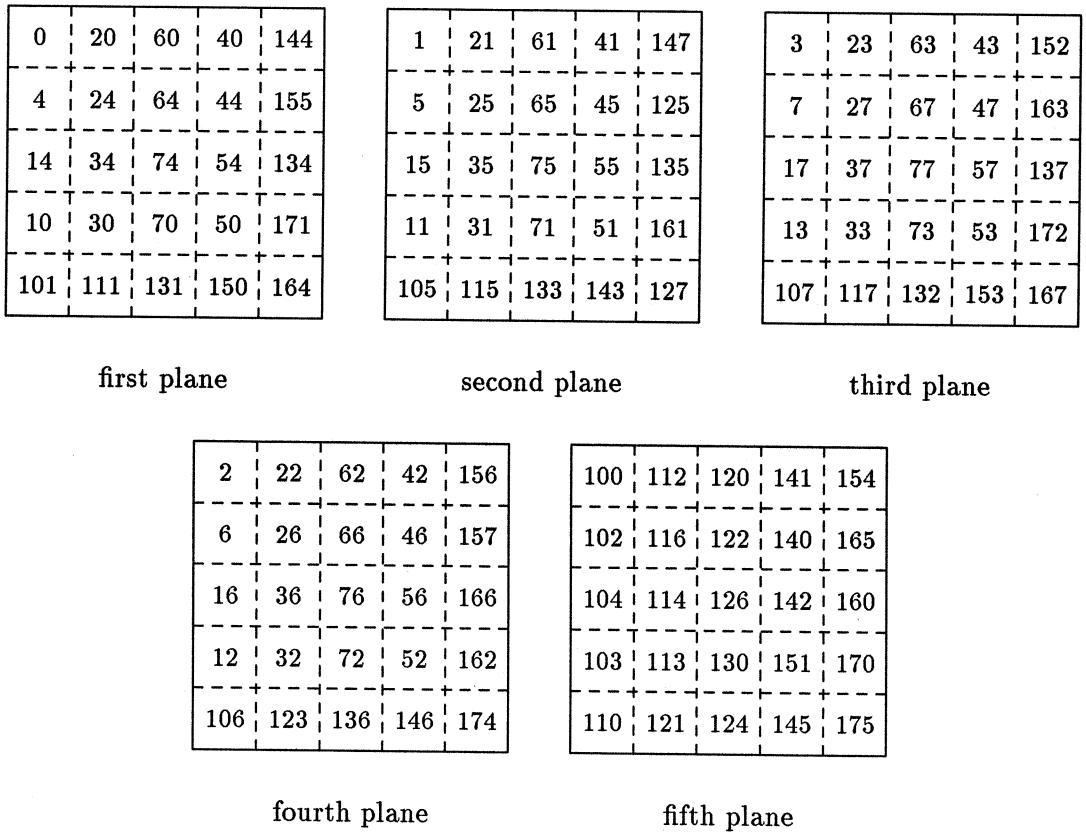


Figure 13: Embedding of a $5 \times 5 \times 5$ mesh in a 7-cube with dilation three. The number is the cube address in octal representation to which the mesh node is mapped.

$\ell_1 \times \ell_2$ embedding	Dilation	Average dilation	Congestion	Average congestion	Active-degree	Node-load
5×3	2	1.27	2	1.27	4	6
9×7	2	1.06	2	1.01	6	6
11×11	2	1.10	2	1.04	6	8

Table 1: Summary of embeddings for various aspect ratios.

4.1.3 Summary of direct embeddings

The characteristics of the two-dimensional direct embeddings are summarized in Table 1.

For the three-dimensional meshes of 128 nodes or less, the $5 \times 5 \times 5$ mesh is the only mesh for which we do not know of a minimal expansion dilation two embedding, if it exists. For three-dimensional meshes with up to 256 nodes, there are four additional meshes for which the same statement applies: $5 \times 7 \times 7$, $3 \times 9 \times 9$, $5 \times 5 \times 10$ and $3 \times 5 \times 17$.

4.2 Modified line compression

Recently, Chan [4] presented a dilation-two embedding for all two-dimensional meshes. The technique is based on a modification of the line compression technique directly applied to Boolean cubes. An intermediate mesh of the form $\lfloor \ell_1 \rfloor_2 \times \lceil \ell_2 \rceil_2$ is embedded in the guest mesh $\ell_1 \times \ell_2$ by the line compression technique. The intermediate mesh can be embedded in the Boolean cube by a binary-reflected Gray code. The intermediate mesh consists of $\lfloor \ell_1 \rfloor_2$ rows of length $\lceil \ell_2 \rceil_2$. For each row a chain of length at least ℓ_2 and at most $\lceil \ell_2 \rceil$ is formed. Each such chain covers all the nodes of a row of the guest mesh. Nodes of the guest mesh that are neighbors on the same chain are at most distance two apart. Successive chains are embedded in the guest mesh by cyclic rotation. The starting position of a row of the intermediate mesh is not the same for all chains, but depends on its rank. With a binary-reflected Gray code encoding of the chains, dilation three is achieved. To reduce the dilation to two, each row of the intermediate mesh is split and embedded into two symmetrical subcubes. The embedding is done such that every two successive nodes in the same row of the intermediate mesh are assigned to corresponding positions of the two symmetrical subcubes. The order in which successive nodes are assigned to these two subcubes is based on a coloring technique (on a derived bipartite graph) such that the distance between any two corresponding nodes of adjacent rows is at most two. Though both the dilation and expansion are minimal, the technique does not necessarily guarantee minimal congestion, node-congestion, or average dilation, for instance. Hence, the techniques described above may still be competitive.

5 Embedding by graph decomposition

Embedding by graph decomposition is based on the results in Section 2.1. We list the general strategy below followed by the strategy specialized to the embeddings of two- and three-dimensional meshes.

1. If the number of nodes along any axis is a power of two then no factoring of the number of nodes along that axis is performed. The embedding is by a binary-reflected Gray code. For instance, by this, embedding a $12 \times 16 \times 20 \times 32$ mesh is reduced to the problem of embedding 12×20 and 16×32 meshes.

2. For the axes with lengths not being powers of two, a decomposition is sought into meshes for which good embeddings are known, and the products of the expansions for the decomposed meshes is minimized. For instance, the embedding of a 12×20 mesh can be reduced to the embedding of a 3×5 and a 4×4 mesh. Embedding a $3 \times 25 \times 3$ mesh can be reduced to the embedding of two 3×5 meshes.

3. If the axes lengths are not powers of two, but can be increased slightly without increasing the size of the cube for a minimal expansion of the original mesh, then the mesh might be extended, and the procedure just mentioned applied to the extended mesh. For instance, a $3 \times 3 \times 23$ mesh can be extended to a $3 \times 3 \times 25$ mesh, which is treated with the scheme above.

Note that the choice between the last two schemes depends on the configuration of the considered mesh and the existing embeddings.

5.1 Two-dimensional Meshes

5.1.1 Decomposition and direct embedding

Any of the existing two-dimensional embeddings, such as the reshaping techniques [15], the direct embeddings [14], and other methods such as those described in [7], [11], [4] and [2] can be used in combination with the decomposition technique to reduce the average dilation and average congestion, in general. The number of edges with dilation two are approximately proportional to the size of the mesh for most known embeddings. For an $\ell_1 \times \ell_2$ mesh, the maximum of $n_1 + n_2$ is determined such that $\ell_i \leq \ell_i' 2^{n_i}$ and $\lceil \ell_1 \ell_2 \rceil_2 = \lceil \ell_1' \ell_2' 2^{n_1+n_2} \rceil_2$. Then, the problem is reduced to embedding an $\ell_1' \times \ell_2'$ mesh, which can always be done with minimal expansion and dilation two by the method in [4], or possibly any other method. For instance, a 17×17 mesh can be extended to 24×20 and then decomposed into a 3×5 mesh and a $2^3 \times 2^2$ mesh. The congestion for our three direct embeddings is two.

Figure 14 shows the embedding of meshes of the form $5 \cdot 2^{n_1} \times 3 \cdot 2^{n_2}$. In the figure, 'r' and 'c' denote the local addresses within a block (subcube). The local addresses are determined by a binary-reflected Gray code $\tilde{G}(x_i, y_i)$ as defined in Equation 1. Figure 15 shows a specific case: the embedding of a 10×6 mesh into a 6-cube. The orientation of the blocks within each subcube is important for the dilation of the edges. A block is reflected along axis i compared to the individual block embedding, if the rank of the block along axis i is an odd number.

Table 2 gives the average dilation for two-dimensional meshes decomposed into a Boolean cube graph and a mesh for which a direct embedding is given above. The average dilation decreases as the size of the mesh increases.

Figure 16-(f) shows the set of pairs (ℓ_1, ℓ_2) for which the decomposition of a mesh into a Boolean cube graph and a mesh with direct embedding given above yields lower, possibly

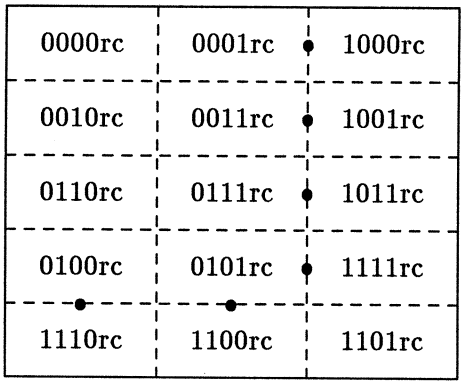


Figure 14: Block addresses of embedding a $5 \cdot 2^{n_1} \times 3 \cdot 2^{n_2}$ mesh into a $(4 + n_1 + n_2)$-cube.

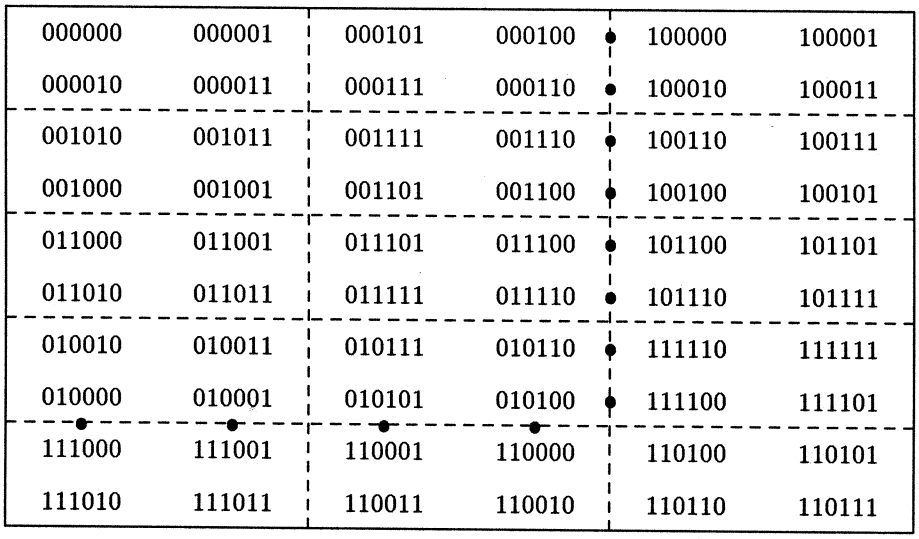


Figure 15: Embedding of a 10×6 mesh into a 6-cube.

$\ell_1 \times \ell_2$ embedding	# of edges w. dil. 2	Average dilation	$\lfloor \ell_i \rfloor_2 = 16$	$\lfloor \ell_i \rfloor_2 = 128$	$\lfloor \ell_i \rfloor_2 = 1024$
$5 \cdot 2^{n_1} \times 3 \cdot 2^{n_2}$	$\frac{4}{5}\ell_1 + \frac{2}{3}\ell_2$	$1 + \frac{12\ell_1+10\ell_2}{15(2\ell_1\ell_2-\ell_1-\ell_2)}$	1.03	1.004	1.0005
$9 \cdot 2^{n_1} \times 7 \cdot 2^{n_2}$	$\frac{1}{3}\ell_1 + \frac{4}{7}\ell_2$	$1 + \frac{7\ell_1+12\ell_2}{21(2\ell_1\ell_2-\ell_1-\ell_2)}$	1.02	1.003	1.0003
$11 \cdot 2^{n_1} \times 11 \cdot 2^{n_2}$	$\frac{5}{11}\ell_1 + \frac{16}{11}\ell_2$	$1 + \frac{55\ell_1+176\ell_2}{121(2\ell_1\ell_2-\ell_1-\ell_2)}$	1.04	1.005	1.0007

Table 2: The average dilation for meshes embedded by graph decomposition into a Boolean cube graph and a mesh embedded by direct mesh embedding.

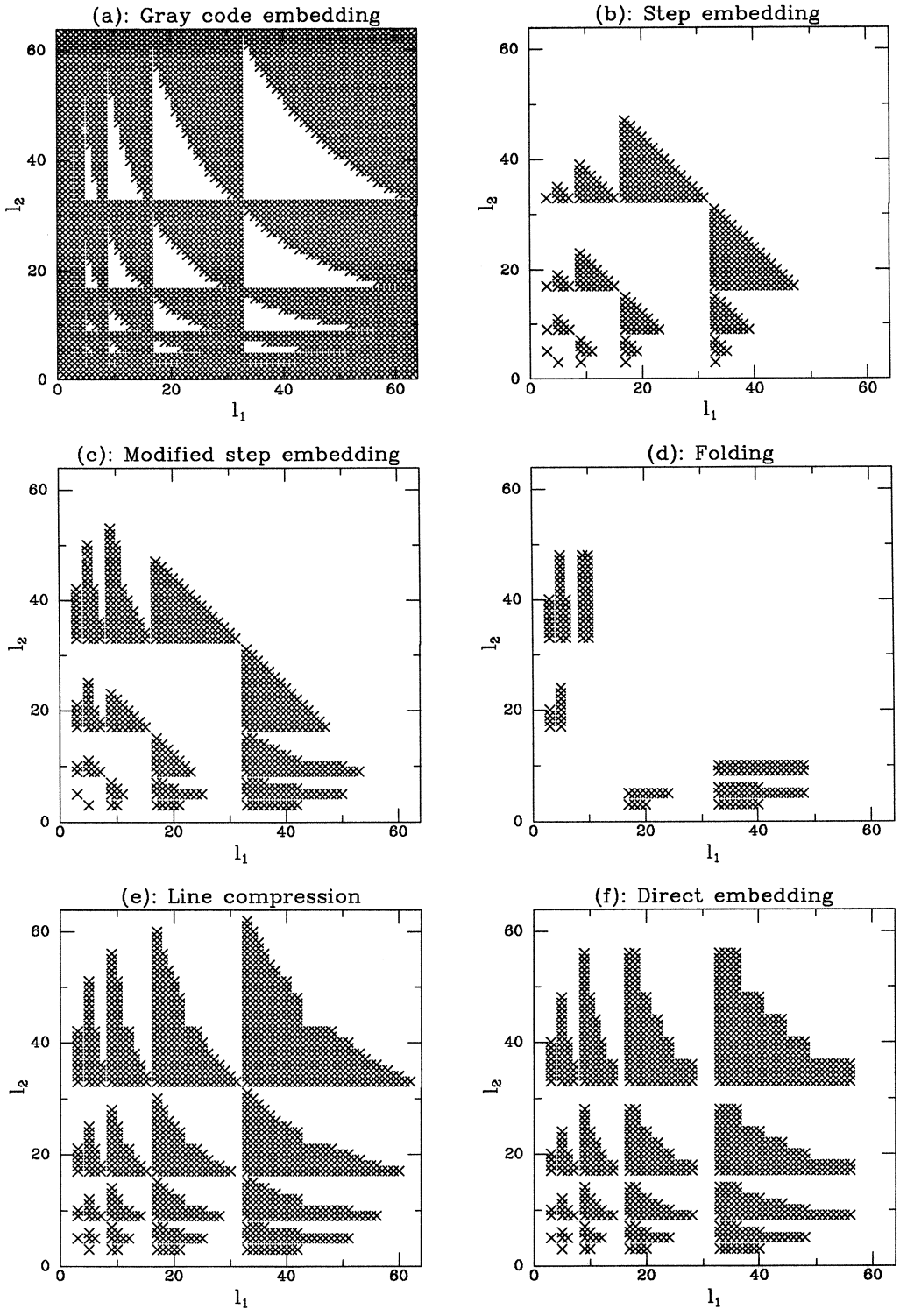


Figure 16: The pairs (ℓ_1, ℓ_2) for which minimal expansion is attained using (a) Gray code embedding, (b) step embedding, (c) modified step embedding, (d) folding, (e) line compression, and (f) direct embedding. For (b) to (f), only the regions for which Gray code embedding can not achieve minimal expansion are considered.

minimal, expansion than a Gray code embedding. The ratio of the number of these pairs to the total number of pairs is $\approx \frac{1}{4}$, independent of the range of $2^n \times 2^n$.

By simply combining the three direct embeddings with Gray code embedding by Equation (1), we have provided minimal expansion, dilation two and congestion two embeddings for about 70% of the meshes for which the (pure) Gray code embedding has an expansion > 2 [14]. The graph decomposition technique together with direct embedding, and Gray code encoding yields dilation-two, congestion-two, minimal-expansion embeddings for 87% of all two-dimensional meshes. The average dilation is 1, asymptotically. The expansion is at most ≈ 2.4 in the worst-case for all two-dimensional meshes. The percentage can actually be increased further by various combinations of the reshaping and direct embeddings and Gray code, by Theorem 2. For instance, a $(2^x + 1)2^{n_1} \times (2^x - 1)2^{n_2}$ mesh is covered by line compression (or combined with Gray code); a 9×25 mesh is embedded by combining two 3×5 meshes.

5.2 Three-dimensional meshes

5.2.1 Decomposition and direct embedding

We have provided two dilation-two embeddings, and one dilation-three embedding for the three-dimensional case. Together with the decomposition technique, and the two-dimensional embedding techniques the majority of three-dimensional meshes are covered. The decomposition and direct embedding may yield minimal-expansion dilation-two embeddings, where the previously mentioned techniques may not. For instance, a $6 \times 11 \times 7$ mesh can be embedded with minimal expansion using this technique, but not Gray code, or a combination of the technique in [4] and Gray code.

Another example where graph decomposition is effective is in the case of embedding a $21 \times 9 \times 5$ mesh. It can be embedded with minimal expansion by combining the $7 \times 9 \times 1$ direct embedding with the $3 \times 1 \times 5$ direct embedding. Another effective decomposition is the product of a $21 \times 3 \times 1$ mesh and a $1 \times 3 \times 5$ mesh.

For three-dimensional meshes it is also possible to perform a dilation two embedding to any pair of axes, and apply Gray code to the third axis. The relative expansion for the three possible choices are

$$\frac{\lceil \ell_1 \ell_2 \rceil_2 \lceil \ell_3 \rceil_2}{\lceil \ell_1 \ell_2 \ell_3 \rceil_2}, \frac{\lceil \ell_2 \ell_3 \rceil_2 \lceil \ell_1 \rceil_2}{\lceil \ell_1 \ell_2 \ell_3 \rceil_2}, \text{ and } \frac{\lceil \ell_3 \ell_1 \rceil_2 \lceil \ell_2 \rceil_2}{\lceil \ell_1 \ell_2 \ell_3 \rceil_2}, \text{ respectively.}$$

The relative expansions are either equal to one or two. Note that more than one relative expansion may be one, such as for a $5 \times 10 \times 11$ mesh, or no relative expansion may be one, such as for a $3 \times 3 \times 25$ mesh. Choosing the two axes that have the lowest values of $\ell_1/\lceil \ell_1 \rceil_2$, $\ell_2/\lceil \ell_2 \rceil_2$, and $\ell_3/\lceil \ell_3 \rceil_2$, for the two-dimensional embedding according to [4] results in the smallest relative expansion of the three choices. For instance, for a $5 \times 6 \times 7$ mesh, the first two axes (of length five and six respectively) should be chosen for the two-dimensional embedding.

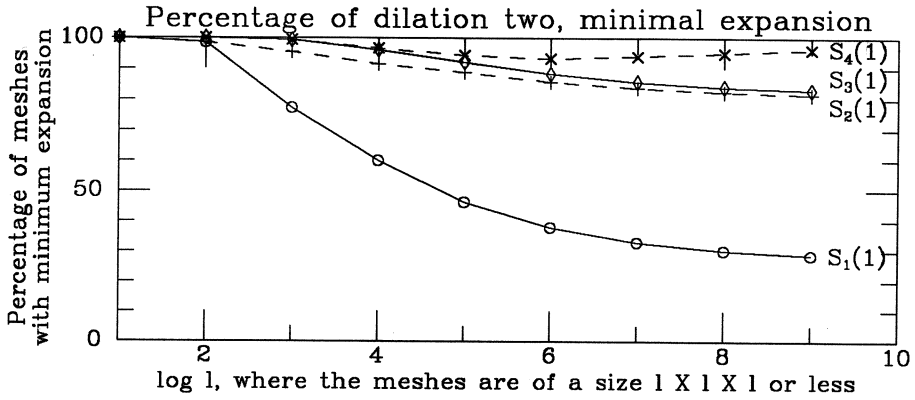


Figure 17: The cumulated percentage of the $\ell_1 \times \ell_2 \times \ell_3$ meshes where $1 \leq \ell_i \leq 2^n$ for $1 \leq n \leq 9$.

5.2.2 The effectiveness of graph decomposition

The fraction of meshes for which the decomposition technique combined with the two- and three-dimensional embedding techniques yield minimal expansion embeddings with a dilation of at most two is given in Figure 17. In the figure, $S_i(\varepsilon)$ is the cumulative percentage of meshes that have a relative expansion ε by applying the embedding methods with an index less than or equal to i below:

1. Apply Gray code embedding.
2. Apply the result in [4] to any pair of axes and apply Gray code to the third axis.
3. Apply the $7 \times 3 \times 3$ (or $3 \times 3 \times 3$) embedding combined with Gray code by Corollary 2.
4. For an $\ell_1 \times \ell_2 \times \ell_3$ mesh, find $\ell_2' \ell_2'' \geq \ell_2$ such that $\lceil \ell_1 \ell_2' \rceil_2 \lceil \ell_2'' \ell_3 \rceil_2 = \lceil \ell_1 \ell_2 \ell_3 \rceil_2$, Theorem 2 and [4]. The procedure is repeated for decomposing ℓ_1 and ℓ_3.

For a mesh of size less than or equal to $512 \times 512 \times 512$, the cumulated percentages grows as the sequence: 28.5%, 81.5%, 82.9%, 96.1%. Applying the method in [4] to any pair of axes, only allows about 81.5% of the meshes to achieve minimal expansion.

The general combining mechanism plays a very important role in achieving an embedding of minimal expansion based on previously known results. Moreover, congestion is preserved through the combining mechanism. Therefore, any mapping combined from the mappings that fall in the 87% domain of the two-dimensional meshes described before have a congestion of one or two.

6 Embeddings for wrap-around meshes

Lemma 5 [22] *Let $\ell_i = \ell_i' \ell_i''$ and ℓ_i be even for all $1 \leq i \leq k$. Then, the $\ell_1 \times \ell_2 \times \cdots \times \ell_k$ wrap-around mesh is a subgraph of the product graph of the $\ell_1' \times \ell_2' \times \cdots \times \ell_k'$ mesh and the $\ell_1'' \times \ell_2'' \times \cdots \times \ell_k''$ mesh (both without wrap-around).*

Proof: Since every $\ell_i' \times \ell_i''$ mesh contains a ring of the same size as a subgraph, if $\ell_i'\ell_i''$ is even [22], the lemma follows from Theorem 1 and Lemma 1. ∎

Let $dil_\varphi(e)$ be the dilation of the edge e under the mapping φ.

Lemma 6 *Let φ_1 be an embedding $G \rightarrow I$ and φ_2 be an embedding $I \rightarrow H$. Then, there exists an embedding function $\varphi : G \rightarrow H$ such that*

$$dil_\varphi(e) = \sum_{e_i \in \varphi_1(e)} dil_{\varphi_2}(e_i).$$

The lemma gives a much tighter upper bound for the dilation than the bound obtained by simply multiplying the two dilations under φ_1 and φ_2.

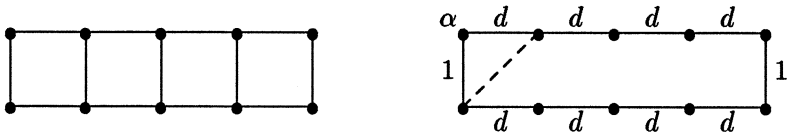
Lemma 7 *An $\ell_1 \times \ell_2 \times \cdots \times \ell_k$ wrap-around mesh M can be embedded into a minimal hypercube with dilation $\leq d+1$, if there exists an embedding φ that maps the $\lceil \ell_1/2 \rceil \times \lceil \ell_2/2 \rceil \times \cdots \times \lceil \ell_k/2 \rceil$ mesh M' into a minimal hypercube with dilation d and $\lceil \prod_{i=1}^{k} \ell_i \rceil_2 = 2^k \lceil \prod_{i=1}^{k} \lceil \ell_i/2 \rceil \rceil_2$. (The dilation is d, if all ℓ_i's are even.)*

Proof: Consider the embedding of a $2\lceil \ell_1/2 \rceil \times 2\lceil \ell_2/2 \rceil \times \cdots \times 2\lceil \ell_k/2 \rceil$ wrap-around mesh $\tilde{M}$. Since the mesh M' can be embedded into a minimal hypercube with dilation d, and a k-cube can be embedded into a k-cube with expansion one and dilation one, it follows from Corollary 1 that the product graph of M' and a k-cube can be embedded into a minimal hypercube with dilation d. Since a k-cube is the same as a k-dimensional mesh of the form $2 \times 2 \times \cdots \times 2$, the product graph of M' and a k-cube contains the wrap-around mesh $\tilde{M}$ as a subgraph, by Lemma 5. Therefore, the wrap-around mesh $\tilde{M}$ can be embedded into a minimal hypercube with dilation d.

We now embed the wrap-around mesh M in the wrap-around mesh $\tilde{M}$ by removing one hyperplane for each coordinate i with ℓ_i being odd. While a hyperplane of mesh nodes is removed from $\tilde{M}$ to become M, the edge in M that passes through the removed hyperplane is simulated by the length-two path in $\tilde{M}$. The hyperplane to be removed is chosen such that the length-two path is mapped to a hypercube with the dilation of edges being one and d, respectively. For instance, the hyperplane to be removed for the ith coordinate can be 0, $\lfloor \ell_i/2 \rfloor$, $\lceil \ell_i/2 \rceil$ or ℓ_i. Since the hyperplane of mesh nodes removed connects to two neighboring hyperplanes through two sets of edges of dilation one and d, respectively, in mapping to a hypercube, the edge of M which is a path of length two in $\tilde{M}$ has a dilation of edges of $d+1$, according to Lemma 6. ∎

Figure 18-(a) demonstrates the ith coordinate of the product graph of the mesh $\tilde{M}$ and the k-cube for which $\lceil \ell_i/2 \rceil = 5$. All the vertical edges have a dilation $\leq d$, and all the horizontal edges have a dilation of one. It is easy to see from Figure 18-(b) that if $\ell_i = 9$, then the node α is removed and the dilations of the two edges incident to the removed node are $\leq d$ and one, respectively. So, the dilation for the new "logical edge" (the dashed edge in the figure) is $d+1$.

Intuitively, the mesh $\tilde{M}$ is partitioned into 2^k blocks of submeshes of the form $\lceil \ell_1/2 \rceil \times \lceil \ell_2/2 \rceil \times \cdots \times \lceil \ell_k/2 \rceil$. The submeshes are labeled $\tilde{M}_i$, $0 \leq i < 2^k$, such that submesh i and



(a) (b)

Figure 18: A linear array of size ℓ_i and ℓ_i being odd, embedded in the product graph of a linear array of size $\lceil \ell_i/2 \rceil$ and a 1-cube, where the latter linear array has a dilation d and the 1-cube has a dilation one in another embedding.

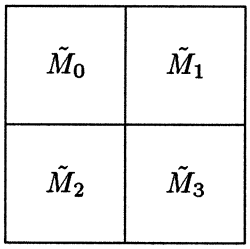


Figure 19: Partitioning for the embedding of an wrap-around mesh.

submesh j are adjacent if $Hamming(i,j) = 1$. The submesh $i = (i_{k-1}i_{k-2}\ldots i_0)$ is reflected with respect to coordinate r for all $i_r = 1$ and for all i's. Then, the same embedding function φ is applied to all submeshes for their embeddings in their respective k-cube. Figure 19 shows the four submeshes for a two-dimensional case, in which the submeshes $\tilde{M}_1$ and $\tilde{M}_3$ are reflected horizontally and the submeshes $\tilde{M}_2$ and $\tilde{M}_3$ are reflected vertically before the embedding function is applied.

Clearly, if all the ℓ_i's are even, then the condition $\lceil \prod_{i=1}^{k} \ell_i \rceil_2 = 2^k \lceil \prod_{i=1}^{k} \lceil \ell_i/2 \rceil \rceil_2$ is satisfied. If this condition holds, then the expansion remains minimal by using a mesh with wrap-around of a slightly larger size (or of the same size) as an intermediate graph.

Lemma 8 *An $\ell_1 \times \ell_2 \times \cdots \times \ell_k$ wrap-around mesh can be embedded into a minimal hypercube with dilation* $\max(d,2)$*, if there exists an embedding that maps the $\lceil \ell_1/4 \rceil \times \lceil \ell_2/4 \rceil \times \cdots \times \lceil \ell_k/4 \rceil$ mesh into a minimal hypercube with dilation d and $\lceil \prod_{i=1}^{k} \ell_i \rceil_2 = 4^k \lceil \prod_{i=1}^{k} \lceil \ell_i/4 \rceil \rceil_2$.*

Proof: Consider the embedding of $4\lceil \ell_1/4 \rceil \times 4\lceil \ell_2/2 \rceil \times \cdots \times 4\lceil \ell_k/4 \rceil$ wrap-around mesh $\tilde{M}$. Apply an argument similar to the one in the proof of Lemma 7.

Figure 20-(a) and (c) shows one axis of the product graph of the mesh $\tilde{M}$ and the k-cube with $\lceil \ell_i/4 \rceil = 5$ and 4, respectively. All the vertical edges have a dilation $\leq d$, and all the horizontal edges have a dilation of one. Figure 20-(b) and (d) show an embedded linear array of size $4\lceil \ell_i/4 \rceil$ (by ignoring the dashed edges). Consider the case where $\ell_i \bmod 4 \neq 0$. We wish to show that by removing one, two and three nodes, respectively, the newly formed "logical edges" have a dilation of $\leq \max(d,2)$. When $\ell_i \bmod 4 = 1$, remove node α. When $\ell_i \bmod 4 = 2$, remove nodes β and γ (but keep node α). When $\ell_i \bmod 4 = 3$, remove all the three nodes α, β and γ. The newly-formed "logical edges" are marked by the dashed edges in the figure. Clearly, all the dashed edges preserve the property of the dilation $\leq \max(d,2)$.

Since the above proof requires that $\lceil \ell_i/4 \rceil \geq 3$, it remains to be proved that if $\lceil \ell_i/4 \rceil = 2$ or 1, the lemma still holds. Figure 20-(e) shows for $\ell_i = 5$, 6, 7 and 8. For $1 \leq \ell_i \leq 4$, the lemma can be derived easily. ∎

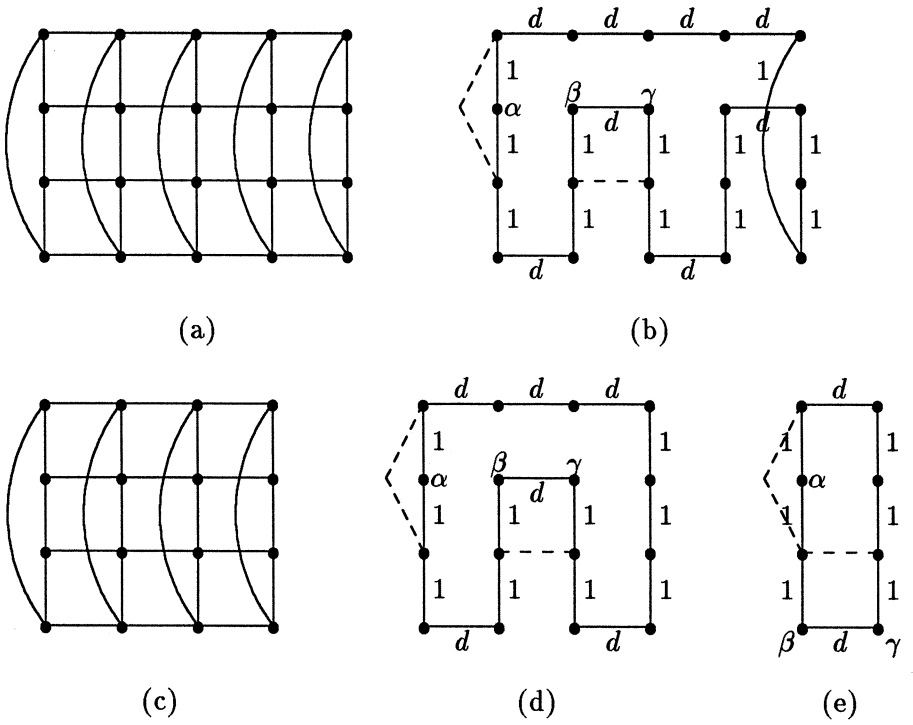


Figure 20: A linear array of size ℓ_i, embedded in the product graph of a linear array of size $\lceil \ell_i/4 \rceil$ and a 2-cube, where the latter linear array has a dilation d and the 2-cube has a dilation one in another embedding.

Note that there exist several ways to embed a ring for Figure 20-(a) and (b) that preserve the dilation of the edges. The selected embedding minimizes the number of horizontal edges used and therefore minimizes the average dilation, in general.

Corollary 4 *Any two-dimensional wrap-around mesh $\ell_1 \times \ell_2$ can be embedded into a minimal hypercube with dilation at most two, if $\lceil \ell_1 \ell_2 \rceil_2 = 16 \lceil \lceil \ell_1/4 \rceil \lceil \ell_2/4 \rceil \rceil_2$ or both ℓ_1 and ℓ_2 are even. Any two-dimensional end-around mesh $\ell_1 \times \ell_2$ can be embedded into a minimal hypercube with dilation at most three, if $\lceil \ell_1 \ell_2 \rceil_2 = 4 \lceil \lceil \ell_1/2 \rceil \lceil \ell_2/2 \rceil \rceil_2$.*

Proof: The former follows from [4], Lemmas 8 and 7. The latter follows from [4] and Lemma 7. ∎

7 Summary

The approach to the mesh embedding problem taken here is that of graph decomposition. The dilation (congestion) of the embedded mesh is the maximum of the dilation (congestion) for the smaller decomposed graphs. Specifically, we show that if an $\ell_1 \times \ell_2 \times \cdots \ell_k$ mesh can be embedded in an n_1-cube with dilation d_1 and congestion c_1, and an $\ell'_1 \times \ell'_2 \times \cdots \times \ell'_k$ mesh can be embedded into an n_2-cube with dilation d_2 and congestion c_2, then the $\ell_1\ell'_1 \times \ell_2\ell'_2 \times \cdots \times \ell_k\ell'_k$ can be embedded into an $(n_1 + n_2)$-cube with dilation $\max(d_1, d_2)$ and congestion $\max(c_1, c_2)$. The graph decomposition technique is likely to yield a smaller

average dilation and average congestion than a direct embedding of the original mesh. Using graph decomposition the average dilation and average congestion are asymptotically one for all two-dimensional meshes.

In particular, we have demonstrated the capability of the graph decomposition technique:

- By applying the graph decomposition technique to the mappings of 3×5, 9×7 and 11×11 meshes and Gray code embedding, 87% of the two-dimensional meshes, asymptotically, can be embedded into a minimal Boolean cube with dilation two and congestion two.
- By applying the graph decomposition technique to the known embeddings for two-dimensional meshes [14, 4] and two three-dimensional meshes presented in the paper, we have attained dilation-two minimal-expansion embeddings into a Boolean cube for 96% of all three-dimensional meshes of a size less than, or equal to, $512 \times 512 \times 512$.

The same approach can be applied to even higher dimensional meshes, and is expected to contain a majority of the domain with dilation two based on the existing two-dimensional and derived three-dimensional mesh embeddings.

The embeddings of wrap-around meshes can be easily constructed out of the embeddings for meshes without wrap-around using the graph decomposition technique. As a special case, for all two-dimensional wrap-around meshes $\ell_1 \times \ell_2$, we have a minimal expansion embedding with dilation two if $\lceil \ell_1 \ell_2 \rceil_2 = 16 \lceil \lceil \ell_1/4 \rceil \lceil \ell_2/4 \rceil \rceil_2$ or both ℓ_1 and ℓ_2 are even; and with dilation three if $\lceil \ell_1 \ell_2 \rceil_2 = 4 \lceil \lceil \ell_1/2 \rceil \lceil \ell_2/2 \rceil \rceil_2$ (where $\lceil x \rceil_2 = 2^{\lceil \log_2 x \rceil}$).

Acknowledgement This work has been supported in part by the Office of Naval Research under Contract N00014-86-K-0310. The authors would like to thank Quentin Stout for pointing out reference [12], Abhiram Ranade for helpful discussion of Lemma 4, David Greenberg for his comments on a draft of this paper, and Eileen Daily for her editorial efforts.

References

[1] Romas Aleliunas and Arnold L. Rosenberg. On embedding rectangular grids in square grids. *IEEE Trans. Computers*, 31(9):907–913, September 1982.

[2] Said Bettayeb, Zevi Miller, and I. Hal Sudborough. Embedding grids into hypercubes. In *Proceedings of '88 AWOC: VLSI, Algorithms and Architectures Conf.*, Springer Verlag's Lecture Notes in Computer Science, no 319, 1988.

[3] J.E. Brandenburg and D.S. Scott. *Embeddings of Communication Trees and Grids into Hypercubes.* Technical Report No. 280182-001, Intel Scientific Computers, 1985.

[4] M.Y. Chan. Dilation-2 embeddings of grids into hypercubes. In *1988 International Conf. on Parallel Processing*, The Pennsylvania State University Press, 1988.

[5] M.Y. Chan. *The Embedding of Grids into Optimal Hypercubes.* Technical Report, Computer Science Dept., University of Texas at Dallas, 1988.

[6] M.Y. Chan. *Embeddings of 3-Dimensional Grids into Optimal Hypercubes.* Technical Report, Computer Science Dept., University of Texas at Dallas, 1988. To appear in the Proceedings of the Fourth Conference on Hypercubes, Concurrent Computers, and Applications, March, 1989.

[7] M.Y. Chan and F.Y.L. Chin. *On Embedding Rectangular Grids.* Technical Report TR-B2-87, Center of Computer Studies and Applications, University of Hong Kong, February 1987. to appear in IEEE Trans. Computers.

[8] Tony Chan. 1986. Personal communication.

[9] John A. Ellis. Embedding rectangular grids into square grids. In *Proceedings of '88 AWOC: VLSI, Algorithms and Architectures Conf.*, Springer Verlag's Lecture Notes in Computer Science, no 319, 1988.

[10] Niall Graham. *Simulations Using the Hypercube Graph.* Technical Report, Computing Research Lab., New Maxico State Univ., 1989. To appear in the Proceedings of the Fourth Conference on Hypercubes, Concurrent Computers, and Applications, March, 1989.

[11] David S. Greenberg. *Minimum Expansion Embeddings of Meshes in Hypercubes.* Technical Report YALEU/DCS/RR-535, Dept. of Computer Science, Yale Univ., New Haven, CT, August 1987.

[12] I. Havel and J. Móravek. B-valuations of graphs. *Czech. Math. J.*, 22:338–351, 1972.

[13] Ching-Tien Ho and S. Lennart Johnsson. Distributed routing algorithms for broadcasting and personalized communication in hypercubes. In *1986 International Conf. on Parallel Processing*, pages 640–648, IEEE Computer Society, 1986. Tech. report YALEU/DCS/RR-483, May 1986.

[14] Ching-Tien Ho and S. Lennart Johnsson. On the embedding of arbitrary meshes in Boolean cubes with expansion two dilation two. In *1987 International Conf. on Parallel Processing*, pages 188–191, IEEE Computer Society, 1987. Report YALEU/DCS/RR-576, April 1987.

[15] S. Lennart Johnsson. Communication efficient basic linear algebra computations on hypercube architectures. *J. Parallel Distributed Comput.*, 4(2):133–172, April 1987. (Tech. Rep. YALEU/DCS/RR-361, Yale Univ., New Haven, CT, January 1985).

[16] S. Lennart Johnsson and Ching-Tien Ho. *Spanning graphs for optimum broadcasting and personalized communication in hypercubes.* Technical Report YALEU/DCS/RR-500, Dept. of Computer Science, Yale Univ., New Haven, CT, November 1986. Revised November 1987, YALEU/DCS/RR-610. To appear in IEEE Trans. Computers.

[17] S. Lennart Johnsson and Peggy Li. *Solutionset for AMA/CS 146.* Technical Report 5085:DF:83, California Institute of Technology, May 1983.

[18] S.R. Kosaraju and M.J. Atallah. *Optimal Simulations between Mesh-Connected Arrays of Processors.* Technical Report, Johns Hopkins University, 1986. appear in JACM, July, 1988, page 635-650.

[19] Ten-Hwang Lai and Alan P. Sprague. *Placement of the Processors of a Hypercube.* Technical Report, Dept. of Computer and Information Science, Ohio State Univ., January 1989.

[20] Charles E. Leiserson. Area-efficient graph layouts (for VLSI). In *Proc. 21st IEEE Symp. Foundations Comput. Sci.*, pages 270–281, IEEE Computer Society, 1980.

[21] Marilynn Livingston and Quentin F. Stout. Embeddings in hypercubes. In *Proceedings Sixth International Conference on Mathematical Modelling*, pages 222–227, 1988. Preprint in August, 1987.

[22] Yuen-Wah E. Ma and Lixin Tao. Embeddings among toruses and meshes. In *1987 International Conf. on Parallel Processing*, pages 178–187, IEEE Computer Society, 1987.

[23] E M. Reingold, J Nievergelt, and N Deo. *Combinatorial Algorithms.* Prentice-Hall, Englewood Cliffs. NJ, 1977.

[24] M. Zubair and S.N. Gupta. *Embeddings on a Boolean Cube.* Technical Report, Dept. of Computer Science, Old Dominion Univ., Norfolk, Virginia, January 1989.