On Embedding Between 2D Meshes of the Same Size

Xiaojun Shen, *Member*, *IEEE*, Weifa Liang, and Qing Hu

Abstract—Mesh is one of the most commonly used interconnection networks and, therefore, embedding between different meshes becomes a basic embedding problem. Not only does an efficient embedding between meshes allow one mesh-connected computing system to efficiently simulate another, but it also provides a useful tool for solving other embedding problems. In this paper, we study how to embed an $s_1 \times t_1$ mesh into an $s_2 \times t_2$ mesh, where $s_i \le t_i$ ($i = 1, 2$), $s_1t_1 = s_2t_2$, such that the minimum dilation and congestion can be achieved. First, we present a lower bound on the dilations and congestions of such embeddings for different cases. Then, we propose an embedding with dilation $\lfloor s_1/s_2 \rfloor + 2$ and congestion $\lfloor s_1/s_2 \rfloor + 4$ for the case $s_1 \ge s_2$, both of which almost match the lower bound $\lceil s_1/s_2 \rceil$. Finally, for the case $s_1 < s_2$, we present an embedding which has a dilation less than or equal to $2\sqrt{s_1}$.

Index Terms—Dilation, embedding, mesh, parallel processing, vertex partition.

1 Introduction

An interconnection network (or network for short) provides connections among processors in a multiprocessor computing system, and plays an important role in the design of parallel algorithms. A network is often represented by a graph G(V, E), where each node represents a processor and an edge represents a communication channel between two processors. The *size* of a network is defined as the number of vertices |V|. Over recent years, many network topologies have been proposed, such as Meshes, Hypercubes, Trees, and Stars. An efficient embedding of network G (called guest) into network H (called host) is a vertex mapping from V(G) to V(H), so that the network H can simulate network G efficiently. An embedding allows an algorithm designed for one network to run on the other network, without redesigning the algorithm. Extensive research has been performed on embeddings, and on the efficiency of embeddings as measured by the four parameters *expansion*, *load*, *dilation*, and *congestion* ([1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], see also Section 2).

An $s \times t$ two-dimensional (2D) mesh, M, is a network in which the vertices can be arranged in a mesh of s rows numbered 0 through $s - 1$ from top to bottom (unless otherwise specified), and t columns numbered 0 through $t - 1$ from left to right. The vertex at row i and column j is denoted $M(i, j)$. Given a mesh, we define the *ratio* as the number of rows over the number of columns. Although higher dimensional meshes can be defined similarly, we will concentrate on 2D meshes in this paper. Among various network topologies, the 2D mesh is one of the most important networks and has received extensive study [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12] because many data structures, especially arrays and matrices, naturally fit into a mesh-connected system. Some well-known real multiprocessor computer systems have been produced based on meshes [18]. The study of embedding between meshes has many applications. A direct application is to allow a mesh to simulate other meshes of various ratios, which means matrices of various shapes can be efficiently mapped to a mesh-connected system. Another application has been pointed out [3], [4] in the design of VLSI, where circuits can be represented by rectangular meshes, which must eventually be manufactured on a square chip. The chip also can be considered a new mesh with a different size and ratio. The critical factors of area and wire length are represented by some embedding parameters. The study of embedding between different meshes also has theoretical impact on other embedding problems. It often serves as an intermediate step to embed a mesh into other networks, such as hypercubes, stars, etc. [5], [6], [7]. For example, Chan's optimal algorithm for embedding [5], [6] a mesh into its ideal hypercube is obtained by first embedding the mesh into another mesh, where the number of rows is equal to a power of two. A similar method is applied in the embedding of meshes into stars [7].

A number of research papers are devoted to embeddings between meshes of different sizes and/or ratios, [1], [2], [3], [4]. Kosaraju and Atallah [1] considered a simulation between k-dimensional meshes, which have the same total number of processors but have different numbers of processors in each dimension. However, they only presented asymptotic lower bounds on the dilations of such embeddings. Aleliunas and Rosenberg [2] dealt with embedding a mesh to another larger sized square mesh (ratio one). They

- *X. Shen is with the Computer Science Telecommunications Program, University of Missouri-Kansas City, 5100 Rockhill Road, Kansas City, MO 64110. E-mail: xshen@cstp.umkc.edu.*
- *W. Liang is with the Department of Computer Science, The Australian National University, Canberra, ACT 0200, Australia. E-mail: wliang@cs.anu.edu.au.*
- *Q. Hu is with FutureNet Technology Inc., 2620 S. California Ave., Monrovia, CA 91016. E-mail: qhu@cstp.umkc.edu.*

For information on obtaining reprints of this article, please send e-mail to: transcom@computer.org, and reference IEEECS Log Number 104663.0.

proposed several embedding schemes and analyzed the relationship between dilation and square size, but didn't guarantee the optimality of dilations. For certain ratios of meshes, Lombardi et al. [3] improved the dilations over the results in [2]. Ellis [4] also studied the problem of embedding a rectangular mesh to the smallest possible square mesh and obtained optimal dilations for some ratios. In general, this method only dealt with square meshes and the expansions were not actually minimized. Some related works can also be found in [5], [6], [7], [8], [9], [10], [11], [12].

In this paper, we will discuss embeddings between 2D meshes of various shapes, but of the same size. Because of the same size, the expansion is minimized. In addition, the embedding scheme for meshes of the same size can easily be extended to deal with meshes of different sizes. Specifically, we consider the embedding of an $s_1 \times t_1$ mesh into an $s_2 \times t_2$ mesh, where $s_i \le t_i$ $(i = 1, 2)$, $s_1 t_1 = s_2 t_2$, $s_1 \ne s_2$, such that both the dilation and congestion are minimized. Since the problem is trivial for the case $s_1 = 1$, we assume $s_1 > 1$. The main results of this paper are:

1) For the case $s_1 > s_2$, obtaining a lower bound $\lceil s_1 / s_2 \rceil$ on both the dilations and the congestions;
2) Introducing the snake-like embedding for the case $s_1 > s_2$ in which the dilation is bounded by $\lfloor s_1 / s_2 \rfloor + 2$ and the congestion is bounded by $\lfloor s_1 / s_2 \rfloor + 4$, almost matching their lower bounds;
3) For the case $1 < s_1 < s_2$, a folding algorithm is presented.

By this embedding, a nontrivial upper bound of dilation is obtained. When $s_2 \bmod s_1 = 0$, this dilation is two, matching its lower bound. The congestion for this case is four. When $s_2 \bmod s_1 \ne 0$, the dilation is upper bounded by $2\sqrt{s_1}$. Therefore, there is still a gap between the upper and lower bounds if $1 < s_1 < s_2$ and $s_2 \bmod s_1 \ne 0$. With results on both dilations and congestions, this journal paper is an improvement on our earlier conference paper [19] where some preliminary results were reported.

The remaining parts of this paper are organized as follows: In Section 2, we prove a lower bound on both dilations and congestions of embedding between meshes. In Section 3, we propose an efficient embedding algorithm called snake-like embedding for 2D meshes, and, in Section 4, we present another embedding algorithm, called folding embedding. In Section 5, we give a brief conclusion.

2 A Lower Bound on Both Dilations and Congestions

An embedding ψ of a guest graph G into a host graph H is a vertex mapping from $V(G)$ to $V(H)$. For any edge $e = (x, y) \in E(G)$, a path from $\psi(x)$ to $\psi(y)$ in H is specified and called the image of e, denoted by $\psi(e)$. The following four parameters are used to evaluate the quality of the embedding ψ: $expansion_\psi = |V(H)|/|V(G)|$; $dilation_\psi = max\{|\psi(e)| \mid e \in E(G)\}$; $load_\psi = max_{v \in V(H)} |\{u \mid u \in V(G) \text{ and } \psi(u) = v\}|$; and $congestion_\psi = max\{cong(e') \mid e' \in E(H)\}$, where $cong(e')$ is the number of edges in G whose images contain e' : $cong(e') = |\{e \mid e' \in \psi(e)\}|$.

The main goal of an embedding is to minimize its expansion, dilation, load, and congestion. Unfortunately, it is very difficult to minimize all of them simultaneously. Thus, we usually try to find a good trade-off among these parameters [16]. In this paper, we will restrict our discussions to embeddings where the load and expansion are equal to one. In searching for efficient embedding algorithms, it is always desirable to know the tight lower bounds on the dilations and congestions for such embeddings. In the following discussion, we will show how to obtain almost tight bounds by using the *vertex partition* method. The vertex partition that will be discussed below is an interesting new technique.

Given a connected graph $G(V, E)$, $|V| = n$, a vertex partition (V_1, V_2) of the vertex set V is called a (p, q) partition, if $|V_1| = p$ and $|V_2| = q$, $p + q = n$. A vertex in V_1 (V_2) is called a *boundary vertex* if at least one of its neighbors belongs to V_2 (V_1). The *boundary sets* of V_1 and V_2 are defined as the sets

$b(V_1) = \{v \mid v \in V_1, v \text{ is a boundary vertex}\}$ and
$b(V_2) = \{v \mid v \in V_2, v \text{ is a boundary vertex}\}$.

The sizes of these two sets, $|b(V_1)|$ and $|b(V_2)|$ are called *boundary lengths.*

In order to obtain a lower bound on the dilations and congestions of embeddings between 2D meshes, we need a lower bound on $bv(V_1, V_2) = max\{|b(V_1)|, |b(V_2)|\}$ for a given (p, q) partition of an $n = s \times t$ $(s \le t)$ mesh M. This problem is similar to the bisection problem [17], which asks for a lower bound on the number of edges between V_1 and V_2. However, the proof method for the bisection problem does not apply to the case of $bv(V_1, V_2)$. We will solve this problem by determining an upper bound on the number of vertices in V_1 given its boundary length $|b(V_1)| = c$.

For ease of presentation, in this section, and only in this section, we assume the row indexing of a mesh M is from bottom to top so that $M(i, j)$ will be consistent with the point (i, j) in a Cartesian system. (For the other sections, we will assume the row indexing is from top to bottom.) The *border* vertices of a mesh are those vertices whose degree is less than four. Let $M(i, j)$ be a vertex at row i and column j, we define *Left Border* = $\{M(i, 0) \mid 0 \le i \le s - 1\}$, *Right Border* = $\{M(i, t - 1) \mid 0 \le i \le s - 1\}$, *Top Border* = $\{M(s - 1, j) \mid 0 \le j \le t - 1\}$, and *Bottom Border* = $\{M(0, j) \mid 0 \le j \le t - 1\}$. Note that each of the four corner vertices, $M(0, 0)$, $M(0, t - 1)$, $M(s - 1, 0)$, $M(s - 1, t - 1)$, belongs to two borders. Two borders are said to be adjacent if they share a common corner vertex.

Let $G(V_1)$ and $G(V_2)$ be subgraphs of $G(V, E)$ induced by V_1 and V_2, respectively. Generally, the subgraphs $G(V_1)$ and $G(V_2)$ consist of several connected components. Given a connected component, we say "this component touches M's border" if at least one vertex of this component is a border vertex of M, and "this component touches M's two (three) borders" if this component contains two (three) vertices in different borders of M.

We observe that, if $bv(V_1, V_2) < s$, then there exists a row which contains only vertices in V_2 (or V_1). Without loss of generality, let this row belong to V_2. Moreover, since $bv(V_1, V_2) < s$ and $s \le t$, there exists at least one column belonging to V_2 also. This row and the column divide M into four disjoint parts. Therefore, we can assume that any connected components of $G(V_1)$ can touch at most two M's borders which are adjacent.

LEMMA 2.1. *Suppose CC is a connected subgraph of M, and*

1) $|b(CC)| < s$, *where* $b(CC)$ *is the boundary set of CC (against the set M-CC).*
2) *CC only touches at most two M's adjacent borders.*

Then, $|CC| \leq c(c+1)/2$, *where* $c = |b(CC)|$.

PROOF. We assume that CC touches M's Left border and Bottom border. If not, we can construct a CC^* by shifting CC such that

1) CC^* touches M's Left border and Bottom border;
2) $|b(CC^*)| \leq |b(CC)|$;
3) $|CC^*| = |CC|$.

Specifically, if CC touches no borders, we can shift it toward the Bottom border without changing CC's shape until CC touches it. Similarly, if CC touches one border, we can shift it along this border until it touches another border. Fig. 1 shows the shifting process. Moreover, if CC touches two borders, we can always assume they are Left and Bottom borders because of the symmetry. Let CC^* be the resulting subgraph after the shifting. It is obvious that $|CC| = |CC^*|$ and $c^* = |b(CC^*)| \leq |b(CC)| = c$. Thus, we only need to prove $|CC^*| \leq c^*(c^*+1)/2$. In other words, we can assume that CC touches M's Left border and Bottom border.

Let $m = max\ \{k \mid M(k, h) \in CC\}$ and $l = max\ \{h \mid M(k, h) \in CC\}$. For any $0 \leq i \leq m$, let $b_r(i)$ be the right-most boundary vertex of CC in row i. Similarly, for any $0 \leq j \leq l$, let $b_c(j)$ be the top-most boundary vertex of CC in column j, as shown in Fig. 2. CC has two important properties.

1) If $b_r(i) = b_c(j) = M(i, j)$ for some i and j, then $CC \cap M' = \{M(i, j)\}$ (a singleton set), where $M' = \{M(i', j') \mid i' \geq i \text{ and } j' \geq j\}$. This is because all vertices $M(i', j)$ $(i' > i)$ and $M(i, j')$ $(j' > j)$ do not belong to CC, and, hence, any vertex in M' except $M(i, j)$ is disconnected from $M(i, j)$ (see Fig. 3). Therefore, $CC \cap M' = \{M(i, j)\}$.
2) If $M(i, j) \in CC$, then $|b(CC)| \geq i + j + 1$. This is proved by the following argument. Let $B_1 = \{b_r(i') \mid 0 \leq i' \leq i\}$ and $B_2 = \{b_c(j') \mid 0 \leq j' \leq j\}$, where $|B_1| = i + 1$ and $|B_2| = j + 1$, $B_1 \cup B_2 \subseteq b(CC)$. We claim that $B_1 \cap B_2 = \Phi$ or $B_1 \cap B_2 = \{M(i, j)\}$. This is because if $b_r(i') = b_c(j') = M(i', j') \in B_1 \cap B_2$ and $M(i', j') \neq M(i, j)$, then we must have $i' < i$ and $j' < j$. By property 1, $M(i, j) \notin CC$, a contradiction. Thus, $|B_1 \cup B_2| = |B_1| + |B_2| - |B_1 \cap B_2| \geq i + j + 1$ and $|b(CC)| \geq |B_1 \cup B_2| \geq i + j + 1$.

Since $|b(CC)| = c$, by Property 2, any $M(i, j) \in CC$ satisfies $i + j \leq c - 1$. Thus, $CC \subseteq M^* = \{M(i, j) \mid i + j \leq c - 1\}$. It is easy to see that $|M^*| = c(c+1)/2$ (M^* is shown in Fig. 4). Therefore, $|CC| \leq |M^*| = c(c+1)/2$. □

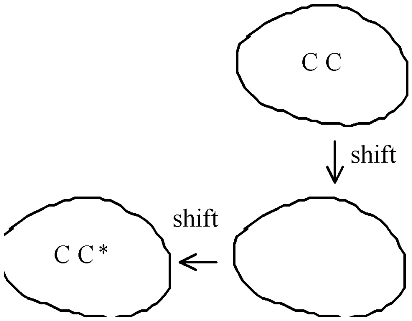


Fig. 1. Shifting *CC* to two adjacent borders.

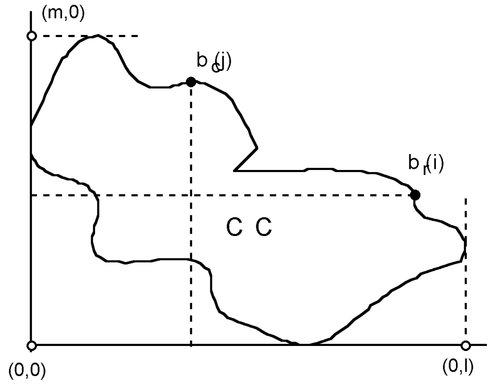


Fig. 2. An illustration of *CC*'s configuration.

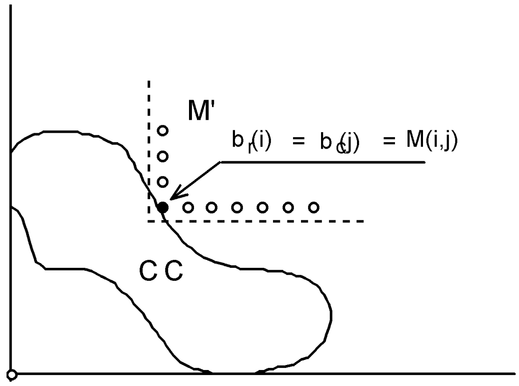


Fig. 3. An illustration of Lemma 2.1.

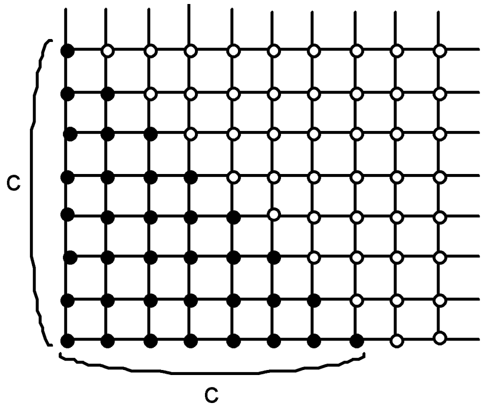


Fig. 4. An illustration of *M**.

COROLLARY 2.1. *Let M be an $s \times t$ mesh ($s \le t$), and (V_1, V_2) be any (p, q) partition of M, $|V_1| = p$, $|V_2| = q$, $p \le q$. If $p > s(s-1)/2$, then $bv(V_1, V_2) \ge s$.*

PROOF. We prove it by contradiction. Suppose $bv(V_1, V_2) \le s - 1$. From previous discussion, we can assume every connected component of $M(V_1)$ touches at most two adjacent borders of M, where $M(V_1)$ is the subgraph induced by V_1. Let $\{CC_1, CC_2, ..., CC_k\}$ be the set of connected components of $M(V_1)$, and $c_i = |b(CC_i)|$ ($1 \le i \le k$). By Lemma 2.1, $|CC_i| \le c_i(c_i + 1)/2$, therefore,

$$|V_1| = p = \sum_{i=1}^{k} |CC_i| \le \sum_{i=1}^{k} c_i(c_i+1)/2.$$

Since $\sum_{i=1}^{k} c_i = |b(V_1)|$,

$$p \le \sum_{i=1}^{k} c_i(c_i+1)/2 \le |b(V_1)|(|b(V_1)|+1)/2.$$

Since $|b(V_1)| \le bv(V_1, V_2) \le s - 1$, $p \le s(s-1)/2$, a contradiction. Obviously, if we assume every connected component of $M(V_2)$ touches at most two adjacent borders, then a contradiction can also be derived. □

Lemma 2.1 and Corollary 2.1 provide the theoretical basis to the proof of the lower bounds on dilations and congestions.

THEOREM 2.1. *Given an $s_1 \times t_1$ mesh M_1 and an $s_2 \times t_2$ mesh M_2, $s_1 \times t_1 = s_2 \times t_2$, $s_i \le t_i$ ($i = 1, 2$), then the tight lower bounds on dilations and congestions of any embedding M_1 into M_2, denoted $L_d(M_1, M_2)$ and $L_c(M_1, M_2)$, respectively, satisfy:*

1) *If $s_1 = 1$, then $L_d(M_1, M_2) = L_c(M_1, M_2) = 1$;*
2) *If $1 < s_1 < s_2$, then $L_d(M_1, M_2) \ge 2$ and $L_c(M_1, M_2) \ge 2$;*
3) *If $s_1 \ge s_2$, then $L_d(M_1, M_2) \ge \lceil s_1/s_2 \rceil$ and $L_c(M_1, M_2) \ge \lceil s_1/s_2 \rceil$.*

PROOF. The proof of 1) is done by using a snake-like embedding (the definition will be given later, in Section 3). For 2), we shall show that M_1 is not isomorphic to M_2. We prove this by contradiction. Suppose M_1 is isomorphic to M_2, and f is a bijection from $V(M_1)$ to $V(M_2)$. The four corner vertices, say u, v, w, x (degree 2) in M_1 must be mapped to the four corner vertices in M_2. Without loss of generality, let the distance between u and v in M_1 be $s_1 - 1$; however, the distance of their images in M_2 is at least $(s_2 - 1) > (s_1 - 1)$, a contradiction. Thus, the dilation of any embedding from M_1 to M_2 is at least two, i.e., $L_d(M_1, M_2) \ge 2$.

The proof of $L_c(M_1, M_2) \ge 2$ is quite simple. Let f be an arbitrary embedding from M_1 to M_2. Since the dilation of f is at least two, there exists an edge (a, b) in M_1 such that its image in M_2 has length at least two. Assume $f((a, b))$ = path (a', b'), where $a' = f(a)$, $b' = f(b)$, and vertex c' is an intermediate vertex on the path (a', b'). The path (a', b') uses exactly two out of the four edges incident with c'. Let $c \in M_1$ be the source of c', $f(c) = c'$ as shown in Fig. 5. Then, each of the images of the four edges incident with c in M_1 must contain one of the four edges incident with c' in M_2, which means that at least one of those four edges in M_2 is contained in at least two paths. Thus, 2) holds.

Now we prove 3). We assume $s_1 > s_2$ since $s_1 = s_2$ implies M_1 and M_2 are identical. We construct a vertex partition of M_2, (V_1, V_2), by deleting all edges between column $\lfloor t_2/2 \rfloor - 1$ and column $\lfloor t_2/2 \rfloor$. Fig. 6. shows two examples for such partitions. Let the vertices on the left side be V_1 and those on the right side be V_2. It is easy to see that $|V_1| = s_2(\lfloor t_2/2 \rfloor) \le |V_2|$.

Let f be any embedding from M_1 to M_2, and (V_1, V_2) be the partition of M_2 described above. Then, $(f^{-1}(V_1), f^{-1}(V_2))$ corresponds to a vertex partition of M_1. It is obvious that $|f^{-1}(V_1)| = |V_1|$ and $|f^{-1}(V_2)| = |V_2|$, because of the bijection f. When t_2 is even, $|V_1| = s_2t_2/2 = s_1t_1/2 > s_1(s_1 - 1)/2$. When t_2 is odd, $|V_1| = s_2(t_2 - 1)/2 = s_2t_2/2 - s_2/2 = s_1t_1/2 - s_2/2 > s_1(s_1 - 1)/2$ since $s_1 > s_2$. By Corollary 2.1, $bv(f^{-1}(V_1), f^{-1}(V_2)) \ge s_1$. Without loss of generality, let $b(f^{-1}(V_1)) \ge s_1$, i.e., $f^{-1}(V_1)$ has s_1 boundary vertices which are adjacent to some vertices of $f^{-1}(V_2)$. In other words, there are at least s_1 edges between $f^{-1}(V_1)$ and $f^{-1}(V_2)$. The images of those s_1 boundary vertices in V_1 must be distributed among at least $\lceil s_1/s_2 \rceil$ columns of M_2, which means one of those images has distance $\ge \lceil s_1/s_2 \rceil$ to reach a vertex in V_2, i.e., at least one edge in M_1 will be mapped to a path in M_2 whose length is at least $\lceil s_1/s_2 \rceil$. Therefore, $L_d(M_1, M_2) \ge \lceil s_1/s_2 \rceil$.

In addition, each image of those s_1 edges between $f^{-1}(V_1)$ and $f^{-1}(V_2)$ must go through an edge between V_1 and V_2 in M_2. There are s_2 such edges, therefore, at least one such edge will be shared by at least $\lceil s_1/s_2 \rceil$ images under embedding f, i.e., $L_c(M_1, M_2) \ge \lceil s_1/s_2 \rceil$. □

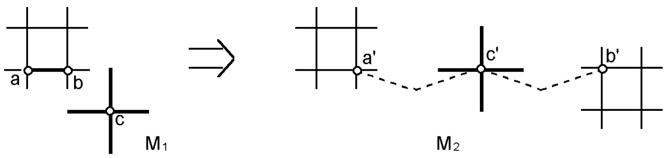


Fig. 5. The images of the five edges (by dark lines) in M_1 must share the four edges (by dark lines) in M_2.

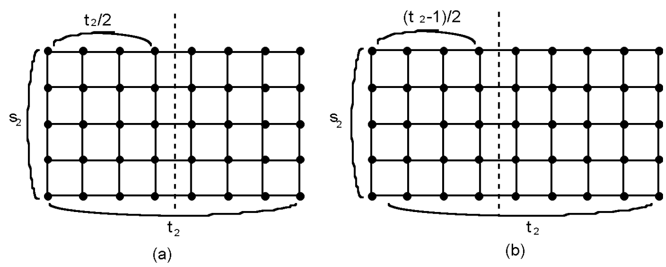


Fig. 6. The partition of M_2: (a) t_2 is an even number; (b) t_2 is an odd number.

3 SNAKE-LIKE EMBEDDING FOR THE CASE $s_1 > s_2$

In this section, we discuss the embedding problem for the case $s_1 > s_2$. The case $s_1 < s_2$ will be discussed in the next section. (Note that M_1 and M_2 are identical if $s_1 = s_2$.) Let M_1 be an $s_1 \times t_1$ mesh, and M_2 be an $s_2 \times t_2$ mesh, where $s_1 \times t_1 = s_2 \times t_2$, $s_i \le t_i$, (i = 1, 2), $s_1 > s_2$. In the following discussion, ψ

is used to denote a mapping from M_1 to M_2, and $\psi(i, j) = (h, k)$ means $M_1(i, j)$ is mapped to $M_2(h, k)$. We also specify $\psi(e)$ where edge $e = (M_1(i, j), M_1(i', j'))$ as the shortest path from $\psi(i, j)$ to $\psi(i', j')$ which takes the horizontal direction first and then turns into the vertical direction. Let $s_1 = \rho s_2 + r$, where $0 \le r < s_2$. We distinguish two cases.

The first case: $r = 0$, $s_1 = \rho s_2$. Because $s_1 t_1 = s_2 t_2$, $t_2 = \rho t_1$, we divide the columns of M_2 into t_1 groups with each group containing consecutive ρ columns. The ρ columns in each group contain exactly $\rho s_2 = s_1$ vertices, forming an $s_2 \times \rho$ submesh. Now, we embed each column of M_1 into a group of columns of M_2 such that two adjacent columns of M_1 will be mapped to two adjacent groups of M_2. This embedding takes snake-like shape as shown in Fig. 7. Namely, a column of M_1 is partitioned into s_2 segments with ρ vertices in each segment, and each segment is mapped to a row of the submesh of M_2, taking the direction from left to right and from right to left alternatively. It is easy to see that both the dilation and the congestion of the snake-like embedding are optimal and equal to $\rho = s_1 / s_2$. The snake-like embedding function ψ for this case has the following algebraic expression:

$$\psi(i, j) = \left(P, j\rho + g(Q, P)\right),$$

where $i = \rho P + Q$, $P \ge 0$ and $0 \le Q < \rho$, and

$$g(Q, P) = \begin{cases} Q, & P \text{ is even}, \\ \rho - Q - 1, & P \text{ is odd}. \end{cases}$$

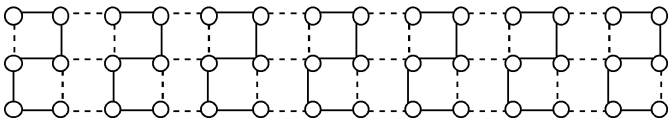


Fig. 7. An example of the snake-like embedding of a 6 × 7 mesh into a 3 × 14 mesh, where the solid lines represent the connections of M_1 and the dotted lines represent the connections of M_2.

REMARK. Suppose we also divide each row of M_2 into segments of ρ vertices each. Let $P_{M2}(i, j)$ be the jth segment of the ith row of M_2, and $P_{M1}(i, j)$ be the ith segment of the jth column of M_1 ($0 \le i \le s_2 - 1$, $0 \le j \le t_1 - 1$). Then, the above embedding is to map the column segment $P_{M1}(i, j)$ to the row segment $P_{M2}(i, j)$. Of course, alternative directions should be used for segments in the same column of M_1. If i is even, we take left to right direction, otherwise, right to left direction.

The second case: $r > 0$, $s_1 = \rho s_2 + r$, $0 < r < s_2$. We will apply the same approach as that for the first case. Since s_1 is not a multiple of s_2, we cannot divide a column into segments of equal length. Thus, we divide each column of M_1 into s_2 segments such that the length of each segment is either ρ or $\rho + 1$. Obviously, there are exactly r segments which contain $\rho + 1$ vertices each. Let $P_{M1}(i, j)$ be the ith segment of the jth column of M_1 ($0 \le i \le s_2 - 1$, $0 \le j \le t_1 - 1$). Since $t_2 = t_1 s_1 / s_2 = t_1(\rho s_2 + r) / s_2 = t_1 \rho + r t_1 / s_2 = t_1 \rho + q$ ($q = r t_1 / s_2$, $0 < q < t_1$). Therefore, we can similarly partition each row of M_2 into t_1 segments of which q segments contain $\rho + 1$ vertices each and the others contain ρ vertices each. Let $P_{M2}(i, j)$ be the jth segment of the ith row of M_2. There are total $r t_1$ column segments in M_1 and $q s_2$ ($= r t_1$) row segments in M_2 which contain $\rho + 1$ vertices. We call them ($\rho + 1$)-segments. We will map $P_{M1}(i, j)$ to $P_{M2}(i, j)$, ($0 \le i \le s_2 - 1$, $0 \le j \le t_1 - 1$). The key question is how to determine which segments should be ($\rho + 1$)-segments and which ones should be ρ-segments so that the dilation is minimized.

Obviously, it must be required that $P_{M1}(i, j)$ is a ($\rho + 1$)-segment if and only if $P_{M2}(i, j)$ is a ($\rho + 1$)-segment. We construct a matrix $[D(i, j)]$, ($0 \le i \le s_2 - 1$, $0 \le j \le t_1 - 1$) where $D(i, j) \in \{0, 1\}$, and $D(i, j) = 1$ indicates that $P_{M1}(i, j)$ and $P_{M2}(i, j)$ will be assigned a ($\rho + 1$)-segment, otherwise, they will be assigned a ρ-segment. Obviously, there should be exactly r ones in each column and q ones in each row of D.

To reduce the dilation, we will distribute ones in the matrix D such that the two ith segments of two adjacent rows, $P_{M2}(i, j)$ and $P_{M2}(i, j + 1)$, will not be out of alignment too much because of the accumulative ones in each row. Also, we want the two jth segments of two adjacent columns to be as close as possible. Let $x = gcd(s_2, t_1)$, where function $gcd(p, q)$ computes the greatest common divisor of p and q. Thus, $s_2 = dx$, $t_1 = cx$, and $gcd(c, d) = 1$. Since s_2 divides $r t_1$ ($q s_2 = r t_1$), we have $rcx = qdx$, or $rc = qd$. Since $gcd(c, d) = 1$, c divides q. Assuming $y = q / c$, then $r = yd$. Matrix D can be divided into $d \times c$ blocks, each of which is an $x \times x$ submatrix, as shown in Fig. 8a. There will be total of $r t_1$ ones in matrix D. We distribute these ones in matrix D such that each block will be assigned xy ($= r t_1 / cd$) ones. The distribution of the xy ones within a block is shown in Fig. 8b.

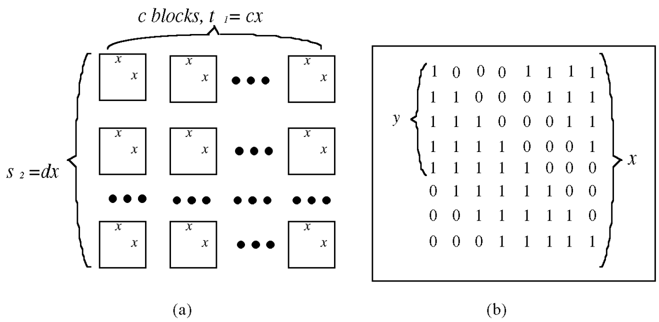


Fig. 8. (a) Matrix D is partitioned into $d \times c$ blocks of size $x \times x$. (b) The distribution of $x \times y$ ones within a block, where $x = 8$, $y = 5$.

The algebraic expression for this distribution is given below. For the 0th column of D, we define

$$D(i, 0) = \begin{cases} 1 & i \bmod x \le y - 1 \\ 0 & i \bmod x > y \end{cases} \quad 0 \le i < s_2.$$

The distribution of ones in the remaining columns is done by the following iterative formula:

$$D(i, k) = \begin{cases} D(i - 1, k - 1), & i \ne 0 \\ D(s_2 - 1, k - 1), & i = 0. \end{cases} \quad 0 < k < t_1,\ 0 \le i < s_2.$$

It is easy to see that $\sum_{i=0}^{s_2 - 1} D(i, k) = yd = r$ for any column k

and $\sum_{j=0}^{t_1-1} D(k,j) = yc = q$ for any row k of M.

The snake-like embedding of M_1 to M_2 will be done with the following steps:

1) Construct an $s_2 \times t_1$ matrix D as shown above;
2) Partition each column of M_1 and each row of M_2 according to the distribution of ones in D;
3) Map $P_{M1}(i, j)$ to $P_{M2}(i, j)$ taking left to right direction if i is even, and taking right to left direction otherwise.

Let $\psi(i, j) = (h, k)$. From the above discussion, it is not difficult to give formal expressions of h and k as follows.

1) Determine h such that the following condition is satisfied:

$$\sum_{u=0}^{h-1}\left(\rho + D(u,j)\right) \le i < \sum_{u=0}^{h}\left(\rho + D(u,j)\right);$$

2) Let $\delta = i - \sum_{u=0}^{h-1}(\rho + D(u,j))$, $0 \le \delta < \rho + D(h, j)$, where δ is the rank of $M_1(i, j)$ in its segment $P_{M1}(h, j)$. Then k can be determined by

$$k = \begin{cases} \sum_{v=0}^{j-1}\left(\rho + D(h,v)\right) + \delta, & h \text{ is even,} \\ \sum_{v=0}^{j}\left(\rho + D(h,v)\right) - \delta - 1, & h \text{ is odd.} \end{cases}$$

Fig. 9 is an example of such an embedding, and Fig. 10 presents a more general graphical example.

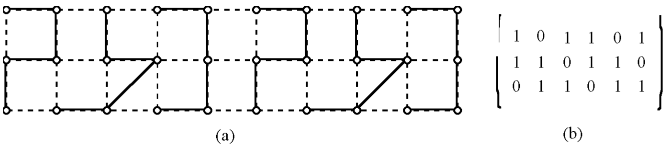


Fig. 9. (a) An example of embedding a 5×6 mesh M_1 into a 3×10 mesh M_2; (b) The matrix D used for the embedding in (a).

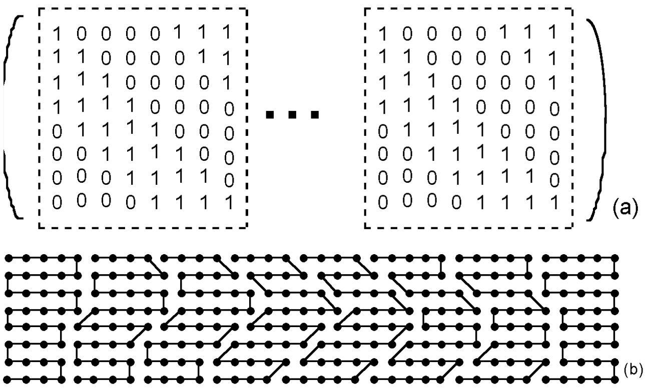


Fig. 10. Embedding a 36×40 mesh into an 8×180 mesh where $\rho = 4$, $x = 8$, $d = 1$, $c = 5$, $y = 4$, $q = 20$, dilation = 6. Note that the 8×40 matrix D is partitioned into five blocks. (a) is the matrix D and (b) illustrates the embedding within a single block.

What remains is to show that the dilation and the congestion of this snake-like embedding almost match their lower bounds. In the following, we will show a lemma which characterizes important properties of the matrix D. This lemma guarantees that the two ith segments of two adjacent rows, $P_{M2}(i, j)$ and $P_{M2}(i, j + 1)$, will be aligned very well, and so will the two jth segments of two adjacent columns. Based on this lemma, the calculation of the dilation and congestion will be straightforward.

LEMMA 3.1. *The matrix D satisfies the following inequalities*:

1) *for any two adjacent columns, j and j* + 1,

$$\left|\sum_{i=0}^{k} D(i,j) - \sum_{i=0}^{k} D(i,j+1)\right| \le 1, \text{ where } 0 \le j < t_{i-1},\ 0 \le k < s_2 - 1; \tag{3-1}$$

2) *for any two adjacent rows, i and i* + 1,

$$\left|\sum_{j=0}^{k} D(i,j) - \sum_{j=0}^{k+1} D(i+1,j)\right| \le 1, \text{ where } 0 \le i < s_2 - 2,\ 0 \le k < t_1 - 2, \tag{3-2}$$

$$\left|\sum_{j=0}^{k} D(i,j) - \sum_{j=0}^{k} D(i+1,j)\right| \le 1, \text{ where } 0 \le i < s_2 - 2,\ 0 \le k < t_1 - 1. \tag{3-3}$$

PROOF. Because $D(i, j) - D(i + 1, j + 1) = 0$ (from the definition of D),

$$\left|\sum_{i=0}^{k} D(i,j) - \sum_{i=0}^{k} D(i,j+1)\right|$$
$$= \left|D(k,j) - D(0,j+1) + \sum_{j=0}^{k-1}\left(D(i,j) - D(i+1,j+1)\right)\right| \le 1,$$
$$\left|\sum_{j=0}^{k} D(i,j) - \sum_{j=0}^{k+1} D(i+1,j)\right|$$
$$= \left|-D(i+1,\ 0) + \sum_{j=0}^{k}\left(D(i,j) - D(i+1,j+1)\right)\right| \le 1,$$

and

$$\left|\sum_{j=0}^{k} D(i,j) - \sum_{j=0}^{k} D(i+1,j)\right|$$
$$= \left|-D(i+1,\ 0) + D(i+1,k+1) + \sum_{j=0}^{k}\left(D(i,j) - D(i+1,j+1)\right)\right|$$
$$\le 1.$$

□

THEOREM 3.1. *Given an* $s_1 \times t_1$ *mesh* M_1 *and an* $s_2 \times t_2$ *mesh* M_2, $s_i \le t_i$, $(i = 1, 2)$, *and* $s_1 > s_2$. *There exists an embedding* ψ *of* M_1 *to* M_2 *with dilation*$_\psi \le \lfloor s_1/s_2 \rfloor + 2$, *and congestion*$_\psi \le \lfloor s_1/s_2 \rfloor + 4$.

PROOF. The proof is done by a direct calculation of dilation$_\psi$ and congestion$_\psi$. The detailed but lengthy calculations are given in the Appendix. □

For convenience, we define function ζ as follows:

$$\zeta(x,y) = \begin{cases} \lfloor x/y \rfloor + 2, & x \bmod y \neq 0, \\ x/y, & x \bmod y = 0. \end{cases}$$

The dilation and congestion of the snake-like embedding for $s_1 > s_2$ equal $\zeta(s_1, s_2)$ and $\zeta(s_1, s_2) + 2$, which differ from the lower bound $\lceil s_1/s_2 \rceil$ by one and three, respectively. Therefore, the snake-like embedding ψ is almost optimal in both dilation and congestion.

4 Folding Embedding for the Case $s_1 \leq s_2$

4.1 A Folding Embedding for the Case $s_2 = \rho s_1$ ($\rho \geq 1$)

Before presenting our algorithm for the general case $s_1 \leq s_2$, we introduce an embedding algorithm for the case $s_2 = \rho s_1$ ($\rho \geq 1$). A naive method is to divide the ith row of M_1 into ρ segments of length t_2 and snake-like embed them into the following ρ rows of M_2: the ith, $(s_1 + i)$th, $(2s_1 + i)$th, ..., $((\rho - 1)s_1 + i)$th. However, this embedding has dilation s_1, because two adjacent segments are separated by $(s_1 - 1)$ rows in M_2. Aleliunas and Rosenberg [2] proposed a folding embedding which is an improvement over the naive embedding. The folding embedding "bends" each segment on its right (or left) part so two adjacent segments will be joined at their ends smoothly. Fig. 11 illustrates this kind of embedding.

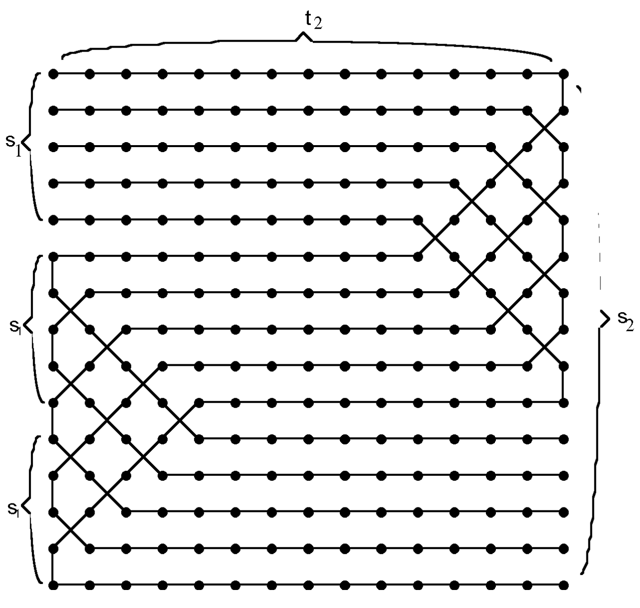


Fig. 11. An illustration of folding embedding which embeds a 5 × 45 mesh into a 15 × 15 mesh.

Let $j = Qt_2 + R$, $Q \geq 0$ and $0 \leq R < t_2$. The explicit mapping function $\psi(i, j) = (h, k)$ for the folding embedding is given below.

$$h = \begin{cases} 2i + Qs_1 + R - t_2 + 1, & Q < \rho - 1 \text{ and } R > t_2 - i + 1, \\ 2i + (Q-1)s_1 + R + 1, & Q > 0 \text{ and } R < s_1 - i + 1, \\ i + Qs_1, & \text{otherwise.} \end{cases}$$

$$k = \begin{cases} R, & Q \text{ is even,} \\ t_2 - R, & Q \text{ is odd.} \end{cases}$$

By the definition of function ψ, we know, if $s_2 \; mod \; s_1 = 0$, then the dilation of the folding embedding equals two. When $s_1 = 1$, the dilation is one. Obviously, this folding embedding is an optimal embedding for the case $s_2 = \rho s_1$. Note that a necessary condition for folding an $s_1 \times t_1$ mesh into an $s_2 \times t_2$ mesh is $s_1 \leq t_2$, for example, a 7 × 9 mesh cannot be embedded into a 21 × 3 mesh by such folding, although seven divides 21. However, for our problem, this condition is always satisfied because $s_1 < s_2 \leq t_2$. It is easy to see that the congestion of such foldings is four.

4.2 Embedding for the General Case $s_1 < s_2$

Although the folding embedding is an efficient method when $s_2 \; mod \; s_1 = 0$, it cannot be applied to the case $s_2 \; mod \; s_1 \neq 0$. Here we introduce two methods for this case. The first method is: Take M_1 as a $t_1 \times s_1$ mesh M_1' and M_2 as a $t_2 \times s_2$ mesh M_2', then embed M_1' into M_2' by snake-like embedding, as shown in Fig. 12. By this method, we obtain an embedding with dilation $\zeta(t_1, t_2) = \zeta(s_2, s_1)$. Note that $\lceil t_1/t_2 \rceil$ is not a lower bound for this case because the assumption $t_i \leq s_i$ is not satisfied. Two is a trivial lower bound for this case.

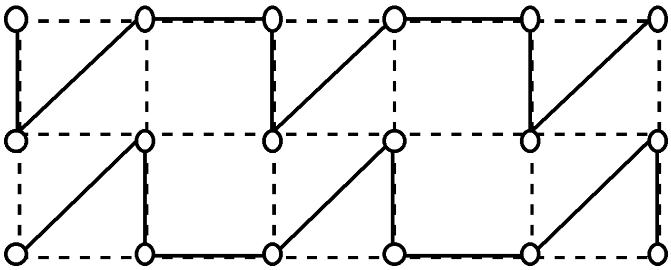


Fig. 12. An example of row snake-like embedding, where M_1 is a 2 × 9 mesh and M_2 is a 3 × 6 mesh.

The second method is to embed M_1 into an $h \times m$ ($h \times m = s_1 \times t_1 = s_2 \times t_2$) auxiliary mesh M' with the same size first, then, embed M' into M_2. The construction of M' should satisfy following relations.

$$h \leq s_1; \tag{4-1}$$

$$s_2 \; mod \; h = 0 \text{ or } t_2 \; mod \; h = 0. \tag{4-2}$$

Note that $h \leq s_1$, we can embed M_1 into M' with dilation $\zeta(s_1, h)$ by snake-like embedding. Meanwhile, since $s_2 \; mod \; h = 0$ (or $t_2 \; mod \; h = 0$) and $h \leq s_1 < s_2$, we also can embed M' into M_2 with dilation two by folding embedding. Thus, we finally obtain an embedding of M_1 into M_2 with dilation $2\zeta(s_1, h)$.

Let I be the set of integers satisfying (4-1) and (4-2) and h be an element of I which minimizes $\zeta(s_1, h)$, i.e., for any $h' \in I$, $\zeta(s_1, h) \leq \zeta(s_1, h')$. Since $gcd(s_1, s_2) \in I$ and $gcd(s_1, t_2) \in I$, $\zeta(s_1, h) \leq min\{\zeta(s_1, gcd(s_1, s_2)), \zeta(s_1, gcd(s_1, t_2))\} = min\{s_1/gcd(s_1, s_2), s_1/gcd(s_1, t_2)\}$. In order to estimate the upper bound of $\zeta(s_1, h)$, we claim that

$$\min\{s_1/\gcd(s_1,s_2), s_1/\gcd(s_1,t_2)\} \leq \sqrt{s_1}.$$

Since $s_1 \times t_1 = s_2 \times t_2$, s_1 can be expressed by $a \times b$, where s_2

TABLE 1
THE CURRENT LOWER BOUNDS AND UPPER BOUNDS ON DILATIONS AND CONGESTIONS OF EMBEDDINGS BETWEEN 2D MESHES OF THE SAME SIZE FOR EACH CASE

case		lower bound of dilation	upper bound of dilation	lower bound of congestion	upper bound of congestion
$s_2 > s_1$	$s_2 = \rho s_1$	ρ	ρ	ρ	ρ
	$s_2 = \rho s_1 + r$	$\rho+1$	$\rho+2$	$\rho+1$	$\rho+4$
$s_2 < s_1$	$s_1 = \rho s_2$	2	2	2	4
	$s_1 = \rho s_2 + r$	2	$\min\{2\zeta(s_1,h), \zeta(s_2,s_1)\}$	2	N/A

mod $a = 0$ and t_2 *mod* $b = 0$. Without loss of generality, we assume $a \ge b$ and $a \ge \sqrt{s_1}$. Thus,

$$\min\{s_1/\gcd(s_1,s_2), s_1/\gcd(s_1,t_2)\} \le s_1/\gcd(s_1,s_2)$$
$$\le s_1/a \le s_1/\sqrt{s_1} = \sqrt{s_1}.$$

From the above discussion, we have an embedding from M_1 into M_2 with dilation $2\zeta(s_1, h)$ which is bounded by $2\sqrt{s_1}$. Between the two embedding methods for the case $s_1 < s_2$, of course, we should choose one which yields a smaller dilation. Therefore, Theorem 4.1 follows.

THEOREM 4.1. *Given an $s_1 \times t_1$ mesh M_1 and an $s_2 \times t_2$ mesh M_2, $s_i \le t_i$, (i = 1, 2), and $s_1 < s_2$. There exists an embedding of M_1 into M_2 with dilation* $\text{dilation}_\psi = \min\{2\zeta(s_1, h), \zeta(s_2, s_1)\}$*, where h satisfies* (4-1) *and* (4-2) *and yields the minimum $\zeta(s_1, h)$.*

What the tight upper bound on dilations and congestions of such embeddings is remains an open problem.

5 CONCLUSION

In this paper, we have studied how to embed an $s_1 \times t_1$ mesh M_1 into an $s_2 \times t_2$ mesh M_2, where $s_1 \times t_1 = s_2 \times t_2$, $s_i < t_i$ (i = 1, 2), such that the dilation and the congestion are minimized. It is assumed that the load equals one and the expansion equals one. We have shown for each case a lower bound on the dilation and a lower bound on the congestion. For the case $s_1 > s_2$, the corresponding embedding algorithm almost achieves the lower bounds. For the case $s_1 < s_2$, an embedding algorithm is given. It achieves optimal dilation two and a small congestion four if s_2 is a multiple of s_1, otherwise, the dilation by this embedding algorithm will be upper bounded by $2\sqrt{s_1}$. The results for dilations and congestions are summarized in Table 1.

APPENDIX

Proof of Theorem 3.1

The proof will be done by directly calculating the dilation and the congestion.

1) The calculation of the dilation.

There are two kinds of edges in M_1, horizontal edges and vertical edges. Each vertical edge in M_1 will be mapped to a horizontal or a vertical edge or an "oblique" edge in M_2 by ψ. Because of Lemma 3.1, the distance of two adjacent vertices connected by an oblique edge is two. Thus, the dilation of ψ is determined by $max\{|\psi(e)| \mid e$ is a horizontal edge in $M_1\}$.

For any horizontal edge $e = (M_1(i_1, j_1), M_1(i_1, j_1 + 1))$, let $\psi(i_1, j_1) = (h_1, k_1)$ and $\psi(i_1, j_1 + 1) = (h_2, k_2)$. $|\psi(e)| = |k_1 - k_2| + |h_1 - h_2|$. By the definition, we know

$$\sum_{i=0}^{h_1-1}\left(\rho + D(i, j_1)\right) \le i_1 < \sum_{i=0}^{h_1}\left(\rho + D(i, j_1)\right)$$
$$\sum_{i=0}^{h_2-1}\left(\rho + D(i, j_1+1)\right) \le i_1 < \sum_{i=0}^{h_2}\left(\rho + D(i, j_1+1)\right) \quad \text{(A-1)}$$

Let

$$r_1 = i_1 - \sum_{i=0}^{h_1-1}\left(\rho + D(i, j_1)\right);\ r_2 = i_1 - \sum_{i=0}^{h_2-1}\left(\rho + D(i, j_1+1)\right). \quad \text{(A-2)}$$

From (3-1), we have either $h_2 = h_1$ or $|h_1 - h_2| = 1$.

LEMMA A.1. *If $h_2 = h_1 + 1$, then $r_1 = \rho + D(h_1, j_1) - 1$, $r_2 = 0$; if $h_1 = h_2 + 1$, then $r_2 = \rho + D(h_2, j_2) - 1$, $r_1 = 0$.*

PROOF. Without loss of generality, we assume $h_2 = h_1 + 1$. If $r_1 < \rho + D(h_1, j_1) - 1$, then, from the definition of r_1,

$$r_1 = i_1 - \sum_{i=0}^{h_1-1}\left(\rho + D(i, j_1)\right) < \rho + D(h_1, j_1) - 1,$$

thus, $i_1 < \sum_{i=0}^{h_1}(\rho + D(i, j_1)) - 1$. From (A-1) and $h_2 = h_1 + 1$,

$$\sum_{i=0}^{h_1}\left(\rho + D(i, j_1+1)\right) \le i_1.$$

Therefore,

$$\sum_{i=0}^{h_1}\left(\rho + D(i, j_1+1)\right) < \sum_{i=0}^{h_1}\left(\rho + D(i, j_1)\right) - 1,$$
$$\sum_{i=0}^{h_1} D(i, j_1+1) < \sum_{i=0}^{h_1} D(i, j_1) - 1,$$

contradicting (3-1). Thus, $r_1 = \rho + D(h_1, j_1) - 1$.

Moreover, if $r_2 > 0$,

$$\sum_{i=0}^{h_1} \left(\rho + D(i, j_1 + 1)\right) < i_1.$$

From (A-1),

$$\sum_{i=0}^{h_1} \left(\rho + D(i, j_1)\right) - \sum_{i=0}^{h_1} \left(\rho + D(i, j_1 + 1)\right) > 1,$$

which contradicts (3-1). Thus, $r_2 = 0$. □

The dilation of the embedding is the maximum distance between $\psi(i_1, j_1)$ and $\psi(i_1, j_1 + 1)$. We have two cases:

1) $h_1 = h_2$:

By Lemma 3.1, we have

$$\begin{aligned} |r_1 - r_2| &= \left|\sum_{i=0}^{h_1-1} \left(\rho + D(i, j_1)\right) - \sum_{i=0}^{h_2-1} \left(\rho + D(i, j_1 + 1)\right)\right| \\ &\le \left|\sum_{i=0}^{h_1-1} \left(\rho + D(i, j_1)\right) - \sum_{i=0}^{h_1-1} \left(\rho + D(i, j_1 + 1)\right)\right| \le 1. \end{aligned} \quad \text{(A-3)}$$

If h_1 is even, then

$$\begin{aligned} \text{dilation}_\psi &= |\psi(e)| = |h_1 - h_2| + |k_1 - k_2| \\ &= \left|\sum_{j=0}^{j_1-1} \left(\rho + D(h_1, j)\right) + r_1 - \left(\sum_{j=0}^{j_1} \left(\rho + D(h_1, j)\right) + r_2\right)\right| \\ &= \left|\sum_{j=0}^{j_1-1} \left(\rho + D(h_1, j)\right) - \sum_{j=0}^{j_1} \left(\rho + D(h_1, j)\right) + r_1 - r_2\right| \\ &\le \rho + D(h_1, j_1) + |r_1 - r_2| \le \rho + 2, \text{ by (A-2).} \end{aligned}$$

If h_1 is odd, then

$$\begin{aligned} \text{dilation}_\psi &= |\psi(e)| = |h_1 - h_2| + |k_1 - k_2| \\ &= \left|\sum_{j=0}^{j_1} \left(\rho + D(h_1, j)\right) - r_1 - 1 - \sum_{j=0}^{j_1+1} \left(\rho + D(h_1, j)\right) + r_2 + 1\right| \\ &\le \rho + D(h_1, j_1 + 1) + |r_2 - r_1| \\ &\le \rho + D(h_1, j_1 + 1) + 1 \le \rho + 2, \text{ by (A-3).} \end{aligned}$$

2) $h_1 \neq h_2$:

Without loss of generality, assume $h_2 = h_1 + 1$. From Lemma A.1, we know

$$r_1 = \rho + D(h_1, j_1) - 1 \text{ and } r_2 = 0.$$

If h_1 is even, then,

$$\begin{aligned} \text{dilation}_\psi &= |\psi(e)| = |h_1 - h_2| + |k_1 - k_2| = 1 + |k_1 - k_2| \\ &= 1 + \left|\sum_{j=0}^{j_1-1} \left(\rho + D(h_1, j)\right) + r_1 - \left(\sum_{j=0}^{j_1+1} \left(\rho + D(h_2, j)\right) - r_2 - 1\right)\right| \\ &= 1 + \left|\sum_{j=0}^{j_1-1} \left(\rho + D(h_1, j)\right) + \rho + D(h_1, j_1) - 1 \right. \\ &\quad \left. - \left(\sum_{j=0}^{j_1+1} \left(\rho + D(h_2, j)\right) - 1\right)\right| \\ &= 1 + \left|\left(\sum_{j=0}^{j_1} \left(\rho + D(h_1, j)\right) - \left(\sum_{j=0}^{j_1+1} \left(\rho + D(h_2, j)\right)\right)\right)\right| \\ &= 1 + \left|\left(\sum_{j=0}^{j_1} D(h_1, j) - \sum_{j=0}^{j_1+1} D(h_2, j)\right) - \rho\right| \\ &\le \rho + 2, \text{ by (3-2).} \end{aligned}$$

If h_1 is odd, then,

$$\begin{aligned} \text{dilation}_\psi &= |\psi(e)| = |h_1 - h_2| + |k_1 - k_2| = 1 + |k_1 - k_2| \\ &= 1 + \left|\sum_{j=0}^{j_1} \left(\rho + D(h_1, j)\right) - r_1 - 1 - \left(\sum_{j=0}^{j_1} \left(p + D(h_2, j) + r_2\right)\right)\right| \\ &= 1 + \left|\sum_{j=0}^{j_1} \left(\rho + D(h_1, j)\right) - \left(\rho + D(h_1, j_1) - 1\right) - 1 \right. \\ &\quad \left. - \sum_{j=0}^{j_1} \left(\rho + D(h_2, j)\right)\right| \\ &= 1 + \left|\sum_{j=0}^{j_1-1} \left(\rho + D(h_1, j)\right) - \sum_{j=0}^{j_1} \left(\rho + D(h_2, j)\right)\right| \\ &= 1 + \left|\left(\sum_{j=0}^{j_1-1} D(h_1, j) - \sum_{j=0}^{j_1} D(h_2, j)\right) - \rho\right| \\ &\le \rho + 2, \text{ by (3-2).} \end{aligned}$$

Therefore, we obtain $\text{dilation}_\psi \le \lfloor s_1 / s_2 \rfloor + 2$ for the case $s_1 > s_2$.

2) The calculation of the congestion.

Under the embedding ψ, each edge (a, b) in M_1 is mapped into a vertex pair $(\psi(a), \psi(b))$ in M_2. $\psi(a, b)$ is specified as the shortest path from $\psi(a)$ to $\psi(b)$ which takes the horizontal direction first and then turns into the vertical direction. Since each vertical edge in M_1 will be mapped to a horizontal, or a vertical edge or an "oblique" edge in M_2, and the distance between two endpoints of an oblique edge is two, any edge in M_2 belongs to images of at most two vertical edges of M_1. Let us consider how many images of horizontal edges of M_1 share an edge $e = (p, q)$ of M_2. If e is horizontal, then it is shared by images of at most $\lfloor s_1 / s_2 \rfloor$ + 2 horizontal edges of M_1 because $\text{dilation}_\psi \le \lfloor s_1 / s_2 \rfloor + 2$. If e is vertical, then, from A-2, it is shared by images of at most two horizontal edges of M_1. Thus, we have $\text{congestion}_\psi \le \lfloor s_1 / s_2 \rfloor + 4$.

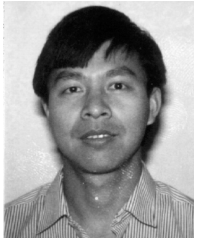
ACKNOWLEDGMENTS

This research is partially supported by UMKC Faculty Research Grant K-2-11265.

REFERENCES

[1] S.R. Kosaraju and M.J. Atallah, "Optimal Simulations Between Mesh-Connected Arrays of Processors," *J. ACM*, vol. 35, pp. 635-650, 1988.
[2] R. Aleliunas and A.L. Rosenberg, "On Embedding Rectangular Grids in Square Grids," *IEEE Trans. Computers*, vol. 31, no. 9, pp. 907-913, Sept. 1982.
[3] F. Lombardi, D. Sciuto, and R. Stefanelli, "An Algorithm for Functional Reconfiguration of Fixed-Size Arrays," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 1,114-1,118, 1988.
[4] J. Ellis, "Embedding Rectangular Grids into Square Grids," *IEEE Trans. Computers*, vol. 40, no. 1, pp. 46-52, Jan. 1991.
[5] M.Y. Chan, "Dilation-2 Embedding of Grids into Hypercubes," *Proc. Int'l Conf. Parallel Processing*, vol. 3, pp. 295-298, 1988.
[6] M.Y. Chan, "Embedding of Grids into Optimal Hypercubes," *SIAM J. Computing*, vol. 20, no. 5, pp. 834-864, 1991.
[7] B. Cong and H. Sudborough, "Dilation-4 Embedding of 2D Meshes into Star Graphs," *Proc. First Int'l Conf. Computer Comm. and Networks* (IC^3N), pp. 6-9, 1992.
[8] M.Y. Chan and F.Y.L. Chin, "On Embedding Rectangular Grids in Hypercubes," *IEEE Trans. Computers*, vol. 37, no. 10, pp. 1,285-1,288, Oct. 1988.
[9] H. Liu and S.S. Huang, "Dilation-6 Embedding of 3D Dimensional Grids into Optimal Hypercubes," *Proc. Int'l Conf. Parallel Processing*, vol. 3, pp. 250-254, 1991.
[10] C.-T. Ho and S.L. Johnsson, "On the Embedding of Arbitrary Meshes in Boolean Cubes with Expansion Two Dilation Two," *Proc. Int'l Conf. Parallel Processing*, pp. 188-191, 1987.
[11] Y. Ma and L. Tao, "Embeddings Among Toruses and Meshes," *Proc. Int'l Conf. Parallel Processing*, pp. 178-187, 1987.
[12] R.A. DeMillo, S.C. Eisenstat, and R.J. Lipton, "Preserving Average Proximity in Arrays," *Comm. ACM*, vol. 21, pp. 228-231, 1978.
[13] S.N. Bhatt, F.R.K. Chung, F.T. Leighton, and A.L. Rosenberg, "Efficient Embeddings of Trees in Hypercubes," *SIAM J. Computing*, vol. 21, no. 1, pp. 151-162, 1992.
[14] A.K. Gupta and S.E. Hambrusch, "Embedding Complete Binary Trees into Butterfly Networks," *IEEE Trans. Computers*, vol. 40, no. 7, pp. 853-863, July 1991.
[15] A.Y. Wu, "Embedding of Tree Networks into Hypercubes," *J. Parallel and Distributed Computing*, vol. 2, pp. 238-249, 1985.
[16] J.-W. Hong, K. Mehlhorn, and A.L. Rosenberg, "Cost Tradeoffs in Graph Embeddings," *J. ACM*, vol. 30, no. 4, pp. 709-728,1983.
[17] F.T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees and Hypercubes*. Morgan Kaufmann, 1992.
[18] K. Hwang and F.A. Briggs, *Computer Architecture and Parallel Processing*. McGraw-Hill, 1984.
[19] "Embedding Between 2D Meshes of the Same Size," *Proc. Fifth IEEE Symp. Parallel and Distributed Processing*, pp. 712-719, Dallas, Tex., Dec. 1-4, 1993.

Xiaojun Shen received his PhD degree in computer science from the University of Illinois, Urbana-Champaign, in 1989. He received his MS degree in computer science from the Nanjing University of Science and Technology, China, in 1982, and his BS degree in numerical analysis from the Qinghua University, Beijing, China, in 1968. He is currently an associate professor in the Computer Science Telecommunications Program at the University of Missouri-Kansas City. His current research interests include parallel processing, discrete mathematics, algorithms, and computer networking.



Weifa Liang received the BS degree from Wuhan University in 1984 and the MS degree from the University of Science and Technology of China (USTC) in 1989, both in computer science. He was a visiting scholar in the Computer Science Telecommunications Program at the University of Missouri-Kansas City from 1991 to 1993. He is currently a PhD candidate in the Department of Computer Science at the Australian National University. His current research interests include parallel processing, parallel graph algorithm design and analysis, parallel architecture, and interconnection networks.

Qing Hu received his BS and MS degrees in computer science from Nanjing University of Science and Technology (formerly East China Institute of Technology), China, in 1982 and 1985, respectively. He was with the Computer Science Department of that university until 1991. In September of 1991, he came to the U.S. to continue his studies and received his PhD degree from the Computer Science Telecommunications Program at the University of Missouri-Kansas City. He is currently with FutureNet Technology Inc., Monrovia, California. His research interests include parallel processing, image processing, pattern recognition, computer vision, and computer networks.