Embedding Complete k-ary Trees into 2-dimensional Meshes and Tori *

Jan Trdlička and Pavel Tvrdík

Department of Computer Science and Engineering
Czech Technical University, Karlovo nám. 13
121 35 Prague, Czech Republic
{trdlicka,tvrdik}@sun.felk.cvut.cz

Abstract. We have designed an algorithm for embedding of complete k-ary trees into 2-dimensional square meshes. The embedding has load 1, optimal dilation (the constant is 3 if $k = 3$ and 2 otherwise), and expansion 2. This solution can be easily converted into an embedding with optimal expansion 1/2 for load 2 while keeping the dilation the same.

Keywords: complete k-ary tree, 2-dimensional square mesh, embedding problem, expansion, load, dilation.

1 Basic definitions

A graph G is a pair $(\mathcal{V}(G), \mathcal{E}(G))$ where $\mathcal{V}(G)$ is the set of *nodes* of G and $\mathcal{E}(G)$ is the set of *edges* of G. An edge with end nodes u and v is denoted by $\langle u, v\rangle$. The set of all paths of graph G is denoted $\mathcal{P}(G)$. An *embedding* of a *guest* graph $G = (\mathcal{V}(G), \mathcal{E}(G))$ into a *host* graph $H = (\mathcal{V}(H), \mathcal{E}(H))$ is defined as a pair of mappings (ρ, ξ) where $\rho : \mathcal{V}(G) \to \mathcal{V}(H)$ and $\xi : \mathcal{E}(G) \to \mathcal{P}(H)$. The quality of an embedding is measured using several parameters .

The *expansion* of (ρ, ξ) is defined as $\frac{|\mathcal{V}(H)|}{|\mathcal{V}(G)|}$. The *dilation* of an edge $e_G \in \mathcal{E}(G)$ is the length of the path $\xi(e_G)$; the *dilation* of (ρ, ξ) is the maximum dilation over all edges of G; the *average dilation* of (ρ, ξ) is the average over all edge dilations. The *load* of a node $v_H \in \mathcal{V}(H)$ is the number of nodes of $\mathcal{V}(G)$ that are mapped onto v_H; the *load* of (ρ, ξ) is the maximum load over all nodes of H. An embedding of G into H has the *optimal load* $z = \left\lceil \frac{|\mathcal{V}(G)|}{|\mathcal{V}(H)|} \right\rceil$ if the load of each node v_H of H is $z - 1 \leq \text{load}(v_H) \leq z$.

For $k \geq 2$, $h \geq 0$, let $CT_{k,h}$ denote the *k-ary complete tree* of height h and $\mu(k, h) = (k^{h+1} - 1)/(k - 1)$ be the number of its nodes. Each internal node of $CT_{k,h}$ has exactly k children and each leaf of $CT_{k,h}$ is at distance h from the root. The *level* of a node v of $CT_{k,h}$, denoted by $l(v)$, is the distance of v from the root. For $0 \leq i \leq h$, define

$$\begin{array}{lll} \mathcal{A}_i = \{v \in \mathcal{V}(CT_{k,h}) : 0 \leq l(v) \leq i\} & \text{and} & \alpha_i = |\mathcal{A}_i|, \\ \mathcal{B}_i = \{v \in \mathcal{V}(CT_{k,h}) : l(v) = i\} & \text{and} & \beta_i = |\mathcal{B}_i|. \end{array}$$

* This research has been supported by GAČR under Grant 102/97/1055.

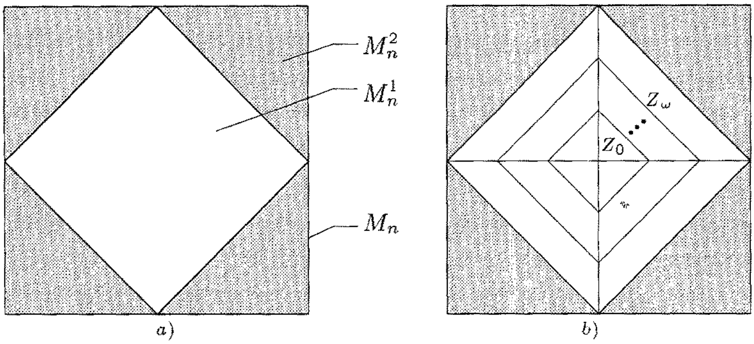

Fig. 1. a) Decomposition of M_n into M_n^1 and M_n^2, b) Decomposition of $\mathcal{V}(M_d^1)$ into zones Z_i, $0 \leq i \leq \omega$.

For $n \geq 1$, let M_n denote the square mesh $[n \times n]$. Vertex set of M_n is $\mathcal{V}(M_n) = \{u = [u_x, u_y] : 1 \leq u_x, u_y \leq n\}$ and edge set of M_n is $\mathcal{E}(M_n) = \{\langle u, v\rangle : (|u_x - v_x| = 1 \text{ and } u_y = v_y) \text{ or } (|u_y - v_y| = 1 \text{ and } u_x = v_x)\}$. Mesh node $c = [\lceil n/2 \rceil, \lceil n/2 \rceil]$ is said to be the *center* of M_n. Let $L(i)$ denote the set of mesh nodes in distance i from c. For odd n, let M_n^1 and M_n^2 be the decomposition of M_n into two subgraphs as depicted on Figure 1a. It follows that $|\mathcal{V}(M_n^1)| = (n^2+1)/2$ and $|\mathcal{V}(M_n^2)| = |\mathcal{V}(M_n)| - |\mathcal{V}(M_n^1)| = (n^2-1)/2$. The minimum odd n such that $|\mathcal{V}(M_n^1)| \geq \mu(k, h)$ will be denoted by $d(k, h)$. It is easy to show that $d(k, h) = 2\left\lceil \left(\sqrt{2\mu(k,h) - 1} - 1\right)/2 \right\rceil + 1$. Since the diameter of $M^1_{d(k,h)}$ is $d(k, h) - 1$, it follows that the lower bound on the dilation of an embedding $CT_{k,h}$ into $M_{d(k,h)}$ with load 1 is $\psi(k, h) = \left\lceil \left\lceil (\sqrt{2\mu(k,h) - 1} - 1)/2 \right\rceil / h \right\rceil$.

2 Related work

Embeddings of $CT_{2,h}$ into meshes have been investigated in several papers. From the diameter argument, it follows that the lower bound on the dilation for an embedding with load 1 is $\Omega(\sqrt{\mu(2,h)}/\log \mu(2,h))$. The well-known H-tree construction from [7] and the improved embeddings in [2, 5] all have load 1 and dilation $O(\sqrt{\mu(2,h)})$, but they embed $CT_{2,h}$ into non-optimal square meshes. Modified H-tree construction from [4, 7] also embeds $CT_{2,h}$ into a non-optimal square mesh, but it achieves optimal load 1 and dilation $O(\sqrt{\mu(2,h)}/\log \mu(2,h))$.

An embedding of $CT_{2,h}$ into its optimal square mesh with load 1 and dilation $O(\sqrt{\mu(2,h)})$ is presented in [8]. In [3], the author proves that $CT_{2,h}$ can be embedded into its optimal square mesh with optimal load 1 and optimal dilation $O(\sqrt{\mu(2,h)}/\log \mu(2,h))$. The general problem of embedding $CT_{k,h}$ into 2-dimensional meshes is discussed in [1].

In this paper, we give an algorithm for embedding of complete k-ary trees into 2-dimensional square meshes with expansion 2, load 1, and optimal dilation. This solution can easily be converted into embedding of complete k-ary trees into 2-dimensional meshes with load 2, expansion 1/2, and optimal dilation. Hence, both solutions are optimal with respect to load, expansion and dilation within constant factors close to one.

3 The embedding algorithm

In the following text, we will assume $k \geq 3$ and $h \geq 0$ and use symbols μ, d, and ψ instead of $\mu(k,h)$, $d(k,h)$, and $\psi(k,h)$. The embedding algorithm is based on a decomposition of the mesh M_d into vertex-disjoint *zones* Z_i defined as follows (see Figure 1b).

Definition 1. Let $\omega = h-1$ if $(h-1)\psi = \frac{d-1}{2}$ and $\omega = h$ otherwise. Let (ρ,ξ) be an embedding of $CT_{k,h}$, $k \geq 3$, $h \geq 0$, into M_d with load 1. For $0 \leq i \leq \omega$, define

$$Z_i = \begin{cases} L(0) & \text{if} \quad i = 0, \\ L((i-1)\psi + 1) \cup \ldots \cup L(i\psi) & \text{if} \quad 0 < i < \omega, \\ L((i-1)\psi + 1) \cup \ldots \cup L(\frac{d-1}{2}) & \text{otherwise.} \end{cases}$$

Obviously, the last zone Z_ω may be narrower than the inner zones Z_i, $0 < i < \omega$. The number of nodes within zones Z_i increases linearly with i.

Lemma 2. *For* $0 \leq i \leq \omega$,

$$|Z_i| = \begin{cases} 1 & \text{if} \quad i = 0, \\ 4i\psi^2 - 2\psi^2 + 2\psi & \text{if} \quad 0 < i < \omega, \\ \frac{1}{2}(d^2-1) - 2(i-1)^2\psi^2 - 2(i-1)\psi & \text{otherwise.} \end{cases}$$

From the definition of d, it follows that $\sum_{r=0}^{\omega} |Z_r| = |\mathcal{V}(M_n^1)| \geq \mu$. To describe the embedding algorithm, we need to specify tree node subsets of $\mathcal{A}_j$ and $\mathcal{B}_j$, $0 \leq j \leq h$, mapped into zones Z_i, $0 \leq i \leq \omega$.

Definition 3. For $0 \leq j \leq h$, define

$$\begin{array}{lcl} \mathcal{A}_j(Z_i) = \{x \in \mathcal{A}_j : \rho(x) \in \mathcal{V}(Z_i)\} & \text{and} & \alpha_j(Z_i) = |\mathcal{A}_j(Z_i)|, \\ \mathcal{B}_j(Z_i) = \{x \in \mathcal{B}_j : \rho(x) \in \mathcal{V}(Z_i)\} & \text{and} & \beta_j(Z_i) = |\mathcal{B}_j(Z_i)|. \end{array}$$

The embedding algorithm embeds the tree nodes level by level. Since $|Z_i|$ grows linearly with i whereas β_i grows exponentially with i, we cannot just simply map all nodes from $\mathcal{B}_i$ into Z_i for $0 < i < \omega$.

The embedding algorithm consists of two phases. In the first phase, successively for $i = 0, \ldots, \lfloor \frac{h}{2} \rfloor$, all the tree nodes from $\mathcal{B}_i$ are embedded into zone Z_i. Hence, up to level $\lfloor \frac{h}{2} \rfloor$, the tree is embedded level by level into successive zones. As we will show later, it follows from Lemma 4 that after the first phase, there are free unloaded mesh nodes in zones Z_i, $0 < i \leq \lfloor \frac{h}{2} \rfloor$.

Hence, in the second phase, successively for $i = \lfloor \frac{h}{2} \rfloor + 1, \ldots, h-1$, we will embed the tree nodes from $\mathcal{B}_i$ so that the tree will grow not only towards the border of the mesh but also backwards to fill in the remaining free parts of the previous zones (see Figure 2 for an example).

Assume that, for given $i \geq \lfloor \frac{h}{2} \rfloor + 1$, $\beta_i(Z_i)$ tree nodes from $\mathcal{B}_i$ are embedded into Z_i. Among their $k\beta_i(Z_i)$ children, $\beta_{i+1}(Z_{i+1})$ of them will be embedded into Z_{i+1} and the remaining children will stay in Z_i, i.e., $k\beta_i(Z_i) = \beta_{i+1}(Z_i) + \beta_{i+1}(Z_{i+1})$. The ratio between $\beta_{i+1}(Z_i)$ and $\beta_{i+1}(Z_{i+1})$ is determined by the following strategy. All the subtrees of nodes from $\mathcal{B}_{i+1}(Z_i)$ will be embedded backwards into zones $Z_{i-1}, Z_{i-2}, \ldots$, level by level. Hence, $\beta_{i+1+p}(Z_{i-p}) = k^p \beta_{i+1}(Z_i)$ for all $p = 1, \ldots, h-i-2$, and the backward phase ends up in zone $Z_{i-(h-i-2)} = Z_{2i-h+2}$, which will be loaded with $k^{h-i-2}\beta_{i+1}(Z_i)$ tree nodes from $\mathcal{B}_{h-1}$. It follows that $\beta_{h-1}(Z_{2i-h+2}) = k^{h-i-2}\beta_{i+1}(Z_i)$. All the $k^{h-i-1}\beta_{i+1}(Z_i)$ nodes, i.e., all the children of nodes $\mathcal{B}_{h-1}(Z_{2i+h-2})$, will be used to fill up the empty parts of both Z_{2i-h+1} and Z_{2i-h+2} in this order and if some of these children still remain, they will be embedded into Z_{2i-h+3}. Since now on, zones Z_{2i-h+1} and Z_{2i-h+2} are full and the algorithm will proceed to fill up the next two zones. Again, it starts by splitting children of $\mathcal{B}_{i+1}(Z_{i+1})$ into $\mathcal{B}_{i+2}(Z_{i+1})$ and $\mathcal{B}_{i+2}(Z_{i+2})$.

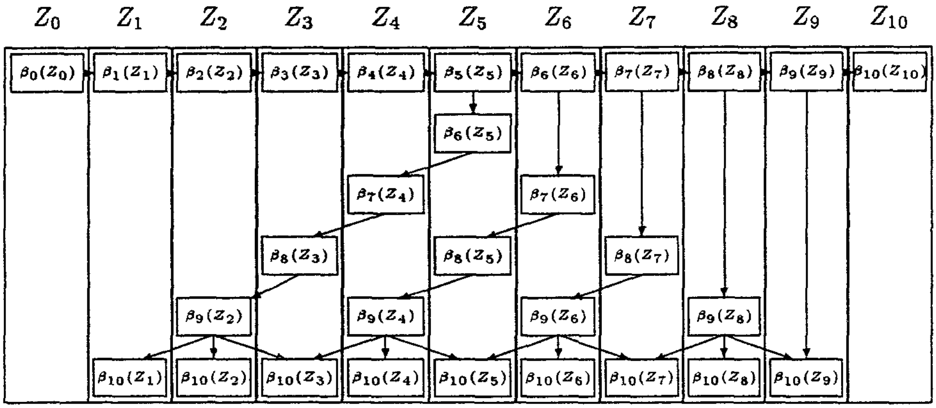

Fig. 2. The distribution of nodes of $CT_{k,10}$ within zones $Z_0 \cup \ldots \cup Z_{10}$

In general, the detailed way how the tree nodes will be distributed across zones of the mesh depends on whether h is even or odd. Assume that h is even (for h odd, the conditions are similar). The distribution of tree levels into zones must satisfy the following constraints (see Figure 2).

1. i is even and $0 < i \leq \frac{h}{2}$. Then

$$\alpha_h(Z_i) = \beta_i(Z_i) + \beta_{h-i+1}(Z_i) + \beta_{h-i+3}(Z_i) + \ldots + \beta_{h-i+(i-1)}(Z_i) + \beta_h(Z_i),$$

where

$$\beta_i(Z_i) = k\beta_{i-1}(Z_{i-1}), \tag{1}$$

$$\beta_{h-i+j+1}(Z_{i-1}) = k\beta_{h-i+j}(Z_i) \tag{2}$$

for all $j = 1, 3, 5, \ldots, i-1$, (equation (2) says that all the children of nodes in $\beta_{h-i+j}(Z_i)$ are embedded back into preceeding zone Z_{i-1}),

$$\beta_{h-1}(Z_i) = \left\lceil \frac{\gamma_h(Z_{i-1}) + \lambda_{h-2}(Z_i)}{k^{\frac{h-i}{2}} + k^{\frac{h-i}{2}-1}} \right\rceil k^{\frac{h-i}{2}-1}. \tag{3}$$

($\gamma_h(Z_{i-1})$ is the number of free mesh nodes in Z_{i-1} after embedding $\alpha_{h-2}(Z_{i-1})$ and all their children $\beta_h(Z_{i-3})$, $\beta_h(Z_{i-2})$ and $\beta_h(Z_{i-1})$, $\lambda_{h-2}(Z_i)$ is the number of free mesh nodes in Z_i after embedding $\alpha_{h-3}(Z_i)$ for $h \geq 4$), and finally

$$\beta_h(Z_i) = |Z_i| - \alpha_{h-1}(Z_i). \tag{4}$$

(the remaining unloaded part of Z_i will be filled up with leaves of $CT_{k,h}$)

2. i is odd and $0 < i \leq \frac{h}{2}$. Then

$$\alpha_h(Z_i) = \beta_i(Z_i) + \beta_{h-i+1}(Z_i) + \beta_{h-i+3}(Z_i) + \ldots + \beta_{h-i+(i-2)}(Z_i) + \beta_h(Z_i),$$

where $\beta_j(Z_i)$ for $j = i, h-i+1, h-i+3, \ldots, h-2, h$ is defined by (1), (2) and (4), and $\beta_{h-1}(Z_i) = 0$.

3. i is even and $\frac{h}{2} < i < h-1$. Then

$$\alpha_h(Z_i) = \beta_i(Z_i) + \beta_{i+1}(Z_i) + \beta_{i+3}(Z_i) + \ldots + \beta_{h-1}(Z_i) + \beta_h(Z_i),$$

where

$$\beta_i(Z_i) = k\beta_{i-1}(Z_{i-1}) - \beta_i(Z_{i-1}) \tag{5}$$

and $\beta_j(Z_i)$ for $j = i+1, i+3, \ldots, h-1, h$ is defined by (2), (3), and (4).

4. i is odd and $\frac{h}{2} < i < h-1$. Then

$$\alpha_h(Z_i) = \beta_i(Z_i) + \beta_{i+1}(Z_i) + \beta_{i+3}(Z_i) + \ldots + \beta_{h-2}(Z_i) + \beta_h(Z_i),$$

where $\beta_j(Z_i)$ for $j = i, i+1, i+3, \ldots, h-2, h$ is defined by (5), (2), and (4), and $\beta_{h-1}(Z_i) = 0$.

4 The analysis of the embedding algorithm

To justify the embedding algorithm, we must prove that zones Z_i are large enough to accommodate the tree nodes distributed to them by the algorithm described above. This is proved by the following two lemmas.

Lemma 4. *For $i = 1, \ldots, h-2$,*

$$\beta_i(Z_i) \quad < \quad \frac{1}{2}|Z_i|.$$

Lemma 5. *Let (ρ, ξ) be an embedding of $CT_{k,h}$, $k \geq 3$, $h \geq 0$, into M_d and $0 < i < h-1$. If both h and i are even or both h and i are odd, then*

$$\frac{1}{k}\sum_{r=i+1}^{h-1} \beta_r(Z_{i-1}) + \left\lceil \frac{\beta_h(Z_{i-1})}{k^{\frac{h-i}{2}}} \right\rceil < \frac{1}{2}|Z_i|.$$

If h is odd and i is even or h is even and i is odd, then

$$\frac{1}{k}\sum_{r=i+1}^{h-2} \beta_r(Z_{i-1}) < \frac{1}{2}|Z_i|.$$

We can summarize by giving exact formulae for computing $\beta_j(Z_i)$, $0 \le i \le \omega$, $0 \le j \le h$ (ω is defined in Definition 1).

Lemma 6.

$$\beta_j(Z_0) = \begin{cases} 1 \text{ if } & j = 0 \\ 0 \text{ otherwise.} \end{cases}$$

For $0 < i \le \omega$

$$\beta_j(Z_i) = \begin{cases} k\beta_{i-1}(Z_{i-1}) - \beta_i(Z_{i-1}) & \text{if} \quad j = i \\ \frac{1}{k}\beta_{j+1}(Z_{i-1}) & \text{if} \quad i < j < h-1 \\ \left\lceil \frac{\gamma_h(Z_{i-1})+\lambda_{h-2}(Z_i)}{k^{\frac{h-i}{2}}+k^{\frac{h-i}{2}-1}} \right\rceil k^{\frac{h-i}{2}-1} & \text{if} \quad j = h-1 \quad \text{and} \quad i < h-1 \quad \text{and} \\ & ((i \text{ and } h \text{ are even}) \text{ or } (i \text{ and } h \text{ are odd})) \\ \lambda_{h-1}(Z_i) & \text{if} \quad j = h \quad \text{and} \quad i \ne h \\ 0 & \text{otherwise,} \end{cases}$$

where

$$\lambda_j(Z_i) = \begin{cases} |Z_i| - \sum_{r=i}^{j} \beta_r(Z_i) & \text{if} \quad i < \omega \\ \mu - \sum_{r=0}^{i-1} |Z_r| - \sum_{r=i}^{j} \beta_r(Z_i) & \text{if} \quad i = \omega \\ 0 & \text{otherwise,} \end{cases}$$

$$\gamma_h(Z_i) = \sum_{r=0}^{i} \beta_h(Z_r) - k \sum_{r=0}^{i-1} \beta_{h-1}(Z_r).$$

Proof: The proof is by a detailed case analysis based on equations (1)-(5). □

The main result of the paper follows.

Theorem 7. *$CT_{k,h}$, $k \ge 3$, $h \ge 0$, can be embedded into mesh M_d with load* 1, *expansion* 2, *and dilation* dil(k,h). *where* dil$(k,h) = 2\psi$ *if* $k \ge 4$ *and* dil$(k,h) = 3\psi$ *if* $k = 3$.

Proof: The exact proof is very technical and can not be included in its full extent here. We will just sketch the strategy leading to optimal dilation.

We have shown how to distribute tree nodes across mesh zones with load 1 by giving formulae for computing numbers $\beta_j(Z_i)$, $0 \le j \le h$, $0 \le i \le \omega$. To achieve the optimal dilation, we use the following strategy.

1. for all $i = 0, \ldots, \omega$, nodes from $B_i(Z_i)$ are embedded into zone Z_i starting from the outermost layers of Z_i,
2. successively for $j = i+1, \ldots, h-1$, nodes from $B_j(Z_i)$ (if they exist) are embedded into the innermost layers of Z_i,
3. if both h and i are even or both h and i are odd, then nodes from $B_h(Z_i)$ will be embedded into remaining inner parts of Z_i,

4. if h is even and i is odd or h is odd and i is even, then the nodes from $\mathcal{B}_h(Z_i)$ can have parents embedded into Z_{i-1} or Z_{i+1}. Say that $\mathcal{B}'_h(Z_i)$ is the subset of nodes from $\mathcal{B}_h(Z_i)$ whose parents are embedded into Z_{i-1} and $\mathcal{B}''_h(Z_i)$ is the subset of nodes from $\mathcal{B}_h(Z_i)$ whose parents are embedded into Z_{i+1}. First, we embed nodes from $\mathcal{B}'_h(Z_i)$ into the remaining innermost parts of Z_i and then nodes from $\mathcal{B}''_h(Z_i)$ into remaining free nodes in Z_i.

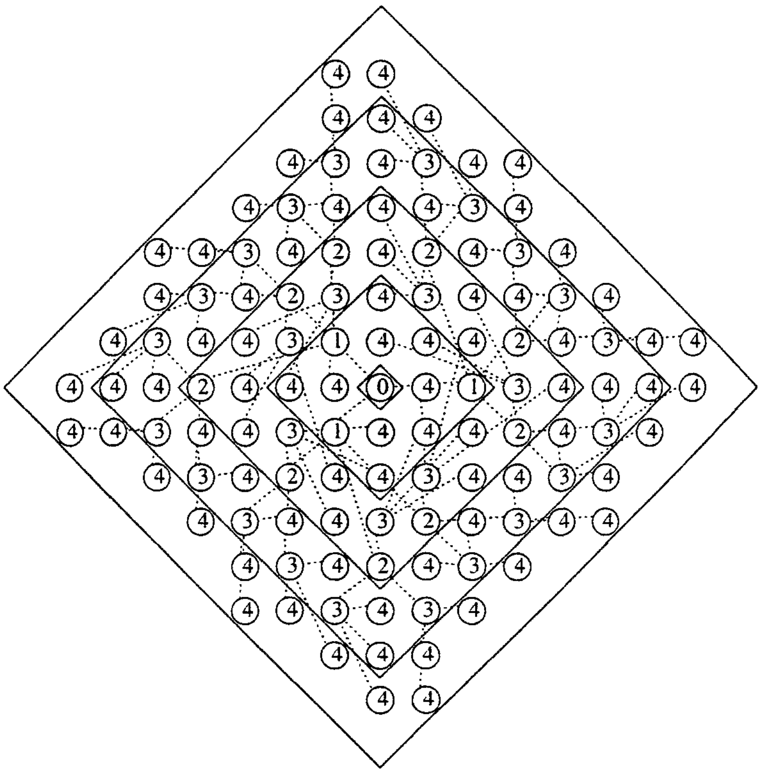

Fig. 3. Embedding of $CT_{3,4}$ into M^1_{17} with dilation 5 and load 1.

As an example, Figure 3 shows the embedding of $CT_{3,4}$ into M^1_{17}. The nodes of $CT_{3,4}$ are labeled by their levels.

5 Embedding of complete k-ary tree into optimal mesh for load 2

We have described the embedding of $CT_{k,h}$ into M^1_d (see Figure 1). This embedding achieves load 1 and dilation $\mathrm{dil}(k,h)$, but the expansion is 2. However, this embedding can be easily converted into an embedding with expansion optimal for load 2 by folding the corners of M^1_d towards the center (see Figure 4).

Lemma 8. *Let* $p = \left\lfloor \frac{\frac{n-1}{2}-1}{2} \right\rfloor = \left\lfloor \frac{n-4}{4} \right\rfloor$. $CT_{k,h}$, $k \geq 3$, $h \geq 0$, *can be embedded into mesh* M_{n-2p} *with load* 2, *expansion* $1/2$, *and dilation* $\mathrm{dil}(k,h)$.

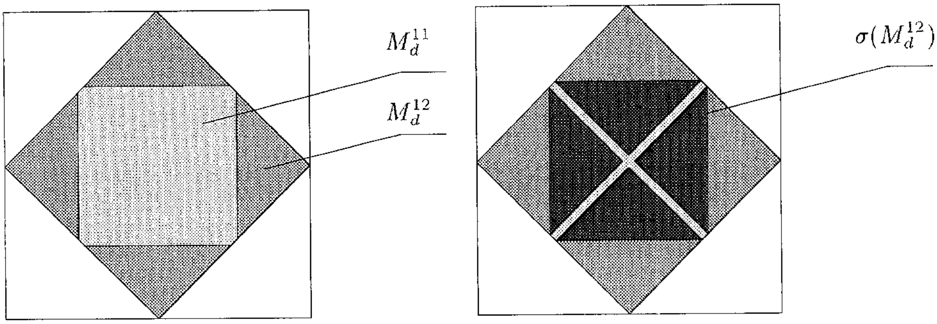

Fig. 4. The transformation of the expansion-2 embedding into an expansion-1 embedding.

6 Conclusions

We have designed an embedding of complete k-ary trees into 2-dimensional square meshes with load 1, expansion 2, and dilation optimal within a multiplicative constant ≤ 2 for $k > 3$ and ≤ 3 for $k = 3$. It can be easily converted into an embedding with optimal expansion $1/2$ for load 2 while keeping the dilation the same. Currently, we are working on the proof of optimality of the edge congestion. We conjecture that the embedding algorithm can be generalized for 3-dimensional square meshes.

References

1. M. J. Fischer and M. S. Paterson. Optimal tree layout. In *Proceedings of the 12th ACM Symposium on the Theory of Computing*, pages 177–189, 1980.
2. D. Gordon. Efficient embeddings of binary trees in VLSI arrays. *IEEE Transactions on Computers*, C-36:1009–1018, 1987.
3. R. Heckmann, R. Klasing, B. Monien, and W. Unger. Optimal embedding of complete binary trees into lines and grids. In *Proceedings of Graph-Theoretic Concepts in Computer Science*, number 570 in Lecture Notes of Computer Science, pages 25–35. Springer Verlag, 1991.
4. M. S. Paterson, W. L. Ruzzo, and L. Snyder. Bounds on minimax edge length for complete binary trees. In *Proceedings of the 13th ACM Symposium on the Theory of Computing*, pages 293–299, 1981.
5. A. D. Singh and H. Y. Youn. Near optimal embedding of binary tree architectures in VLSI. In *Proceedings of the 8th International Conference on Distributed Computing Systems*, pages 86–93, 1988.
6. J. Trdlička and P. Tvrdík. Embedding complete k-ary trees into 2-dimensional meshes and tori. Technical report, Dept. of CS&E, Czech Technical University, Prague, To be published.
7. J. D. Ullman. Computational aspects of VLSI. *Computer Science Press*, 1984.
8. P. Zienicke. Embedding of treelike graphs into 2-dimensional meshes. In *Proceedings of the Conference on the Graph Theoretical Concepts in Computer Science*, number 484 in Lecture Notes of Computer Science, pages 182–192, 1990.