Bounding Dilation of Separator-Based Graph Embeddings into Grids

Akira Matsubayashi[1,a)]

Abstract: In this report, we address a classical problem of embedding a guest graph with minimum dilation into a multidimensional grid of the same size as that of the guest graph, and propose a relatively simple embedding bounding dilation based on graph separators. Specifically, we prove that any graph with N nodes, maximum node degree $\Delta \geq 2$, and with a node-separator of size $O(n^{\alpha})$ can be embedded with a dilation of $O(N^{1/d} \log \Delta / \log N)$ into a grid of a fixed dimension $d \geq 2$ with at least N nodes, where α is a fixed number with $0 \leq \alpha < 1$. This dilation matches a trivial lower bound within a constant factor. A remarkable merit of the proposed embedding is that it can be used to bound the dilation of another embedding algorithm. Combining the proposed embedding with a previous embedding algorithm bounding the edge-congestion, we can obtain an edge-congestion of $O(\Delta)$ as well as the dilation $O(N^{1/d} \log \Delta / \log N)$ if $d > 1/(1 - \alpha)$. This congestion achieves constant ratio approximation. For $d \leq 1/(1 - \alpha)$, we present a trade-off between tight upper bounds of dilation and edge-congestion. Specifically, we prove a dilation of $O(\frac{N^{1/d} \log \Delta}{\epsilon \log N})$ and an edge-congestion of $O(\Delta(N^{\alpha-1+\frac{1}{d}+\epsilon} + \log N))$ for any $1/\log N \leq \epsilon < 1 - \alpha$. These dilation and edge-congestion match existential lower bounds within a constant factor for $\epsilon = \Omega(1)$ and $\epsilon = 1/\log N$, respectively. Besides, there exists a guest graph for which better dilation and edge-congestion cannot be obtained for $\epsilon = \log\log N/\log N$. The above results improve and generalize a number of previous results.

Keywords: graph embedding, dilation, congestion, grid, separator

1. Introduction

The *graph embedding* of a guest graph into a host graph is to map (typically one-to-one) nodes and edges of the guest graph onto nodes and paths of the host graph, respectively, so that an edge of the guest graph is mapped onto a path connecting the images of end-nodes of the edge. The graph embedding problem is to embed a guest graph into a host graph with certain constraints and/or optimization criteria. This problem has applications such as efficient VLSI layout and parallel computation. I.e., the problem of efficiently laying out VLSI can be formulated as the graph embedding problem with modeling a design rule on wafers and a circuit to be laid out as host and guest graphs, respectively. Also, the problem of efficiently implementing a parallel algorithm on a message passing parallel computer system consisting of processing elements connected by an interconnection network can be formulated as the graph embedding problem with modeling the interconnection network and interprocess communication in the parallel algorithm as host and guest graphs, respectively. The graph embedding problem has a history over 30 years. See for a survey, e.g., [22]. The major criteria to measure the efficiency of an embedding are dilation, node-congestion, and edge-congestion. In this paper, we consider the problem of embedding a guest graph with the minimum dilation into a d-dimensional grid with $d \geq 2$ and the same size as that of the guest graph. Embeddings into grids important for both VLSI layout and parallel computation. Actually, design rules on wafers in VLSI are usually modeled as 2-dimensional grids. As for parallel computation, multidimensional grid networks, including hypercubes, are popular for interconnection networks. The setting that host and guest graphs have the same number of nodes is important for parallel computation because the processing elements are expensive resource and idling some of them is wasteful.

1 Division of Electrical Engineering and Computer Science, Kanazawa University Kakuma-machi, Kanazawa, 920–1192 Japan
a) mbayashi@t.kanazawa-u.ac.jp

Previous Results

Table 1 summarizes previous results of graph embeddings minimizing dilation for various combinations of guest graphs and host grids.

VLSI layout has been studied through formulating the layout as the graph embedding into a 2-dimensional grid with objective of minimizing the grid under constrained congestion-1 routing [25]. Leiserson [18] and Valiant [26] independently proposed such embeddings based on graph separators. In particular, it was proved in [18] that any N-node graph with maximum node degree at most 4 and an edge-separator of size $O(n^{\alpha})$ can be laid out in an area of $O(N)$ if $\alpha < 1/2$, $O(N \log^2 N)$ if $\alpha = 1/2$, and $O(N^{2\alpha})$ if $\alpha > 1/2$. A separator of a graph G is a set S of either nodes or edges whose removal partitions the node set $V(G)$ of G into two subsets of roughly the same size with no edge between the subsets. The graph G is said to have a hereditary separator of size $s(n)$ if $|S| \leq s(|V(G)|)$ and any subgraph of G recursively has a hereditary separator of size $s(n)$. Separators are important tools to design divide-and-conquer algorithms and have been ex-

Table 1 Previous results of graph embeddings minimizing dilation.

Guest Graph N: # nodes, Δ: max degree s: separator size	Host Grid # nodes	 dimension	Dilation	Congestion	
binary tree	any	2	NP-hard for 1	1	[10]
tree	2^d	given d	NP-hard for 1	1	[27]
connected planar graph	N	2	any	NP-hard for 1	[8]
connected graph	$2^{\lceil \log_2 N \rceil}$	$\lceil \log_2 N \rceil$	any	NP-hard for 1	[15]
complete binary tree	$N+1$	2	$O(\sqrt{N}/\log N)$	$O(\sqrt{N}/\log N)$	[11]
binary tree	$2^{\lceil \log_2 N \rceil}$	$\lceil \log_2 N \rceil$	8	$O(1)$	[12]
2-D $h \times w$-grid ($h \le w$)	$h'w' \ge N^*$	2	$\lceil h/h' \rceil + 1$	$\lceil h/h' \rceil + 1$	[24]
2-D $h \times w$-grid ($h \le w$)	$h'w' \ge N^\dagger$	2	5	5	[24]
2-D grid	$2^{\lceil \log_2 N \rceil}$	$\lceil \log_2 N \rceil$	3	2	[23]
$\Delta \le 4$, $s = O(n^\alpha)$, $\alpha < 1/2$	$O(N)$	2	$O(\sqrt{N}/\log N)$	1	[4]
$\Delta \le 4$, $s = O(\sqrt{n})$	$O(N \log^2 N)$	2	$O(\frac{\sqrt{N}\log N}{\log\log N})$	1	[4]
$\Delta \le 4$, $s = O(n^\alpha)$, $\alpha > 1/2$	$O(N^{2\alpha})$	2	$O(N^\alpha)$	1	[4]
tree width t	$2^{\lceil \log_2 N \rceil}$	$\lceil \log_2 N \rceil$	$O(\log(\Delta t))$	$O(\Delta^4 t^3)$	[13]
$s = \log^{O(1)} N$	$2^{\lceil \log_2 N \rceil}$	$\lceil \log_2 N \rceil$	$O(\log \Delta)$	$\Delta^{O(1)}$	[14]
$\Delta = O(1)$	N	$d = O(1)$	$O(N^{1/d} \log N)$	$O(N^{1/d} \log N)$	[17]

* $h' \times w'$-grid with $h' < h \le w < w'$
† $h' \times w'$-grid with $h < h' \le w' < w$

tensively studied. It is well-known that any planar graph has a node-separator of size $O(\sqrt{n})$ [19]. This was generalized in [1] so that any graph with an excluded minor of a fixed size has a node-separator of size $O(\sqrt{n})$. Moreover, graphs with a fixed treewidth, such as trees, outerplanar graphs, and series-parallel graphs have a node-separator of a fixed size [16]. Bhatt and Leighton [4] achieved a better layout with several nice properties including reduced dilation as well as the same or better area as that of [18] by introducing a special type of edge-separators called *bifurcators*. An approximation algorithm for VLSI layout was proposed in [6]. Separator-based graph embeddings on hypercubes were presented in [3], [14], [21]. In particular, Heun and Mayr [14] proved that any N-node graph with maximum node degree Δ and an extended edge-bisector of polylogarithmic size can be embedded into a $\lceil \log_2 N \rceil$-dimensional cube with a dilation of $O(\log \Delta)$ and an edge-congestion of $\Delta^{O(1)}$.

A quite general embedding based on the multicommodity flow was presented by Leighton and Rao [17], who proved that any N-node bounded degree graph G can be embedded into an N-node bounded degree graph H with both dilation and edge-congestion of $O((\log N)/\alpha)$, where α is the *flux* of H, i.e., $\min_{U \subset V(H)} \frac{|\{(u,v) \in E(H) | u \in U,\ v \in V(H) \setminus U\}|}{\min\{|U|, |V(H) \setminus U|\}}$. This implies that G can be embedded into an N-node d-dimensional grid with both dilation and edge-congestion of $O(N^{1/d} \log N)$ for any fixed d.

Contributions and Technical Overview

All the previous results minimizing both dilation and edge-congestion have drawbacks such as host grids of large size or dimension, guest graphs of a fairly restricted class, and a gap between upper and lower bounds on either dilation or congestion. In this paper, we bound both dilation and edge-congestion to existential lower bounds within constant factors for guest graphs of a wide class and host grids having the optimal size and a fixed dimension more than 1.

First, we propose a relatively simple embedding bounding dilation based on graph separators. Specifically, we prove the following theorem:

Theorem 1 *Suppose that a guest graph G has N nodes, maximum node degree $\Delta \ge 2$, and a hereditary node-separator of size $O(n^\alpha)$, where α is a fixed number with $0 \le \alpha < 1$. Moreover, suppose that a host grid M has a fixed dimension $d \ge 2$, at least N nodes, and constant aspect ratio. Then, G can be embedded into M with a dilation of $O(N^{1/d} \log \Delta / \log N)$.*

The dilation of Theorem 1 matches a trivial existential lower bound, the diameter of M divided by the diameter of G, within a constant factor. The basic idea of Theorem 1 is to construct a complete binary tree structure of node sets of G using the node-separator, and embed G by applying the embedding algorithm provided in [11] to the tree structure. We can obtain the desired dilation with the fact that the algorithm of [11] can embed a complete binary tree with nearly optimal dilation, and with careful choice of the size of node sets of the tree structure. Similar techniques of embeddings via tree structures were previously proposed in the literature ([3], [12], [13], [14]). However, our structure is much simpler than the previous ones. We discuss the tree structure and prove Theorem 1 in Sects. 3 and 4, respectively.

Second, combining the embedding of Theorem 1 with a previous embedding algorithm bounding the edge-congestion [20], we also bound the edge-congestion as well as the dilation. Specifically, we prove the following theorem:

Theorem 2 *Suppose that a guest graph G has N nodes, maximum node degree $\Delta \ge 2$, and a hereditary node-separator of size $O(n^\alpha)$, where α is a fixed number with $0 \le \alpha < 1$. Moreover, suppose that a grid M has a fixed dimension $d \ge 2$, at least N nodes, and constant aspect ratio. Then, G can be embedded into M with a dilation of $O(N^{1/d} \log \Delta / \log N)$ and an edge-congestion of $O(\Delta)$ if $d > 1/(1-\alpha)$, and with a dilation of $O(\frac{N^{1/d} \log \Delta}{\epsilon \log N})$ and an edge-congestion of $O(\Delta(N^{\alpha - 1 + \frac{1}{d} + \epsilon} + \log N))$ if $d \le 1/(1-\alpha)$, where $1/\log N \le \epsilon < 1 - \alpha$.*

The embedding of Theorem 1, in fact, determines just locations of the node sets in the tree structure, neither a specific position of each node nor routing for edges. Therefore, we can use any algorithms for embedding each node and routing edges. For these purposes, we adopt another separator-based embedding algorithm bounding edge-congestion and a routing algorithm that are presented in [20]. Theorem 2 is proved in Sect. 5.

The edge-congestion of Theorem 2 achieves constant ratio approximation for a fixed $d > 1/(1-\alpha)$. This is because any embedding requires an edge-congestion of $\Delta/(2d)$. For $d \leq 1/(1-\alpha)$, Theorem 2 provides a trade-off between tight upper bounds of dilation and edge-congestion. Actually, the dilation and edge-congestion of Theorem 2 match existential lower bounds within a constant factor for $\epsilon = \Omega(1)$ and $\epsilon = 1/\log N$, respectively. Besides, it is known that any embedding of an N-node mesh of trees G, having $\Delta = O(1)$, into a 2-dimensional grid with edge-congestion 1 requires a dilation of $\Omega(\sqrt{N}\log N/\log\log N)$ and an $\Omega(N\log^2 N)$ nodes of the grid [4]. This implies that any embedding of G into a 2-dimensional N-node grid requires a dilation of $\Omega(\sqrt{N}/\log\log N)$ and an edge-congestion of $\Omega(\log N)$. This is because we can easily transform an embedding into a 2-dimensional N-node grid with a dilation δ and an edge-congestion c into another embedding into a 2-dimensional $O(c^2N)$-node grid with a dilation $c\delta$ and the edge-congestion 1, by replacing each row and each column of the N-node grid with $O(c)$ rows and $O(c)$ columns, respectively. In this sense, we cannot simultaneously improve the dilation and edge-congestion of Theorem 2 for $\epsilon = \log\log N/\log N$. We note that because G has a hereditary node-separator of size $O(\sqrt{n})$, by Theorem 1, G can be embedded into a 2-dimensional N-node grid with a dilation of $O(\sqrt{N}/\log N)$. However, such a dilation involves an edge-congestion of $\Omega(\log^2 N/\log\log N)$. To the author's best knowledge, this is the first observation of a trade-off between dilation and edge-congestion in embeddings into grids.

Our separator-based embedding algorithm performs in a polynomial time on the condition that a separator of the guest graph is given. Although finding a separator of minimum size is generally NP-hard [5], [9], approximation algorithms presented in [2], [7], [17] can be applied to our algorithm.

2. Preliminaries

For a graph G, $V(G)$ and $E(G)$ are the node set and edge set of G, respectively. We denote the set of integers $\{i \mid 1 \leq i \leq \ell\}$ by $[\ell]$. For a d-dimensional vector $v := (x_i)_{i\in[d]}$, let $\pi_j(v) := x_j$ and $\bar{\pi}_j(v) := (x_i)_{i\in[d]\setminus\{j\}}$ for $j \in [d]$. We use π_j and $\bar{\pi}_j$ also for a set of vectors and for a graph whose nodes are vectors. I.e., for a set V of d-dimensional vectors, we denote $\{\pi_j(v) \mid v \in V\}$ and $\{\bar{\pi}_j(v) \mid v \in V\}$ as $\pi_j(V)$ and $\bar{\pi}_j(V)$, respectively. Moreover, for a graph G with $V(G) = V$, we denote the graph with the node set $\bar{\pi}_j(V(G))$ and edge multiset $\{(\bar{\pi}_j(u), \bar{\pi}_j(v)) \mid (u,v) \in E(G)\}$ as $\bar{\pi}_j(G)$. For positive integers $\ell_1, \ldots, \ell_d$, the *d-dimensional $\ell_1 \times \cdots \times \ell_d$-grid*, denoted as $M(\ell_i)_{i\in[d]}$, is a graph with the node set $\prod_{i\in[d]}[\ell_i]$, i.e., the Cartesian product of sets $[l_1], \ldots, [l_d]$, and edge set $\{(u,v) \mid \exists j \in [d]\ \pi_j(u) = \pi_j(v) \pm 1, \bar{\pi}_j(u) = \bar{\pi}_j(v)\}$. The *aspect ratio* of $M(\ell_i)_{i\in[d]}$ is $\max_{i,j\in[d]}\{\ell_j/\ell_i\}$. An edge (u,v) of $M(\ell_i)_{i\in[d]}$ with $\pi_j(u) = \pi_j(v) \pm 1$ is called a *dimension-j edge*. The grid $M(\ell_i)_{i\in[d]}$ is called the *d-dimensional cube* if $\ell_i = 2$ for every $i \in [d]$.

A *routing request* on a graph H is a pair of nodes, a *source* and *target*, of H. A multiset of routing requests can be represented as

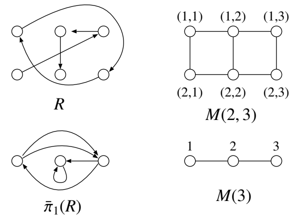

Fig. 1 A routing graph R on $M(2,3)$ and $\bar{\pi}_1(R)$ on $M(3)$.

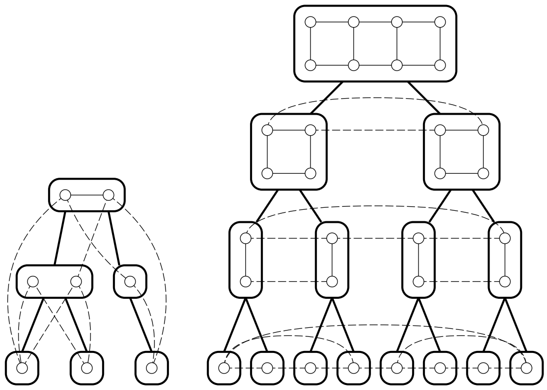

Fig. 2 A (1/2)-node-decomposition tree (left) and a (1/2)-edge-decomposition tree (right) for $M(2,4)$.

a *routing graph* R with the node set $V(H)$ and directed edges joining the sources and targets of all the routing requests. It should be noted that R may have parallel edges and loops. In particular, if H is a d-dimensional grid, then $\bar{\pi}_j(R)$ is a routing graph with the multiset of edges $(\bar{\pi}_j(u), \bar{\pi}_j(v))$ for every $(u,v) \in E(R)$ on the $(d-1)$-dimensional grid with node set $\bar{\pi}_j(V(H))$ (Fig 1). R is called a *p-q routing graph* if the maximum outdegree and indegree of R are at most p and q, respectively. We define a *routing* of R as a mapping ρ that maps each edge $(u,v) \in E(R)$ onto a set of edges of H inducing a path connecting u and v. We denote $\rho((u,v))$ simply as $\rho(u,v)$. The *dilation* and *edge-congestion* of ρ are $\max_{e\in E(R)} |\rho(e)|$ and $\max_{e'\in E(H)} |\{e \in E(R) \mid e' \in \rho(e)\}|$, respectively.

An *embedding* $\langle\phi,\rho\rangle$ of a graph G into a graph H is a pair of mappings consisting of a one-to-one mapping $\phi : V(G) \to V(H)$ and a routing ρ of an arbitrary orientation of the graph with the node set $V(H)$ and edge set $\{(\phi(u), \phi(v)) \mid (u,v) \in E(G)\}$. The *dilation* and *edge-congestion* of the embedding $\langle\phi,\rho\rangle$ are defined as the dilation and edge-congestion of ρ, respectively. The minimum dilation of an embedding of G into a path is called the *bandwidth*.

The node- and edge- separators are formally defined as follows: Let $1/2 \leq \beta < 1$ and $s(n)$ be a non-decreasing function. For a graph G, a set S of nodes (edges, resp.) are called a *β-cut nodes (edge, resp.)* if G is partitioned into two subgraphs G_1 and G_2 with at most $\beta|V(G)|$ nodes ($\lceil\beta|V(G)|\rceil$ nodes, resp.) and with no edges connecting G_1 and G_2. S is called a *node-* or *edge-bisector* if $|V(G_1)| = |V(G_2)|$, and S is a set of nodes or edges, respectively. A graph G has a *recursive β-node(edge, resp.)-separator of size $s(n)$* if $|V(G)| = 1$, or if G has a β-cut nodes (edges, resp.) S with $|S| \leq s(|V(G)|)$, and the two subgraphs of G partitioned using S recursively have a β-node(edge, resp.)-separator of size $s(n)$. A *hereditary β-node(edge, resp.)-separator of size $s(n)$* is defined in the same manner, except that

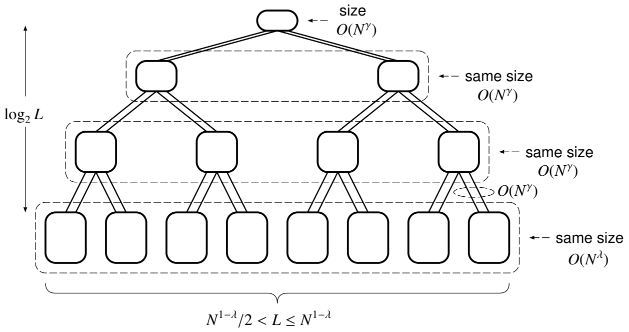

Fig. 3 Tree Structure.

not only the partitioned subgraphs but also all of the subgraphs of G must recursively have a β-node(edge, resp.)-separator of size $s(n)$. By definition, a hereditary separator is also a recursive separator. The process of partitioning G into isolated nodes using a recursive separator is typically referred to as a *decomposition tree*. In this paper, we define a *node-decomposition tree* as the tree such that the root is the cut nodes S to partition G, and the subtrees of the root are recursively the node-decomposition trees for two partitioned subgraphs by S. An *edge-decomposition tree* is defined similarly, except that the root is not cut edges but G itself. I.e., nodes of the edge-decomposition tree are subgraphs of G appeared in partitioning G using a recursive edge-separator (Fig. 2).

3. Tree Structure

In this section, we introduce a tree structure of a guest graph, called *a* (γ, λ)-*tree*, and provide an algorithm to construct the tree structure using a hereditary node-separator of the guest graph.

3.1 Definition

For an N-node graph G and $\gamma < \lambda < 1$, a (γ, λ)-tree of G is a complete binary tree $\mathcal{T}$ that has a set of subsets of $V(G)$ as its nodes and satisfies the following conditions (Fig. 3):

Condition 1 *(1) Every node of G is contained in a single node of $\mathcal{T}$, and every edge of G joins nodes contained in a single node or two adjacent nodes of $\mathcal{T}$.*

(2) $\mathcal{T}$ has L leaves with $L \leq N^{1-\lambda} < 2L$. These leaves have the equal number $\Theta(N^\lambda)$ of nodes of G. We note that the height of $\mathcal{T}$ is equal to $\log_2 L$.

(3) Internal nodes of $\mathcal{T}$ at the same depth (the distance to the root) have the equal number $O(N^\gamma)$ of nodes of G.

(4) For adjacent nodes U and V in $\mathcal{T}$, the number of edges of G connecting nodes of U and nodes of V is $O(N^\gamma)$.

3.2 Construction using Node-Separator

We construct a (γ, λ)-tree using a hereditary node-separator as follows:

Lemma 3 *Any graph G with N nodes, maximum node degree $\Delta \geq 2$, and with a hereditary β-node-separator of size Cn^α ($C > 0$, $0 < \alpha < 1$, $1/2 \leq \beta < 1$) has a $(\alpha + (1-\lambda)\log_2 \Delta, \lambda)$-tree for any λ with $\frac{\log_2 \Delta + \alpha}{\log_2 \Delta + 1} < \lambda < 1$.*

Proof We initially find a node-bisector B of G. This can be done in a similar way as one for constructing a (1/2)-node-decomposition tree [4]. I.e., we first construct a linear layout of nodes of G so that an internal node of the node-decomposition tree $\mathcal{D}$ associated with an hereditary β-node-separator of G is located between the two subtrees of the internal node. Then, we set B to $U \in V(\mathcal{D})$ containing a node of G located at the center of the layout and the ancestors of U. At this point, B is a (1/2)-cut nodes of G. To make B a node-bisector, we move any nodes of the larger subgraph to B so that the resulting two subgraphs have the equal size.

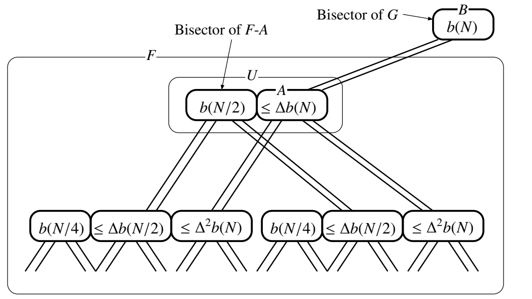

Fig. 4 Construction of Tree Structure.

We set B as the root of a tree structure $\mathcal{T}$ and the two subgraphs of G partitioned by B as the leaves. Assume inductively that we have constructed nodes of $\mathcal{T}$ of height $i \geq 1$ satisfying Condition 1 except the bounds on the size of a node of $\mathcal{T}$ and number of edges of G between adjacent nodes of $\mathcal{T}$. I.e., we have $\mathcal{T}$ with 2^i leaves such that all nodes at depth j with $0 \leq j \leq i$ have the equal size. Unless $2^{i+1} > N^{1-\lambda}$, we grow $\mathcal{T}$ by partitioning every leaf F by a node-bisector U that contains all nodes of G adjacent to nodes in the parent P of F. The node-bisector U of F is simply the union of the nodes A contained in F adjacent to nodes in P and a node-bisector of $F - A$ constructed in the same way as done for B. After constructing all internal nodes at depth i, if they are not balanced, i.e., if an internal node U at depth i has the size smaller than the maximum size X of another internal node at depth i, then we move any $(X - |U|)/2$ nodes from each leaf of U to U, by which all leaves as well as internal nodes at depth i are balanced. We illustrate the construction in Fig. 4.

By the inductive assumption and the above construction, the updated $\mathcal{T}$ also satisfies Condition 1 except the bounds on the size of a node of $\mathcal{T}$ and number of edges of G between adjacent nodes of $\mathcal{T}$. In the rest of the proof, we derive these bounds. Let $\gamma := \alpha + (1-\lambda)\log_2 \Delta$. The size $b(N)$ of the node-bisector B of G is at most $2\sum_{i=0}^{O(\log N)} C(\beta^i N)^\alpha = O(CN^\alpha)$. This is because the size of B is at most the sum of the sizes of a node of $\mathcal{D}$ and its at most $O(\log N)$ ancestors, plus the number of nodes moved from the larger subgraph to B. If I_i is the size of an internal node of $\mathcal{T}$ at depth i, then we have the recurrence $I_i \leq \Delta I_{i-1} + b(N/2^i)$ and $I_0 \leq b(N)$, yielding

$$I_i \leq \sum_{j=0}^{i} \Delta^{i-j} b(N/2^j) = \sum_{j=0}^{i} \Delta^{i-j} O(C(N/2^j)^\alpha)$$

$$= O\left(CN^\alpha \Delta^i \sum_{j=0}^{i} (2^\alpha \Delta)^{-j}\right) = O(CN^\alpha \Delta^i).$$

Thus, we have

$$\max_i\{I_i\} = I_{\log_2 L-1} = O\left(CN^\alpha \Delta^{\log_2 L-1}\right)$$
$$= O\left(CN^\alpha L^{\log_2 \Delta}/\Delta\right) = O\left(CN^{\alpha+(1-\lambda)\log_2 \Delta}\right) = O(N^\gamma).$$

Moreover, it follows that the maximum number of edges of G joining a node of $\mathcal{T}$ is at most $\Delta \cdot \max_i\{I_i\} = \Delta I_{\log_2 L-1} = O(CN^\alpha L^{\log_2 \Delta}) = O(CN^{\alpha+(1-\lambda)\log_2 \Delta}) = O(N^\gamma)$.

It remains to bound the size of leaves of $\mathcal{T}$. Because we grow the height i unless $2^{i+1} > N^{1-\lambda}$, the number L of leaves satisfies $L = 2^i \le N^{1-\lambda} < 2L$. Therefore, the size of a leaf is at most $N/L < 2N^\lambda$. To show that the size of a leaf is at least $\Omega(N^\lambda)$, it suffices to prove that the total size of all internal nodes is $o(N)$, because this implies that the total size of leaves is $\Omega(N)$ and the size of a leaf is $\Omega(N)/L = \Omega(N^\lambda)$. The total size of all internal nodes is at most

$$\sum_{i=0}^{\log_2 L-1} 2^i I_i = \sum_{i=0}^{\log_2 L-1} O(CN^\alpha (2\Delta)^i)$$
$$= O\left(CN^\alpha (2\Delta)^{\log_2 L}\right) = O\left(CN^\alpha L^{\log_2 \Delta+1}\right)$$
$$= O\left(CN^{\alpha+(1-\lambda)(\log_2 \Delta+1)}\right) = O(CN^{\gamma+1-\lambda}),$$

which is $o(N)$ because $\gamma - \lambda = \alpha + (1-\lambda)\log_2 \Delta - \lambda = (\alpha + \log_2 \Delta) - (1 + \log_2 \Delta)\lambda < 0$ by the assumption of λ. □

4. Embedding with Bounded Dilation

In this section, we prove Theorem 1 by showing the following lemma:

Lemma 4 *An N-node graph with a (γ, λ)-tree can be embedded into a grid with at least N nodes, a dimention $d \ge 2$, and with a constant aspect ratio, with a dilation of $O(\frac{d^2}{1-\lambda} \cdot \frac{N^{1/d}}{\log N})$.*

In fact, Theorem 1 is obtained by setting $\lambda := \frac{\log_2 \Delta+1-\epsilon}{\log_2 \Delta+1}$ and $0 < \epsilon < 1 - \alpha$ with $\epsilon = \Omega(1)$, and combining Lemmas 3 and 4. We note that Theorem 1 for the case $\alpha = 0$ is implied by the case $\alpha > 0$.

We prove Lemma 4 by providing a desired algorithm. Broadly, we embed each leaf of a (γ, λ)-tree $\mathcal{T}$ of an N-node guest graph G into each of L subgrids with $\Theta(N/L)$-nodes, called *blocks*, which are obtained by dividing each dimension of the host grid M into $L^{1/d}$ segments of length $\Theta((N/L)^{1/d})$. We map also $L-1$ internal nodes of $\mathcal{T}$ to the blocks as if an $(L-1)$-node complete binary tree were embedded into an L-node grid K using an algorithm presented in [11]. The algorithm of [11] achieves a nearly optimal dilation of $O(dL^{1/d}/\log L)$ with evenly distributing the leaves of the complete binary tree among K. Therefore, we can locate any pair of adjacent nodes of $\mathcal{T}$ on M within a distance of the bandwidth of an L-node complete binary tree multiplied by the diameter of a d-dimensional $\Theta(N/L)$-node grid. Thus, we can bound the maximum dilation of an edge of G by

$$O\left(\frac{dL^{1/d}}{\log L}\right) \cdot O\left(d\left(\frac{N}{L}\right)^{1/d}\right) = O\left(\frac{d^2}{1-\lambda} \cdot \frac{N^{1/d}}{\log N}\right). \quad (1)$$

The rest of this section is devoted to a formal proof of Lemma 4. We review the algorithm of [11] in Sect. 4.1. We define and analyze our algorithm in Sect. 4.2.

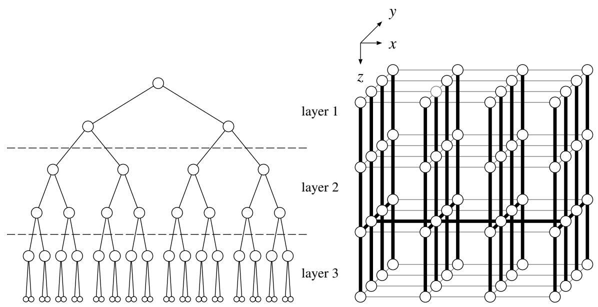

Fig. 5 Embedding a 63-node comlete binary tree into $M(4,4,4)$. Subtrees in layer 1, 2, and 3 are mapped on 1-dimensional grids along x-, y-, z-axis, respectively.

4.1 Embedding of Complete Binary Trees

For $d = 2$ and a given complete binary tree with odd height, the algorithm of [11] splits the tree at half of the height, yielding an upper complete binary tree and $\sqrt{N}$ lower complete binary trees, where N is the number of the nodes of the given tree plus 1. The upper and lower trees have $\sqrt{N}-1$ nodes. The algorithm embeds the upper tree onto the center row of a $\sqrt{N} \times \sqrt{N}$-grid and the lower trees the lower trees onto columns with leaving nodes on the center row empty. This embedding achieves the dilation $O(\sqrt{N}/\log N)$ because each of upper or lower complete binary trees can be embedded onto a path in such a way that the following conditions are satisfied:

Condition 2 *(1) The dilation is exactly equal to the bandwidth. The bandwidth of a complete binary tree is equal to a trivial lower bound $\lceil \frac{(\#\,nodes)-1}{diameter} \rceil$ [11].*
(2) The root is mapped at a middle node of the path.
(3) If $x_1, x_2 \ldots$ with $x_i < x_{i+1}$ for $i \ge 1$ are positions to which the leaves of the complete binary tree are mapped, then $2i - (bandwidth) \le x_i \le 2i + (bandwidth)$ for all $i \ge 1$.

If the given complete binary tree has an even height, then it is similarly embedded into $\sqrt{N} \times 2\sqrt{N}$-grid. The aspect ratio of the underlying grid can be reduced within a constant overheads of dilation and edge-congestion using embeddings of grids into grids [24]. Moreover, as mentioned in [11], the above algorithm can be generalized to a larger dimension $d \ge 3$, in which the dilation is $O(dN^{1/d}/\log N)$. This can be done by splitting a given complete binary tree into d layers with (roughly) the equal height, i.e., removing edges joining nodes at height $\frac{h+1}{d} \cdot i - 1$ and nodes at height $\frac{h+1}{d} \cdot i$ for all $1 \le i < d$, where h is the height of the complete binary tree, and by embedding each layer into line(s) of each dimension (Fig 5). The dilation is derived from the fact that an $N^{1/d}$-node complete binary tree has the bandwidth $O(N^{1/d}/\log(N^{1/d})) = O(dN^{1/d}/\log N)$.

4.2 Embedding of Graphs with (γ, λ)-Tree

4.2.1 Algorithm

We define below our algorithm to embed a given guest graph G with a (γ, λ)-tree $\mathcal{T}$ with L leaves into a host grid M. For simplicity, we ignore rounding of numbers that should be integers, because such rounding is not essential for our result.

(1) Let K be an $L^{1/d} \times \cdots \times L^{1/d}$-grid.
(2) Let a mapping τ map $L-1$ internal nodes of $\mathcal{T}$ to $V(K)$

according to the algorithm of [11].

(3) Split the subtree of $\mathcal{T}$ induced by the internal nodes into d layers with the equal height $\frac{\log_2 L}{d} - 1$ by removing edges joining nodes at height $\frac{\log_2 L}{d} \cdot j$ and nodes at height $\frac{\log_2 L}{d} \cdot j - 1$ for $1 \leq j < d$. We note that the ith layer to the top contains $L^{i/d}$ subtrees with $L^{1/d} - 1$ nodes, which are embedded by τ into $L^{1/d}$-node 1-dimensional grids (i.e., paths) of the same dimension in K. We assume without loss of generality that the subtrees in the ith layer are embedded into paths of dimension i.

(4) We make τ map L leaves of $\mathcal{T}$ to $V(K)$ as follows: Let F_1 and F_2 be leaves of $\mathcal{T}$ having a common parent U, which is a leaf of a subtree $\mathcal{S}$ in the lowest layer. Suppose that τ embeds $\mathcal{S}$ into a 1-dimensional grid P in K. Since $\mathcal{S}$ is embedded so that Condition 2 is satisfied, if U is the ith leaf of $\mathcal{S}$ appeared on P, then the position of U on P is between $2i - \Theta(dL^{1/d}/\log L)$ and $2i + \Theta(dL^{1/d}/\log L)$. Note that $\mathcal{S}$ has $L^{1/d} - 1$ nodes, and hence, its bandwidth is $\Theta(L^{1/d}/\log L^{1/d}) = \Theta(dL^{1/d}/\log L)$. We define $\tau(F_1)$ and $\tau(F_2)$ as the $(2i-1)$st and ith positions on P, respectively.

(5) We partition M into L *blocks*, i.e., subgrids. Roughly, we divide each dimension into $L^{1/d}$ segments of the equal length, by which we can regard M as an $L^{1/d} \times \cdots \times L^{1/d}$-grid of $\Theta(N/L)$-node blocks. We aim to embed nodes of G contained in a leaf $F \in V(\mathcal{T})$, together with nodes in an internal node $U \in V(\mathcal{T})$ with $\tau(U) = \tau(F)$, into the block corresponding to the node $\tau(F)$ in K. Therefore, we partition M into L blocks with exactly the same numbers of nodes as those to be mapped to the blocks. We do this by dividing dimensions in the increasing order. I.e., we first divide dimension 1, along which the subtree of $\mathcal{T}$ contained in the top layer is embedded by τ, so that the resulting $L^{1/d}$ subgrids have the same number of nodes as those to be mapped to the subgrids. For each of the resulting $L^{1/d}$ grids, we then similarly divide the dimension 2, along which the subtrees in the second layer (to the top) are embedded. We repeat this process for all dimensions.

(6) For each $u \in V(K)$, map nodes of G contained in a leaf $F \in V(\mathcal{T})$ with $\tau(F) = u$ and in an internal node $U \in V(\mathcal{T})$ with $\tau(U) = u$, to the block (obtained in the previous step) corresponding to u.

4.2.2 Analysis

Any two adjacent internal nodes of $\mathcal{T}$ are mapped by τ within a distance of $\Theta(dL^{1/d}/\log L)$ on K. Moreover, any adjacent leaf F of $\mathcal{T}$ and its parent U are also mapped within a distance of $\Theta(dL^{1/d}/\log L)$ on K, since for some i, U and F are located on positions between $2i - \Theta(dL^{1/d}/\log L)$ and $2i + \Theta(dL^{1/d}/\log L)$, and $2i-1$ or $2i$ of a 1-dimensional grid in K as mentioned in Step 4. Thus, any two nodes of $\mathcal{T}$ are mapped by τ and τ within a distance of $\Theta(dL^{1/d}/\log L)$.

On the other hand, L blocks in M constructed in Step 5 have a diameter $\Theta(d(N/L)^{1/d})$ since M has a constant aspect ratio and each block has $\Theta(N^{\lambda}) = \Theta(N/N^{1-\lambda}) = \Theta(N/L)$ nodes. We note that the number of nodes of G contained in an internal node of $\mathcal{T}$ is at most $O(N^{\gamma}) = O(N^{\lambda})$. Moreover, we claim the following lemma:

Lemma 5 *For any two blocks B_u and B_v of M corresponding two adjacent nodes u and v of K, the distance between a node in B_u and a node in B_v is at most $O(d(N/L)^{1/d})$.*

Proof Suppose that (u, v) is a dimension-i edge of K. This implies that B_u and B_v are separated in dividing dimension i of a subgrid D of M in Step 5. Since dimensions from 1 to $i-1$ have been already divided, D has side length $\Theta((N/L)^{1/d}) = \Theta(N^{\lambda/d})$ for such dimensions, whereas $\Theta(N^{1/d})$ for dimensions from i to d. If $i = d$, then because B_u and B_v are adjacent in D without no dislocation along dimension other than i, the lemma holds. If $i < d$, then B_u and B_v will possibly have dislocation along dimensions from $i+1$ to d. We prove the amount of such dislocation is at most $O((N/L)^{1/d})$ for each dimension.

When D is partitioned into $L^{1/d}$ subgrids by dividing dimension i, B_u and B_v are contained in consecutive subgrids, say, D_u and D_v, respectively. The numbers of nodes of D_u and D_v differs at most $\Theta(N^{\gamma})$. This follows from the following reasons: D contains a 1-dimensional grid of blocks along dimension i onto which a subtree $\mathcal{S}$ of $\mathcal{T}$ in the ith layer and (unless $i = 1$) a node of $\mathcal{T}$ in a higher layer are mapped. Each of D_u and D_v contains a block onto which either one node of $\mathcal{S}$ or the node in the higher layer is mapped, and blocks onto which all descendants of a leaf of $\mathcal{S}$ are mapped. Nodes of $\mathcal{T}$ at the same height have the equal size. Therefore, the difference of $|D_u|$ and $|D_v|$ is caused only by the difference of sizes of nodes of $\mathcal{S}$ and the node in the higher layer. This difference is at most the size $\Theta(N^{\gamma})$ of an internal node of $\mathcal{T}$. Assume without loss of generality that $|D_v| = |D_u| + O(N^{\gamma})$. If ℓ_j is the side length of D_u in each dimension $j \in [d]$, then D_v has side length $\ell_i + \delta$ in dimension i, where

$$\delta := O\left(\frac{N^{\gamma}}{\prod_{j \in [d] \setminus \{i\}} \ell_j}\right) = O\left(\frac{N^{\gamma}}{N^{\lambda(i-1)/d} N^{(d-i)/d}}\right) = O(N^{\gamma - \lambda + (2\lambda - 1)/d}).$$

Suppose that D_u and D_v are partitioned into smaller grids $D_u^1, \ldots, D_u^{L^{1/d}}$ and $D_v^1, \ldots, D_v^{L^{1/d}}$, respectively, by dividing dimension $i+1$ in Step 5. Then, $|D_u^h| = |D_v^h| + O(N^{\gamma})$ for some $h \in [d]$ and $|D_u^p| = |D_v^p|$ for $p \in [d] \setminus \{h\}$. Let k_p be the side length of D_u^p in dimension $i+1$. Note that D_u^p has side length ℓ_j for each dimension $j \neq i+1$. For $p \in [d] \setminus \{h\}$, the side length of D_v^p in dimension $i+1$ is shorter than k_p, because $|D_u^p| = |D_v^p|$ and D_v^p has longer side length in dimension i than D_u^p has. If δ_p is the shortened amount, then it follows from $|D_u^p| = |D_v^p|$ that $k_p \prod_{j \in [d] \setminus \{i+1\}} \ell_j = (k_p - \delta_p)(\ell_i + \delta) \prod_{j \in [d] \setminus \{i, i+1\}} \ell_j$, by which we obtain

$$\delta_p = \frac{k_p \delta}{\ell_i + \delta} = \frac{\Theta(N^{\lambda/d})\delta}{\Theta(N^{\lambda/d}) + \delta} = O(\delta).$$

The side length of B_v^h in dimension $i+1$ is longer than k_h since $|D_u^h| = |D_v^h| + O(N^{\gamma})$. If δ_h is the lengthened amount, then $\delta_h = O(N^{\gamma/d})$ obviously.

Therefore, maximum amount of dislocation between B_u and B_v along dimension $i+1$ is at most $\sum_p \delta_p = (L^{1/d} - 1)O(\delta) + O(N^{\gamma/d}) = O(N^{\lambda/d} N^{\gamma - \lambda + (2\lambda - 1)/d}) + O(N^{\gamma/d}) = O(N^{\lambda/d})$. We can similarly estimate dislocation amount also for dimensions from $i+2$ to d. Thus, the distance of a node of B_u and B_v is at most $(d-i)O(N^{\lambda/d}) = O(d(N/L)^{1/d})$. □

With the above discussion, we can estimate the dilation as done in (1), and hence, obtain Lemma 4.

5. Bounding Dilation and Edge-Congestion

In the embedding algorithm provided in the previous section, we just defined a mapping of nodes of a (γ, λ)-tree $\mathcal{T}$ of a guest graph G to L blocks, i.e., we did not define a specific position for each node of G. Moreover, we did not specifically define routings for edges of G. An edge of G joins two nodes of G contained in either a single node or distinct nodes of a (γ, λ)-tree $\mathcal{T}$. In the previous section, edges of the former and latter types were implicitly routed with shortest paths inside a block and through at most $O(dL^{1/d}/\log L)$ blocks, respectively. We can obtain an embedding with bounded edge-congestion, as well as bounded dilation, if we can bound edge-congestion in embedding each node of $\mathcal{T}$ into a block and routing edges of the latter type.

In this section, we prove Theorem 2 by combining our embedding with the previous routing and embedding algorithms of [20] with bounded edge-congestion. In Sect. 5.1, we review the previous results that will be combined with our embedding. We define and analyze the combined algorithm in Sect. 5.2.

5.1 Routing and Embedding with Bounded Edge-Congestion

Suppose that a guest graph G has N nodes, maximum node degree Δ, and a recursive node-separator of size $O(n^{\alpha})$, where α is a fixed number with $0 \leq \alpha < 1$. In the embedding of [20], we first construct a recursive edge-separator with bounded *expansion*, i.e., the number of outgoing edges from a subgraph in each recursive step. Specifically, we can construct an edge-decomposition tree $\mathcal{D}$ of G with expansion of $O(\Delta n^{\alpha})$ using the recursive edge-separator of size $O(n^{\alpha})$. Then, we embed G into a d-dimensional host grid with at least N nodes and a constant aspect ratio, with a dilation of $O(dN^{1/d})$ and an edge-congestion of

$$\begin{cases} O(d^2\Delta) & \text{if } d > 1/(1-\alpha), \text{ and} \\ O(\Delta(N^{\alpha-1+\frac{1}{d}} + \log N + d^2)) & \text{if } d \leq 1/(1-\alpha). \end{cases} \tag{2}$$

This is a recursive algorithm, which takes as input a subgraph G' of G appeared as a node of $\mathcal{D}$, together with the multiset X of $O(\Delta|V(G')|^{\alpha})$ nodes of G' incident to outgoing edges from G' (i.e., a node appears in X as many times as the number of outgoing edges incident to the node), and produces an embedding of G' into a host grid M', together with a mapping ψ from X to $V(M')$ satisfying the following condition:

Condition 3 *Nodes in X are evenly distributed across a dimension of M' by ψ. Specifically,* $\max_{v \in \bar{\pi}_i(V(M'))} |\{s \in X \mid \bar{\pi}_i(\psi(s)) = v\}| = \lceil |X|/|V(M')|^{1-1/d} \rceil$ *for some $i \in [d]$.*

This property is essential to route outgoing edges of G' with the desired edge-congestion in ancestor processes, because if X was mapped into a dense "ball" in M', then we would need an edge-congestion of $|X|$ divided by the surface area of the ball, i.e., $\Omega(|X|/|X|^{1-1/d}) = \Omega(|X|^{1/d})$, which could be much larger than the desired congestion. On the contrary, if sources and targets of routing requests are distributed across a dimension, then we can route the requests with nearly tight edge-congestion as stated in the following lemma:

Lemma A ([20]) *Let R be a routing graph on $M := M(\ell_i)_{i\in[d]}$ with $d \geq 2$ and aspect ratio μ. If $\bar{\pi}_h(R)$ is a p-q routing graph with node set $\bar{\pi}_h(V(M))$ for some $h \in [d]$, then R can be routed on M with a dilation at most $2\sum_{i=1}^{d} \ell_i$ and an edge-congestion at most $2\lceil \mu \cdot \max\{p, q\} \rceil$.*

We use Lemma A in a slightly different form. If the host grid has constant aspect ratio, and if sources and targets are distributed across different dimensions, then we can route the requests with similar dilation and edge-congestion via some $(d-1)$-dimensional hyperplane across both these dimensions. Thus, we have the following lemma:

Lemma 6 *Let R be a routing graph with multisets S of sources and T of targets on a grid M with N nodes, a dimension $d \geq 2$, and with constant aspect ratio. Then, R can be routed on M with a dilation of $O(dN^{1/d})$ and an edge-congestion of $O(\max\{\bar{\pi}_i(S), \bar{\pi}_j(T)\}/N^{1-1/d})$, where i and j are any dimensions of M.*

5.2 Combined Embedding

5.2.1 Algorithm

We replace Step 6 of the algorithm in Sect. 4 with the following procedures:

(6) For each $u \in V(K)$, let G_u be the subgraph of G induced by nodes contained in a leaf $F \in V(\mathcal{T})$ with $\tau(F) = u$ and in an internal node $U \in V(\mathcal{T})$ with $\tau(U) = u$. Construct an embedding of G_u into the block B_u corresponding to u using the embedding algorithm of [20], in such a way that the multiset X_u of nodes of G_u incident to outgoing edges from G_u is evenly distributed by a mapping $\psi_u : X_u \to V(B_u)$ across a dimension. We note that since G has a hereditary node-separator of size $O(n^{\alpha})$, G_u has an edge-decomposition tree with expansion $O(\Delta n^{\alpha})$. Because $|X_u| = O(|V(G_u)|^{\gamma/\lambda})$ by $|X_u| = O(N^{\gamma})$ and $|V(G_u)| = \Theta(N^{\gamma})$, putting G_u and outgoing edges from G_u together, G_u can be regarded as a node of a larger edge-decomposition tree of expansion $\max\{O(\Delta n^{\alpha}), O(n^{\gamma/\lambda})\}$.

(7) Let $D^1, \ldots, D^{L^{1/d}}$ be the subgrids obtained from M by partitioning dimension 1 in Step 5. For each $j \in [L^{1/d}]$, there exists a block B^j in D^j to which a node of the subtree $\mathcal{S}$ of $\mathcal{T}$ in the top layer or no internal node of $\mathcal{T}$ is mapped. We route edges of G between two nodes of $\mathcal{S}$ through $B^1, \ldots, B^{L^{1/d}}$. Since there may be dislocation between consecutive blocks as mentioned in the proof of Lemma 5, we actually route the edges through a shortest "bending" sequence Q of blocks containing $B^1, \ldots, B^{L^{1/d}}$. In each block B in Q, more specifically, we define intermediate nodes that evenly distributes edges passing through B on a $(d-1)$-dimensional hyper plane. Then, we apply Lemma 6 on every consecutive blocks in Q. For edges of G between a leaf of $\mathcal{S}$ and its child, we partially route the edges through Q to the subgrid D^j containing the destination block.

(8) For each D^j, we similarly route edges of G between nodes of the subtree $\mathcal{S}^j$ of $\mathcal{T}$ that is in the second layer (to the top) and

mapped to D^j, as well as edges between the root of $\mathcal{S}^j$ and its root, and edges between leaves of $\mathcal{S}^j$ and their children. We repeat this process for all layers $\mathcal{T}$.

5.2.2 Analysis

We prove the above algorithm achieves a dilation and an edge-congestion as stated in the following lemma:

Lemma 7 *An N-node graph with a (γ, λ)-tree can be embedded into a grid with at least N nodes, a dimension $d \geq 2$, and with a constant aspect ratio, with a dilation of $O(\frac{d^2}{1-\lambda} \cdot \frac{N^{1/d}}{\log N})$ and an edge-congestion of*

$$O\left(\frac{d^2}{1-\lambda} \cdot \frac{N^{\gamma-\lambda+1/d}}{\log N}\right) + \begin{cases} O(\Delta) & \text{if } \gamma/\lambda < 1 - 1/d, \\ O(\Delta(N^{\gamma-(1-1/d)\lambda} + \log N + d^2)) & \text{if } \gamma/\lambda \geq 1 - 1/d. \end{cases}$$

Proof Because Lemma 6 achieves a dilation of $O(\text{diam})$ of the underlying grid, we obtain the similar dilation as that of Lemma 4. In what follows, we estimate the edge-congestion.

In Step 6, G_u is regarded as a node of an edge-decomposition tree of expansion $\max\{O(\Delta n^\alpha), O(n^{\gamma/\lambda})\}$, which is in fact $O(\Delta n^{\gamma/\lambda})$ because $\gamma/\lambda = (\alpha + (1-\lambda)\log_2 \Delta)/\lambda > \alpha$. Therefore, it follows from (2) that the edge-congestion of embedding of G_u is $O(\Delta)$ if $\gamma/\lambda < 1 - 1/d$, and $O(\Delta(N^{\lambda(\gamma/\lambda-1+1/d)} + \log N + d^2)) = O(\Delta(N^{\gamma-(1-1/d)\lambda} + \log N + d^2))$ if $\gamma/\lambda \geq 1 - 1/d$.

In Step 7, because any two nodes of $\mathcal{T}$ are mapped within a distance of $\Theta(dL^{1/d}/\log L)$, and because the number of outgoing edges from G_u is $O(N^\gamma)$, the number of edges of G between nodes of $\mathcal{S}$ passing through a block is at most

$$\Theta\left(\frac{dL^{1/d}}{\log L}\right) \cdot O(N^\gamma) = O\left(\frac{d}{1-\lambda} \cdot \frac{N^{\gamma+(1-\lambda)/d}}{\log N}\right).$$

Because these edges are evenly distributed across a dimension in the block, and each block has a constant aspect ratio, it follows from Lemma 6 that the edge-congestion is

$$O\left(\frac{d}{1-\lambda} \cdot \frac{N^{\gamma+(1-\lambda)/d}}{N^{\lambda(1-1/d)} \log N}\right) = O\left(\frac{d}{1-\lambda} \cdot \frac{N^{\gamma-\lambda+1/d}}{\log N}\right).$$

Since this routing is repeated for d layers, putting the edge-congestion in Steps 6 and 7 together, we have the desired edge-congestion.

□

To prove Theorem 2, we set $\lambda := \frac{\log_2 \Delta + 1 - \epsilon}{\log_2 \Delta + 1}$ with $0 < \epsilon < 1 - \alpha$. For the case $d > 1/(1-\alpha)$, we set $\epsilon := 1 - \alpha - 1/d > 0$. Then, $\gamma - (1-1/d)\lambda < \gamma - \lambda + 1/d = \alpha + (1-\lambda)\log_2 \Delta - \lambda + 1/d = \alpha - 1 + 1/d + \epsilon = 0$. By Lemma 7, therefore, we obtain a dilation of $O(\frac{d^2 \log \Delta}{1-\alpha-1/d} \cdot \frac{N^{1/d}}{\log N}) = O(\frac{N^{1/d} \log \Delta}{\log N})$ and an edge-congestion of $O(\frac{d \log \Delta}{1-\alpha-1/d} \cdot \frac{1}{\log N}) + O(\Delta) = O(\Delta)$.

For the case $d \leq 1/(1-\alpha)$, by Lemma 7, we obtain a dilation of $O(\frac{N^{1/d} \log \Delta}{\epsilon \log N})$ and an edge-congestion of

$$O\left(\frac{d^2 \log \Delta}{\epsilon} \cdot \frac{N^{\gamma-\lambda+1/d}}{\log N}\right) + O(\Delta(N^{\gamma-\lambda+1/d} + \log N + d^2)) = O(\Delta(N^{\alpha-1+1/d+\epsilon} + \log N))$$

for any $1/\log N \leq \epsilon < 1 - \alpha$. Thus, we have obtained Theorem 2.

References

[1] Alon, N., Seymour, P. and Thomas, R.: A Separator Theorem for Non-planar Graphs, *J. Amer. Math. Soc.*, Vol. 3, No. 4, pp. 801–808 (1990).
[2] Arora, S., Rao, S. and Vazirani, U.: Expander Flows, Geometric Embeddings and Graph Partitioning, *J. ACM*, Vol. 56, No. 2, p. article 5 (2009).
[3] Bhatt, S. N., Chung, F. R. K., Leighton, F. T. and Rosenberg, A. L.: Efficient Embeddings of Trees in Hypercubes, *SIAM J. Comput.*, Vol. 21, No. 1, pp. 151–162 (1992).
[4] Bhatt, S. N. and Leighton, F. T.: A Framework for Solving VLSI Graph Layout Problems, *J. Comput. Syst. Sci.*, Vol. 28, pp. 300–343 (1984).
[5] Bui, T. N. and Jones, C.: Finding good approximate vertex and edge partitions is NP-hard, *Inform. Process. Lett.*, Vol. 42, pp. 153–159 (1992).
[6] Even, G., Guha, S. and Schieber, B.: Improved Approximations of Crossings in Graph Drawings and VLSI Layout Areas, *SIAM J. Comput.*, Vol. 32, No. 1, pp. 231–252 (2003).
[7] Feige, U., Hajiaghayi, M. and Lee, J. R.: Improved Approximation Algorithms for Minimum Weight Vertex Separators, *SIAM J. Comput.*, Vol. 38, No. 2, pp. 629–657 (2008).
[8] Formann, M. and Wagner, F.: The VLSI Layout Problem in Various Embedding Models, *Proc. 16th International Workshop on Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Science, Vol. 484, Springer, pp. 130–139 (1991).
[9] Garey, M. R., Johnson, D. S. and Stockmeyer, L.: Some Simplified NP-complete Graph Problems, *Theoret. Comput. Sci.*, Vol. 1, No. 3, pp. 237–267 (1976).
[10] Gregori, A.: Unit-Length Embedding of Binary Trees on a Square Grid, *Inform. Process. Lett.*, Vol. 31, pp. 167–173 (1989).
[11] Heckmann, R., Klasing, R., Monien, B. and Unger, W.: Optimal Embedding of Complete Binary Trees into Lines and Grids, *J. Parallel Dist. Comp.*, Vol. 49, No. 1, pp. 40–56 (1998).
[12] Heun, V. and Mayr, E. W.: A New Efficient Algorithm for Embedding an Arbitrary Binary Tree into Its Optimal Hypercube, *J. Alg.*, Vol. 20, pp. 375–399 (1996).
[13] Heun, V. and Mayr, E. W.: Embedding Graphs with Bounded Treewidth into Their Optimal Hypercubes, *J. Algorithms*, Vol. 43, pp. 17–50 (2002).
[14] Heun, V. and Mayr, E. W.: Efficient Embeddings into Hypercube-like Topologies, *The Computer Journal*, Vol. 46, pp. 632–644 (2003).
[15] Kim, Y. M. and Lai, T.-H.: The Complexity of Congestion-1 Embedding in a Hypercube, *J. Algorithms*, Vol. 12, pp. 246–280 (1991).
[16] Kloks, T.: *Treewidth—Computations and Approximations*, Lecture Notes in Computer Science, Vol. 842, Springer (1994).
[17] Leighton, T. and Rao, S.: Multicommodity Max-Flow Min-Cut Theorems and Their Use in Designing Approximation Algorithms, *J. ACM*, Vol. 46, No. 6, pp. 787–832 (1999).
[18] Leiserson, C. E.: *Area Efficient VLSI Computation*, MIT Press (1983).
[19] Lipton, R. J. and Tarjan, R. E.: A Separator Theorem for Planar Graphs, *SIAM J. Appl. Math.*, Vol. 36, No. 2, pp. 177–189 (1979).
[20] Matsubayashi, A.: Separator-Based Graph Embedding into Multidimensional Grids with Small Edge-Congestion (in preparation).
[21] Obrenić, B.: An Approach to Emulating Separable Graphs, *Math. Systems Theory*, Vol. 27, pp. 41–63 (1994).
[22] Rosenberg, A. L. and Heath, L. S.: *Graph Separators, with Applications*, Kluwer Academic/Plenum Publishers (2001).
[23] Röttger, M. and Schroeder, U.-P.: Embedding 2-Dimensional Grids into Optimal Hypercubes with Edge-Congestion 1 or 2, *Parallel Process. Lett.*, Vol. 8, No. 2, pp. 231–242 (1998).
[24] Röttger, M. and Schroeder, U.-P.: Efficient Embeddings of Grids into Grids, *Discrete Appl. Math.*, Vol. 108, No. 1–2, pp. 143–173 (2001).
[25] Ullman, J. D.: *Computational Aspects of VLSI*, Computer Science Press (1984).
[26] Valiant, L. G.: Universality Considerations in VLSI Circuits, *IEEE Trans. Comput.*, Vol. C-30, No. 2, pp. 135–140 (1981).
[27] Wagner, A. and Corneil, D. G.: Embedding Trees in a Hypercube is NP-Complete, *SIAM J. Comput.*, Vol. 19, No. 4, pp. 570–590 (1990).