論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Graph embedding with small congestion and its applications to parallel and VLSI computation
著者(和文)	松林昭
Author(English)	Matsubayashi Akira
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第3183号, 授与年月日:1996年3月26日, 学位の種別:課程博士, 審査員:
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第3183号, Conferred date:1996/3/26, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Graph Embedding with Small Congestion and Its Applications to Parallel and VLSI Computation

Akira Matsubayashi

Department of Electrical and Electronic Engineering
Graduate School of Science and Engineering
Tokyo Institute of Technology

1996

Contents

Chapter 1

Introduction

1.1 Backgrounds

The *graph embedding problem* is to embed a *guest graph* into a *host graph* with certain constraints and/or optimization criteria, such as dilation, congestion (edge-congestion), load, and expansion. A wide variety of problems in the field of parallel computation and VLSI layout have been studied as the graph embedding problem.

In an actual parallel machine with a large number of processing elements, each processing element is connected with the limited number of processing elements by connection links. In fact, the computational performance of a parallel machine depends on the "structure" of interconnections of processing elements, and there are various structures for parallel machines such as grids, hypercubes, and so on [17]. Therefore, it is very important for effective utilization of parallel machines to implement parallel algorithms on the parallel machines efficiently.

The structure of a parallel machine can be represented by the graph called the *processor interconnection graph*, whose vertices and edges represent processing elements and connection links, respectively. The structure of a parallel algorithm also can be represented by the graph called the *communication graph*, whose vertices and edges represent processes and communications between processes, respectively. Therefore, the problem of efficiently implementing parallel algorithms on parallel machines can be modeled as the graph embedding problem, in which guest graphs and host graphs represent communication graphs and processor interconnection graphs, respectively. Moreover, we can also consider the graph embedding problem in which guest graphs represent one class of processor interconnection graphs and host graphs represent the other class of processor interconnection graphs. This is a model of the problem of efficiently implementing parallel algorithms designed for one parallel machine into another parallel machine.

It is well-known that the dilation and/or congestion of the embedding are lower bounds on the communication overhead. The load of the embedding is a lower bound on the computation delay of processing elements, and it is significant when the number of processes is more than that of processing elements, i.e. the expansion is less than one. Therefore, the purpose of implementing parallel algorithms on parallel machines is to minimize the dilation, congestion, load, and expansion. The results on graph embedding problems associated with parallel computing have been studied in a great deal of literature e.g. [4][1][2][18][5].

The problem of efficiently laying out VLSI systems onto VLSI chips have also been extensively studied as the graph embedding problem [21][29][3][8][9][6]. In the graph embedding problem associated with VLSI layout, a guest graph represents a planar (hyper)graph modeling connection requirements of a system, and a host graph usually represents a rectangular grid [29][8][9][6], a hexagonal array [9], or a path [11][31][14][15], which model slots for modules and routing areas for wires on wafer. The dilation, congestion, and expansion of the layout corresponds to wire length, wire congestion, and the layout area, respectively. The layout into paths are often called the *linear layout*. In linear layout, the dilation and the congestion are called *bandwidth* and *cutwidth*, respectively. Frequently, the load and congestion are fixed at one as constraints of the layout since, in certain design rules, at most one module and at most one wire can be placed on a slot and a routing area, respectively. In addition, crossing number is often considered as one of the optimization criteria or constraints of the layout [29][7].

In this thesis, we consider the minimal congestion embeddings of graphs with unit load. It is suggested in [12] and [16] that the communication overhead is essentially independent of dilation in architectures that utilizes circuit switching and "worm-hole" routing, such as Intel iPSC/2, iPSC/860, Paragon, iWarp, and the CM-2, CM-5. In particular, parallel algorithms implemented on such machines with congestion one can achieve same performance as implemented with unit dilation. In VLSI layout, the minimal congestion embeddings are crucial in the sense that the congestion is a lower bound for the number of layers. In addition, the minimal cutwidth linear layout is extensively investigated [15][14][31].

1.2 Graph Embedding

Let G be a graph and let $V(G)$ and $E(G)$ denote the vertex set and edge set of G, respectively. We denote by $\Delta(G)$ the maximum degree of a vertex in G. A tree T is said to be *binary* if $\Delta(T) \leq 3$. An *embedding* $\langle \phi, \rho \rangle$ of a graph G into a graph H is defined by a one-to-one mapping $\phi : V(G) \rightarrow V(H)$, together with a mapping ρ that maps each edge $(u, v) \in E(G)$ onto a path $\rho(u, v)$ in H that connects $\phi(u)$ and $\phi(v)$. The *congestion* of an edge $e' \in E(H)$ under $\langle \phi, \rho \rangle$ is the number of edges e in G such that $\rho(e)$ contains e'. The congestion of an embedding $\langle \phi, \rho \rangle$ is the maximum congestion of an edge in H. The *one dimensional n-grid* denoted by $M(n)$ is the graph with vertex set $\{0, 1, \ldots, n-1\}$ and edge set $\{(i, i+1) \mid 0 \leq i \leq n-2\}$. A Cartesian product $M(n_1) \times M(n_2)$ is called a *two dimensional $n_1 \times n_2$-grid* and denoted by $M(n_1, n_2)$. We define that $n_1 n_2$ is the *area* of $M(n_1, n_2)$. $M(2, n)$ is called an *n-ladder* and denoted by $L(n)$. The embedding of a graph G into a two dimensional grid H is called a *layout* of G into H if it has unit congestion. A layout $\langle \phi, \rho \rangle$ of G into H is said to be *planar* if $\rho(e_1)$ and $\rho(e_2)$ are internally vertex-disjoint for any distinct $e_1, e_2 \in E(G)$. The *n-cube* (*n-dimensional cube*) $Q(n)$ is the graph with 2^n vertices labeled 0 through $2^n - 1$ such that two vertices are joined by an edge if and only if their labels in the binary representation differ by exactly one bit.

We can show by combining the results of Formann and Wagner [8] and Kramer and Leeuwen [13] that the problem of determining, for a planar graph G with maximum vertex degree at most 4 and integers m and n, whether G is embeddable in an $m \times n$ grid with unit congestion is NP-hard. We consider the following problem, which is a variant of the problem above.

Graph k-Layout

Instance A planar graph G with $\Delta(G) \leq 4$ and an integer n.

Question Does there exist a layout of G into $M(k, n)$?

We prove that the GRAPH k-LAYOUT is NP-complete for any fixed $k \geq 3$. GRAPH 1-LAYOUT can be trivially solved in polynomial time. Although we do not know the complexity of GRAPH 2-LAYOUT, we consider a closely related problem of laying out a graph into a ladder. We show a necessary and sufficient condition for a graph to be laid out into $L(\infty)$ as follows:

> A graph G can be laid out into $L(\infty)$ if and only if $\Delta(G) \leq 3$ and $G[S]$ has proper-path-width at most 2, where $S = \{v \in V(G) \mid \deg_G(v) \geq 2\}$.

In connection with the characterization, we describe a linear time algorithm for computing the proper-path-decomposition of width at most 2. Based on the characterization and the algorithm, we can obtain a linear time algorithm for deciding if a given graph can be laid out into $L(\infty)$. In fact, we show a linear time algorithm for laying out a graph G satisfying the condition into $L(|V(G)|)$. We review the proper-path-width and summarize our results on the proper-path-width in Subsection 1.3.

Kim and Lai [12] showed that for a given N-vertex graph G and a hypercube it is NP-complete to determine whether G is embeddable in the hypercube with unit congestion, but G can be embedded with unit congestion in $Q(6\lceil \log N \rceil)$ if $\Delta(G) \leq 6\lceil \log N \rceil$. They posed the question of whether G can be embedded with unit congestion in a hypercube of dimension less than $6\lceil \log N \rceil$. We answer the question by proving the following:

> Every N-vertex graph G can be embedded with unit congestion in $Q(2\lceil \log N \rceil)$ if $\Delta(G) \leq 2\lceil \log N \rceil$.

The basic idea of the embedding is quite simple. We adopt a plain labeling of vertices and a simple routing for edges, and the embedding can be constructed in polynomial time.

Bhatt, Chung, Leighton, and Rosenberg [2] showed that every N-vertex binary tree can be embedded in $Q(\lceil \log N \rceil)$ with dilation and congestion both $O(1)$. Although their embedding is optimal to within a constant factor, there is much room for reducing the dilation and/or congestion. They posed the question of finding a simple embedding of binary trees into hypercubes with smaller dilation and/or congestion. Monien and Sudborough [18] partially answer the question by proving that every N-vertex binary tree can be embedded in $Q(\lceil \log N \rceil)$ with dilation at most 5. We also partially answer the question by proving the following:

> Every N-vertex binary tree can be embedded in $Q(\lceil \log N \rceil)$ with congestion at most 5.

This is the first result that shows a simple embedding of a binary tree into an optimal sized hypercube with explicit small congestion of 5. The embedding is quite simple. We use the postorder labeling of vertices and a greedy (shortest path) routing for edges, and the embedding can be constructed in polynomial time. It is interesting that such a simple embedding guarantees a small congestion of 5.

1.3 Proper-Path-Decomposition

For a graph G, a sequence $\mathcal{X} = (X_1, \ldots, X_r)$ of subsets of $V(G)$ is called a *proper-path-decomposition* of G if $\mathcal{X}$ satisfies the following conditions.

(a) $X_i \not\subseteq X_j \ (i \neq j)$;

(b) $\bigcup_{1 \leq i \leq r} X_i = V(G)$;

(c) $\forall (u, v) \in E(G)\ \exists i$ s.t. $u, v \in X_i$;

(d) $X_a \cap X_c \subseteq X_b \ (1 \leq a \leq b \leq c \leq r)$;

(e) $|X_a \cap X_c| \leq |X_b| - 2$ if $|X_b| \geq 2 \ (1 \leq a < b < c \leq r)$.

The *width* of $\mathcal{X}$ is $\max_{1 \leq i \leq r} |X_i| - 1$. The *proper-path-width* of G is the minimum width over all proper-path-decompositions of G, and denoted by $ppw(G)$. A proper-path-decomposition of G is said to be *optimal* if it has width of $ppw(G)$. A proper-path-decomposition of width k is called a k-proper-path-decomposition. Proper-path-width was introduced by Takahashi, Ueno, and Kajitani [23] as a variant of path-width introduced by Robertson and Seymour. The proper-path-width not only plays an important role for the graph layout into ladders as mentioned in Subsection 1.2, but also has various applications such as VLSI layout, search games [25], and other graph embedding problems [27][28].

It is shown in [25] that the problem of determining, given a graph G and an integer k, whether $ppw(G) \leq k$ is NP-complete, while the problem can be solved in polynomial time for trees. In fact, the optimal proper-path-decomposition of an N-vertex tree can be computed in $O(N \log N)$ time. Although it is known that the problem is in P if k is a fixed integer [24], we do not have an explicit polynomial time algorithm to solve the problem for a fixed integer $k \geq 3$. If $k = 1$ then the problem can be solved trivially in polynomial time. This is because G has proper-path-width 1 if and only if G is a collection of paths. We have an $O(N \log^2 N)$ time algorithm to solve the problem for $k = 2$ based on the minor theory, which is mentioned in [24]. However, the algorithm is neither practical nor constructive since the time complexity involves an enormous constant factor and the algorithm provides no proper-path-decomposition. As mentioned in Subsection 1.2, it is a key procedure in the application to the graph layout into ladders to compute the proper-path-decomposition of width at most 2 of a given graph with maximum vertex degree at most 3.

We show a necessary and sufficient condition for a graph with maximum vertex degree at most 3 to have proper-path-width at most 2, and based on the condition, we give a practical linear time algorithm for computing a proper-path-decomposition of width at most 2.

1.4 Thesis Outline

This thesis is organized as follows.

In Chapter 2, we discuss the results on the proper-path-width which will be used in Chapter 3. We characterize graphs with maximum vertex degree at most 3 and proper-path-width at most 2. Based on the characterization, we construct a practical linear time algorithm for computing a proper-path-decomposition with width at most 2 of a graph with maximum vertex degree at most 3.

In Chapter 3, we show the complexity results on graph embeddings into grids. We first state Graph k-Layout problem. To prove that Graph k-Layout ($k \geq 3$) is NP-complete, we construct a pseudo-polynomial reduction from 3-Partition which is well known to be NP-complete in the strong sense to Graph k-Layout for $k \geq 3$. Moreover, based on the results on channel routing problem, we show that Graph k-Layout is in NP.

We next consider the problem of laying out a graph into a ladder, which is closely related with GRAPH 2-LAYOUT, and show a necessary and sufficient condition for a graph to be laid out into $L(\infty)$. Based on the characterization and the algorithm described in Chapter 2, we construct a linear time algorithm which decides if a given graph G can be laid out into $L(\infty)$ and lays out G into $L(|V(G)|)$ whenever G satisfies the condition. In addition, we estimate the tight upper and lower bounds for the minimum area of a ladder into which an N-vertex graph G can be laid out.

In Chapter 4, we show some results on graph embeddings into hypercubes. First, we prove that every N-vertex graph G can be embedded with unit congestion in $Q(2\lceil \log N \rceil)$ if $\Delta(G) \leq 2\lceil \log N \rceil$. This is done by constructing an embedding $\langle \phi_1, \rho_1 \rangle$ of G into $Q(2\lceil \log N \rceil)$ with unit congestion. Next, we prove that every N-vertex binary tree can be embedded in $Q(\lceil \log N \rceil)$ with congestion at most 5. This is done by constructing an embedding $\langle \phi_2, \rho_2 \rangle$ of a binary tree into $Q(\lceil \log N \rceil)$ and analyzing the congestion of $\langle \phi_2, \rho_2 \rangle$. In addition, this analysis is shown to be tight possible by constructing an example of binary trees for which the congestion of $\langle \phi_2, \rho_2 \rangle$ is 5.

A summary of the results is given in Chapter 5.

Chapter 2

Proper-Path-Decomposition of Width 2

2.1 Introduction

Let G be a graph and let $V(G)$ and $E(G)$ denote the vertex set and edge set of G, respectively. For a graph G, a sequence $\mathcal{X} = (X_1, \ldots, X_r)$ of subsets of $V(G)$ is called a *proper-path-decomposition* of G if $\mathcal{X}$ satisfies the following conditions.

Condition 2.1

(a) $X_i \not\subseteq X_j$ $(i \neq j)$;

(b) $\bigcup_{1 \leq i \leq r} X_i = V(G)$;

(c) *for any* $(u, v) \in E(G)$, *there exists an* i *such that* $u, v \in X_i$;

(d) *for all* a, b, *and* c *with* $1 \leq a \leq b \leq c \leq r$, $X_a \cap X_c \subseteq X_b$;

(e) *for all* a, b, *and* c *with* $1 \leq a < b < c \leq r$, $|X_a \cap X_c| \leq |X_b| - 2$ *if* $|X_b| \geq 2$.

The *width* of $\mathcal{X}$ is $\max_{1 \leq i \leq r} |X_i| - 1$. The *proper-path-width* of G is the minimum width over all proper-path-decompositions of G, and denoted by $ppw(G)$. A proper-path-decomposition is said to be *optimal* if it has width of $ppw(G)$. A proper-path-decomposition of width k is called a k-proper-path-decomposition. Proper-path-width was introduced by Takahashi, Ueno, and Kajitani [23] as a variant of path-width introduced by Robertson and Seymour [20]. The proper-path-width not only plays an important role for the graph layout into ladders as mentioned in Subsection 1.2, but also has various applications such as VLSI layout, search games [25], and graph embeddings [27][28].

It is shown in [25] that the problem of determining, given a graph G and an integer k, whether $ppw(G) \leq k$ is NP-complete, while the problem can be solved in polynomial time for trees. In fact, the optimal proper-path-decomposition of an N-vertex tree can be computed in $O(N \log N)$ time [26]. Although it is known that the problem is in P if k is a fixed integer [24], we do not have an explicit polynomial time algorithm to solve the problem for a fixed integer $k \geq 3$. If $k = 1$ then the problem can be solved trivially in polynomial time. This is because G has proper-path-width 1 if and only if G is a collection of paths. We have an $O(N \log^2 N)$ time algorithm to solve the problem for $k = 2$ based on the minor theory, which is mentioned in [24]. However, the algorithm is neither practical nor constructive since the time complexity involves an enormous constant factor and the algorithm provides no optimal proper-path-decomposition. As mentioned in Subsection 1.2, it is a key procedure for constructing the efficient graph layout into ladders to compute the proper-path-decomposition of width at most 2 of a given graph with maximum vertex degree at most 3.

We show a necessary and sufficient condition for a graph with proper-path-width at most 2, and based on the condition, we give a practical linear time algorithm for computing a proper-path-decomposition of width at most 2.

2.2 Preliminaries

For a sequence $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ of elements, X_1 and X_r are called the *head* of $\mathcal{X}$ and its *tail*, respectively. We denote the sequence without elements by *nul*. For sequences $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ and $\mathcal{Y} = (Y_1, Y_2, \ldots, Y_q)$, we define that $\mathcal{X} + \mathcal{Y} = (X_1, X_2, \ldots, X_r, Y_1, Y_2, \ldots, Y_q)$. For a sequence $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ of subsets of a set Ω and $W \subseteq \Omega$, we define that $\mathcal{X} \cup W = (X_1 \cup W, X_2 \cup W, \ldots, X_r \cup W)$ and $\mathcal{X} \cap W = (X_1 \cap W, X_2 \cap W, \ldots, X_r \cap W)$.

$N_G(v)$ is the set of vertices adjacent to a vertex v in a graph G. $\Gamma_G(v)$ is the set of edges incident to a vertex v in G. $|\Gamma_G(v)|$ is called the *degree* of v and denoted by $\deg_G(v)$.

Let $\Delta(G) = \max\{\deg_G(v) \mid v \in V(G)\}$. For $U \subseteq V(G)$, let $G[U]$ be the subgraph of G induced by U, and let $G - U$ denote $G[V(G) - U]$. Similarly, for $S \subseteq E(G)$, let $G[S]$ be the subgraph of G induced by S, and let $G - S$ denote the graph obtained from G by deleting S. For graphs G and H, $G \cup H$ is the graph with vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H)$, and $G \cap H$ is $G[V(G) \cap V(H)]$, or $H[V(G) \cap V(H)]$. Although a path is a graph, we often denote a path by a sequence of vertices in which consecutive two vertices are adjacent in the path.

A vertex v of G is a *cut vertex* if $E(G)$ can be partitioned into two nonempty subsets E_1 and E_2 such that $G[E_1]$ and $G[E_2]$ have just the vertex v in common. A connected graph that has no cut vertices is called a *block*. Every block with at least three vertices is 2-*connected*. A *block of a graph* is a subgraph that is a block and is maximal with respect to this property.

A planar graph is *outer planar* if it has a planar drawing in which the outer region includes all of its vertices. An edge is *outer* if it is included in the outer region, and is *inner* otherwise. For a subset $U = \{u_1, \dots, u_l\}$ of vertices of an outer planar graph G, $G[U]$ is an *end-region* of G if (u_i, u_{i+1}) $(1 \leq i \leq l-1)$ is an outer edge, u_1 and u_l are adjacent, and u_i $(1 < i < l)$ is incident to no inner edge. Any 2-connected outer planar graph has at least one end-region, and it has at least two end-regions if it has an inner edge.

A graph H is a *minor* of a graph G if H is isomorphic to a graph obtained from a subgraph of G by contracting edges. A family $\mathcal{F}$ of graphs is said to be *minor-closed* if the following condition holds: If $G \in \mathcal{F}$ and H is a minor of G then $H \in \mathcal{F}$. A graph G is a *minimal forbidden minor* for a minor-closed family $\mathcal{F}$ of graphs if $G \notin \mathcal{F}$ and any proper minor of G is in $\mathcal{F}$. $\mathcal{F}$ is characterized by the minimal forbidden minors for $\mathcal{F}$. That is, a graph G is in $\mathcal{F}$ if and only if no minimal forbidden minor for $\mathcal{F}$ is a minor of G. For a positive integer k, the family $\mathcal{P}_k$ of graphs with proper-path-width at most k is minor-closed. All the minimal forbidden minors for $\mathcal{P}_1$ are K_3 and $K_{1,3}$ [23], and all for $\mathcal{P}_2$ are 36 graphs shown in Figure 2.1 [24].

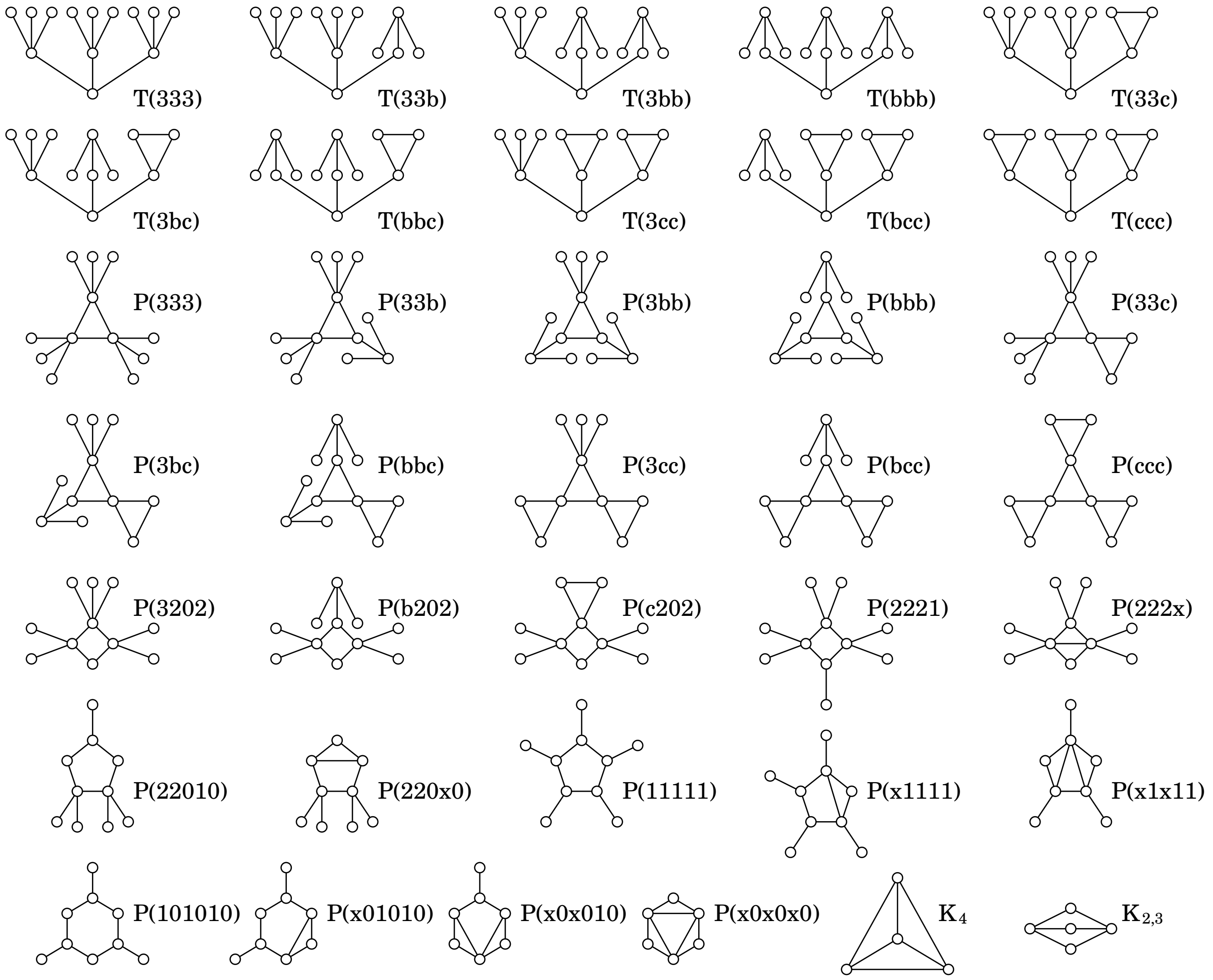

Figure 2.1: Minimal forbidden minors for $\mathcal{P}_2$.

2.3 Algorithm for Proper-Path-Decomposition of Width 2

In this section, we show a necessary and sufficient condition for a graph with maximum vertex degree 3 to have proper-path-width 2, and based on the condition, we give a practical linear time algorithm for computing a proper-path-decomposition of width at most 2.

Suppose that G' is a graph obtained from a graph G by deleting self-loops and replacing multiple edges with a single edge. A proper-path-decomposition of G' is also that of G, and vice versa, by definition. Therefore, an optimal proper-path-decomposition of G' is also that of G. The optimal proper-path-decomposition of a graph can be obtained by concatenating optimal proper-path-decompositions of connected components. From these facts, we assume that the graphs considered in this section are simple and connected.

2.3.1 Binary Tree

An algorithm for computing the optimal proper-path-decomposition of an N-vertex tree T is shown in [25]. Since this algorithm computes $ppw(T)$ in $O(N)$ time and provides the optimal proper-path-decomposition of T in $O(N\,ppw(T))$ time, we can compute the 2-proper-path-decomposition of T with $ppw(T) = 2$ in linear time.

In this subsection, we show algorithms for computing the proper-path-decomposition of a binary tree with width at most 2 satisfying some conditions. These algorithms will be used to construct the algorithm for general graphs.

Lemma 2.1 *For a path $P = (p_0, \ldots, p_l)$, there exists a 1-proper-path-decomposition $\mathcal{X} = (X_1, \ldots, X_r)$ of P such that $p_0 \in X_1$ and $p_l \in X_r$.*

Proof Let $\mathcal{X} = (X_1, \ldots, X_l)$ with $X_i = \{p_{i-1}, p_i\}$ $(1 \leq i \leq l)$ if $l \geq 1$, $\mathcal{X} = (\{p_0\})$ otherwise. $\mathcal{X}$ is clearly a desired proper-path-decomposition. □

Algorithm `PPD_PATH` shown in Figure 2.2 is the formal description of the procedure written in the proof of Lemma 2.1. Trivially, `PPD_PATH` can be executed in linear time.

The following lemma is a characterization of a tree with proper-path-width at most k, and is a basis for our algorithm for binary trees.

`Procedure PPD_PATH` (P)

$$\left[\begin{array}{ll} \text{Input:} & \text{a path } P=(p_0,p_1,\ldots,p_l); \\ \text{Output:} & \text{the 1-proper-path-decomposition } (X_1,X_2,\ldots,X_r) \text{ of } P \text{ such that} \\ & p_0 \in X_1 \text{ and } p_l \in X_r; \end{array}\right]$$

1. `if` $l = 0$ `then return` ($(\{p_0\})$);
2. `for each` $1 \leq i \leq l$ `do`

   $X_i := \{p_{i-1}, p_i\}$;

   `endfor` ;
3. `return` ($(X_1, X_2, \ldots, X_l)$);

`End`

Figure 2.2: Algorithm for computing the 1-proper-path-decomposition of a path

Lemma A (Tayu and Ueno [28]) *For a tree T and an integer $k \geq 2$, $ppw(T) \leq k$ if and only if there exists a path P in T such that $ppw(T - V(P)) \leq k - 1$.* □

k-spine of T is a path satisfying the condition of Lemma A.

Lemma 2.2 *For a binary tree T with $ppw(T) = 2$ and its 2-spine $P = (p_0, \ldots, p_l)$ such that $\deg_T(p_0) = \deg_T(p_l) = 1$, there exists a 2-proper-path-decomposition $\mathcal{X} = (X_1, \ldots, X_r)$ of T such that $p_0 \in X_1 - \bigcup_{1<i\leq r} X_i$ and $p_l \in X_r - \bigcup_{1\leq i<r} X_i$.*

Proof Since P is a 2-spine of T, it follows from Lemma A that $ppw(T - V(P)) \leq 1$. Thus, each connected component of $T - V(P)$ is a path. For $0 < i < l$, at most one connected component H_i of $T - V(P)$ has a vertex adjacent to p_i since $\Delta(T) \leq 3$. Let $I = \{i \mid 0 < i < l, \deg_T(p_i) = 3\}$. We define the sequence $\mathcal{X}$ of subsets of $V(T)$ as follows:

$$\mathcal{X} = (S_1) + \mathcal{Y}_1 + (S_2) + \cdots + (S_{l-1}) + \mathcal{Y}_{l-1} + (S_l), \text{ where}$$
$$\text{for } 1 \leq i \leq l,\ S_i = \begin{cases} \{p_{i-1}, p_i\} \cup V(H_i) & \text{if } i \in I \text{ and } |V(H_i)| = 1 \\ \{p_{i-1}, p_i\} & \text{otherwise} \end{cases}$$
$$\text{for } 1 \leq i < l,\ \mathcal{Y}_i = \begin{cases} \texttt{PPD\_PATH}(H_i) \cup \{p_i\} & \text{if } i \in I \text{ and } |V(H_i)| \geq 2 \\ nul & \text{otherwise} \end{cases}$$

We show that $\mathcal{X}$ is a desired 2-proper-path-decomposition. The following claim can be easily observed from the definition of $\mathcal{X}$.

Claim 2.3

1. *p_0 appears in S_1.*
2. *For $0 < i < l$, p_i appears in $S_{i-1} \cap S_i$ and every element of $\mathcal{Y}_i$.*
3. *p_l appears in S_l.*
4. *For $i \in I$ and $|V(H_i)| \geq 2$, $v \in V(H_i)$ appears in at most two consecutive elements of $\mathcal{Y}_i$.*
5. *For $i \in I$ and $|V(H_i)| = 1$, $v \in V(H_i)$ appears in S_i.*

It is clear by Claim 2.3 that $\mathcal{X}$ satisfies (a), (b), and (c) in Condition 2.1. Moreover, $\mathcal{X}$ satisfies (d) in Condition 2.1 since we can observe that any vertex in T appears in consecutive elements of $\mathcal{X}$. In what follows, we show that $\mathcal{X}$ satisfies (e) in Condition 2.1. If $X_a \cap X_c = \emptyset$ for all a, b, and c with $1 \leq a < b < c \leq r$ then the condition is clearly satisfied. Thus, we assume that there exist a and c with $1 < a + 1 \leq c - 1 < r$ such that $X_a \cap X_c \neq \emptyset$. Since any vertex in $V(T) - \{p_i \mid i \in I, |V(H_i)| \geq 2\}$ appears in at most two consecutive elements of $\mathcal{X}$, there exists p_i such that $i \in I$ and $p_i \in X_a \cap X_c$. Since $(X_a, \ldots, X_c)$ is a subsequence of $(S_{i-1}) + \mathcal{Y}_i + (S_i)$, no vertices in $V(P) - \{p_i\}$ are contained in $X_a \cap X_c$. Moreover, each element of $\mathcal{Y}_i$ has just three elements since $|V(H_i)| \geq 2$. For any b with $a < b < c$, we have that $|X_b| = 3$ since X_b is an element of $\mathcal{Y}_i$. Thus, we have that $|X_a \cap X_c| = |\{p_i\}| = 1 \leq |X_b| - 2$ for any b with $a < b < c$. Therefore, $\mathcal{X}$ satisfies (e) in Condition 2.1. It is clear that $\mathcal{X}$ has width at most 2 and that $p_0 \in X_1 - \bigcup_{1<i\leq r} X_i$ and $p_l \in X_r - \bigcup_{1\leq i<r} X_i$. Therefore, $\mathcal{X}$ is a desired proper-path-decomposition. □

We describe Algorithm `PPD_SPINE` based on Lemma 2.2 in Figure 2.3. The following corollary is immediate.

Corollary 2.4 *Given a binary tree T and a 2-spine $P = (p_0, \ldots, p_l)$ of T,* `PPD_SPINE` *outputs in linear time the proper-path-decomposition $(X_1, \ldots, X_r)$ of T with width at most 2 such that $p_0 \in X_1 - \bigcup_{1<i\leq r} X_i$ and $p_l \in X_r - \bigcup_{1\leq i<r} X_i$.* □

2.3.2 2-Connected Graph

In this subsection, we show a necessary and sufficient condition for a 2-connected graph G with $\Delta(G) \leq 3$ to have $ppw(G) = 2$, and based on this condition, we give an algorithm for computing a 2-proper-path-decomposition of G.

Procedure `PPD_SPINE` (T, P)

Input:	a binary tree T;
	a 2-spine $P = (p_0, \ldots, p_l)$ of T;
Output:	the proper-path-decomposition $(X_1, \ldots, X_r)$ of T with width at most 2 such that $p_0 \in X_1 - \bigcup_{1<i\le r} X_i$ and $p_l \in X_r - \bigcup_{1\le i<r} X_i$;

1. `for` $i := 1$ `to` l `do`

   $S_i := \{p_{i-1}, p_i\}$;

   `endfor` ;

2. `for` $i := 1$ `to` $l-1$ `do`

   `if` $\deg_T(p_i) = 3$ `then`

   (a) `let` H_i be the connected component in $T - V(P)$ which has a vertex adjacent to p_i in T;

   (b) `if` $|V(H_i)| = 1$ `then`
   - $S_i := \{p_{i-1}, p_i\} \cup V(H_i)$;
   - $\mathcal{Y}_i := nul$;

   `else`
   - $\mathcal{Y}_i :=$ `PPD_PATH`$(H_i) \cup \{p_i\}$;

   `else`

   (a) $\mathcal{Y}_i := nul$;

   `endfor` ;

3. $\mathcal{X} := (S_1) + \mathcal{Y}_1 + (S_2) + \cdots + (S_{l-1}) + \mathcal{Y}_{l-1} + (S_l)$;

4. `return` ($\mathcal{X}$);

`End`

Figure 2.3: Algorithm for computing the 2-proper-path-decomposition of a binary tree with its 2-spine.

Lemma 2.5 *For a 2-connected graph G with $\Delta(G) \leq 3$, $ppw(G) = 2$ if and only if G is outer planar and has at most two end-regions.*

Proof First, we assume that $ppw(G) = 2$. Then none of $P(x0x0x0)$, K_4, and $K_{2,3}$ is a minor of G. It is well-known that the family of outer planar graphs is minor-closed and that K_4 and $K_{2,3}$ are the minimal forbidden minors for the family of outer planar graphs. Thus, G is outer planar. Moreover, G has at most two end-regions since $P(x0x0x0)$ is not a minor of G.

Next, we assume that G is outer planar and has at most two end-regions. Let e_s and e_t be any edges in G satisfying the following condition:

Condition 2.2

(a) *e_s and e_t are outer edges contained in distinct end-regions if G has two end-regions.*

(b) *e_s and e_t are a matching of G if $|V(G)| \geq 4$.*

It suffices to show the following claim.

Claim 2.6 *There exists a 2-proper-path-decomposition $\mathcal{X} = (X_1, \ldots, X_r)$ of G such that*

$$e_s \in E(G[X_1]) - E(G[\bigcup_{1<i\leq r} X_i]) \text{ and } \tag{2.1}$$

$$e_t \in E(G[X_r]) - E(G[\bigcup_{1\leq i<r} X_i]). \tag{2.2}$$

We prove this claim by induction on $|V(G)|$.

If $|V(G)| = 3$ then $\mathcal{X} = (V(G))$ is clearly a desired proper-path-decomposition.

Assume that $|V(G)| = 4$. Since G is outer planar, at least one vertex s incident to e_s has degree 2. Let t be the vertex not adjacent to s. Since G is simple, it follows that $\deg_G(t) = 2$. Moreover, t is incident to e_t. Then $\mathcal{X} = (V(G) - \{t\}, V(G) - \{s\})$ is clearly a desired proper-path-decomposition.

We assume that the claim holds for any G' with $|V(G)|-1 \geq 4$ vertices and for any pair of edges in G' satisfying Condition 2.2. Since $|V(G)| \geq 5$, there exists $e \in \{e_s, e_t\}$ which is incident to a vertex s such that $\deg_G(s) = 2$ and that s is not adjacent to an end-vertex of $e' \in \{e_s, e_t\} - \{e\}$. Suppose that $e = (s, y)$ and $N_G(s) - \{y\} = \{x\}$. Let G' be the graph obtained from G by contracting the edge (s, x). We denote by x the resulting vertex. G' is clearly an outer planar graph with at most two end-regions. By the definitions of e and e', (x, y) and e' are distinct edges in G' satisfying Condition 2.2. Therefore, by induction hypothesis, there exists a 2-proper-path-decomposition $\mathcal{Y} = (Y_1, \ldots, Y_l)$ of G' such that

$$\begin{aligned} (x,y) &\in E(G'[Y_1]) - E(G'[\bigcup_{1<i\leq l} Y_i]) \text{ and} & (2.3)\\ e' &\in E(G'[Y_l]) - E(G'[\bigcup_{1\leq i<l} Y_i]). & (2.4) \end{aligned}$$

We show that $\mathcal{X} = (\{s, x, y\}) + \mathcal{Y}$ is a desired 2-proper-path-decomposition of G.

We first show that $\mathcal{X}$ satisfies (2.1) and (2.2). Since

$$s \notin Y_i \ (1 \leq i \leq l), \tag{2.5}$$

we have that

$$e \in E(G[\{s,x,y\}]) - E(G[\bigcup_{1\leq i\leq l} Y_i]). \tag{2.6}$$

It follows from (2.4) and (2.6) that $\mathcal{X}$ satisfies (2.1) and (2.2).

We next show that $\mathcal{X}$ is a 2-proper-path-decomposition of G. $\mathcal{X}$ clearly satisfies (a), (b), and (c) in Condition 2.1. Since $\mathcal{Y}$ is a proper-path-decomposition of G', it follows that

$$\begin{aligned} Y_a \cap Y_c &\subseteq Y_b \ (1 \leq a \leq b \leq c \leq l), & (2.7)\\ |Y_a \cap Y_c| &\leq |Y_b| - 2 \ (1 \leq a < b < c \leq l). \end{aligned}$$

Thus, to show that $\mathcal{X}$ satisfies (d) and (e) in Condition 2.1, it suffices to prove that $\{s, x, y\} \cap Y_c \subseteq Y_b$ and $|\{s, x, y\} \cap Y_c| \leq |Y_b| - 2$ for $1 \leq b < c \leq l$. It follows from (2.3) that

$$\begin{aligned} \{x,y\} &\subseteq Y_1, & (2.8)\\ \{x,y\} &\not\subseteq \bigcup_{1<i\leq l} Y_i. & (2.9) \end{aligned}$$

It follows from (2.5), (2.7), and (2.8) that $\{s, x, y\} \cap Y_c = \{x, y\} \cap Y_c \subseteq Y_1 \cap Y_c \subseteq Y_b$ for $1 \leq b < c \leq l$. It follows from (2.5) and (2.9) that $|\{s, x, y\} \cap Y_c| \leq 1$ for $1 < c \leq l$.

Moreover, we have that $|Y_i| = 3$ for $1 \leq i \leq l$ by the definition of $\mathcal{Y}$. Thus, we have $|\{s, x, y\} \cap Y_c| \leq |Y_b| - 2$ for $1 \leq b < c \leq l$

Therefore, $\mathcal{X}$ is a desired 2-proper-path-decomposition of G, and we conclude that the lemma holds. □

We describe in Figure 2.4 Algorithm `PPD_2CG` based on Lemma 2.5.

Corollary 2.7 *Given an outer planar graph G with at most two end-regions and any edges e_s and e_t in G satisfying Condition 2.2,* `PPD_OPG` *outputs in linear time the 2-proper-path-decomposition $(X_1, \ldots, X_r)$ of G satisfying (2.1) and (2.2).*

Proof The correctness of `PPD_OPG` is immediate from the proof of Lemma 2.5. `PPD_OPG` involves $|V(G)|$ recursive calls each of which consists of constant time operations. Therefore, `PPD_OPG` can be executed in linear time.

□

It is well-known that we can determine if a given graph is outer planar in linear time. Therefore, `PPD_2CG` can be executed in linear time by Corollary 2.7.

2.3.3 General Graph

In this subsection, we show a necessary and sufficient condition for a general graph G with $\Delta(G) \leq 3$ to have $ppw(G) \leq 2$ based on the results described in Subsection 2.3.1 and 2.3.2, and we give an algorithm for computing a 2-proper-path-decomposition of G.

The following lemma will be used extensively throughout this subsection.

Lemma 2.8 *Let $\mathcal{X} = (X_1, \ldots, X_r)$ be a 2-proper-path-decomposition of a graph G with $ppw(G) = 2$. For a path P connecting $s \in X_1$ and $t \in X_r$, every connected component of $G - V(P)$ is a path.*

Proof Suppose that $\mathcal{Y} = (Y_1, \ldots, Y_r)$ is $\mathcal{X} \cap (V(G) - V(P))$. It suffices to show that the sequence $\mathcal{Y}'$ obtained from $\mathcal{Y}$ by deleting redundant elements is a 1-proper-path-decomposition of $G - V(P)$. $\mathcal{Y}$ clearly satisfies (b), (c), and (d) in Condition 2.1 for $G - V(P)$. Thus, $\mathcal{Y}'$ satisfies (a), (b), (c), and (d) in Condition 2.1 for $G - V(P)$. To show that $\mathcal{Y}'$ satisfies (e) in Condition 2.1, it suffices to prove that both of the following statements holds: (i) $|Y_i| \leq 2$ for any $1 \leq i \leq r$; (ii) $Y_a = Y_c$ or $|Y_a \cap Y_c| = 0$ for all a

`Procedure PPD_2CG` (G)

Input:	a 2-connected graph G;
Output:	the 2-proper-path-decomposition $(X_1, \ldots, X_r)$ of G;

1. `if` G is not outer planar `then reject` ;
2. `if` G has more than two end-regions `then reject` ;
3. `let` e_s and e_t be any edges satisfying Condition 2.2;
4. `return` (`PPD_OPG`(G, e_s, e_t));

`End`

`Procedure PPD_OPG` (G, e_s, e_t)

Input:	an outer planar graph G with at most two end-regions;
	edges e_s and e_t satisfying Condition 2.2;
Output:	the 2-proper-path-decomposition $(X_1, \ldots, X_r)$ of G satisfying (2.1) and (2.2);

1. `if` $|V(G)| = 3$ `then return` ($(V(G))$);
2. `if` $|V(G)| = 4$ `then`
   - `let` s be a vertex incident to e_s such that $\deg_G(s) = 2$;
   - `let` t be the vertex not adjacent to s;
   - $\mathcal{X} := (V(G) - \{t\}, V(G) - \{s\})$;
   - `return` ($\mathcal{X}$);
3. `let` $e \in \{e_s, e_t\}$ which is incident to a vertex that s such that $\deg_G(s) = 2$ and s is not adjacent to an end-vertex of $e' \in \{e_s, e_t\} - \{e\}$;
4. `let` y be the vertex joined to s by e and $x \in N_G(s) - \{y\}$;
5. `let` G' be the graph obtained from G by contracting (s, x);
6. `return` ($(\{s, x, y\}) +$ `PPD_OPG`$(G', (x, y), e_t)$);

`End`

Figure 2.4: Algorithm for computing the 2-proper-path-decomposition of a 2-connected graph.

and c with $1 < a+1 \leq c-1 < r$. Every X_i $(1 \leq i \leq r)$ contains a vertex of P since end-vertices s and t of P are contained in X_1 and X_t, respectively, and $\mathcal{X}$ satisfies (c) and (d) in Condition 2.1. Since $\mathcal{X}$ has width 2, we have that $|Y_i| \leq 2$, i.e. (i) holds.

Since $\mathcal{X}$ satisfies (e) in Condition 2.1, we have that

$$|X_a \cap X_c| \leq |X_b| - 2 \leq 3 - 2 = 1 \tag{2.10}$$

for any a, b, and c with $1 \leq a < b < c \leq r$. For a, b, and c with $1 \leq a < b < c \leq r$, let $p_a \in X_a \cap V(P)$, $p_b \in X_b \cap V(P)$, and $p_c \in X_c \cap V(P)$.

Case 1 $p_a = p_c$. It follows from (2.10) that $|X_a \cap X_c| = 1$. Thus, we have $|Y_a \cap Y_c| = 0$.

Case 2 $p_a \neq p_c$. We assume that $|Y_a \cap Y_c| = 1$, and show that $Y_a = Y_c$. Let $v \in Y_a \cap Y_c$. It follows from (d) in Condition 2.1 that $v \in Y_b \subset X_b$. Now we show that $X_b - (V(P) \cup \{v\}) = \emptyset$. We prove this by contradiction. Assume that $X_b - (V(P) \cup \{v\}) \neq \emptyset$. Since $|X_b| \leq 3$, it follows from the assumption that $X_b \cap V(P) = \{p_b\}$. Since P connects $s \in X_1$ and $t \in X_r$, it follows from $1 < b < r$ that $p_b \in X_{b-1} \cap X_{b+1}$. Moreover, since $v \in Y_a \cap Y_c$ and $\mathcal{X}$ satisfies (d) in Condition 2.1, we have that $v \in X_{b-1} \cap X_{b+1}$. Thus, we have that $|X_{b-1} \cap X_{b+1}| \geq |\{p_b, v\}| = 2$, contradicting (2.10). Therefore, it follows that $X_b - (V(P) \cup \{v\}) = \emptyset$. Since this holds for any b with $a < b < c$, we have $Y_a = Y_{a+1} = \cdots = Y_c = \{v\}$.

Therefore, (ii) holds. $\square$

Throughout this subsection, we assume that $\Delta(G) = 3$. Let U be the set of cut vertices of G. We define that $A = U - U'$, where U' is the set of cut vertices contained only in blocks each of which consists of a single edge. A vertex contained in A is called a *connection point* of G. Since a connection point of G is a cut vertex of G, $E(G)$ can be partitioned into disjoint sets $E_1, \ldots, E_m$ such that $G[E_i]$ and $G[E_j]$ share at most one connection point of G for any distinct i and j with $1 \leq i \leq m$ and $1 \leq j \leq m$. Let $\mathcal{D} = \{G[E_i] \mid 1 \leq i \leq m\}$. We define that $\mathcal{H}$ is the set of 2-connected components in $\mathcal{D}$. A component of $\mathcal{D}$ is called a *tree component* of G if the component is a tree with maximum vertex degree 3. $\mathcal{T}$ denotes the set of tree components of G. A component of $\mathcal{D}$ is called a *path component* of G if the component is a path. $\mathcal{P}$ denotes the set of path components of G.

Now we show a necessary and sufficient condition for G to have $ppw(G) \leq 2$.

Theorem 2.9 *For a graph G with $\Delta(G) \leq 3$, $ppw(G) \leq 2$ if and only if G has a sequence $\mathcal{C} = (C_1, C_2, \ldots, C_m)$ of distinct components in $\mathcal{D}$ and a sequence $\mathcal{A} = (a_0, a_1, \ldots, a_m)$ of distinct vertices in $V(G)$ such that the following condition is satisfied. Let $\mathcal{D}' = \mathcal{D} - \{C_i \mid 1 \leq i \leq m\}$.*

Condition 2.3

(a) *$V(C_i) \cap V(C_{i+1}) = \{a_i\}$ for $1 \leq i < m$, $a_0 \in V(C_1)$, and $a_m \in V(C_m)$.*

(b) *$\deg_G(a_0) \leq 2$ and $\deg_G(a_m) \leq 2$.*

(c) *For $1 \leq i \leq m$, if $C_i \in \mathcal{H}$, then C_i is an outer planar graph with at most two end-regions. Moreover, each end-region contains a_{i-1} or a_i.*

(d) *For $1 \leq i \leq m$, if $C_i \in \mathcal{T}$, then the path in C_i connecting a_{i-1} and a_i is a 2-spine of C_i.*

(e) *$\mathcal{D}' \subseteq \mathcal{P}$*

(f) *There exists a one-to-one mapping $f : \mathcal{D}' \rightarrow \{i \mid 1 \leq i \leq m\} \times \{0, 1\}$ satisfying the following statement.*

> *For $P \in \mathcal{D}'$, $f(P) = (i, j)$ if and only if $C_i \in \mathcal{H}$ and there exists x such that x is an end vertex of P and that $(x, a_{i-j}) \in E(C_i)$.* (*)

Proof of Necessity for Theorem 2.9

We first show the necessity. Condition 2.3 is trivially satisfied if $ppw(G) = 1$ or $|V(G)| \leq 3$. Moreover, if $|\mathcal{D}| = 1$ then there exist a_0 and a_1 satisfying Condition 2.3 for $\mathcal{C} = (D)$ ($D \in \mathcal{D}$) by Lemmas A and 2.5. Therefore, we assume that $ppw(G) = 2$, $|V(G)| \geq 4$, and $|\mathcal{D}| \geq 2$. There exists a 2-proper-path-decomposition $\mathcal{X} = (X_1, \ldots, X_r)$ of G. Since $\mathcal{X}$ satisfies (a) in Condition 2.1 and $|V(G)| \geq 4$, there exist $s \in X_1 - X_2$ and $t \in X_r - X_{r-1}$. It should be noted that $\mathcal{H} \neq \emptyset$ from the assumption that $|\mathcal{D}| \geq 2$.

Claim 2.10 *$\deg_G(s) \leq 2$ and $\deg_G(t) \leq 2$.*

Proof $|X_1| \leq 3$ and $|X_r| \leq 3$ since $\mathcal{X}$ has width 2. Thus, we have $\deg_G(s) \leq 2$ and $\deg_G(t) \leq 2$ since $\mathcal{X}$ satisfies (c) in Condition 2.1. □

We define that S is a path connecting s and t.

Claim 2.11 *For $H \in \mathcal{H}$, $H \cap S$ is a path with at least two vertices.*

Proof It suffices to show that $|V(H) \cap V(S)| \geq 2$ and that $H \cap S$ is connected.

We first show that $|V(H) \cap V(S)| \geq 2$. We prove this by contradiction. If $V(H) \cap V(S) = \emptyset$, then $G - V(S)$ has a cycle since H has a cycle. However, this contradicts Lemma 2.8. We next assume that $V(H) \cap V(S) = \{x\}$. If $x \in V(S) - \{s, t\}$ then we have $\deg_G(x) = \deg_H(x) + \deg_S(x) \geq 2 + 2 = 4$. However, this is a contradiction since $\Delta(G) = 3$. If $x \in \{s, t\}$ then we have $\deg_G(x) = \deg_H(x) + \deg_S(x) \geq 2 + 1 = 3$. However, this also contradicts Claim 2.10. Thus, we have that $|V(H) \cap V(S)| \geq 2$.

We next show that $H \cap S$ is connected. We again prove this by contradiction. Suppose that $H \cap S$ has disjoint connected components P_1 and P_2. Since P_1 and P_2 are vertex-disjoint subgraphs of S and S is connected, there exists a path P_3 connecting the end-vertex of P_1 and that of P_2 in $S - (E(P_1) \cup E(P_2))$. $H \cup P_3$ is clearly a 2-connected component of G. Since P_1 and P_2 are disjoint connected components of $H \cap S$, P_3 has an edge $e \notin E(H)$. This means that H is a proper subgraph of $H \cup P_3$. However, this contradicts the fact that H is a block of G. Therefore, $H \cap S$ is connected. □

Suppose that $\mathcal{H} = \{H_1, \ldots, H_l\}$ and that s_i and t_i are end-vertices of the path $H_i \cap S$ for $1 \leq i \leq l$. Since $\Delta(G) = 3$, H_i and H_j are vertex-disjoint for any distinct $H_i, H_j \in \mathcal{H}$ $(1 \leq i < j \leq l)$. Thus, $H_i \cap S$ and $H_j \cap S$ are also vertex-disjoint. We may assume without loss of generality that $d(s_1) < d(t_1) < d(s_2) < d(t_2) < \cdots < d(s_l) < d(t_l)$, where $d(v)$ is the number of edges of the subpath of S connecting s and $v \in V(S)$. We define that R_i is the subpath of S connecting t_i and s_{i+1} $(1 \leq i < l)$. Since $E(H_i) \cap E(R_j) = \emptyset$ for any i and j with $1 \leq i \leq l$ and $1 \leq j < l$, there exists a component $K_i \in \mathcal{T} \cup \mathcal{P}$ containing R_i as a subgraph for $1 \leq i < l$. Similarly, there exists a component $K_0 \in \mathcal{T} \cup \mathcal{P}$ containing the subpath R_0 of S connecting s and s_1 if $s \neq s_1$. Moreover, there exists a component $K_l \in \mathcal{T} \cup \mathcal{P}$ containing the subpath R_l of S connecting t_l and t if $t_l \neq t$.

We define the sequence $\mathcal{C}$ of components in $\mathcal{D}$ and the sequence $\mathcal{A}$ of vertices of G as follows:

$$\begin{aligned}
\mathcal{C} &= \begin{cases} \mathcal{C}' & \text{if } s_1 = s \text{ and } t_l = t \\ (K_0) + \mathcal{C}' & \text{if } s_1 \neq s \text{ and } t_l = t \\ \mathcal{C}' + (K_l) & \text{if } s_1 = s \text{ and } t_l \neq t \\ (K_0) + \mathcal{C}' + (K_l) & \text{if } s_1 \neq s \text{ and } t_l \neq t \end{cases}, \text{ where} \\
\mathcal{C}' &= (H_1, K_1, H_2, K_2, \ldots, K_{l-1}, H_l). \\
\mathcal{A} &= \begin{cases} \mathcal{A}' & \text{if } s_1 = s \text{ and } t_l = t \\ (s) + \mathcal{A}' & \text{if } s_1 \neq s \text{ and } t_l = t \\ \mathcal{A}' + (t) & \text{if } s_1 = s \text{ and } t_l \neq t \\ (s) + \mathcal{A}' + (t) & \text{if } s_1 \neq s \text{ and } t_l \neq t \end{cases}, \text{ where} \\
\mathcal{A}' &= (s_1, t_1, s_2, t_2, \ldots, s_l, t_l).
\end{aligned}$$

Suppose that $\mathcal{C} = (C_1, \ldots, C_m)$ and $\mathcal{A} = (a_0, \ldots, a_m)$. We show that $\mathcal{C}$ and $\mathcal{A}$ satisfies Condition 2.3.

$\mathcal{C}$ and $\mathcal{A}$ clearly satisfies (a) in Condition 2.3 by definition. Moreover, (b) in Condition 2.3 is satisfied from Claim 2.10. The following claim is used to show that $\mathcal{C}$ and $\mathcal{A}$ satisfies (c) in Condition 2.3.

Claim 2.12 *For a 2-connected component $H \in \mathcal{H}$ with two end-regions, each end-region contains an end-vertex of $H \cap S$.*

Proof It follows from Claim 2.11 that there exist distinct end-vertices u and v of $H \cap S$. We assume without loss of generality that $d(u) < d(v)$. Let P_s be the subpath of S connecting s and u, and let P_t be the subpath of S connecting v and t. If H has an end-region Z which contains neither u nor v, then there exists a path $\overline{P}$ in H which connects u and v and contains no vertices in Z. Since P_s, P_t, and $\overline{P}$ are internally vertex-disjoint, $S' = P_s \cup \overline{P} \cup P_t$ is a path connecting s and t. Since S' and Z are vertex-disjoint, $G - V(S')$ contains a cycle as a subgraph. However, this contradicts Lemma 2.8 and the assumption that $ppw(G) = 2$. Thus, each end-region contains an end-vertex of $H \cap S$. □

We see in the following claim that $\mathcal{C}$ and $\mathcal{A}$ satisfies (c) in Condition 2.3.

Claim 2.13 *If $C_i \in \mathcal{H}$ $(1 \leq i \leq m)$, then C_i is an outer planar graph with at most two end-regions. Moreover, each end-region contains a_{i-1} or a_i.*

Proof Suppose that $C_i \in \mathcal{H}$ $(1 \leq i \leq m)$. Since $ppw(G) = 2$, we have that $ppw(C_i) = 2$. Thus, it follows from Lemma 2.5 that C_i is an outer planar graph with at most two end-regions. Moreover, it follows from Claim 2.12 that each end-region contains a_{i-1} or a_i. □

We see in the following claim that $\mathcal{C}$ and $\mathcal{A}$ satisfies (d) in Condition 2.3.

Claim 2.14 *If $C_i \in \mathcal{T}$ $(1 \leq i \leq m)$, then the path in C_i connecting a_{i-1} and a_i is a 2-spine of C_i.*

Proof Let S_i be the path in C_i connecting a_{i-1} and a_i. Every connected component of $G - V(S)$ is a path by Lemma 2.8. Since S_i is a subpath of S, every connected component of $C_i - V(S_i)$ is a path. This means that S_i is a 2-spine of C_i. □

We see in the following claim that $\mathcal{C}$ and $\mathcal{A}$ satisfies (e) in Condition 2.3.

Claim 2.15 $\mathcal{D}' \subseteq \mathcal{P}$.

Proof Every 2-connected component of G is an element of $\mathcal{C}$ by Claim 2.11 and the definition of $\mathcal{C}$. Thus, it suffices to show that every tree component of G is an element of $\mathcal{C}$. We prove this by contradiction. Assume that $T \in \mathcal{T} - \{C_i \mid 1 \leq i \leq m\}$. It follows from the assumption that $|\mathcal{D}| \geq 2$ that, for $c \in V(T) \cap A$, there exists $H \in \mathcal{H}$ such that $V(H) \cap V(T) = \{c\}$. It follows from Claim 2.11 that $H \in \{C_i \mid 1 \leq i \leq m\}$. Suppose that $H = C_i$ $(1 \leq i \leq m)$. Since $T \notin \{C_i \mid 1 \leq i \leq m\}$, it follows from Claim 2.10 and the assumption that $\Delta(G) = 3$ that c is not an element of $\mathcal{A}$. Since C_i is 2-connected, there exists a path $\overline{P}$ in C_i which connects a_{i-1} and a_i and does not contain c. Let P_s be the subpath of S connecting s and a_{i-1}, and let P_t be the subpath of S connecting a_i and t. Since P_s, P_t, and $\overline{P}$ are internally vertex-disjoint, $S' = P_s \cup \overline{P} \cup P_t$ is a path connecting s and t. Since S' and T are vertex-disjoint and T has a vertex with degree 3, $G - V(S')$ has a vertex with degree 3. However, this contradicts Lemma 2.8. □

We prove by a sequence of claims that $\mathcal{C}$ and $\mathcal{A}$ satisfies (f) in Condition 2.3. Let $c(P)$ be a unique element of $V(P) \cap A$ for $P \in \mathcal{D}'$.

Claim 2.16 *For $P \in \mathcal{D}'$, there exists a unique $C_i \in \mathcal{H}$ $(1 \leq i \leq m)$ such that $V(C_i) \cap V(P) = \{c(P)\}$. Moreover, $(c(P), a_{i-1}) \in E(C_i)$ or $(c(P), a_i) \in E(C_i)$.*

Proof It is clear by $\Delta(G) \le 3$ and by the definition of path components that, for $P \in \mathcal{D}'$, there exists a unique $C_i \in \mathcal{H}$ $(1 \le i \le m)$ such that $V(C_i) \cap V(P) = \{c(P)\}$. We show that $(c(P), a_{i-1}) \in E(C_i)$ or $(c(P), a_i) \in E(C_i)$. We prove this by contradiction. Assume that $(c(P), a_{i-1}) \notin E(C_i)$ and $(c(P), a_i) \notin E(C_i)$. If $a_{i-1} \neq s$ then $\deg_{C_i \cup C_{i-1}}(a_{i-1}) = 3$. Thus, we have $c(P) \neq a_{i-1}$ since $\Delta(G) = 3$. If $a_{i-1} = s$ then $\deg_{C_i}(a_{i-1}) = 2$. Thus, we have $c(P) \neq a_{i-1}$ by Claim 2.10. Therefore, it follows that $c(P) \neq a_{i-1}$. We can show by a similar argument that $c(P) \neq a_i$. Thus, neither a_{i-1} nor a_i is contained in $N_G(c(P)) \cup \{c(P)\}$. Since C_i is outer planar and $\Delta(G) \le 3$, $c(P)$ incident to just two outer edges of C_i and to exactly one edge of P. Therefore, there exists a path $\overline{P}$ in C_i which connects a_{i-1} and a_i and does not contain a vertex in $N_G(c(P)) \cup \{c(P)\}$. Let P_s be the subpath of S connecting s and a_{i-1}, and let P_t be the subpath of S connecting a_i and t. Since P_s, P_t, and $\overline{P}$ are internally vertex-disjoint, $S' = P_s \cup \overline{P} \cup P_t$ is a path connecting s and t. Since $V(S')$ and $N_G(c(P)) \cup \{c(P)\}$ are disjoint, $G - V(S')$ has $c(P)$ with degree 3. However, this contradicts Lemma 2.8. $\square$

Claim 2.17 *For distinct $P_1, P_2 \in \mathcal{D}'$, $c(P_1) \neq c(P_2)$.*

Proof Each $c(P_i)$ $(i = 1, 2)$ is contained in a 2-connected component of G by Claim 2.16. If $c(P_1) = c(P_2)$ then $\deg_G(c(P_i)) \ge 4$ $(i = 1, 2)$, contradicting the assumption that $\Delta(G) = 3$. $\square$

Claim 2.18 *If C_1 is 2-connected, then $|\{P \in \mathcal{D}' \mid c(P) \in N_G(a_0)\}| \le 1$.*

Proof Since $s = a_0$, it follows from Claim 2.10 that $|N_G(a_0)| = 2$. Suppose that $N_G(a_0) = \{u, v\}$. We prove the claim by contradiction. Assume that there exist distinct $P_1, P_2 \in \mathcal{D}'$ such that $\{c(P_1), c(P_2)\} \subseteq N_G(a_0)$. By Claim 2.17, we may assume without loss of generality that $c(P_1) = u$, $c(P_2) = v$. It follows from Claim 2.16 that $a_1 \notin \{u, v\}$. If (u, v) is an outer edge of C_1, then $V(C_1) = \{a_0, u, v\}$ and a_1 is either u or v, a contradiction. If (u, v) is an inner edge of C_1, then $\deg_{C_1}(u) \ge 3$ since each vertex in a 2-connected component is incident to two outer edges. Thus, $\deg_G(u) = \deg_{C_1}(u) + \deg_{P_1}(u) \ge 3 + 1 = 4$, contradicting the assumption that $\Delta(G) = 3$. Therefore, u and v are not adjacent.

Since $s \in X_1 - X_2$ and $\mathcal{X}$ satisfies (c) in Condition 2.1, we have that $X_1 = \{s, u, v\}$. Suppose that $\Gamma_G(u) = \{(s, u), e_1, e_2\}$ and $\Gamma_G(v) = \{(s, v), e_3, e_4\}$. It should be noted that since u and v are not adjacent, e_1, e_2, e_3, and e_4 are distinct edges. Let $j(i) = \max\{j \mid X_j \text{ contains both end-vertices of } e_i\}$ for $1 \leq i \leq 4$. Moreover, let γ be i which maximize $j(i)$. We assume without loss of generality that $\gamma \in \{3, 4\}$. If $j(\gamma) = j(1) = j(2)$, then $G[X_{j(\gamma)}]$ is isomorphic to K_3 since $|X_{j(\gamma)}| \leq 3$ and G is simple. However, this is a contradiction because u and v are not adjacent. Thus, it follows that $j(\gamma) > j(1)$ or $j(\gamma) > j(2)$. We assume without loss of generality that $j(1) \leq j(2) \leq j(\gamma)$. Since both end-vertices of e_1 are not contained in X_1, we have that $j(1) \geq 2$. Since $\{u, v\} \in X_1$, $u \in X_{j(2)}$, $v \in X_{j(\gamma)}$, and $\mathcal{X}$ satisfies (d) in Condition 2.1, it follows that $\{u, v\} \subseteq X_1 \cap X_{j(1)} \cap X_{j(2)}$.

Case 1 $j(1) < j(2)$. Since $|X_{j(1)}| \leq 3$, we have $|X_{j(1)}| - |X_1 \cap X_{j(2)}| \leq 3 - 2 = 1$, contradicting that $\mathcal{X}$ satisfies (e) in Condition 2.1.

Case 2 $j(1) = j(2)$. Suppose that $e_1 = (u, x_1)$ and $e_2 = (u, x_2)$. In this case, it follows that $\{u, v, x_1, x_2\} \subseteq X_{j(1)}$. u, v, x_1, and x_2 are distinct since u and v are not adjacent and G is simple. Thus, we have that $|X_{j(1)}| \geq 4$, contradicting that $\mathcal{X}$ has width 2.

Therefore, we conclude that $|\{P \in \mathcal{D}' \mid c(P) \in N_G(a_0)\}| \leq 1$. □

Claim 2.19 *Suppose that $C_1 \in \mathcal{H}$. If there exist distinct $P_1, P_2 \in \mathcal{D}'$ such that both $c(P_1)$ and $c(P_2)$ are adjacent to a_1, then $c(P_1)$ or $c(P_2)$ is adjacent to a_0.*

Proof We show the claim by contradiction. Assume that there exist distinct $P_1, P_2 \in \mathcal{D}'$ such that both $c(P_1)$ and $c(P_2)$ are adjacent to a_1 and that neither $c(P_1)$ nor $c(P_2)$ is adjacent to a_0. Let L be the subgraph of G induced by all the outer edges of C_1. Since $s = a_0$ and $C_1 \in \mathcal{H}$, it follows from Claim 2.10 that $|N_G(a_0)| = 2$. Suppose that $N_G(a_0) = \{u, v\}$. It follows from the assumption and Claims 2.16 and 2.17 that a_0, a_1, u, v, $c(P_1)$, and $c(P_2)$ are distinct vertices.

If $(u, v) \in E(G)$, then $P(x01010)$ shown in Figure 2.1 is a minor of the subgraph $L \cup P_1 \cup P_2 \cup G[\{u, v\}]$ of G. This means that $(u, v) \notin E(G)$ and that the graph G' obtained from G by joining u and v by an additional edge has proper-path-width more than 2.

On the other hand, since $a_0 = s \in X_1 - X_2$ and $\mathcal{X}$ satisfies (c) in Condition 2.1, we have that $X_1 = \{a_0, u, v\}$. Therefore, $\mathcal{X}$ is a 2-proper-path-decomposition of G', i.e. $ppw(G') = 2$, a contradiction. □

The proofs of the following Claims 2.20 and 2.21 can be accomplished by similar arguments for Claims 2.18 and 2.19, and is omitted.

Claim 2.20 *If C_m is 2-connected, then $|\{P \in \mathcal{D}' \mid c(P) \in N_G(a_m)\}| \leq 1$.* □

Claim 2.21 *Suppose that $C_m \in \mathcal{H}$. If there exist distinct $P_1, P_2 \in \mathcal{D}'$ such that both $c(P_1)$ and $c(P_2)$ are adjacent to a_{m-1}, then $c(P_1)$ or $c(P_2)$ is adjacent to a_m.* □

Claim 2.22 *Suppose that $C_i \in \mathcal{H}$ $(1 < i < m)$. If there exists distinct $P_1, P_2 \in \mathcal{D}'$ such that both $c(P_1)$ and $c(P_2)$ are adjacent to $a \in \{a_{i-1}, a_i\}$, $c(P_1)$ or $c(P_2)$ adjacent to $a' \in \{a_{i-1}, a_i\} - \{a\}$.*

Proof We show the claim by contradiction. Assume that there exist distinct $P_1, P_2 \in \mathcal{D}'$ such that both $c(P_1)$ and $c(P_2)$ are adjacent to $a \in \{a_{i-1}, a_i\}$ and that neither $c(P_1)$ nor $c(P_2)$ is adjacent to $a' \in \{a_{i-1}, a_i\} - \{a\}$. Let L be the subgraph of G induced by all the outer edges of C_i. Since $1 < i < m$ and $C_i \in \mathcal{H}$, it follows from Claim 2.10 that $|N_G(a')| = 2$. Suppose that $N_G(a') = \{u, v\}$. It follows from the assumption and Claims 2.16 and 2.17 that a, a', u, v, $c(P_1)$, and $c(P_2)$ are distinct vertices. Moreover, there exists $b \in V(G) - V(C_i)$ adjacent to a' since $1 < i < m$. Therefore, $P(101010)$ shown in Figure 2.1 is a minor of the subgraph $L \cup P_1 \cup P_2 \cup G[\{a', b\}]$ of G, contradicting the assumption that $ppw(G) \leq 2$. □

Claim 2.23 *$\mathcal{C}$ and $\mathcal{A}$ satisfies (f) in Condition 2.3.*

Proof It follows from Claim 2.16 that there exists a mapping f satisfying $(*)$. Moreover, it follows from Claims 2.17 through 2.22 that there exists one-to-one mapping f satisfying $(*)$. □

Thus, $\mathcal{C}$ and $\mathcal{A}$ satisfies Condition 2.3. Therefore, the proof of necessity for Theorem 2.9 is completed.

Proof of Sufficiency for Theorem 2.9

We next show the sufficiency. Assume that G has a sequence $\mathcal{C} = (C_1, C_2, \ldots, C_m)$ of components in $\mathcal{D}$ and a sequence $\mathcal{A} = (a_0, a_1, \ldots, a_m)$ of vertices in $V(G)$ such that Condition 2.3 is satisfied. If $C_1 \in \mathcal{T}$ and $\deg_G(a_0) = 2$ then we can easily find a vertex $a'_0 \in V(C_1)$ such that $\deg_G(a'_0) = 1$ and that the path connecting a'_0 and a_1 is a 2-spine of C_1. Moreover, $\mathcal{C}$ and the sequence $(a'_0, a_1, \ldots, a_m)$ satisfies Condition 2.3. Thus, we assume without loss of generality that, if $C_1 \in \mathcal{T}$, then $\deg_G(a_0) = 1$. Similarly, we assume without loss of generality that, if $C_m \in \mathcal{T}$, then $\deg_G(a_m) = 1$.

For $C_i \in \mathcal{H}$ $(1 \le i \le m)$, we define that l_i and r_i are distinct outer edges in C_i incident to a_{i-1} and a_i, respectively, such that:

(i) $l_i = (a_{i-1}, c(P))$ if there exists $P \in \mathcal{D}'$ such that $f(P) = (i, 1)$;

(ii) $r_i = (a_i, c(P))$ if there exists $P \in \mathcal{D}'$ such that $f(P) = (i, 0)$;

(iii) l_i and r_i are contained in distinct end-regions if C_i has two end-regions;

(iv) l_i and r_i are a matching of C_i if $|V(C_i)| \ge 4$.

Since $\mathcal{C}$ and $\mathcal{A}$ satisfies (f) in Condition 2.3, for every $C_i \in \mathcal{H}$, l_i and r_i satisfies Condition 2.2.

We show that the sequence $\mathcal{X} = (X_1, \ldots, X_r)$ of subsets of $V(G)$ defined as follows is a 2-proper-path-decomposition of G.

$$\mathcal{X} = \sum_{1 \le i \le m} \mathcal{L}^i + \mathcal{Y}^i + \mathcal{R}^i, \text{ where}$$

for $1 \le i \le m$,

$$\mathcal{Y}^i = \begin{cases} \texttt{PPD\_SPINE}(C_i, \text{path connecting } a_{i-1} \text{ and } a_i) & \text{if } C_i \in \mathcal{T} \cup \mathcal{P} \\ \texttt{PPD\_OPG}(C_i, l_i, r_i) & \text{if } C_i \in \mathcal{H} \end{cases}$$

$$\mathcal{L}^i = \begin{cases} \texttt{PPD\_PATH}(P = (p_0, \ldots, c(P))) \cup \{a_{i-1}\} & \text{if } \exists P \in \mathcal{D}' \text{ such that } f(P) = (i, 1) \\ nul & \text{otherwise} \end{cases}$$

$$\mathcal{R}^i = \begin{cases} \texttt{PPD\_PATH}(P = (c(P), \ldots, p_l))) \cup \{a_i\} & \text{if } \exists P \in \mathcal{D}' \text{ such that } f(P) = (i, 0) \\ nul & \text{otherwise} \end{cases}$$

Since G satisfies Condition 2.3, $\mathcal{X}$ satisfies (a), (b), and (c) in Condition 2.1 by definition. Moreover, every element of $\mathcal{X}$ contains at most three vertices of G. Thus, it suffices to show that $\mathcal{X}$ satisfies (d) and (e) in Condition 2.1. By the definition of `PPD_PATH` and Corollary 2.4 and 2.7, we can observe the following claim.

Claim 2.24

(i) *For* $1 \leq i \leq m$*,* $v \in V(C_i) - (\{a_{i-1}, a_i\} \cup \{c(P) \mid P \in \mathcal{D}'\})$ *appears in consecutive elements of* $\mathcal{Y}^i$*.*

(ii) *For* $P \in \mathcal{D}'$*,* $v \in V(P) - \{c(P)\}$ *appears in at most two consecutive elements of* $\mathcal{X}$*.*

(iii) *For* $0 \leq i \leq m$*,* a_i *is contained in every element of* $(\textit{the tail of } \mathcal{Y}^i) + \mathcal{R}^i + \mathcal{L}^{i+1} + (\textit{the head of } \mathcal{Y}^{i+1})$*, where* $\mathcal{Y}^0 = \mathcal{R}^0 = \mathcal{Y}^{m+1} = \mathcal{L}^{m+1} = nul$*.*

(iv) *For* $P \in \mathcal{D}'$ *with* $f(P) = (i, 1)$*,* $c(P)$ *appears in the tail of* $\mathcal{L}^i$ *and in consecutive elements of* $\mathcal{Y}^i$ *including its head.*

(v) *For* $P \in \mathcal{D}'$ *with* $f(P) = (i, 0)$*,* $c(P)$ *appears in the head of* $\mathcal{R}^i$ *and in consecutive elements of* $\mathcal{Y}^i$ *including its tail.*

□

It follows from Claim 2.24 that every vertex in G appears in consecutive elements of $\mathcal{X}$. Thus, $\mathcal{X}$ satisfies (d) in Condition 2.1.

It remains to show that $\mathcal{X}$ satisfies (e) in Condition 2.1. If $X_a \cap X_c = \emptyset$ for all a and c with $1 < a+1 \leq c-1 < r$, then this is immediate. Thus, we assume that there exist a and c with $1 < a+1 \leq c-1 < r$ such that $X_a \cap X_c \neq \emptyset$. For $1 \leq i \leq m$, $\mathcal{Y}^i$ is a proper-path-decomposition of C_i. Thus, we have that $|X_a \cap X_c| \leq |X_b| - 2$ for any b with $a < b < c$ if there exists i with $1 \leq i \leq m$ such that both X_a and X_c are elements of $\mathcal{Y}^i$. Therefore, we assume that there exists no i with $1 \leq i \leq m$ such that both X_a and X_c are elements of $\mathcal{Y}^i$. It follows by Claim 2.24 that $X_a \cap X_c \subseteq \{a_i \mid 0 \leq i \leq m\} \cup \{c(P) \mid P \in \mathcal{D}'\}$. We see the following two claims.

Claim 2.25 $|X_a \cap X_c| = 1$.

Proof By Claim 2.24, $X_a \cap X_c$ contains at most one vertex in $\mathcal{A}$ and at most one vertex in $\{c(P) \mid P \in \mathcal{D}'\}$. Thus, it suffices to show that both $a_i (0 \leq i \leq m)$ and $c(P)$ are not contained in $X_a \cap X_c$. We prove this by contradiction. Assume that there exist i $(0 \leq i \leq m)$ and $P \in \mathcal{D}'$ such that $\{a_i, c(P)\} \subseteq X_a \cap X_c$. By Claim 2.24, $f(P) = (i, 0)$ or $f(P) = (i+1, 1)$. We may assume without loss of generality that $f(P) = (i, 0)$. Then, both X_a and X_c are elements of $\mathcal{Y}^i +$ (the head of $\mathcal{R}^i$). Suppose that $\mathcal{Y}^i = (Y_1^i, \ldots, Y_r^i)$. Since both X_a and X_c are not elements of $\mathcal{Y}^i$, X_c is the head of $\mathcal{R}^i$. Moreover, X_a is not the tail of $\mathcal{Y}^i$ since $c - a \geq 2$. This means that both a_i and $c(P)$ are contained in an element of $\mathcal{Y}^i$ except the tail. However, this is impossible since $(a_i, c(P)) = r_i$ and $r_i \in E(G[Y_r^i]) - E(G[\bigcup_{1 \leq j < r} Y_j^i])$ by Corollary 2.7. □

Claim 2.26 $|X_b| = 3$ *for any* b *with* $a < b < c$.

Proof Let b be any integer such that $a < b < c$. If there exists i $(1 \leq i \leq m)$ such that X_b is an element of $\mathcal{Y}^i$ and that $C_i \in \mathcal{H}$, then $|X_b| = 3$ by the definition of `PPD_OPG`. If there exists i $(1 \leq i \leq m)$ such that X_b is an element of $\mathcal{L}^i$ or $\mathcal{R}^i$, then $|X_b| = 3$ by the definition of `PPD_PATH` and by the fact that $|V(P)| \geq 2$ for any $P \in \mathcal{D}'$. Thus, it suffices to show that X_b is not an element of $\mathcal{Y}^i$ such that $C_i \in \mathcal{T} \cup \mathcal{P}$. We prove this by contradiction. Assume that X_b is an element of $\mathcal{Y}^i$ $(1 \leq i \leq m)$ such that $C_i \in \mathcal{T} \cup \mathcal{P}$. It follows from the assumption and Claim 2.25 that either $X_a \cap X_c = \{a_{i-1}\}$ or $X_a \cap X_c = \{a_i\}$. We assume without loss of generality that $X_a \cap X_c = \{a_i\}$. Since X_b is an element of $\mathcal{Y}^i$, it follows that X_a is an element of $\mathcal{Y}^i$ except the tail. This means that a_i is contained in an element of $\mathcal{Y}^i$ except the tail. However, this is impossible since a_i is an end-vertex of 2-spine of C_i and a_i appears only in the tail of $\mathcal{Y}^i$ by Corollary 2.4. □

It follows from Claims 2.25 and 2.26 that $|X_a \cap X_c| - |X_b| = 3 - 2 = 1$ for $a < b < c$. Thus, $\mathcal{X}$ satisfies (e) in Condition 2.1.

Therefore, $\mathcal{X}$ is a 2-proper-path-decomposition of G and the proof of sufficiency for Theorem 2.9 is completed.

We describe in Figure 2.5 Algorithm `PPD_GENERAL` based on Theorem 2.9. It is well-known that we can find all blocks of a graph in linear time. Thus, step 2 can be executed in linear time. To find a_0 and a_m in step 3, we need the algorithm to find a 2-spine of a binary tree, which has not been described yet. Although the details are not mentioned here, it should be noted that this can be done in linear time by using a simple postorder searching and the algorithm in [25], which outputs, for a rooted binary tree, the proper-path-width of every subtree rooted at a vertex. The other operations in `PPD_GENERAL` clearly executed in linear time.

Procedure PPD_GENERAL (G)

$\left[\begin{array}{ll} \text{Input:} & \text{a connected graph } G \text{ with } \Delta(G) \leq 3; \\ \text{Output:} & \text{the 2-proper-path-decomposition of } G; \end{array}\right]$

1. **if** $\Delta(G) \leq 2$ **then return** (PPD_PATH(G));
2. **let** $\mathcal{H}$ be the set of 2-connected components of G;
   let $\mathcal{T}$ be the set of tree components of G;
   let $\mathcal{P}$ be the set of path components of G;
   let $\mathcal{D} = \mathcal{H} \cup \mathcal{T} \cup \mathcal{P}$;
3. find a sequence $\mathcal{C} = (C_1, C_2, \ldots, C_m)$ of components in $\mathcal{D}$ and a sequence $\mathcal{A} = (a_0, a_1, \ldots, a_m)$ of vertices in $V(G)$ such that Condition 2.3 and the conditions in the following are satisfied;
   - $\deg_G(a_0) = 1$ if $C_1 \in \mathcal{T}$;
   - $\deg_G(a_m) = 1$ if $C_m \in \mathcal{T}$;
4. **if** $\mathcal{C}$ and $\mathcal{A}$ do not exist **then reject** ;
5. $\mathcal{D}' := \mathcal{D} - \bigcup_{1 \leq i \leq m} C_i$;
6. **for each** $C_i \in H$ **do**
   (a) find distinct outer edges l_i and r_i in C_i incident to a_{i-1} and a_i, respectively, such that:
      i. $l_i = (a_{i-1}, c(P))$ if $\exists P \in \mathcal{D}'$ such that $f(P) = (i, 1)$;
      ii. $r_i = (a_i, c(P))$ if $\exists P \in \mathcal{D}'$ such that $f(P) = (i, 0)$;
      iii. l_i and r_i are contained in distinct end-regions if C_i has two end-regions;
      iv. l_i and r_i are a matching of C_i if $|C_i| \geq 4$.

   endfor
7. **for** $i = 1$ to m **do**
   (a) **if** $C_i \in \mathcal{T} \cup \mathcal{P}$ **then** $\mathcal{Y}^i :=$ PPD_SPINE(C_i, path connecting a_{i-1} and a_i);
   else $\mathcal{Y}^i :=$ PPD_OPG(C_i, l_i, r_i);
   (b) **if** $\exists P \in \mathcal{D}'$ such that $f(P) = (i, 1)$ **then** $\mathcal{L}^i :=$ PPD_PATH($P = (p_0, \ldots, c(P))$) $\cup \{a_{i-1}\}$;
   else $\mathcal{L}^i := nul$;
   (c) **if** $\exists P \in \mathcal{D}'$ such that $f(P) = (i, 0)$ **then** $\mathcal{R}^i :=$ PPD_PATH($P = (c(P), \ldots, p_l)$) $\cup \{a_i\}$;
   else $\mathcal{R}^i := nul$;

   endfor
8. **return** ($\sum_{1 \leq i \leq m} \mathcal{L}^i + \mathcal{Y}^i + \mathcal{R}^i$);

End

Figure 2.5: Algorithm for computing the 2-proper-path-decomposition of a general graph graph.

Chapter 3

Embedding into Grids

3.1 Introduction

The problem of efficiently implementing parallel algorithms on parallel machines and the problem of efficiently laying out VLSI systems onto VLSI chips have been studied as the graph embedding problem, which is to embed a guest graph within a host graph with certain constraints and/or optimization criteria. For the former problem, guest graphs and host graphs represent parallel algorithms and parallel machines, respectively, and the purpose is to minimize communication overhead, such as dilation and/or congestion of the embedding. For the latter problem, a guest graph represents connection requirements of a system and a host graph usually represents a rectangular grid modeling wafer. In VLSI layout, there are various criteria such as wire length, wire congestion, crossing number, and the layout area.

We consider minimal congestion embeddings of graphs into grids. The grids are well-known not only as a model of VLSI chips but also as one of the most popular processor interconnection graphs for parallel machines. It is well-known that the minimal congestion embedding is very important for a grid-connected parallel machine that uses circuit switching for node-to-node communication. In VLSI layout, the minimal congestion embeddings are crucial in the sense that the congestion is a lower bound for the number of layers.

Let G be a graph and let $V(G)$ and $E(G)$ denote the vertex set and edge set of G, respectively. We denote by $\Delta(G)$ the maximum degree of a vertex in G. An *embedding* $\langle \phi, \rho \rangle$ of a graph G into a graph H is defined by a one-to-one mapping $\phi : V(G) \rightarrow V(H)$, together with a mapping ρ that maps each edge $(u, v) \in E(G)$ onto a path $\rho(u, v)$ in H that connects $\phi(u)$ and $\phi(v)$. The *congestion* of an edge $e' \in E(H)$ under $\langle \phi, \rho \rangle$ is the number of edges e in G such that $\rho(e)$ contains e'. The congestion of an embedding $\langle \phi, \rho \rangle$ is the maximum congestion of an edge in H. The *one dimensional* n*-grid* denoted by $M(n)$ is the graph with vertex set $\{0, 1, \ldots, n-1\}$ and edge set $\{(i, i+1) \mid 0 \leq i \leq n-2\}$. A Cartesian product $M(n_1) \times M(n_2)$ is called a *two dimensional* $n_1 \times n_2$*-grid* and denoted by $M(n_1, n_2)$. We define that $n_1 n_2$ is the *area* of $M(n_1, n_2)$. $M(2, n)$ is called an n*-ladder* and denoted by $L(n)$. The embedding of a graph G into a two dimensional grid H is called a *layout* of G into H if it has unit congestion. A layout $\langle \phi, \rho \rangle$ of G into H is said to be *planar* if $\rho(e_1)$ and $\rho(e_2)$ are internally vertex-disjoint for any distinct $e_1, e_2 \in E(G)$.

Formann and Wagner [8] showed that the following problem is NP-complete.

Graph Layout I

Instance A planar graph G with $\Delta(G) \leq 4$ and an integer A.

Question Does there exist a layout of G into the grid of area at most A?

Kramer and Leeuwen [13] showed that Graph Layout I can be reduced to the following problem:

Graph Layout II

Instance A planar graph G with $\Delta(G) \leq 4$ and integers m, n.

Question Does there exist a layout of G into $M(m, n)$?

and thus Graph Layout II is NP-hard[1] .

We consider the following problem which is a variant of Graph Layout II:

Graph k-Layout

Instance A planar graph G with $\Delta(G) \leq 4$ and an integer n.

Question Does there exist a layout of G into $M(k, n)$?

[1] [13] claimed that Graph Layout II is in NP without proof. However, this is not trivial as mentioned in Subsection 3.3.2.

We prove that the GRAPH k-LAYOUT is NP-complete for any fixed $k \geq 3$. GRAPH 1-LAYOUT can be trivially solved in polynomial time. Although we do not know the complexity of GRAPH 2-LAYOUT, we consider a closely related problem of laying out a graph into a ladder. We show a necessary and sufficient condition for a graph to be laid out into $L(\infty)$ and show that the graph satisfying the condition can be laid out into $L(|V(G)|)$. Based on the characterization, we suggest a linear time algorithm for deciding if a given graph can be laid out into $L(\infty)$.

This chapter is organized as follows. Some definitions are given in Section 3.2. In Section 3.3, we prove the NP-completeness of GRAPH k-LAYOUT for any fixed integer $k \geq 3$. In Section 3.4, we review the proper-path-width of graphs and show some lemmas used in the following section. In Section 3.5, we give a necessary and sufficient condition for a graph to be laid out into $L(\infty)$. We conclude this chapter with some remarks in Section 3.6.

3.2 Preliminaries

$\Gamma_G(v)$ is the set of edges incident to a vertex v in a graph G. $|\Gamma_G(v)|$ is called the *degree* of v and denoted by $\deg_G(v)$. For $S \subseteq V(G)$, let $\Gamma_G(S) = \bigcup\{\Gamma_G(v) \mid v \in S\}$. $G[S]$ is the subgraph of G induced by $S \subseteq V(G)$.

For graphs G and H, $G \cup H$ is the graph with vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H)$. We write $G \subseteq H$ if G is a subgraph of H. For an embedding $\varepsilon = \langle \phi, \rho \rangle$ of G into H and $G' \subseteq G$, let $\varepsilon(G') = \bigcup_{e \in E(G')} \rho(e)$.

Let $M = M(n_1, n_2)$. For a vertex $(i, j) \in V(M)$, let $l_1(i, j) = i$ and $l_2(i, j) = j$. Let $R_i^M = \{(i, j) \in V(M) \mid 0 \leq j \leq n_2 - 1\}$ and $C_j^M = \{(i, j) \in V(M) \mid 0 \leq i \leq n_1 - 1\}$. Subgraphs $M[R_i^M]$ and $M[C_j^M]$ are called the ith *row* and the jth *column* of M, respectively. For an embedding $\langle \phi, \rho \rangle$ of M and a vertex $(i, j) \in V(M)$, we denote $\phi((i, j))$ simply by $\phi(i, j)$.

3.3 NP-Completeness of GRAPH k-LAYOUT

We prove the following theorem in this section.

Theorem 3.1 GRAPH k-LAYOUT *is NP-complete for any fixed integer* $k \geq 3$.

We prove in Subsection 3.3.1 that GRAPH k-LAYOUT ($k \geq 3$) is NP-hard by constructing a pseudo-polynomial reduction from 3-PARTITION which is well-known to be NP-complete in the strong sense to GRAPH k-LAYOUT. We show that GRAPH k-LAYOUT is in NP in Subsection 3.3.2.

3.3.1 NP-Hardness of GRAPH k-LAYOUT

3-PARTITION is defined as follows.

3-PARTITION

Instance A positive integer B, and a set of $3m$ integers $A = \{a_0, a_1, \ldots, a_{3m-1}\}$, such that $B/4 < a_x < B/2$ and $\sum_{x=0}^{3m-1} a_x = mB$.

Question Can A be partitioned into m disjoint sets $A_0, \ldots, A_{m-1}$ such that $\sum_{a \in A_y} a = B$ for $0 \leq y \leq m-1$?

For given integers $B, a_0, \ldots, a_{3m-1}$ as an instance of 3-PARTITION, we construct the instance of GRAPH k-LAYOUT as follows:

$$\begin{aligned} G(A,B) &= F(B,m,k) \cup \bigcup_{0 \leq x \leq 3m-1} M(a_x), \\ n(A,B) &= m(B+k+1)+k+1, \end{aligned}$$

where $F(B,m,k)$ is the graph obtained from $M(k, n(A,B))$ by removing the vertex $(1,j)$ and joining $(0,j)$ and $(2,j)$ by an edge for each $j = (B+k+1)y+z+k+1$ $(0 \leq y \leq m-1, 0 \leq z \leq B-1)$. Figure 3.1 shows $F(B,m,k)$. It should be noted that G is well-defined if $k \geq 3$.

Throughout this subsection, $k \geq 3$ is a fixed integer. For $0 \leq y \leq m$, we define that $J_y^M = \{(B+k+1)y+z \mid 0 \leq z \leq k\}$, $J_y^{\overline{M}} = \{(B+k+1)y+z \mid 1 \leq z \leq k-1\}$, $M_y = F(B,m,k)[\{(i,j) \mid 0 \leq i \leq k-1, j \in J_y^M\}]$, and $\overline{M}_y = F(B,m,k)[\{(i,j) \mid 0 \leq i \leq k-1, j \in J_y^{\overline{M}}\}]$. It should be noted that M_y is isomorphic to $M(k,k+1)$ for each $0 \leq y \leq m$. Moreover, for $0 \leq y \leq m-1$, we define that $J_y^D = \{(B+k+1)y+z+k \mid 0 \leq z \leq B+1\}$, and $D_y = F(B,m,k)[\{(i,j) \mid 0 \leq i \leq k-1, j \in J_y^D\}]$.

Now we show that A can be partitioned into disjoint sets $A_0, \ldots, A_{m-1}$ such that $\sum_{a \in A_y} a = B$ for $0 \leq y \leq m-1$ if and only if there exists an layout of $G(A,B)$ into $H = M(k, n(A,B))$ by a series of lemmas.

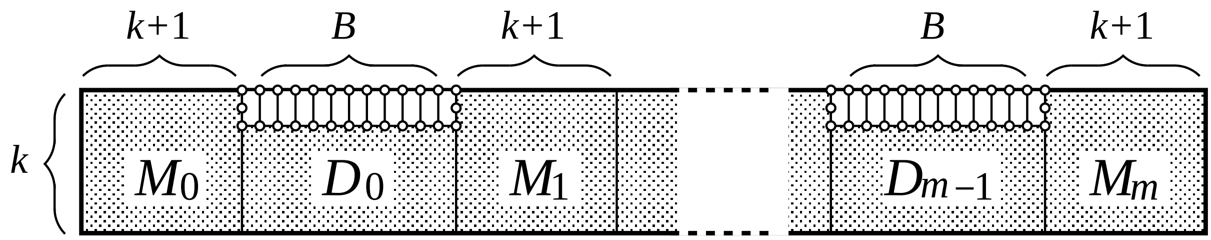

Figure 3.1: $F(B, m, k)$. The gray area is grid-connected.

Lemma 3.2 *For any layout* $\varepsilon = \langle \phi, \rho \rangle$ *of* $M = M(k, k+1)$ *into* $H = M(k, n(A, B))$*,*

$$0 \le \forall i \le k-1 \ \exists i' \ : \ \varepsilon(M[R_i^M - \{(i,0),(i,k)\}]) \subseteq H[R_{i'}^H], \tag{3.1}$$

$$1 \le \forall j \le k-1 \ \exists j' \ : \ \varepsilon(M[C_j^M]) = H[C_{j'}^H]. \tag{3.2}$$

Proof For $0 \le i \le k-1$ and $0 \le j \le k$, let $P_i^R = \varepsilon(M[R_i^M])$ and $P_j^C = \varepsilon(M[C_j^M])$. Let

$$\begin{aligned} q_1 &= \min_{0 \le j \le k} \max_{0 \le i \le k-1} l_2(\phi(i,j)), \\ q_2 &= \max_{0 \le j \le k} \min_{0 \le i \le k-1} l_2(\phi(i,j)), \end{aligned}$$

and

$$\begin{aligned} j_1 &\in \{0 \le j \le k \mid \max_{0 \le i \le k-1} l_2(\phi(i,j)) = q_1\}, \\ j_2 &\in \{0 \le j \le k \mid \min_{0 \le i \le k-1} l_2(\phi(i,j)) = q_2\}. \end{aligned}$$

It follows from the definitions of q_1 and q_2 that

$$0 \le \forall j \le k \ \exists v_1 \in C_j^M : q_1 \le l_2(\phi(v_1)), \text{ and } \exists v_2 \in C_j^M : q_2 \ge l_2(\phi(v_2)). \tag{3.3}$$

Claim 3.3 $q_1 < q_2$.

Proof If $q_1 > q_2$ then it follows from (3.3) that $P_0^C, \ldots, P_k^C$ are $k+1$ edge-disjoint trails across the columns between the q_1st column and the q_2nd column of H. However, this is impossible since H has just k rows. Thus, we have $q_1 \le q_2$.

It remains to show that $q_1 \ne q_2$. We prove this by contradiction. If $q_1 = q_2 = q$ then it follows from (3.3) that $P_0^C, \ldots, P_k^C$ are $k+1$ edge-disjoint trails across the qth column of H. Thus $0 < q < n(A,B) - 1$, for otherwise, $q = 0$ or $q = n(A,B) - 1$, and we have that $\phi(C_j^M) \cap C_q^H \neq \emptyset$ for every $0 \leq j \leq k$, contradicting that ϕ is one-to-one since $|\{C_j^M\}| > |C_q^H|$. We define that

$$\begin{aligned} E^- &= \{((i,q-1),(i,q)) \in E(H) \mid 0 \leq i \leq k-1\}, \\ E^+ &= \{((i,q),(i,q+1)) \in E(H) \mid 0 \leq i \leq k-1\}. \end{aligned}$$

For each $0 \leq j \leq k$, if $\phi(C_j^M) \cap C_q^H = \emptyset$ then there exist $v_1, v_2 \in C_j^M$ such that $l_2(\phi(v_2)) < q < l_2(\phi(v_1))$ from (3.3). Thus, it follows that for any $0 \leq j \leq k$,

$$\phi(C_j^M) \cap C_q^H \neq \emptyset \text{ or} \tag{3.4}$$

$$E(P_j^C) \cap E^- \neq \emptyset \text{ and } E(P_j^C) \cap E^+ \neq \emptyset. \tag{3.5}$$

Claim 3.4 *For any* $0 \leq j \leq k$,

$$E(P_j^C) \cap (E^- \cup E^+) \neq \emptyset. \tag{3.6}$$

Proof If there exists $0 \leq j' \leq k$ such that $E(P_{j'}^C) \cap (E^- \cup E^+) = \emptyset$, then $P_{j'}^C$ is identical with $H[C_q^H]$. This means that a vertex with degree at least 3 in $C_{j'}^M$ is mapped into $\{(i,q) \in V(H) \mid 1 \leq i \leq k-2\}$, and that a vertex with degree at least 2 in $C_{j'}^M$ is mapped into $\{(i,q) \in V(H) \mid i = 0 \text{ or } k-1\}$. Thus, both (3.4) and (3.5) do not hold for any $j \neq j'$ $(0 \leq j \leq k)$, a contradiction. Therefore, (3.6) holds for any $0 \leq j \leq k$.

End of proof of Claim 3.4

Claim 3.5 $j_1 \neq j_2$.

Proof If $j_1 = j_2$, then $\phi(C_{j_1}^M) = C_q^H$ by definition. Thus, for every $j \neq j_1$ $(0 \leq j \leq k)$, (3.5) holds since (3.4) does not hold. However, since $(0,q) \in \phi(C_{j_1}^M)$ and $\deg_H(0,q) = 3$, P_j^C does not pass through $(0,q)$ for every $j \neq j_1$ $(0 \leq j \leq k)$. Thus P_j^C does not pass through $\Gamma_H(0,q)$ for every $j \neq j_1$ $(0 \leq j \leq k)$. Since $P_0^C, \ldots, P_k^C$ are edge-disjoint, it follows from (3.5) that

$$\sum_{0 \leq j \leq k-1} |E(P_j^C) \cap E^-| + \sum_{0 \leq j \leq k-1} |E(P_j^C) \cap E^+| + |\Gamma_H(0,q) \cap (E^- \cup E^+)| \geq k + k + 2 = 2k + 2.$$

However, this is a contradiction since the left hand side of the inequality is no more than $|E^- \cup E^+| = 2k$. Therefore, we have $j_1 \neq j_2$. *End of proof of Claim 3.5*

Let

$$\begin{aligned} C_1 &= \{v \in C^M_{j_1} \mid l_2(\phi(v)) = q\}, \text{ and} \\ C_2 &= \{v \in C^M_{j_2} \mid l_2(\phi(v)) = q\}. \end{aligned}$$

Since

$$\begin{aligned} \forall v \in C^M_{j_1} - C_1 &: \; l_2(\phi(v)) < q, \text{ and} \\ \forall v \in C^M_{j_2} - C_2 &: \; l_2(\phi(v)) > q \end{aligned}$$

by definition, it follows that

$$\forall i \in X_1 \; : \; E(P^R_i) \cap E^- \neq \emptyset, \tag{3.7}$$

$$\forall i \in X_2 \; : \; E(P^R_i) \cap E^+ \neq \emptyset, \tag{3.8}$$

where

$$\begin{aligned} X_1 &= \{0 \le i \le k-1 \mid (i, j_1) \in C^M_{j_1} - C_1\}, \\ X_2 &= \{0 \le i \le k-1 \mid (i, j_2) \in C^M_{j_2} - C_2\}. \end{aligned}$$

Since $P^C_0, \ldots, P^C_k$ and $P^R_0, \ldots, P^R_{k-1}$ are edge-disjoint, we have

$$\sum_{0 \le j \le k} |E(P^C_j) \cap (E^- \cup E^+)| + \sum_{i \in X_1} |E(P^R_i) \cap E^-| + \sum_{i \in X_2} |E(P^R_i) \cap E^+| \le |E^- \cup E^+| = 2k. \tag{3.9}$$

Since $j_1 \neq j_2$, it follows that $|C_1| + |C_2| = |C_1 \cup C_2| \le |C^H_q| = k$. Thus, it follows from (3.6), (3.7), and (3.8) that

$$\begin{aligned} (\text{the left hand side of (3.9)}) &\ge (k+1) + |X_1| + |X_2| \\ &= (k+1) + (k - |C_1|) + (k - |C_2|) \\ &\ge (k+1) + 2k - k \\ &\ge 2k + 1, \end{aligned}$$

a contradiction. This proves that $q_1 \neq q_2$.

Therefore, we have $q_1 < q_2$. *End of proof of Claim 3.3*

Thus $P_0^R, \ldots, P_{k-1}^R$ are k edge-disjoint trails across the columns between the q_1st column and the q_2nd column of H. Each P_i^R $(0 \le i \le k-1)$ passes through only edges in one row of $H' = H[\bigcup_{q_1 \le j \le q_2} C_j^H]$ since H has just k rows. Thus, it follows from (3.3) that for any $0 \le j \le k$ $(j \notin \{j_1, j_2\})$ P_j^C passes through only column edges of H'. Therefore, we have $\{j_1, j_2\} = \{0, k\}$, and (3.1) and (3.2) hold. □

Throughout this subsection, we assume that $\varepsilon = \langle \phi, \rho \rangle$ is a layout of $F(B, m, k)$ into $H = M(k, n(A, B))$. We may assume without loss of generality that

$$l_1(\phi(0,1)) \le l_1(\phi(k-1,k-1)), \text{ and } l_2(\phi(0,1)) \le l_2(\phi(k-1,k-1)). \tag{3.10}$$

Lemma 3.6 *For any* $0 \le y \le m$,

$$0 \le \forall i \le k-1 \ \exists i' \ : \ \varepsilon(F(B,m,k)[\{(i,j) \mid j \in J_y^{\overline{M}}\}]) \subseteq H[R_{i'}^H], \tag{3.11}$$

$$\forall j \in J_y^{\overline{M}} \ \exists j' \ : \ \varepsilon(F(B,m,k)[\{(i,j) \mid 0 \le i \le k-1\}]) = H[C_{j'}^H]. \tag{3.12}$$

Proof Immediate from Lemma 3.2. □

Corollary 3.7 *For any* $0 \le y \le m$ *and* $e \in E(F(B,m,k)) - E(\overline{M}_y)$, $\rho(e)$ *does not pass through an edge of* $\varepsilon(\overline{M}_y)$. □

Lemma 3.8 *For any* $j \in J_y^{\overline{M}}$ *and* $j' \in J_{y'}^{\overline{M}}$ $(j < j',\ 0 \le y \le y' \le m)$,

$$l_2(\phi(0,j)) < l_2(\phi(0,j')). \tag{3.13}$$

Proof We first consider the case when $y = 0$. It follows from Lemma 3.6 and assumption (3.10) that

$$l_2(\phi(0,1)) < l_2(\phi(0,2)) < \cdots < l_2(\phi(0,k-1)).$$

Thus, (3.13) holds for any $j, j' \in J_0^{\overline{M}}$ $(j < j')$. Furthermore, for any $j \in J_0^{\overline{M}}$ and $j' \in J_{y'}^{\overline{M}}$ $(j < j',\ 0 < y' \le m)$, (3.13) follows from Corollary 3.7.

We next consider the case when $y > 0$. Suppose $j \in J_y^{\overline{M}}$ and $j' \in J_{y'}^{\overline{M}}$ $(j < j',\ y \le y' \le m)$. Let $P = \varepsilon(F(B,m,k)[\{(0,l) \mid k-1 \le l \le j\}])$. Since $l_2(\phi(0,k-1)) < l_2(\phi(0,j'))$, if $l_2(\phi(0,j)) > l_2(\phi(0,j'))$ then P passes through a vertex in $C_{l_2(\phi(0,j'))}^H$. This means that $\varepsilon(F(B,m,k)[\{(i,j') \mid 0 \le i \le k-1\}]) \ne C_{l_2(\phi(0,j'))}^H$, contradicting to (3.12).

Therefore, we have $l_2(\phi(0,j)) < l_2(\phi(0,j'))$ for any $j \in J_y^{\overline{M}}$ and $j' \in J_{y'}^{\overline{M}}$ $(j < j',\ 0 \le y \le y' \le m)$. □

Lemma 3.9 *For any* $j \in J_y^{\overline{M}}$ *(*$0 \leq y \leq m$*) and* j', j'' *such that* $0 \leq j' < j < j'' \leq n(A, B) - 1$,

$$\max_{0 \leq i \leq k-1} l_2(\phi(i, j')) < l_2(\phi(0, j)) < \min_{0 \leq i \leq k-1} l_2(\phi(i, j'')). \tag{3.14}$$

Proof Immediate from (3.12), Corollary 3.7, and Lemma 3.8. □

Lemma 3.10 *For any* $0 \leq y \leq m-1$ *and any* $j, j' \in J_y^D$ *(*$j < j'$*),* $l_2(\phi(0, j)) < l_2(\phi(0, j'))$.

Proof It follows from Lemma 3.9 that

$$l_2(\phi(0, (B + k + 1)y + k - 1)) < l_2(\phi(0, j)) < l_2(\phi(0, (B + k + 1)(y + 1) + 1)) \tag{3.15}$$

for any $j \in J_y^D$. Fix $j, j' \in J_y^D$ ($j < j'$) and let $q = l_2(\phi(0, j))$. We define that

$$\begin{aligned} E^- &= \{((i, q - 1), (i, q)) \in E(H) \mid 0 \leq i \leq k - 1\}, \\ E^+ &= \{((i, q), (i, q + 1)) \in E(H) \mid 0 \leq i \leq k - 1\}. \end{aligned}$$

For $i \in \{0, 2, \ldots k-1\}$, let $P_i^R = \varepsilon(F(B, m, k)[\{(i, (B+k+1)y+z+k) \mid -1 \leq z \leq B+2\}])$. Since $P_3^R, \ldots, P_{k-1}^R$ are edge-disjoint and each P_i^R ($3 \leq i \leq k - 1$) contains at least one edge in E^- and at least one edge in E^+ from (3.15), it follows that

$$\sum_{3 \leq i \leq k-1} |E(P_i^R) \cap E^-| \geq k - 3, \tag{3.16}$$

$$\sum_{3 \leq i \leq k-1} |E(P_i^R) \cap E^+| \geq k - 3. \tag{3.17}$$

First assume that $l_2(\phi(0, j)) > l_2(\phi(0, j'))$. It follows from (3.15) that P_0^R contains at least 3 edges in E^-, and P_2^R contains at least one edge in E^-. Since $P_0^R, P_2^R, \ldots, P_{k-1}^R$ are edge-disjoint, it follows from (3.16) that

$$\sum_{i \in \{0, 2, \ldots, k-1\}} |E(P_i^R) \cap E^-| \geq 3 + k - 2 = k + 1,$$

which is a contradiction since the left hand side of the inequality is no more than $|E^-| = k$.

Next assume that $l_2(\phi(0, j)) = l_2(\phi(0, j'))$.

Assume that $l_2(\phi(2, j)) = l_2(\phi(2, j')) = q$. Since all the vertices in $U = \{(0, j), (0, j'), (2, j), (2, j')\} \subset V(F(B, m, k))$ have degree at least 3, none of $P_3^R, \ldots, P_{k-1}^R$

passes through a vertex in $\phi(U)$. Thus none of $P_3^R, \ldots, P_{k-1}^R$ passes through an edge in $\Gamma_H(\phi(U))$. Since $|\Gamma_H(\phi(U)) \cap E^-| \geq 4$ by the assumption that $\phi(U) \subseteq C_q^H$, it follows from (3.16) that

$$\sum_{3 \leq i \leq k-1} |E(P_i^R) \cap E^-| + |\Gamma_H(\phi(U)) \cap E^-| \geq k - 3 + 4 = k + 1.$$

This is a contradiction since the left hand side of the inequality is no more than $|E^-| = k$.

Thus, we conclude that $l_2(\phi(2,j)) \neq q$ or $l_2(\phi(2,j')) \neq q$. We assume without loss of generality that $l_2(\phi(2,j)) \neq q$ and show a contradiction. For $i \in \{0,2\}$, let $P_i^{R-} = \varepsilon(F(B,m,k)[\{(i,l) \mid (B+k+1)y+k-1 \leq l \leq j\}])$, and $P_i^{R+} = \varepsilon(F(B,m,k)[\{(i,l) \mid j \leq l \leq (B+k+1)(y+1)+1\}])$. Moreover, let $P_j^C = \varepsilon(F(B,m,k)[\{(i,j) \mid 0 \leq i \leq k-1\}])$.

Case 1 $l_2(\phi(2,j)) < q$: Each of P_0^{R-}, P_2^{R+}, and P_j^C contains at least one edge in E^-, and they together with $P_3^R, \ldots, P_{k-1}^R$ are edge-disjoint. Moreover, none of P_0^{R-}, P_2^{R+}, P_j^C, and $P_3^R, \ldots, P_{k-1}^R$ passes through $\phi(0,j')$. Thus none of P_0^{R-}, P_2^{R+}, P_j^C, and $P_3^R, \ldots, P_{k-1}^R$ passes through an edge in $\Gamma_H(\phi(0,j'))$. Thus, it follows from (3.16) that

$$\begin{aligned}&\sum_{3 \leq i \leq k-1} |E(P_i^R) \cap E^-| + |\Gamma_H(\phi(0,j')) \cap E^-| + \\ &|E(P_0^{R-}) \cap E^-| + |E(P_2^{R+}) \cap E^-| + |E(P_j^C) \cap E^-| \\ &\geq k - 3 + 4 = k + 1.\end{aligned}$$

This is a contradiction since the left hand side of the inequality is no more than $|E^-| = k$.

Case 2 $l_2(\phi(2,j)) > q$: Let $P' = \varepsilon(F(B,m,k)[\{(2,j),(2,j'),(0,j')\}])$. Each of P_0^{R+}, P_2^{R-}, P_j^C, and P' contains at least one edge in E^+, and they together with $P_3^R, \ldots, P_{k-1}^R$ are edge-disjoint. Thus, it follows from (3.16) that

$$\begin{aligned}&\sum_{3 \leq i \leq k-1} |E(P_i^R) \cap E^+| + |E(P_0^{R+}) \cap E^+| + \\ &|E(P_2^{R-}) \cap E^+| + |E(P_j^C) \cap E^+| + |E(P') \cap E^+| \\ &\geq k - 3 + 4 = k + 1.\end{aligned}$$

This is again a contradiction since the left hand side of the inequality is no more than $|E^+| = k$.

Therefore, we conclude that $l_2(\phi(0,j)) < l_2(\phi(0,j'))$. □

Lemma 3.11 *For any* $0 \le y \le m$*,*

$$\forall j \in J_y^{\overline{M}} \;:\; \phi(\{(i,j) \mid 0 \le i \le k-1\}) = C_j^H, \tag{3.18}$$
$$\forall j \in J_y^D \;:\; \phi(\{(i,j) \mid 0 \le i \le k-1\}) \subset \{(i,l) \in V(H) \mid 0 \le i \le k-1, l \in J_y^D\} \tag{3.19}$$

Proof It follows from Lemmas 3.9 and 3.10 that $l_2(\phi(0,j)) < l_2(\phi(0,j'))$ for any $0 \le j < j' \le n(A,B)-1$. Since H has just $n(A,B)$ columns, we have $l_2(\phi(0,j)) = j$ for any $0 \le j \le n(A,B)-1$. Thus, (3.18) holds by Lemma 3.6, and (3.19) holds by (3.18) and Lemma 3.9. □

Now we are ready to prove the following.

Lemma 3.12 Graph k-Layout *is NP-hard for any fixed integer* $k \ge 3$*.*

Proof We first assume that A can be partitioned into disjoint sets $A_0, \ldots, A_{m-1}$ such that $\sum_{a \in A_y} a = B$ for $0 \le y \le m-1$. We construct a layout $\langle \phi', \rho' \rangle$ of $G(A,B)$ into H as follows: By the definition of $F(B,m,k)$, $F(B,m,k)$ has a planar layout into H such that $\phi'(i,j) = (i,j)$. For each $0 \le y \le m-1$, we layout $M(a_x)$ into $H[\{(1,(B+k+1)y+z+k+1) \mid 0 \le z \le B-1\}]$ if $a_x \in A_y$. We can construct such layout by the assumption that A can be partitioned into disjoint sets $A_0, \ldots, A_{m-1}$ such that $\sum_{a \in A_y} a = B$ for $0 \le y \le m-1$. Thus, we have obtained the desired layout.

Conversely, we assume that there exists a layout $\varepsilon' = \langle \phi', \rho' \rangle$ of $G(A,B)$ into H. For $0 \le y \le m-1$, let $U_y = U'_y - \phi'(V(F(B,m,k)))$, where $U'_y = \{(i,j) \in V(H) \mid 0 \le i \le k-1, j \in J_y^D\}$. It follows from Lemma 3.11 that $|U_y| = B$ for $0 \le y \le m-1$. Let $U = \bigcup_{0 \le y \le m-1} U_y$. Every $M(a_x)$ $(0 \le x \le 3m-1)$ is mapped into either U_y or $U - U_y$ by Lemma 3.11 and the structure of $F(B,m,k)$. This means that A can be partitioned into disjoint sets $A_0, \ldots, A_{m-1}$ such that $\sum_{a \in A_y} a = B$ for $0 \le y \le m-1$.

The reduction is pseudo-polynomial since $G(A,B)$ has $kn(A,B) = O(Bm)$ vertices. Thus, Graph k-Layout is NP-hard for any fixed integer $k \ge 3$ since 3-Partition is NP-complete in the strong sense. □

3.3.2 Graph k-Layout is in NP

In this subsection, we prove that Graph k-Layout is in NP. This is not trivial in the sense that every layout of G into H itself may not be a witness of polynomial size if n is much greater than $|V(G)|$. However, the following lemma guarantees that there exists a witness of polynomial size for any instance.

Lemma 3.13 *A graph G which can be laid out into $M(k,n)$ can be laid out into $M(k, 2k|V(G)|)$.*

Proof Let $\varepsilon = \langle \phi, \rho \rangle$ be a layout of G into $H = M(k,n)$. Let $J = \{j \mid \phi(V(G)) \cap C_j^H \neq \emptyset\}$, and we suppose $J = \{j_1, \ldots, j_{|J|}\}$ where $j_1 < \cdots < j_{|J|}$. Obviously, $|J| \leq |V(G)|$. For $1 \leq l \leq |J|-1$, let $E_l = \{((i, j_l), (i, j_l+1)) \in E(H) \mid 0 \leq i \leq k-1\} \bigcup \{((i, j_{l+1}-1), (i, j_{l+1})) \in E(H) \mid 0 \leq i \leq k-1\}$, and $M_l = H[\bigcup_{j_l \leq j \leq j_{l+1}} C_j^H]$. Moreover, let $M_0 = H[\bigcup_{0 \leq j \leq j_1} C_j^H]$, and $M_{|J|} = H[\bigcup_{j_{|J|} \leq j \leq n-1} C_j^H]$.

Suppose that M_l ($1 \leq l \leq |J|-1$) has more than $2k+1$ columns. If an image of ρ contains an edge in E_l then the image forms one or more subtrail(s) contained in M_l called "net(s)" each of which contains exactly two edges in E_l. Notice that the image contains the even number of edges in E_l since no vertex of G is mapped by ϕ into $V(M_l) - (C_{j_l}^H \cup C_{j_{l+1}}^H)$. Thus, for $1 \leq l \leq |J|-1$, the layout forms a solution of a "channel routing problem" on M_l by considering a vertex in $C_{j_l}^H \cup C_{j_{l+1}}^H$ to be a "terminal" which is connected by a net in M_l. It is known that for a fixed channel length k, if there exists a routing for an instance then there exists a routing with channel width at most $2k-1$ [10]. Thus, we can compact M_l by applying the result so that it has at most $2k+1$ columns.

For M_l ($l \in \{0, |J|\}$), terminals are on only single side of the channel, i.e. C_l^H, and it is easy to see that channel width $\lfloor k/2 \rfloor$ are sufficient for such case. It follows that we can compact M_l so that it has at most $\lfloor k/2 \rfloor + 1$ columns.

Thus, we can obtain a layout of G into $M(k,x)$, where

$$\begin{aligned} x &\leq (2k-1)(|J|-1) + 2\lfloor k/2 \rfloor + |J| \\ &\leq 2k|J| - (2k-1) + k \\ &\leq 2k|V(G)|. \end{aligned}$$

□

Lemma 3.14 GRAPH k-LAYOUT *is in NP.*

Proof Suppose that there exists a layout ε of G into $M(k, n)$. Then $\Delta(G) \leq 4$ obviously. From Lemma 3.13, we can assume that n is at most $2k|V(G)|$. Thus, we can check that ε is a layout in $O(|E(M(k,n))||E(G)| + |V(G)|) = O(2kn \cdot 2|V(G)| + |V(G)|) = O(k^2|V(G)|^2)$ time. □

3.4 Proper-Path-Decomposition

In this section, we show some lemmas related on proper-path-decomposition used in the following section.

A k-proper-path-decomposition $(X_1, X_2, \ldots, X_r)$ is said to be *full* if $|X_i| = k + 1$ $(1 \leq i \leq r)$ and $|X_j \cap X_{j+1}| = k$ $(1 \leq i \leq r - 1)$ [25]. The following lemma is shown in [25].

Lemma A *For any graph G with $ppw(G) = k$, there exists a full k-proper-path-decomposition of G.*

The following lemma will be used in the next section.

Lemma 3.15 *Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a full proper-path-decomposition. For $2 \leq i \leq r - 1$, there exist a unique $s_i \in X_i - X_{i-1}$ and a unique $t_i \in X_i - X_{i+1}$ $(s_i \neq t_i)$. Moreover, $X_i - \{s_i, t_i\} = X_{i-1} \cap X_{i+1}$.*

Proof It is obvious from the definition (a) that there exist $s_i \in X_i - X_{i-1}$ and $t_i \in X_i - X_{i+1}$ for $2 \leq i \leq r - 1$. Since $\mathcal{X}$ is full, it follows that $|X_i - X_{i-1}| = |X_i - X_{i+1}| = 1$, so we have $X_i - \{s_i\} \subset X_{i-1}$ and $X_i - \{t_i\} \subset X_{i+1}$. Thus, $X_i - \{s_i, t_i\} \subseteq X_{i-1} \cap X_{i+1}$. It follows from the definition (e) that $|X_i| - 2 \geq |X_{i-1} \cap X_{i+1}| \geq |X_i| - |\{s_i, t_i\}|$. Therefore, we have $s_i \neq t_i$ and $X_i - \{s_i, t_i\} = X_{i-1} \cap X_{i+1}$ for $2 \leq i \leq r - 1$. □

Corollary 3.16 *Let $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ be a full 2-proper-path-decomposition. For $2 \leq i \leq r - 1$, there exist a unique $s_i \in X_i - X_{i-1}$, a unique $t_i \in X_i - X_{i+1}$ $(s_i \neq t_i)$, and a unique $v_i \in X_{i-1} \cap X_{i+1}$.* □

3.5 Graph Layout into Ladders

In this section, we show a necessary and sufficient condition for a graph G to be laid out into $L(\infty)$ based on the proper-path-width of G, and show that G satisfying the condition is embeddable into $L(|V(G)|)$. Based on the characterization, we suggest a linear time algorithm for deciding if a given graph can be laid out into $L(\infty)$.

Lemma 3.17 *If a graph G can be laid out into $L(\infty)$, then $\Delta(G) \leq 3$ and $ppw(G[S]) \leq 2$, where $S = \{v \in V(G) \mid \deg_G(v) \geq 2\}$.*

Proof Suppose that there exists a layout $\langle \phi, \rho \rangle$ of G into $L(\infty)$. Then, we have $\Delta(G) \leq 3$ since $\Delta(L(\infty)) \leq 3$. Moreover, for $(u, v) \in E(G)$ and $w \in V(G) - \{u, v\}$, $\deg_G(w) \leq 1$ if $\rho(u, v)$ contains $\phi(w)$. Thus, $\rho(e_1)$ and $\rho(e_2)$ are internally vertex-disjoint for any distinct edges $e_1, e_2 \in E(G[S])$. This means that $G[S]$ is homeomorphic to a subgraph of $L(\infty)$. It is not difficult to see that $ppw(L(n)) \leq 2$ for any positive integer n. Therefore, we have $ppw(G[S]) \leq 2$. □

Lemma 3.18 *For a graph G such that $\Delta(G) \leq 3$, $|V(G)| \geq 2$, and $ppw(G) \leq 2$, there exists a planar layout of G into $L(|V(G)| - 1)$.*

Proof We denote $L(|V(G)| - 1)$ simply by L. It is easy to see that there exists an desired layout of G into L if $ppw(G) = 1$ or $|V(G)| \leq 3$. Thus we assume that $ppw(G) = 2$ and $|V(G)| \geq 4$, and we will construct a desired layout $\varepsilon = \langle \phi, \rho \rangle$.

There exists a full 2-proper-path-decomposition $\mathcal{X} = (X_1, X_2, \ldots, X_r)$ of G from the assumption that $ppw(G) = 2$ and Lemma A. It should be noted that $r = |V(G)| - 2 \geq 2$. The following is an algorithm for laying out G into $L(|V(G)| - 1)$.

Phase 1 Denote $s_i \in X_i - X_{i-1}$, $t_i \in X_i - X_{i+1}$, and $v_i = X_{i-1} \cap X_{i+1}$ for $2 \leq i \leq r - 1$ according to Corollary 3.16. In addition, let t_1 be a unique element in $X_1 - X_2$, s_r be a unique element in $X_r - X_{r-1}$, $v_r = v_{r-1} (\in X_r)$, and $t_r = s_{r-1}$.

Phase 2 Set $\phi(t_1) = (0, 0)$, $\phi(v_2) = (1, 1)$, and $\phi(t_2) = (0, 1)$. If (t_1, v_2), (t_1, t_2), and (v_2, t_2) are contained in $E(G)$, then set $\rho(t_1, v_2) = L[\{(0, 0), (1, 0), (1, 1)\}]$, $\rho(t_1, t_2) = L[\{(0, 0), (0, 1)\}]$, and $\rho(v_2, t_2) = L[\{(1, 1), (0, 1)\}]$.

Phase 3 Execute the following for $i = 2$ to r:

(a) Set $\phi(s_i) = (l_1(\phi(t_i)), i)$. Let

$$\begin{aligned} P_1 &= L[\{(l_1(\phi(t_i)), j) \mid l_2(\phi(t_i)) \leq j \leq i\}], \\ P_2 &= L[\{(l_1(\phi(v_i)), j) \mid l_2(\phi(v_i)) \leq j \leq i\}], \\ P_3 &= L[C_i^L]. \end{aligned}$$

(b) If $(t_i, s_i) \in E(G)$, then set $\rho(t_i, s_i) = P_1$.

(c) If $(s_i, v_i) \in E(G)$ and no $s_{i'}$ $(i' > i)$ is adjacent to v_i, set $\rho(v_i, s_i) = P_2 \cup P_3$.

(d) If $(s_i, v_i) \in E(G)$ and there exists $s_{i'}$ $(i' > i)$ adjacent to v_i, reset $\phi(v_i) = (l_1(x), i)$ and $\rho(s_i, v_i) = P_3$, where x is the vertex in L into which v_i was mapped before reseting. Moreover, if there exists $y \in V(G) - \{s_i, s_{i'}\}$ adjacent to v_i, then reset $\rho(y, v_i) = P_0 \cup P_2$, where P_0 is the trail in L in which (y, v_i) was mapped before reseting.

Let $Y_i = \bigcup_{1 \leq j \leq i} X_j$. We show that ε is the planar layout of G into L by induction on the number of steps in Phase 3. It should be noted that, up to step i in Phase 3, $G[Y_i]$ is laid out into L and that $\phi(v_i)$ may be reset later.

The layout of $G[Y_1]$ defined in Phase 1 is obviously desired one. We assume that ε is the planar layout of $G[Y_{i-1}]$ into $L(|Y_{i-1}| - 1)$ for step $i - 1$, and show that this is also true for step i. Notice that $|Y_i| = i + 2$.

We first show that $\varepsilon(G) \subseteq L(|Y_i| - 1)$. It is easy to see that ϕ is an injection of Y_i since $l_1(\phi(t_i)) \neq l_1(\phi(v_i))$. $\phi(Y_{i-1}) \subseteq \bigcup_{0 \leq j \leq i-1} C_j^L$ by induction hypothesis. After step i, $\phi(s_i) \in C_i^L$ and $\phi(v_i) \in \bigcup_{0 \leq j \leq i} C_j^L$ since $t_i \in Y_{i-1}$. This means that $\phi(Y_i) \subset V(L(|Y_i| - 1))$. Moreover, the images of ρ defined in step i are contained in $P_1 \cup P_2 \cup P_3$, and $P_1 \cup P_2 \cup P_3 \subseteq L(|Y_i| - 1)$. Thus, we conclude that $\varepsilon(G) \subseteq L(|Y_i| - 1)$.

We next show that ε is the planar layout. Notice that P_1, P_2, and P_3 are internally vertex-disjoint. P_1 and $\rho(e)$ are internally vertex-disjoint for any $e \in E(G[Y_{i-1}])$ since neither vertices nor edges in $\varepsilon(G[Y_{i-1}])$ are contained in $L[\{(l_1(\phi(t_i)), j) \mid j \geq l_2(\phi(t_i))\}]$ except $\phi(t_i)$. If $(s_i, v_i) \notin E(G)$ then ε is the planar layout since $\varepsilon(G[Y_i]) \subseteq \varepsilon(G[Y_{i-1}]) \cup P_1$. If $(s_i, v_i) \in E(G)$ then P_2, P_3, and $\rho(e)$ are internally vertex-disjoint for any $e \in E(G[Y_{i-1}])$

since neither vertices nor edges in $\varepsilon(G[Y_{i-1}])$ are contained in $L[\{(l_1(\phi(v_i)), j) \mid j \geq l_2(\phi(v_i))\}]$ except $\phi(v_i)$. Thus, we conclude that ε is the planar layout. □

Lemma 3.19 *For a graph G such that $\Delta(G) \leq 3$, $|S| \geq 2$, and $ppw(G[S]) \leq 2$, there exists a layout of G into $L(|V(G)| - 1)$, where $S = \{v \in V(G) \mid \deg_G(v) \geq 2\}$.*

Proof It follows from Lemma 3.18 and the assumption that $\Delta(G[S]) \leq 3$, $|S| \geq 2$, and $ppw(G[S]) \leq 2$ that there exists a planar layout of $G[S]$ into $L(|S|-1)$. Let $v \in V(G)-S$, and let $u \in V(G)$ be a vertex adjacent to v if such u exists. Since $\deg_{G[S]}(u) \leq 2$, We can map v and (u, v) by adding a new column next to the column containing $\phi(u)$ so that the congestion of the resulting embedding is one. Thus, we can obtain the layout of G into $L(|V(G)| - 1)$ since the number of additional columns is at most $|V(G) - S|$. □

We have the following theorem from Lemmas 3.17 and 3.19.

Theorem 3.20 *A graph G can be laid out into $L(\infty)$ if and only if $\Delta(G) \leq 3$ and $ppw(G[S]) \leq 2$, where $S = \{v \in V(G) \mid \deg_G(v) \geq 2\}$.* □

Based on this theorem, we can obtain a linear time algorithm for deciding if a given graph G can be laid out into $L(\infty)$ by using the algorithm `PPD_GENERAL` described in Chapter 2. If a full 2-proper-path-decomposition of $G[S]$ is given, the algorithm obtained from the proofs of Lemmas 3.18 and 3.19 provides a layout of G into $L(|V(G)|)$ in $O(|V(G)|)$ time. For a graph G with $\Delta(G) \leq 3$ and $ppw(G) \leq 2$, we can construct in linear time a full 2-proper-path-decomposition of G from the output of the algorithm `PPD_GENERAL`, although the details are omitted here. Therefore, our algorithm can be modified so that it lays out G satisfying the condition of Theorem 3.20 into $L(|V(G)|)$ in $O(|V(G)|)$ time.

3.6 Concluding Remarks

Let $A(G)$ be the minimum area of a ladder into which an N-vertex graph G can be laid out. We can easily modify the algorithm obtained from the proofs of Lemmas 3.18 and 3.19 so that it lays out G into $L(N-2)$ if G has at least 5 vertices with degree at least 2. Thus we have $A(G) \leq 2(N-2)$. This is the tight bound for $A(G)$ as described in the following corollary.

Corollary 3.21 *If an N-vertex graph G has at least 5 vertices with degree at least 2 then $N \leq A(G) \leq 2(N-2)$. Moreover, these are tight bounds, i.e. there exist graphs with $A(G) = N$ and graphs with $A(G) = 2(N-2)$.*

Proof The lower bound is trivial. It is not difficult to see that the graph G shown in Figure 3.2 has $A(G) = 2(N-2)$. □

Figure 3.2: A graph G with $A(G) = 2(N-2)$.

Chapter 4

Embedding into Hypercubes

4.1 Introduction

The problem of efficiently implementing parallel algorithms on parallel machines has been studied as the graph embedding problem, which is to embed the communication graph underlying a parallel algorithm within the processor interconnection graph for a parallel machine with minimal communication overhead. It is well-known that the dilation and/or congestion of the embedding are lower bounds on the communication delay, and the problem of embedding a guest graph within a host graph with minimal dilation and/or congestion has been extensively studied.

We consider minimal congestion embeddings of graphs in hypercubes, which are well-known as one of the most popular processor interconnection graphs for parallel machines. It was pointed out by Kim and Lai [12] that minimal congestion embeddings are very important for a hypercube that uses circuit switching for node-to-node communication such as Intel iPSC/2 [19].

Let G be a graph and let $V(G)$ and $E(G)$ denote the vertex set and edge set of G, respectively. We denote by $\Delta(G)$ the maximum degree of a vertex in G. A tree T is said to be *binary* if $\Delta(T) \leq 3$. An *embedding* $\langle \phi, \rho \rangle$ of a graph G into a graph H is defined by a one-to-one mapping $\phi : V(G) \rightarrow V(H)$, together with a mapping ρ that maps each edge $(u, v) \in E(G)$ onto a path $\rho(u, v)$ in H that connects $\phi(u)$ and $\phi(v)$. ϕ and ρ are called the labeling and routing of an embedding $\langle \phi, \rho \rangle$, respectively. The *dilation* of an edge $e \in E(G)$ under $\langle \phi, \rho \rangle$ is the length of the path $\rho(e)$. The dilation of an embedding $\langle \phi, \rho \rangle$ is the maximum dilation of an edge in G. The *congestion* of an edge $e' \in E(H)$

under $\langle \phi, \rho \rangle$ is the number of edges e in G such that $\rho(e)$ contains e'. The congestion of an embedding $\langle \phi, \rho \rangle$ is the maximum congestion of an edge in H. The *n-cube* (*n-dimensional cube*) $Q(n)$ is the graph with 2^n vertices labeled 0 through $2^n - 1$ such that two vertices are joined by an edge if and only if their labels in the binary representation differ by exactly one bit. We assume that the bits are numbered 0 through $n-1$. An edge (u, v) in $Q(n)$ is called an *i-edge* (*i-dimensional edge*) if the labels of u and v in the binary representation differ in the ith bit ($0 \le i \le n-1$). It is well-known that $Q(n)$ is n-connected.

Kim and Lai [12] showed that for a given N-vertex graph G and a hypercube it is NP-complete to determine whether G is embeddable in the hypercube with unit congestion, but G can be embedded with unit congestion in $Q(6\lceil \log N \rceil)$ if $\Delta(G) \le 6\lceil \log N \rceil$. They posed the question of whether G can be embedded with unit congestion in a hypercube of dimension less than $6\lceil \log N \rceil$. We answer the question by proving the following theorem.

Theorem 4.1 *Every N-vertex graph G can be embedded with unit congestion in $Q(2\lceil \log N \rceil)$ if $\Delta(G) \le 2\lceil \log N \rceil$.*

The basic idea of the embedding is quite simple. We adopt a plain labeling of vertices and a simple routing for edges, and the embedding can be constructed in polynomial time. We do not know whether G can be embedded with unit congestion in a hypercube of dimension less than $2\lceil \log N \rceil$. However, we can show that some graphs can be embedded with unit congestion in hypercubes of asymptotically smaller dimensions. More precisely, we can easily show by combining the results of Saad and Shultz [22] and Valiant [29] that every N-vertex tree T with $\Delta(T) \le 4$ can be embedded with unit congestion in a hypercube of dimension $\log N + O(1)$, and every N-vertex planar graph G with $\Delta(G) \le 4$ can be embedded with unit congestion in a hypercube of dimension $\log N + 2 \log \log N + O(1)$.

Bhatt, Chung, Leighton, and Rosenberg [2] showed that every N-vertex binary tree can be embedded in $Q(\lceil \log N \rceil)$ with dilation and congestion both $O(1)$. Although their embedding is optimal to within a constant factor, there is much room for reducing the dilation and/or congestion. They posed the question of finding a simple embedding of binary trees into hypercubes with smaller dilation and/or congestion. Monien and Sudborough [18] partially answer the question by proving that every N-vertex binary tree can be embedded in $Q(\lceil \log N \rceil)$ with dilation at most 5. We also partially answer the question by proving the following theorem.

Theorem 4.2 *Every N-vertex binary tree can be embedded in $Q(\lceil \log N \rceil)$ with congestion at most* 5.

Theorem 4.2 is the first result that shows a simple embedding of a binary tree into an optimal sized hypercube with explicit small congestion of 5. The embedding is quite simple. We use the postorder labeling of vertices and a greedy (shortest path) routing for edges, and the embedding can be constructed in polynomial time. It is interesting that such a simple embedding guarantees a small congestion of 5. We do not know an N-vertex binary tree that cannot be embedded in $Q(\lceil \log N \rceil)$ with unit congestion except $K_{1,3}$ (a complete bipartite graph). The author verified that every N-vertex binary tree except $K_{1,3}$ can be embedded in $Q(\lceil \log N \rceil)$ with unit congestion if $N \leq 16$. In this connection, based on some conjecture, Wagner [30] mentioned a heuristic algorithm which would embed every N-vertex binary tree into $Q(\lceil \log N \rceil)$ with dilation and congestion both at most 2.

The chapter is organized as follows. We prove Theorems 4.1 and 4.2 in Sections 4.2 and 4.3, respectively. In Section 4.4, we conclude with remarks on dilations of our embeddings and some other remarks.

4.2 General Graph Embedding

Let $V(G) = \{0, 1, \ldots, N-1\}$ and $n = \lceil \log N \rceil$. We assume that $\Delta(G) \leq 2n$. We construct an embedding $\langle \phi_1, \rho_1 \rangle$ of G into $Q(2n)$ with unit congestion. We define the labeling ϕ_1 in Section 4.2.1. In Section 4.2.2, we consider an arc coloring of a digraph associated with G. We define the routing ρ_1 in Section 4.2.3 based on the results in Section 4.2.2. We analyze the congestion of embedding $\langle \phi_1, \rho_1 \rangle$ in Section 4.2.4.

4.2.1 Labeling ϕ_1

The labeling $\phi_1 : V(G) \to V(Q(2n))$ is defined as follows. For each $u \in V(G)$, $\phi_1(u) = u(2^n + 1)$. That is, the binary representation of $\phi_1(u)$ is the concatenation of two copies of the binary representation of u.

4.2.2 Arc Coloring

In this section, we consider an arc coloring of a digraph associated with G which will be used to define routing ρ_1. The associated digraph D of G is the digraph obtained from G by replacing each edge e of G by two oppositely oriented arcs with the same ends as e. We denote the vertex set and arc set of D by $V(D)$ and $A(D)$, respectively. We denote an arc a by $[u, v]$ if u is the tail of a, and v is its head. Let $\Gamma_D^+(u)$ denote the set of arcs with tail u, and $d_D^+(u) = |\Gamma_D^+(u)|$. Let $\Gamma_D^-(u)$ denote the set of arcs with head u, and $d_D^-(u) = |\Gamma_D^-(u)|$. Since $\Delta(G) \leq 2n$, $d_D^+(u) \leq 2n$ and $d_D^-(u) \leq 2n$ for any $u \in V(D)$.

We construct a coloring C of the arcs of D with two colors $\{0, 1\}$ such that both of the following two conditions are satisfied. We denote by $C[u, v]$ the color of an arc $[u, v]$ assigned by C. Define that $X_C^0(w) = \{[w, x] \mid [w, x] \in \Gamma_D^+(w), C[w, x] = 0\}$, and $X_C^1(w) = \{[w, y] \mid [w, y] \in \Gamma_D^+(w), C[w, y] = 1\}$.

Condition 4.1 *For each edge $(u, v) \in E(G)$, $C[u, v] = 0$ if and only if $C[v, u] = 1$.*

Condition 4.2 *For any vertex $u \in V(D)$, $|X_C^0(u)| \leq n$ and $|X_C^1(u)| \leq n$*

Lemma 4.3 *There exists a 2-arc coloring of D satisfying Conditions 4.1 and 4.2.*

Proof It is well-known that G has an orientation D' such that $|d_{D'}^+(u) - d_{D'}^-(u)| \leq 1$ for any $u \in V(D')$. It follows that $d_{D'}^+(u) \leq n$ and $d_{D'}^-(u) \leq n$ for any $u \in V(D')$ since $\Delta(G) \leq 2n$. Moreover, for each $(u, v) \in E(G)$, exactly one of the associated arcs $[u, v]$ and $[v, u]$ of D is contained in $\Gamma_{D'}^+(u) \cup \Gamma_{D'}^-(u)$. Thus, $|\Gamma_D^+(u) \cap \Gamma_{D'}^+(u)| \leq n$ and $|\Gamma_D^+(u) - \Gamma_{D'}^+(u)| = |\Gamma_D^-(u) \cap \Gamma_{D'}^-(u)| \leq n$ for any $u \in V(D)$. For each vertex $u \in V(D)$, we assign color 0 to the arcs in $\Gamma_D^+(u) \cap \Gamma_{D'}^+(u)$, and color 1 to the arcs in $\Gamma_D^+(u) - \Gamma_{D'}^+(u)$. The resulting 2-arc coloring of D satisfies Conditions 4.1 and 4.2. □

4.2.3 Routing ρ_1

For two vertices w and w' of G, let $m(w, w')$ be the vertex of $Q(2n)$ labeled with $w2^n + w'$. There exists a 2-arc coloring C of D satisfying Conditions 4.1 and 4.2 by Lemma 4.3. For a vertex $w \in V(G)$, suppose that $X_C^0(w) = \{[w, x_1], [w, x_2], \ldots, [w, x_k]\}$, and $X_C^1(w) = \{[w, y_1], [w, y_2], \ldots, [w, y_l]\}$, where $k = |X_C^0(w)|$ and $l = |X_C^1(w)|$. $k \leq n$

and $l \leq n$ since C satisfies Condition 4.2. Let $Q^0_w(n)$ and $Q^1_w(n)$ be the n-dimensional subcubes of $Q(2n)$ induced by the vertices $\{w2^n + i \mid 0 \leq i \leq 2^n - 1\}$ and the vertices $\{i2^n + w \mid 0 \leq i \leq 2^n - 1\}$, respectively. Notice that $\phi_1(w) \in V(Q^0_w(n)) \cap V(Q^1_w(n))$ and that $m(w, w') \in V(Q^0_w(n)) \cap V(Q^1_{w'}(n))$. Since $Q^0_w(n)$ is n-connected, there exist k vertex-disjoint paths P_i in $Q^0_w(n)$ connecting $\phi_1(w)$ and $m(w, x_i)$ $(1 \leq i \leq k)$. Define that $P[w, x_i] = P_i$ $(1 \leq i \leq k)$. Also, since $Q^1_w(n)$ is n-connected, there exist l vertex-disjoint paths P'_j in $Q^1_w(n)$ connecting $\phi_1(w)$ and $m(y_j, w)$ $(1 \leq j \leq l)$. Define that $P[w, y_j] = P'_j$ $(1 \leq j \leq l)$.

Now we define the routing ρ_1. Let (u, v) be an edge of G. We may assume that $C[u, v] = 0$ and $C[v, u] = 1$ since C satisfies Condition 4.1. Define the path $\rho_1(u, v)$ connecting $\phi_1(u)$ and $\phi_1(v)$ in $Q(2n)$ as the concatenation of $P[u, v]$ connecting $\phi_1(u)$ and $m(u, v)$ in $Q^0_u(n)$ and $P[v, u]$ connecting $\phi_1(v)$ and $m(u, v)$ in $Q^1_v(n)$.

Notice that the embedding $\langle \phi_1, \rho_1 \rangle$ defined above can be constructed in polynomial time.

4.2.4 Congestion of $\langle \phi_1, \rho_1 \rangle$

Lemma 4.4 *The congestion of $\langle \phi_1, \rho_1 \rangle$ is one.*

Proof It suffices to show that $P[u, v]$ and $P[s, t]$ are edge-disjoint for any distinct arcs $[u, v], [s, t] \in A(D)$.

Case 1 $C[u, v] \neq C[s, t]$. We may assume without loss of generality that $C[u, v] = 0$ and $C[s, t] = 1$. Since $Q^0_u(n)$ and $Q^1_s(n)$ are edge-disjoint, and $P[u, v]$ and $P[s, t]$ are contained in $Q^0_u(n)$ and $Q^1_s(n)$, respectively, $P[u, v]$ and $P[s, t]$ are edge-disjoint.

Case 2 $C[u, v] = C[s, t]$. We assume that $C[u, v] = C[s, t] = 0$. The proof for the case when $C[u, v] = C[s, t] = 1$ can be accomplished by a similar argument, and is omitted.

Case 2.1 $u \neq s$. Since $Q^0_u(n)$ and $Q^0_s(n)$ are vertex-disjoint, and $P[u, v]$ and $P[s, t]$ are contained in $Q^0_u(n)$ and $Q^0_s(n)$, respectively, $P[u, v]$ and $P[s, t]$ are edge-disjoint.

Case 2.2 $u = s$. Since $[u, v], [u, t] \in X^0_C(u)$, $P[u, v]$ and $P[u, t]$ are edge-disjoint by definition.

□

4.3 Binary Tree Embedding

Let T be an N-vertex binary tree and $n = \lceil \log N \rceil$. We construct an embedding $\langle \phi_2, \rho_2 \rangle$ of T into $Q(n)$ with congestion at most 5. We define $\langle \phi_2, \rho_2 \rangle$ in Section 4.3.1. In Section 4.3.2, we show some lemmas on the postorder numbering. In Section 4.3.3, we analyze the congestion of $\langle \phi_2, \rho_2 \rangle$ based on the results of Section 4.3.2.

4.3.1 Embedding $\langle \phi_2, \rho_2 \rangle$

The embedding we propose here is quite simple. We choose a vertex of T with degree at most two as the root of T, and we suppose that T is a rooted tree. Without loss of generality, we assume that for each vertex u of T, the number of left descendants of u (i.e., the number of vertices of left subtree rooted at u) is not less than that of right descendants of u. Give each vertex of T a number from 0 through $N - 1$ according to the postorder numbering of T so that the left most leaf has the number 0.

We define the labeling $\phi_2 : V(T) \to V(Q(n))$ as follows. For each $u \in V(T)$, $\phi_2(u)$ is the vertex of $Q(n)$ labeled with the postorder number of u.

We define the routing ρ_2 as follows. Let (u, v) be an edge of T, and $\phi_2(u) < \phi_2(v)$. The path $\rho_2(u, v)$ connecting $\phi_2(u)$ and $\phi_2(v)$ in $Q(n)$ starts at $\phi_2(u)$, passes through i-edges in the increasing order of i such that the binary representations of $\phi_2(u)$ and $\phi_2(v)$ differ in the ith bit. Thus, ρ_2 is a greedy (shortest path) routing for edges.

Notice that the embedding $\langle \phi_2, \rho_2 \rangle$ defined above can be constructed in polynomial time.

In what follows, for each $u \in V(T)$, we denote the postorder number of u and $\phi_2(u)$ simply by u. In addition, if we denote an edge of T by (u, v), we assume that $u < v$.

4.3.2 Properties of Postorder Numbering

The following lemmas on the postorder numbering will be used in the next section to analyze the congestion of $\langle \phi_2, \rho_2 \rangle$.

Lemma 4.5 *For any distinct edges* $(u, v), (s, t) \in E(T)$ $(u \le s)$, $u < s < t \le v$ *or* $u < v \le s < t$.

Proof Since the vertices of T are labeled according to the postorder numbering, each $y \in V(T)$ is adjacent to at most one vertex with a label more than y. Thus, $u \ne s$ and we may assume that $u < s$. Define that $I = \{x \in V(T) \mid u < x < v\}$. I is the set of right descendants of v if u is the left child of v, and I is the empty set if u is the right child of v. It follows that any $x \in I$ is adjacent only to vertices of $I \cup \{v\}$. Thus, if $s \in I$ then $t \in I \cup \{v\}$. This means that $u < s < t \le v$. If $s \notin I$, we have $u < v \le s < t$ by the assumption that $u < s$ and the definition of I. □

Lemma 4.6 *For any distinct edges* $(u, v), (s, t) \in E(T)$ $(u < s < t \le v)$, $t - s \le s - u + 1$.

Proof Since $u < s < t \le v$, u is the left child of v and both s and t are right descendants of v. If s is the right child of t then $t - s = 1$ and the lemma is immediate. Thus, we assume that s is the left child of t. Let m_L and m_R be the numbers of left descendants and right descendants of t, respectively, and let w be the vertex with the minimum postorder number in the descendants of s. It follows that

$$w - u \ge 1. \tag{4.1}$$

Since $m_L - 1$ is the number of descendants of s and $m_L \ge m_R$,

$$s - w = m_L - 1 \ge m_R - 1. \tag{4.2}$$

Since s is the left child of t,

$$m_R = t - s - 1. \tag{4.3}$$

From (4.1), (4.2), and (4.3), we have $t - s \le s - u + 1$, as desired. □

4.3.3 Congestion of $\langle \phi_2, \rho_2 \rangle$

In this section, we show that the congestion of $\langle \phi_2, \rho_2 \rangle$ is no more than 5. We will prove this by a series of lemmas. Let $\mathrm{bit}(m, k)$ denote the number (0 or 1) in the kth bit $(k \ge 0)$ in the binary representation of a non-negative integer m. For each edge $(u, v) \in E(T)$ and an integer k $(0 \le k \le n - 1)$, define that $\mathrm{dir}((u, v), k) = \mathrm{bit}(v, k) - \mathrm{bit}(u, k)$. If some paths in $Q(n)$ contain an edge $d \in E(Q(n))$ then the paths are said to *share* d. We can easily see the following lemma from the definition of ρ_2.

Lemma 4.7 *For any distinct edges $(u, v), (s, t) \in E(T)$, $\rho_2(u, v)$ and $\rho_2(s, t)$ share a k-edge in $Q(n)$ if and only if the following three conditions are satisfied.*

Condition 4.3 $\mathrm{dir}((u, v), k) \neq 0$ *and* $\mathrm{dir}((s, t), k) \neq 0$.

Condition 4.4 *If $k < n - 1$, the $(n - k - 1)$-bit strings consisting of the $(k + 1)$st bit through the $(n - 1)$st bit in the binary representations of u and s are identical.*

Condition 4.5 *If $k > 0$, the k-bit strings consisting of the 0th bit through the $(k - 1)$st bit in the binary representations of v and t are identical.* □

Lemma 4.8 *For any distinct edges $(u, v), (s, t) \in E(T)$ such that*

$$u < s < t < v \text{ and } \mathrm{dir}((u, v), k) = \mathrm{dir}((s, t), k), \tag{4.4}$$

if $\rho_2(u, v)$ and $\rho_2(s, t)$ share a k-edge in $Q(n)$ then

$$t - s \leq 2^k, \text{ and} \tag{4.5}$$

$$v - u > 2^{k+1}. \tag{4.6}$$

Proof We have $\mathrm{bit}(u, k) = \mathrm{bit}(s, k) \neq \mathrm{bit}(v, k) = \mathrm{bit}(t, k)$ from (4.4) and Lemma 4.7 (Condition 4.3). Thus, $s - u < 2^k$ and $v - t \geq 2^{k+1}$ by Lemma 4.7 (Conditions 4.4 and 4.5). Therefore, we have (4.5) by Lemma 4.6, and (4.6) since $u < t$. □

Lemma 4.9 *For any distinct edges $(u, v), (s, t) \in E(T)$ such that*

$$u < s < t = v, \tag{4.7}$$

if $\rho_2(u, v)$ and $\rho_2(s, t)$ share a k-edge in $Q(n)$ then

$$t - s \leq 2^k. \tag{4.8}$$

Proof Since $t = v$, $\mathrm{bit}(u, k) = \mathrm{bit}(s, k) \neq \mathrm{bit}(v, k) = \mathrm{bit}(t, k)$ by Lemma 4.7 (Condition 4.3). Therefore, $s - u < 2^k$ by Lemma 4.7 (Condition 4.4). By Lemma 4.6, we have (4.8). □

Lemma 4.10 *For any distinct edges* $(u, v), (s, t) \in E(T)$ *such that*

$$u < s < t < v \text{ and } \mathrm{dir}((u, v), k) \neq \mathrm{dir}((s, t), k), \tag{4.9}$$

if $\rho_2(u, v)$ *and* $\rho_2(s, t)$ *share a* k*-edge in* $Q(n)$ *then*

$$t - s \leq 2^{k+1}. \tag{4.10}$$

Proof $s - u < 2^{k+1}$ by Lemma 4.7 (Condition 4.4). Thus, we have (4.10) by Lemma 4.6. □

Lemma 4.11 *For any distinct edges* $(u, v), (s, t) \in E(T)$ *such that*

$$u < v \leq s < t, \tag{4.11}$$

if $\rho_2(u, v)$ *and* $\rho_2(s, t)$ *share a* k*-edge in* $Q(n)$ *then*

$$v - u < 2^{k+1}. \tag{4.12}$$

Proof $s - u < 2^{k+1}$ by Lemma 4.7 (Condition 4.4). Since $v \leq s$, we have (4.12). □

Lemma 4.12 *Any distinct edges* $(u, v), (s, t) \in E(T)$ $(u < s)$ *satisfy exactly one of (4.4), (4.7), (4.9), and (4.11).*

Proof Immediate from Lemma 4.5. □

Lemma 4.13 *For any distinct edges* $(u, v), (s, t) \in E(T)$ $(u < s)$ *such that* $\rho_2(u, v)$ *and* $\rho_2(s, t)$ *share a* k*-edge in* $Q(n)$*,* (u, v) *and* (s, t) *satisfy either (4.4) or (4.7) if and only if* $\mathrm{dir}((u, v), k) = \mathrm{dir}((s, t), k) \neq 0$*, and* (u, v) *and* (s, t) *satisfy either (4.9) or (4.11) if and only if* $\mathrm{dir}((u, v), k) = 1$ *and* $\mathrm{dir}((s, t), k) = -1$.

Proof We first show the necessities. If (u, v) and (s, t) satisfy either (4.4) or (4.7) then $\mathrm{dir}((u, v), k) = \mathrm{dir}((s, t), k) \neq 0$ from the proofs of Lemmas 4.8 and 4.9. If (u, v) and (s, t) satisfy (4.9) then $\mathrm{dir}((u, v), k) = 1$ and $\mathrm{dir}((s, t), k) = -1$ by Lemma 4.7 (Conditions 4.3 and 4.4). Assume that (u, v) and (s, t) satisfy (4.11). If $k < n - 1$ then the $(n - k - 1)$-bit strings consisting of the $(k+1)$st bit through the $(n-1)$st bit in the binary representations of u, v, and s are identical by Lemma 4.7 (Condition 4.4). Thus, $\mathrm{dir}((u, v), k) = 1$ and $\mathrm{dir}((s, t), k) = -1$ by Lemma 4.7 (Condition 4.3).

The sufficiencies are immediate from Lemma 4.12 and the necessities. □

For distinct edges $e_1, e_2, \ldots,$ and e_l in T, suppose that $\rho_2(e_1), \rho_2(e_2), \ldots,$ and $\rho_2(e_l)$ share a k-edge $d \in E(Q(n))$. If $\mathrm{dir}(e_1, k) = \mathrm{dir}(e_2, k) = \ldots = \mathrm{dir}(e_l, k) \neq 0$ then $\rho_2(e_1), \rho_2(e_2), \ldots,$ and $\rho_2(e_l)$ are said to share d in the *same direction.*

Lemma 4.14 *For any distinct edges (u, v), (s, t), and (w, x) in T which are a matching, $\rho_2(u, v)$, $\rho_2(s, t)$, and $\rho_2(w, x)$ do not share an edge in the same direction.*

Proof We may assume without loss of generality that $u < s < w$. Assume that $\rho_2(u, v)$ and $\rho_2(s, t)$ share a k-edge $e \in E(Q(n))$ in the same direction. Since (u, v) and (s, t) are a matching of T, we have $u < s < t < v$ from Lemma 4.13. Thus, it follows from Lemma 4.8 that $t - s \leq 2^k$. If $\rho_2(s, t)$ and $\rho_2(w, x)$ share e in the same direction, we have $s < w < x < t$ from Lemma 4.13, and it follows from Lemma 4.8 that $t - s > 2^{k+1}$, a contradiction. □

Lemma 4.15 *For any distinct edges (u, v), (s, t), and (w, x) in T which are incident to a vertex, $\rho_2(u, v)$, $\rho_2(s, t)$, and $\rho_2(w, x)$ do not share an edge in the same direction.*

Proof Suppose that $\rho_2(u, v)$, $\rho_2(s, t)$, and $\rho_2(w, x)$ share an edge in the same direction. Then we have $v = t = x$ by Lemma 4.13. Therefore $u < v$, $s < v$ and $w < v$. This is a contradiction, however, since each $y \in V(T)$ is adjacent to at most two vertices with labels less than y by the definition of the postorder numbering. □

Let d be a k-edge of $Q(n)$. We define that

$$\begin{aligned} H^+(d) &= \{e \mid e \in E(T), \mathrm{dir}(e, k) = 1, \rho_2(e) \text{ contains } d\}, \\ H^-(d) &= \{e \mid e \in E(T), \mathrm{dir}(e, k) = -1, \rho_2(e) \text{ contains } d\}. \end{aligned}$$

Lemma 4.16 $|H^+(d)| \leq 3$ *and* $|H^-(d)| \leq 3$ *for any* $d \in E(Q(n))$. *That is, the congestion of $\langle \phi_2, \rho_2 \rangle$ is at most* 6.

Proof Suppose that d is a k-edge ($0 \leq k \leq n-1$). If all edges in $H^+(d)$ are incident to a vertex then $|H^+(d)| \leq 2$ by Lemma 4.15. We next consider the case that there are edges $(u, v), (s, t) \in H^+(d)$ $(u < s)$ which are a matching of T. Then we have $u < s < t < v$ by Lemma 4.13, and it follows from Lemma 4.8 that

$$v - u > 2^{k+1}. \tag{4.13}$$

Suppose that there exists an edge $(w, x) \in H^+(d) - \{(u, v), (s, t)\}$. By Lemma 4.14, (w, x) is adjacent to (u, v) or (s, t).

If (w, x) is adjacent to (u, v) then $x = v$ from Lemma 4.13. Thus we have $x - w \leq 2^k$ by Lemma 4.9 and (4.13). Since $t < v = x$, it follows from Lemma 4.13 that $w < s < t < x$ for (w, x) and (s, t). Thus, we have $x - w > 2^{k+1}$ from Lemma 4.8, which is a contradiction. Therefore, (w, x) is adjacent to (s, t), and $x = t$ from Lemma 4.13. In addition, (w, x) is the only edge in $H^+(d)$ adjacent to (s, t) by Lemma 4.15. Thus we conclude $|H^+(d)| \leq 3$.

Similarly, we can show that $|H^-(d)| \leq 3$. □

Lemma 4.17 *The congestion of* $\langle \phi_2, \rho_2 \rangle$ *is at most* 5.

Proof $|H^+(d)| \leq 3$ and $|H^-(d)| \leq 3$ for any $d \in E(Q(n))$ by Lemma 4.16. If $|H^+(d)| \leq 2$ and $|H^-(d)| \leq 2$ for any $d \in E(Q(n))$ then the lemma is immediate.

Suppose first that $|H^+(d)| = 3$ for a k-edge $d \in E(Q(n))$. Then $H^+(d)$ contains non-adjacent two edges from the proof of Lemma 4.16. Let (u, v) be one of such edges which satisfies (4.13). Then, we have $v - u > 2^{k+1}$. Let (s, t) be an edge in $H^-(d)$. It follows from Lemma 4.13 that we have either $u < s < t < v$ or $u < v \leq s < t$ for (u, v) and (s, t). However, if $u < v \leq s < t$ then $v - u < 2^{k+1}$ from Lemma 4.11, which is a contradiction. Thus, $u < s < t < v$ and we have

$$t - s \leq 2^{k+1} \tag{4.14}$$

by Lemma 4.10. Suppose (w, x) and (y, z) are any distinct edges in $H^-(d)$ $(w < y)$. We have $x - w \leq 2^{k+1}$ from (4.14). It follows that $x = z$, for otherwise $w < y < z < x$ from Lemma 4.13, and we have $x - w > 2^{k+1}$ by Lemma 4.8, which is a contradiction. Therefore, $|H^-(d)| \leq 2$ by Lemma 4.15.

Suppose next that $|H^-(d)| = 3$ for a k-edge $d \in E(Q(n))$. Then there exists an edge $(s, t) \in H^-(d)$ such that $t - s > 2^{k+1}$. Let (u, v) be an edge in $H^+(d)$. It follows from Lemma 4.13 that we have either $u < s < t < v$ or $u < v \leq s < t$ for (u, v) and (s, t). However, if $u < s < t < v$ then $t - s \leq 2^{k+1}$ from Lemma 4.10, which is a contradiction. Thus, $u < v \leq s < t$ and we have

$$v - u < 2^{k+1} \tag{4.15}$$

by Lemma 4.11. Suppose (w, x) and (y, z) are any distinct edges in $H^+(d)$ $(w < y)$. We have $x - w < 2^{k+1}$ from (4.15). It follows that $x = z$, for otherwise $w < y < z < x$ from Lemma 4.13, and we have $x - w > 2^{k+1}$ by Lemma 4.8, which is a contradiction. Therefore, $|H^+(d)| \leq 2$ by Lemma 4.15.

Thus, we conclude that the congestion of $\langle \phi_2, \rho_2 \rangle$ is at most 5. □

4.4 Concluding Remarks

Although $\langle \phi_1, \rho_1 \rangle$ may have a large dilation, we can also construct an embedding of G into $Q(2n)$ with dilation at most $2n + 2$ and unit congestion using a more sophisticated routing. It should be noted that the dilation of $\langle \phi_2, \rho_2 \rangle$ is at most the diameter of the hypercube since ρ_2 is a shortest path routing.

Our analysis of the congestion of $\langle \phi_2, \rho_2 \rangle$ is tight possible. That is, there exist binary trees for which the congestion of $\langle \phi_2, \rho_2 \rangle$ is exactly 5. For the tree shown in Figure 4.1, the image paths of five bold edges by ρ_2 share $(10000, 10100) \in E(Q(6))$. This is also true when we choose any vertex in the right subtree (represented as the gray triangle) as the root. Moreover, the same situation occurs if the root is not in the right subtree. Thus the congestion of $\langle \phi_2, \rho_2 \rangle$ for the tree is independent of the choice of the root.

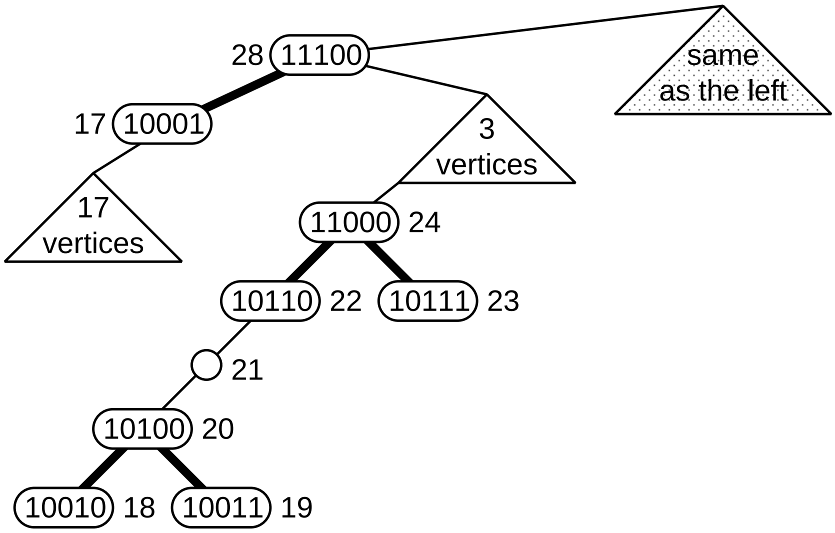

Figure 4.1: An example with 58 vertices of binary trees for which the congestion of $\langle \phi_2, \rho_2 \rangle$ is 5.

Chapter 5

Conclusion

In this thesis, we investigate the small congestion embeddings of graphs into grids and hypercubes.

In Chapter 2, we discuss the results on the proper-path-width which is used in Chapter 3. We show a necessary and sufficient condition for a graph with maximum vertex degree at most 3 and with proper-path-width at most 2. Based on the characterization, we give a practical linear time algorithm for computing a proper-path-decomposition with width at most 2 of a graph with maximum vertex degree at most 3.

We do not have any polynomial time algorithm to compute an optimal proper-path-decomposition of a given graph with bounded proper-path-width.

In Chapter 3, we show the complexity results on graph embeddings into grids. First, we prove that GRAPH k-LAYOUT is NP-complete for any fixed $k \geq 3$. We do not know the time complexity of GRAPH 2-LAYOUT. Next, we consider the problem of laying out a graph into a ladder, which is closely related with GRAPH 2-LAYOUT, and show a necessary and sufficient condition for a graph to be laid out into $L(\infty)$. Based on the characterization and the algorithm described in Chapter 2, we show a linear time algorithm which decides if a given graph G can be laid out into $L(\infty)$ and lays out G into $L(|V(G)|)$ whenever G satisfies the condition. In addition, we give the tight upper and lower bounds for the minimum area of a ladder into which an N-vertex graph G can be laid out.

It is still open (i) whether every N-vertex binary tree can be embedded into N-vertex grid with $O(1)$ congestion; (ii) whether any N-vertex binary tree can be embedded into

$N + o(1)$-vertex grid with unit congestion.

In Chapter 4, we show some results on graph embeddings into hypercubes. First, we prove that every N-vertex graph G can be embedded with unit congestion in $Q(2\lceil\log N\rceil)$ if $\Delta(G) \leq 2\lceil\log N\rceil$. Next, we prove that every N-vertex binary tree can be embedded in $Q(\lceil\log N\rceil)$ with congestion at most 5. The latter is the first result that shows a simple embedding of a binary tree into an optimal sized hypercube with explicit small congestion of 5. The embeddings proposed here are quite simple and can be constructed in polynomial time.

We do not know an N-vertex binary tree that cannot be embedded in $Q(\lceil\log N\rceil)$ with unit congestion except $K_{1,3}$.

Acknowledgment

I would like to express my gratitude to Professor Yoji Kajitani and Associate Professor Shuichi Ueno for many valuable comments and encouragement. I am very grateful to Dr. Atsushi Takahashi for helpful discussions. I am thankful to everyone who spent precious time together in Kajitani and Ueno laboratory at Tokyo Institute of Technology.

Bibliography

[1] F. Berman and L. Snyder. On mapping parallel algorithms into parallel architectures. *J. Parallel Distrib. Comput.*, 4:439–458, 1987.

[2] S. N. Bhatt, F. R. K. Chung, F. Thomson Leighton, and A. L. Rosenberg. Efficient embeddings of trees in hypercubes. *SIAM J. Comput.*, 21(1):151–162, 1992.

[3] S. N. Bhatt and F. Thomson Leighton. A framework for solving VLSI graph layout problems. *J. Comput. System Sciences*, 28:300–343, 1984.

[4] S. H. Bokhari. On the mapping problem. *IEEE Trans. Computers*, C-30(3):207–214, March 1981.

[5] M. Y. Chan. Embedding of grids into optimal hypercubes. *SIAM J. Comput.*, 20(5):834–864, October 1991.

[6] P. Czerwinski and V. Ramachandran. Optimal VLSI graph embeddings in variable aspect ratio rectangles. *Algorithmica*, 3:487–510, 1988.

[7] D. Dolev, T. Leighton, and H. Trickey. Planar embedding of planar graphs. In F. P. Preparata, editor, *Advances in Computing Research*, volume 2, pages 147–161. JAI Press, 1984.

[8] M. Formann and F. Wagner. The VLSI layout problem in various embedding models. In R. H. Möhring, editor, *Graph-Theoretic Concepts in Computer Science*, volume 484 of *Lecture Notes in Computer Science*, pages 130–139. Springer-Verlag, New York, Berlin, 1991.

[9] D. Gordon. Efficient embeddings of binary trees in VLSI arrays. *IEEE Trans. Comput.*, C-36(9):1009–1018, September 1987.

[10] P. Groeneveld. Necessary and sufficient conditions for the routability of classical channels. *Integration, the VLSI journal*, 16:59–74, 1993.

[11] R. Heckmann, R. Klasing, B. Monien, and W. Unger. Optimal embedding of complete binary trees into lines and grids. In G. Schmidt and R. Berghammer, editors, *Graph-Theoretic Concepts in Computer Science*, volume 570 of *Lecture Notes in Computer Science*, pages 25–35. Springer-Verlag, New York, Berlin, 1992.

[12] Y. M. Kim and T. H. Lai. The complexity of congestion-1 embedding in a hypercube. *J. of Algorithm*, 12:246–280, 1991.

[13] M. R. Kramer and J. van Leeuwen. The complexity of wire-routing and finding minimum area layouts for arbitrary VLSI circuits. In F. P. Preparata, editor, *Advances in Computing Research*, volume 2, pages 129–146. JAI Press, 1984.

[14] F. Makedon and I. H. Sudborough. On minimizing width in linear layouts. *Disc. Appl. Math.*, 23:243–263, 1989.

[15] F. S. Makedon, C. H. Papadimitriou, and I. H. Sudborough. Topological bandwidth. *SIAM J. Alg. Disc. Meth.*, 6(3):418–444, 1985.

[16] O. A. McBryan. An overview of message passing environments. *Parallel Computing*, 20(4):417–444, April 1984.

[17] B. Monien and I. H. Sudborough. Comparing interconnection networks. In M. P. Chytil, L. Janiga, and V. Koubek, editors, *Mathematical Foundations of Computer Science 1988*, volume 324 of *Lecture Notes in Computer Science*, pages 138–153. Springer-Verlag, New York, Berlin, 1988.

[18] B. Monien and I. H. Sudborough. Simulating binary trees on hypercubes. In J. H. Reif, editor, *VLSI Algorithms and Architectures*, volume 319 of *Lecture Notes in Computer Science*, pages 170–180. Springer-Verlag, New York, Berlin, 1988.

[19] S. F. Nugent. The iPSC/2 direct-connect communications technology. In *Third Conference on Hypercube Computers and Applications, Pasadena, CA*, January 1988.

[20] N. Robertson and P. D. Seymour. Graph minors. I. Excluding a forest. *J. Combinatorial Theory(B)*, 35:39–61, 1983.

[21] W. L. Ruzzo and L. Snyder. Minimum edge length planar embeddings of trees. In H. T. Kung, R. Spruoll, and G. steele, editors, *VLSI Systems and Computations*, pages 119–123. Computer Science Press, Rockville, MD, 1981.

[22] Y. Saad and M. H. Shultz. Topological properties of hypercubes. *IEEE Trans. Comput.*, 37(7):862–872, 1988.

[23] A. Takahashi, S. Ueno, and Y. Kajitani. Minimal acyclic forbidden minors for the family of graphs with bounded path-width. *Discrete Math.*, 127:293–304, 1994.

[24] A. Takahashi, S. Ueno, and Y. Kajitani. Minimal forbidden minors for the family of graphs with proper-path-width at most two. *IEICE Tras. Fundamentals*, E78-A(12):1828–1839, December 1995.

[25] A. Takahashi, S. Ueno, and Y. Kajitani. Mixed searching and proper-path-width. *Theoretical Computer Science*, 137:253–268, 1995.

[26] A. Takahashi, S. Ueno, and Y. Kajitani. On the proper-path-decomposition of trees. *IEICE Tras. Fundamentals*, E78-A(1):131–136, January 1995.

[27] S. Tayu and S. Ueno. On embedding binary trees in hypercubes. *IEICE Technical Report*, 92(236):77–84, September 1992.

[28] S. Tayu and S. Ueno. Efficient embeddings of binary trees with bounded pathwidth into paths and grids. *IPSJ SIG Notes*, 95(39):1–8, May 1995.

[29] L. G. Valiant. Universality considerations in VLSI circuits. *IEEE Trans. Comput.*, C-30(2):135–140, 1981.

[30] A. Wagner. Embedding all binary trees in the hypercube. *J. Parallel Distrib. Comput.*, 18:33–43, 1993.

[31] M. Yannakakis. A polynomial algorithm for the min-cut linear arrangement of trees. *J. Association for Computing Machinery*, 32(4):950–988, 1985.

Publications

Papers

[1] Akira Matsubayashi, Shuichi Ueno. On the complexity of embedding of graphs into grids with minimum congestion. *IEICE Transactions on Fundamentals*, E79-A(4):469–476, April 1996.

[2] Akira Matsubayashi, Shuichi Ueno. Small congestion embedding of graphs into hypercubes. submitted to *Networks*.

[3] Akira Matsubayashi, Shuichi Ueno. An efficient algorithm for computing proper-path-decomposition of width two. in preparation.

Presentations

[1] Akira Matsubayashi, Shuichi Ueno. Small congestion embedding of graphs into hypercubes. *IPSJ SIG Notes*, Vol.94, No.69, pp.57–64, July 1994.

[2] Akira Matsubayashi, Shuichi Ueno. On the complexity of embedding of graphs into grids with minimum congestion. *Proceedings of 51st IPSJ Annual Convention*, Vol.1, pp.65–66, September 1995. in Japanese.

[3] Akira Matsubayashi, Shuichi Ueno. An efficient algorithm for computing proper-path-decomposition of width two. *IPSJ SIG Notes*, Vol.95, No.109, pp.9–16, November 1995. in Japanese.