



Available online at <http://scik.org>

J. Math. Comput. Sci. 6 (2016), No. 2, 216-229

ISSN: 1927-5307

## LAYOUT OF PLANAR PRODUCTS

YULI CHEN

Fuzhou University Zhicheng College, Fuzhou, China

Copyright © 2015 Yuli Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Abstract.** Graph layout problems are a particular class of combinatorial optimization problems whose goal is to find a linear layout of an input graph in such a way that a certain objective function is optimized. In this paper, planar direct products and their stack and queue layouts are determined. The stack and queue numbers are directly motivated by VLSI layout problems.

**Keywords:** Planar direct product; Stack layouts; Queue layouts .

**2010 AMS Subject Classification:** 90C27.

### 1. Introduction

Graph layout problems are a particular class of combinatorial optimization problems whose goal is to find a linear layout of an input graph so that an objective function is optimized. A linear layout is simply a bijective mapping of the vertex set of a graph onto a chain of cardinality equal to the order of the graph. A large number of problems in different areas of science and engineering may be formulated as graph layout problems. These include parallel computer architecture, see [6] references therein, VLSI circuit layout, see [13] references therein, and information retrieval, see [11] references therein.

---

\*Corresponding author.

Received September 23, 2015

Layout problems for integrated circuitry arose as simplified models for actual layout problems in VLSI.

Given a set of modules, the VLSI layout problem consists in placing the modules on a board in a nonoverlapping manner and wiring together the terminals on the different modules according to a given wiring specification and in such a way that the wires do not interfere among them. There are two stages in VLSI layout: placement and routing. The placement problem consists in placing the modules on a board; the routing problem consists in wiring together the terminals on different modules that should be connected. A VLSI circuit can be modeled adequately by a graph, where the edges represent the wires and the vertices represent modules. This is certainly an oversimplification, but understanding and solving problems in this simple model help obtain better solutions for the actual model.

Various parameters that occur in graph layout problems are bandwidth, cutwidth, vertex separation and bipartition. There is a great amount of current literature in these direction. The reader is referred to [5].

## 2. Preliminaries

The work of this paper concentrates on the stack and queue layouts of planar products.

**Definition 1.1.** For a graph  $G$ , a *vertex ordering* or a *linear layout* is a bijection

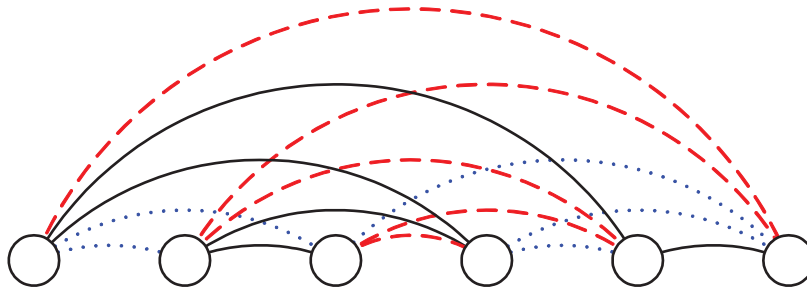
$$\sigma : V(G) \rightarrow \{1, 2, \dots, |G|\}.$$

For a layout  $\sigma$ , its *reverse layout* is

$$\sigma^{-1}(u) = |G| - \sigma(u) + 1, u \in V(G).$$

For a linear layout  $\sigma$  and an edge  $e = uv \in E(G)$ , if  $\sigma(u) < \sigma(v)$  then the *left end* of  $e$  is  $e^- = u$  and the *right end* of  $e$  is  $e^+ = v$ . For a pair of edges  $e, f \in E(G)$  one of the three situations will occur:

- (1)  $e$  and  $f$  *cross* if  $e^- < f^- < e^+ < f^+$ ;
- (2)  $e$  and  $f$  *nest* and  $f$  is *nested within*  $e$  if  $e^- < f^- < f^+ < e^+$ ;
- (3)  $e$  and  $f$  are *disjoint* if  $e^- < e^+ < f^- < f^+$ .

FIGURE 1. A 3-stack Layout of  $K_6$ .

It will be agreed in addition that edges with a common end do not cross, do not nest, and are not disjoint.

Let  $F \subseteq E(G)$ . If no two edges in  $F$  cross then  $F$  is called a *stack* in  $\sigma$ . In a traversal of  $\sigma$  from left to right, the left and right ends of the edges in a stack, are traversed in a *last in–first out* (usually abbreviated LIFO) order. Hence the natural term “stack” from data structures. If no two edges of  $F$  nest then  $F$  is called a *queue*. In a traversal of  $\sigma$  from left to right, the left and right ends of the edges in a stack, are traversed in a *first in–first out* (abbreviated FIFO) order. Hence term “queue” as in data structures.

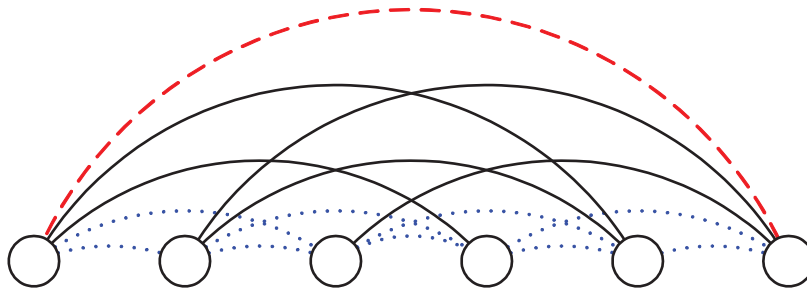
**Definition 1.2.** A linear layout  $\sigma$  of  $G$ , together with a partition  $\{E_1, E_2, \dots, E_k\}$  of  $E(G)$ , is called a *k-stack layout* of  $G$  if for each  $i$ ,  $1 \leq i \leq k$ ,  $E_i$  is a stack.

A 3-stack Layout of  $K_6$  is illustrated in Figure 1.

A linear layout  $\sigma$  of  $G$ , together with a partition  $\{E_1, E_2, \dots, E_k\}$  of  $E(G)$ , is called a *k-queue layout* of  $G$  if for each  $i$ ,  $1 \leq i \leq k$ ,  $E_i$  is a queue. A 3-queue Layout of  $K_6$  is illustrated in Figure 2.

A graph admitting a *k-stack* (respectively, *queue*) layout is called a *k-stack (queue) graph*. The *stack-number* (respectively *queue-number*) of a graph  $G$ , denoted by  $s(G)$  (respectively,  $q(G)$ ) is the minimum  $k$  such that  $G$  is a *k-stack* (respectively *k-queue*) graph. Observing that the vertices of the components  $G_1, G_2, \dots, G_k$  of a disconnected graph  $G$  can be grouped by components along the line, it follows that

$$s(G) = \max \{s(G_1), s(G_2), \dots, s(G_k)\}$$

FIGURE 2. A 3-queue Layout of  $K_6$ .

and

$$q(G) = \max \{q(G_1), q(G_2), \dots, q(G_k)\}.$$

Henceforth, consider only connected simple graphs.

A stack layout is sometimes called a *book embedding*, and stack-number, *book-thickness*, *fixed outer-thickness* or *page-number*. An  $n$ -book embedding of a graph  $G$  is an embedding of  $G$  in a book with  $n$  pages such that  $V(G)$  lie on the spine and each  $e \in E(G)$  embeds in a single page so that no two edge cross. The book thickness, or page number,  $p(G)$  of a graph  $G$  is the smallest number  $n$  of pages of a book so that  $G$  has an embedding in the book. Sometimes a restriction on the maximum number of edges on a page is also considered.

Consider a vertex ordering  $\sigma = \{V_1, V_2, \dots, V_n\}$  of a graph  $G$ . For each edge  $v_i v_j \in E(G)$ , let the width of  $v_i v_j$  in  $\sigma$  be  $|i - j|$ , the maximum width of an edge of  $G$  in  $\sigma$ . The bandwidth of  $G$ , denoted by  $bw(G)$ , is the minimum bandwidth over all vertex orderings of  $G$ .

### 3. Recent results in layout

As in [3], four types of embedding problems have been of interest in engineering.

The first type of problems are characterizations of graphs that can be embedded in books with a small number of pages. The second type of problems consider upper bounds on the number of pages required by graphs of bounded degree. For example, for  $d > 2$ , if is  $G$   $d$ -regular graphs

of order  $n$  then

$$s(G) \leq \min \left\{ \frac{n}{2}, O(dn^{1/2}) \right\}.$$

and there exist  $d$ -regular graphs with  $|G| = n$  that cannot be embedded in fewer than

$$\Omega \left( \frac{n^{1/2-1/d}}{\log^2 n} \right)$$

pages. The third type of problems involve near-optimal embeddings of various families of graphs, including trees, grids, X-trees, cyclic shifters, permutation networks, and complete graphs. For example, every  $d$ -ary tree of order  $n$  may be embedded in a book having one page, with page width

$$\left\lceil \frac{d}{2} \right\rceil \cdot \log n.$$

The fourth type of problems considers tradeoff between the number of pages and the widths of the pages. For example, every one-page embedding of the "ladder" graph with depth  $n$  requires width  $n/2$ , but there exist 2-page embeddings with page width 2 for the graph.

The main motivation of stack layout (book-embedding) and queue layout problem arose in numerous applications in VLSI design technology [3], in particular in electronic design automation (EDA).

One of these applications is the sorting with parallel stacks. The problem of realization of fixed permutations of  $\{1, 2, \dots, n\}$  with noncommunicating stacks was studied in [6,15]. In a realization, initially each number is pushed, in the order 1 to  $n$ , onto any one of the stacks. After all the numbers are on stacks, the stacks are popped to form the permutation. The bipartite graph  $G$  with  $V(G) = \{a_1, \dots, a_n, b_1, \dots, b_n\}$  and  $E(G) = [\{a_1, \dots, a_n\}, \{b_1, \dots, b_n\}]$  models the problem. A realization of a permutation  $\sigma$  on  $\{1, 2, \dots, n\}$  with  $k$  parallel stacks corresponds to stack layouts and queue layouts in the order  $a_1, \dots, a_n, b_{\sigma(1)}, \dots, b_{\sigma(n)}$ .

Another of these applications is the single-row routing.

In routing multilayer printed circuit boards (PCBs), a simplification may be obtained by a decomposition [13]. Circuit elements may be arranged in a regular grid, with wiring channels separating rows and columns of elements. The circuit net list may then be decomposed (additional dummy elements may be introduced) in such a way that every net connects elements in a single row or in a single column. The PCB can now be routed by routing each of its rows

and each of its columns independently. A variant in which a net running from the top of a row around to its bottom or change layers en route is not allowed was considered in [11]. This corresponds directly to a stack layouts problem for graphs of small degrees.

Study of fault-tolerant process or arrays is a third type of these applications. The Diogenes approach to the design of fault-tolerant arrays of identical processing elements (PEs, for short) uses “stacks of wires” to configure around faulty processing elements [2,12]. The processing elements are laid out in a (logical, if not physical) line, with some number of “bundles” of wires running above the line of the elements. Lines of processing elements are then scanned and faulty elements and working elements are determined. As each working element is encountered, it is connected to the bundles of wires through a network of switches, thereby the working element is connected to the working elements that have already been found, preparing it for eventual connection to those that will be found. To simplify the configuration process, each bundle is considered as behaving like a stack. After determining the faulty and working elements, line of processing elements are then scanned from right to left. As a working element of degree 1 is encountered, it is connected to line 1 in the bundle, simultaneously making lines 1 through  $d - 1$  “shift up” to “become” lines 2 through  $d$ ; switches disconnect the left parts of the lines from the right parts so that local connectivity remains the same. The bundle has thus behaved like a stack being pushed [3]. As a working element of degree at least 2 is encountered, it is connected to the stack/bundle in two stages. First it is connected to lines 1 and 2 of the bundle, simultaneously making lines 3 through  $d$  “shift down” to “become” lines 1 through  $d - 2$ ; again switches ensure that proper local connectivity is maintained. The bundle here behaves like a stack being twice popped. Second, the processing element pushes a connection onto the stack. In this case, pops amount to an element adopting two children that lie to its right in the line, while pushes amount to the element to be adopted by some higher level vertex that lies to its left. The process just described lays a  $d$ -ary tree out in preorder and, hence, uses at most  $d$  lines. It is clear from this procedure that the configuration process corresponds directly to the stack layouts problem.

These formulations of stack layout problem suggest at least two variants: fixed layout (as in sorting with parallel stacks and single-row routing); layout as part of the problem (as in the

construction of fault-tolerant processor arrays). Problems considered in the present paper are where placement of the vertices is not given beforehand.

The pinwheel graph is defined to be one with

$$\begin{aligned} V(P(n)) &:= \{a_i, b_i : 1 \leq i \leq n\}, \\ E(P(n)) &:= \{a_i b_i : 1 \leq i \leq n\} \cup \{a_i b_{n-i+1} : 1 \leq i \leq n\} \\ &\quad \cup \{a_i a_{i+1} : 1 \leq i \leq n\} \cup \{b_i b_{i+1} : 1 \leq i \leq n\} \end{aligned}$$

with addition modulo  $n$ . Clearly,  $P(3) = K_{3,3}$ , and for  $n \geq 3$ ,  $P(n)$  is not planar. It is not hard to see that  $s(P(n)) \leq 3$ .

If  $T$  is a  $d$ -ary tree of order  $n$  then  $s(T) = 1$  [3]. An  $X$ -tree with depth- $d$  admits a 2-stack layout. The definition of the  $n$ -cube is folklore. For  $n \geq 2$ ,  $s(Q_n) \leq n - 1$ . And, as a last illustrative example,  $s(K_n) = \frac{n}{2}$ .

Note also that every series-parallel graph admits a 2-stack layout.

The only graphs with  $s(G) = 0$  are discrete graphs (i.e., the complements of complete graphs). Graphs with  $s = 1$  are precisely connected outerplanar graphs, see [1,9] references therein.

**Lemma 1.1.**[9]  $s(G) = 1$  if and only if  $G$  is outerplanar.

Let  $G$  be a 1-stack graph with  $|G| = n$ . Then  $\|G\| \leq 2n - 3$ , since it has at most  $n$  edges for a hamilton cycle and at most  $n - 3$  edges in the interior of the cycle. A graph is called *subhamiltonian* if it is a subgraph of a hamiltonian graph. Graphs with  $s \leq 2$  are precisely subhamiltonian graphs.

**Lemma 1.2.**[9]  $s(G) \leq 2$  if and only if  $G$  is subhamiltonian.

Each planar graph has a 4-stack layout [9].

**Lemma 1.3.**[9] If  $G$  is a planar graph, then  $s(G) \leq 4$ .

If, in addition to planarity, a graph is also bipartite, then it has a 2-stack layout [9].

**Lemma 1.4.**[3] If  $G$  is a planar bipartite graph, then  $s(G) \leq 2$ .

The following result was obtained in [3].

**Lemma 1.5.**  $s(G) \leq 2$  if and only if  $G$  is a subgraph of a planar Hamiltonian graph.

Hence if  $G$  is not a planar, then  $s(G) \geq 3$ .

The extremal problems concerning the maximum number of edges in a graph if it admits a certain layout were considered in [1,4,8].

**Lemma 1.6.** If  $s(G) \leq s$  and  $|G| = n$  then  $\|G\| \leq (s+1)n - 3s$ , and this bound is tight for all even  $n \geq 4$  and all  $1 \leq s \leq n/2$ .

For the maximum number of edges in a  $k$ -queue graph, stremal graphs for  $k = 1$  was determined in [7,10]. The proof in [7] is based on a characterisation of 1-queue graphs as the arched levelled planar graphs. The proof in [10] is based on a relationship between queue layouts and “staircase covers of matrices”. These results have been summarised in [5], where the following was also included.

**Lemma 1.7.** A queue in a graph with  $|G| = n$  has at most  $2n - 3$  edges.

A direct generalization of this result is that every  $k$ -queue graph has at most  $k(2n - 3)$  edges [7]. The following improved upper bound was given in [10]. This bound is tight for all values of  $n$  and  $k$ .

**Lemma 1.8.** If  $G$  is graph with  $q(G) = k$  and  $|G| = n$ , then  $\|G\| \leq 2kn - k(2k + 1)$ . For every  $k$  and  $n \geq 2k$ , there exists graph with  $|G| = n$ ,  $q(G) = k$  and  $\|G\| = 2kn - k(2k + 1)$ .

Table 1 [5] summarizes some known bounds on the stack number and queue number of various classes of graphs. A blank entry indicates that a more general result provides the best known bound.

## 4. Layout of planar products

In this section we consider planarity criterion for direct products of graphs. In order to state and prove the main theorem of this section, we recall the concepts of a contraction and a minor of a graph.

A *contraction*  $f$  of a graph  $G$  is a partition  $\{V_1, V_2, \dots, V_r\}$  of  $V(G)$  such that for each  $i = 1, 2, \dots, r$ , the subgraph  $G|_{V_i}$  induced by  $V_i$  is connected. The resulting graph  $H$  with  $V(H) = \{V_1, V_2, \dots, V_r\}$  and edges  $V_i V_j$  for  $[V_i, V_j] \neq \emptyset$  where  $i \neq j$  is also called a *contraction* (graph) of  $G$ . The mapping  $f : G \rightarrow H$  is called a *contraction* (mapping). If there exists  $K \subseteq G$  and a contraction  $f : K \rightarrow H$ , then we call  $H$  a *minor* of  $G$  and is denoted  $H \leq G$ . Notice that the binary relation  $\leq$  over the class of finite undirected graphs is reflexive and transitive.



TABLE 1. Upper bounds on the stack number and queue number

Graph Family	Stack number	Queue number
n vertices	$\lfloor \frac{n}{2} \rfloor$	$\lfloor \frac{n}{2} \rfloor$
m edges	$O(\sqrt{m})$	$e\sqrt{m}$
proper minor-closed	bounded	
genus $\gamma$	$O(\sqrt{\gamma})$	
tree-width $w$	$w + 1$	$3^w 6^{(4^w - 3w - 1)/9} - 1$
tree-width $w$ , max. degree $\Delta$		$36\Delta w$
path-width $p$		$p$
band-width $b$	$b - 1$	$\lfloor \frac{b}{2} \rfloor$
track-number $t$		$t - 1$
toroidal	7	
planar	4	
bipartite planar	2	
2-trees	2	3
Halin	2	3
X-trees	2	2
outerplanar	1	2
arched levelled planar	2	1
trees	1	1

Denote by  $G \square H$  the *direct product* of graphs  $G$  and  $H$ . Thus

$$V(G \square H) = V(G) \times V(H)$$

$$W(G \square H) = \{(u, v)(x, y) : u = x, vy \in E(H) \text{ or } ux \in E(G), v = y\}.$$

**Theorem 4.1.** Let  $G$  and  $H$  be connected graphs. Then  $G \square H$  is planar if and only if

- (1)  $|G|, |H| \geq 3$  and  $\Delta(G), \Delta(H) \leq 2$ ; that is,  $G$  and  $H$  are a path or a cycle;
- (2)  $|H| = 2$  and  $G$  is an outerplanar graph; or
- (3)  $|H| = 1$  and  $G$  is planar.

**Proof.** Suppose that  $G$  and  $H$  are connected graphs with  $|G|, |H| \geq 3$ .

If  $G$  has a vertex of degree at least 3, then let this vertex be labeled  $x_1$  and let  $N(x_1) = \{x_2, x_3, x_4\}$ . Since  $H$  is connected and  $|H| \geq 3$ ,  $H$  has a path of length at least 3. Hence let  $P = abc \subseteq H$ . Let

$$J = (G \square H) |_{\{a,b,c\} \times \{x_1, x_2, x_3, x_4\}} - (a, x_1)(b, x_1) - (b, x_1)(c, x_1).$$

Then  $J \subseteq G \square H$ . Now

$$J / \{(a, x_i), (c, x_i) : i = 1, 2, 3, 4\} \simeq K_{3,3}.$$

Hence  $K_{3,3} \leq G \square H$  and therefore  $G \square H$  is not planar.

Suppose then that  $\Delta(G), \Delta(H) \leq 2$ . Then clearly  $G \square H$  is planar. Note that since  $G$  and  $H$  are both connected and of order at least 3, they both contain a path of order at least 3.

Hence it may be assumed that  $|H| \leq 2$ . The case that  $|H| = 1$  reduces to whether  $G$  is planar. Hence  $|H| = 2$  and  $H \simeq K_2$ .

If  $G$  is an outerplanar graph, then  $G \square H$  is planar.

Suppose that  $G$  is not outerplanar. Since  $G$  is planar (for otherwise  $G \square H$  is certainly not planar), hence  $K_4 \leq G$ . Then  $K_5 \leq K_4 \square K_2 \leq G \square H$ , and therefore  $G \square H$  is not planar.

This theorem will be used in determining layouts of planar products.

## 5. Stack and queue layouts on direct product

The direct product  $G \square H$  of two graphs  $G$  and  $H$  is defined on the direct product  $V(G) \times V(H)$  of the vertex sets of the factors. The edge set  $E(G \square H)$  is the set of all pairs  $[(u, v), (x, y)]$  of vertices for which either  $u = x$  and  $[v, y] \in E(H)$  or  $[u, x] \in E(G)$  and  $v = y$ .

For instance,  $C_4 = K_2 \square C_4$ .

**Theorem 5.1.** Let  $P$  be a path and  $T$  be a tree. Then

- (1)  $sn(P \square T) \leq 3$ .
- (2)  $sn(P \square T) = 1$  if and only if  $|P| = 1$ ,  $T$  is any tree; or,  $|P| = 2$  and  $T = P_1, P_2$ .
- (3)  $sn(P \square T) = 2$  if and only if  $|P| > 2$ ,  $T$  is a path with at least three vertices.
- (4)  $sn(P \square T) = 3$  if and only if  $|P| > 2$ ,  $T$  is a tree with  $\delta(T) \geq 3$ .

**Proof.** That  $sn(P_2 \square P_2) = 1$  is obvious;  $sn(P_n \square P_m) = 2(m, n \geq 3)$ , since the vertices of  $P_n \square P_m$  may be considered to lie in the line according to the ordering  $\sigma, \sigma^{-1}, \sigma, \sigma^{-1}, \dots$  from left to right, where  $\sigma$  is the layout of the path which gives the stack-number  $sn(P) = 1$  and  $\sigma^{-1} = |V| - \sigma(u) + 1$  (for all  $u \in V(P)$ ) is the reversed layout of  $\sigma$ , then the partition of edges is  $E_1 = \{uv \in E \mid \sigma(u) + \sigma(v) = 2km + 1, k = 1, 3, 5, \dots, \lceil n + 1/2 \rceil\}$  and  $E_2 = \{uv \in E \mid \sigma(u) + \sigma(v) = 2km + 1, k = 2, 4, 6, \dots, \lceil n + 1/2 \rceil\}$ , which is the minimal layout of  $P_n \square P_m$  in two stacks.

Let  $\varphi$  be the layout of  $T$  such that  $sn(T) = 1$  in  $\varphi$ . We consider the stack-number of  $P \square T$ . Let the vertices of  $P \square T$  lie in the line according to  $\varphi, \varphi^{-1}, \varphi, \varphi^{-1}, \dots$ , and let  $P$  be a path with length  $l \geq 3$ ,  $T$  be a tree and there is a vertex of degree 3 in  $T$ . We know that  $sn(T) = 1$ , for any  $x \in V(P)$ , denote  $E_3 = \bigcup_{x \in V(P)} E(x \square T)$ , then  $E_1, E_2, E_3$  become a 3-stack partition of  $E(P \square T)$ , so we can get  $sn(P \square T) \leq 3$ . According to the supposition above, the direct product of graph  $P$  and  $T$  is not a planar, since  $P \square T$  includes a  $K_{3,3}$  minor, then  $sn(P \square T) \geq 3$  using the lemma 2. On the other hand, we can find a layout such that  $sn(P \square T) \leq 3$  according to the method mentioned above. So we can get a 3-stack layout of  $P \square T$ , and the bound of  $sn(P \square T) \leq 3$  is tight.  $sn(P \square T) = 3$  when  $T$  is tree with  $\delta(T) \geq 3$ , because  $P \square T$  is not a planar when  $P$  is a path with at least three vertices and  $T$  is a tree with  $\delta(T) \geq 3$  according to theorem 8 in this paper.

Meanwhile, according to the method of layout,  $sn(P \square T) = 2$  if and only if any edge in  $(T, \varphi)$  can be added to one of the two stacks  $E_1, E_2$  to become one stack. If there is a vertex of degree 3 in  $T$ , denoted  $v$ , then  $v$  is adjacent to at least one vertex denoted  $u$  where  $u \notin \{x : \varphi(x) = \varphi(v) - 1, \varphi(v) + 1\}$ , then the edge  $uv$  will cross either stack  $E_1$  or  $E_2$ . So  $sn(P \square T) = 2$  if and only if  $T$  is a path of length  $l \geq 1$ . Then we proved this theorem.

**Theorem 5.2.** Let  $P$  be a path and  $G$  be a connected simple graph. Let  $sn(G) = s$ , then  $sn(P \square G) = f(s) \leq s + 2$ . The bound is tight.

- (1)  $f = s + 2$  when  $bw(G) \geq 2$ ;
- (2)  $f = s + 1$  when  $bw(G) = 1$ .

**Proof.**  $sn(P \square G) = 1$  when  $G$  is a graph with one vertex.  $sn(P \square G) = 2$  when  $G = P_n, (m, n > 2)$  or  $G = C_n$  using the theorems 1 and 2. Then we consider the stack number of  $P \square G$  where  $G$  is an arbitrary simple connected graph. Let  $\varphi$  be the layout of  $G$  with the minimal stack number  $sn(G) = s$  in  $\varphi, \{E_1, E_2, \dots, E_s\}$  is a partition of  $E(G)$ , and let the vertices of  $P \square G$  lie in the line according to  $\varphi, \varphi^{-1}, \varphi, \varphi^{-1}, \dots$ , then  $\{E_1, E_2, \dots, E_s, E_a, E_b\}$  is the partition of  $E(G)$ , where  $E_a, E_b$  is a partition of  $E(P \square P)$  using the method mentioned in the lemma above. So we get the upper bound of  $sn(P \square G)$ .

**Theorem 5.3.** Let  $P$  be a path and  $G$  be a connected simple graph. Then  $qn(P \square G) \leq qn(G) + 1$ .

**Proof.** Let  $\varphi$  be the vertex ordering in a  $qn(G)$ -queue layout of  $G$  with the partition  $Q_1, Q_2, \dots, Q_q$  of  $E(G)$ , we can make the vertex ordering of  $P \square G$  according to  $\varphi, \varphi, \dots, \varphi$ , then the  $P \square G$  admits  $(qn(G) + 1)$ -queue layout with partition  $Q_1, Q_2, \dots, Q_q, Q_0$  of  $E(P \square G)$  where  $Q_0$  is the  $P_m$ -edges in  $P \square G$ .  $qn(P \square G) = qn(G)$  when  $G$  is a graph with one vertex.

**Theorem 5.4.** Let  $C$  be a cycle, then  $sn(C_m \square C_n) \leq 3$ , when  $m$  or  $n$  is even.

**Proof.** For  $n = 2k$ . Let  $\varphi$  be the layout of  $C_m$  such that  $sn(C_m) = 1$  in  $\varphi$ . We consider the stack-number of  $C_m \square C_n$ . Let the vertices of  $C_m \square C_n$  lie in the line according to  $\varphi, \varphi^{-1}, \varphi, \varphi^{-1}, \dots$ , then the partition of edges is  $E_1 = \{uv \in E \mid \sigma(u) + \sigma(v) = 2km + 1, k = 1, 3, 5, \dots, \lceil n + 1/2 \rceil\}$  and  $E_2 = \{uv \in E \mid \sigma(u) + \sigma(v) = 2km + 1, k = 2, 4, 6, \dots, \lceil n + 1/2 \rceil\}$ ,  $E_3 = \bigcup_{x \in V(C_m)} E(x \square C_m)$ . So we can get a 3 – stack layout of  $C_m \square C_n$ , and the bound of  $sn(C_m \square C_n) \leq 3$  is tight.

**Theorem 5.5.** Let  $C_{2m}$  be a even cycle and  $G$  be a connected simple graph, then  $sn(C_{2m} \square G) \leq sn(G) + 2$ .

**Proof.** Let  $\varphi$  be the layout of  $G$  with the minimal stack number  $sn(G) = s$  in  $\varphi, \{E_1, E_2, \dots, E_s\}$  is a partition of  $E(G)$ , and let the vertices of  $C_{2m} \square G$  lie in the line according to  $\varphi, \varphi^{-1}, \varphi, \varphi^{-1}, \dots$ , then  $\{E_1, E_2, \dots, E_s, E_a, E_b\}$  is the partition of  $E(G)$ , where  $E_a, E_b$  is a partition

of  $E(C_{2m} \square G)$  using the method mentioned in the theorem above. So we get the upper bound of  $sn(C_{2m} \square G)$ .

**Theorem 5.6.** Let  $C$  be a cycle and  $G$  be a connected simple graph. Then  $qn(C \square G) \leq qn(G) + 2$ .

**Proof.** Let  $\varphi$  be the vertex ordering in a  $qn(G)$ -queue layout of  $G$  with the partition  $Q_1, Q_2, \dots, Q_q$  of  $E(G)$ , we can make the vertex ordering of  $C \square G$  according to  $\varphi, \varphi, \dots, \varphi$ , then the  $C \square G$  admits  $(qn(G) + 2)$ -queue layout with partition  $Q_1, Q_2, \dots, Q_q, Q_0$  of  $E(C \square G)$  where  $Q_0$  is the  $C_m$ -edges in  $C \square G$ , and  $qn(Q_0) = 2$ .  $qn(C \square G) = qn(G)$  when  $G$  is a graph with one vertex.

### Conflict of Interests

The authors declare that there is no conflict of interests.

### Acknowledgements

This work was supported by the Foundation of Fujian Education Bureau (No. JA14357).

### REFERENCES

- [1] F.R. Bernhart, and P.C. Kainen, The book thickness of a graph, *J. Combin. Theory*, 27(3), (1979), 320–331.
- [2] F.R.K. Chung, F.T. Leighton, and A.L. Rosenberg, DIOGENES-A methodology for designing fault-tolerant processor arrays, *13th Internat. Conf. on Fault-Tolerant Computing*, (1983), 36–32.
- [3] F.R.K. Chung, F.T. Leighton, and A.L. Rosenberg, Embedding graphs in books: a layout problem with applications to VLSI design, *Siam J. Alg. Disc. Meth*, 8(1), (1987), 33–58.
- [4] G.A. Cottafava, and O. D’antona, Book-thickness and book-coarseness of graphs, *In Proc. 5th International Symp. on Network Theory (Sarajevo)*, (1984), 337–340.
- [5] V. Dujmović, and D.R. Wood, On linear layouts of graphs, *Discrete Mathematics and Theoretical Computer Science*, 6, (2004), 339–358.
- [6] S. Even, and A. Itai, Queues, tacks, and graphs, *Theory of Machines and Computations*, (Z. Kohavi and A. Paz, eds), Academic Press, New York, 71–86, 1971.
- [7] L.S. Heath, and A.L. Rosenberg, Laying out graphs using queues, *SIAM J. Comput.*, 21(5), (1992), 927–958.
- [8] C.D. Keys, Graphs critical for maximal bookthickness, *Pi Mu Epsilon J.*, 6, (1975), 79–84.
- [9] S. Overbay, Graphs with small book thickness, *Missouri J. Math. Sci.*, 19(2), (2007), 121–130.
- [10] S.V. Pemmaraju, Exploring the powers of stacks and queues via graph layouts, PhD thesis, Virginia Polytechnic Institute and State University, USA, 1992.

- [11] R. Raghavan, and S. Sahni, Single row routing, *IEEE Trans. Comput.*, 32, (1983), 209–220.
- [12] A.L. Rosenberg, The Diogenes approach to testable fault-tolerant arrays of processors, *IEEE Trans. Comput.*, 32, (1983), 902–910.
- [13] H.C. So, Some theoretical results on the routing of multilayer printed-wiring boards. *IEEE Intl. Symp. on Circuits and Systems*, (1974), 296–303.
- [14] M.M. Syslo, Characterizations of outerplanar graphs, *Discrete Math.*, 26, (1979), 47–53.
- [15] R.E. Tarjan, Sorting using networks of queues and stacks, *J. Assoc. Comput. Mach.*, 19, (1972), 341–346.