A-007

Stack-queue mixed layouts of graph subdivisions

Hikoe Enomoto[1] **Miki Miyauchi**[2]

[1]Graduate School of Economics, Waseda University
(1-6-1, Nishi-Waseda, Shinjuku-ku, Tokyo, 169-8050 Japan
eno3632w@ruri.waseda.jp)

[2]NTT Communication Science Laboratories, NTT Corporation
(3-1, Morinosato Wakamiya, Atsugi-shi, Kanagawa Pref., 243-0198 Japan
miyauchi.miki@lab.ntt.co.jp)

Abstract This paper studies the problem of stack-queue mixed layouts of graph subdivisions. Dujmović and Wood showed that for every integer $s, q > 0$, every graph G has an s-stack q-queue subdivision layout with $4\lceil\log_{(s+q)q} \mathrm{sn}(G)\rceil$ (resp. $2+4\lceil\log_{(s+q)q} \mathrm{qn}(G)\rceil$) division vertices per edge, where $\mathrm{sn}(G)$ (resp. $\mathrm{qn}(G)$) is the stack number (resp. queue number) of G. This paper improves these results by showing that for every integer $s, q > 0$, every graph G has an s-stack q-queue mixed subdivision layout with $2\lceil\log_\alpha \mathrm{sn}(G)\rceil + 2$ (resp. $2\lceil\log_\alpha \mathrm{qn}(G)\rceil + 4$) division vertices per edge, where α is a function of s and q satisfying $\alpha > \sqrt{(s+q)q}$.

Keywords graph layout, number of subdivisions of graphs, stack layout of graphs, queue layout of graphs, stack-queue mixed layout of graphs.

1 Introduction

A vertex ordering of a graph G is a total order of the vertex set $V(G)$. In a vertex ordering $<$ of a graph G, let $L(e)$ and $R(e)$ denote the endpoints of each edge $e \in E(G)$ such that $L(e) < R(e)$. Consider two edges $e, f \in E(G)$. If $L(e) < L(f) < R(e) < R(f)$ then e and f *cross*, and if $L(e) < L(f) < R(f) < R(e)$ then e and f *nest*. A *stack* (resp. *queue*) is a set of edges $E \subset E(G)$ such that no two edges in E cross (nest). Observe that when traversing the vertex ordering, edges in a stack (queue) appear in LIFO (FIFO) order - hence the names. Note that if two edges of a graph share a vertex, then they neither cross nor nest.

For an integer $k > 0$, a *k-stack (queue) layout of G* consists of a vertex ordering of G and a partition $\{E_i \mid 1 \leq i \leq k\}$ of $E(G)$, such that each E_i (which is called a *page*) is a stack (queue). The *stack number* sn(G) of a graph G is the minimum k such that there is a k-stack layout of G. The *queue number* qn(G) of a graph G is the minimum k such that there is a k-queue layout of G.

Applications and results regarding stack and queue layouts can be found in [3, 7, 9] etc. Stack layouts of graphs can be regarded as book embeddings of graphs. Thus, results for book embedding problems ([1, 2, 4, 5, 6, 8, 10] etc.) also hold true for stack layout problems.

Stack and queue layouts are generalized through the notion of a mixed layout. Here each edge of a graph is assigned to a stack or to a queue that is defined with respect to a common vertex ordering. Such a layout is called an s-stack q-queue layout, if there are s stacks and q queues, Part of the motivation for studying stack-queue mixed layouts is that they model the double-ended queue (dequeue) data structure, since a dequeue may be simulated by two stacks and one queue.

This paper studies stack-queue mixed layouts of graph subdivisions. Every graph can be embedded in a stack-queue mixed layout with the limited number of stacks and queues by subdividing edges. Therefore, the next goal is to reduce the number of subdivisions as much as possible when we subdivide each edges. Thus,it is interesting to determine the minimum number of division vertices in a stack-queue mixed layout of a subdivision of a given graph.

Dujmović and Wood [3] showed the following theorem:

Theorem 1 (Dujmović and Wood [**3**])

(1) *For all integers $s \geq 1$ and $q \geq 1$, every graph G has an s-stack q-queue mixed subdivision with at most* $4\lceil \log_{(s+q)q} \mathrm{sn(G)} \rceil$ *division vertices per edge.*

(2) *For all integers $s \geq 1$ and $q \geq 1$, every graph G has an s-stack q-queue mixed subdivision with at most* $2 + 4\lceil \log_{(s+q)q} \mathrm{qn(G)} \rceil$ *division vertices per edge.*

Let $\alpha = \frac{s+q-1+\sqrt{(s+q-1)^2+4q}}{2}$ be the positive root of $X^2 - (s+q-1)X - q = 0$, and β be the negative root. Define

$$h(n) = \min\{k \mid \frac{\alpha^{k+1} - \beta^{k+1} + \alpha^k - \beta^k}{\alpha - \beta} \geq n\}.$$

The following theorem improves Theorem 1, because $\alpha > \sqrt{(s+q)q}$ and $h(n) \leq \lceil \log_\alpha n \rceil$.

Theorem 2 *For all integers $s \geq 1$ and $q \geq 1$, every graph G has s-stack q-queue mixed subdivisions with at most* $2h(\mathrm{sn}(G))+2$ *and* $2h(\mathrm{qn}(G))+4$ *division vertices per edge, respectively.*

We prove Theorem 2 by constructing a 1-stack 1-queue mixed layout of a newly defined (s,q)-ary tree. This construction method is not a direct extension of the one given in Dujmović and Wood [3], because not all leaves are laid out at the right end in our layout.

2 Newly defined (s,q)-ary tree and its 1-stack 1-queue mixed layout

For all integers $s, q > 0$, consider a rooted tree (named (s,q)-ary tree T) directed from the root to leaves, which has two types of edges called stack-type edges and queue-type edges as follows. In the tree T, all leaves are at the same depth h. Note that every node except the root has indegree 1 in a rooted tree. The root has $s+q$ outgoing edges, s of them are stack-type and q of them are queue-type. For every non-leaf node of T, if the type of incoming edge is stack, then it is incident with $s-1$ outgoing stack-type edges and q outgoing queue-type edges and if the type of incoming edge is queue, then it is connected with s outgoing stack-type edges and q outgoing queue-type edges.

Then, we symbolize the node set $V(T)$ and an edge set $E(G)$ of this (s,q)-ary tree T with depth h as follows:

$$V(T) = \{a_1 a_2 \cdots a_k \mid 0 \leq k \leq h, 0 \leq a_i < s+q, a_{i+1} \neq a_i \text{ if } a_i < s\}.$$

$$E(T) = \{(v, va) | v, va \in V(T), 0 \leq a < s+q\}.$$

For a string

$$a = a_1 \cdots a_h,$$

let $a(i)$ be the string consisting of the first i digits of a,

$$a(i) = a_1 \cdots a_i$$

and $a(0)$ be the empty string ϵ.

Then, for two strings $a, b \in R$, let define a lexicographic-like ordering $a \prec_R b$ of a string set R, which has only stack-type edges when one of the following conditions holds where $a \neq b(= b_1 ... b_\ell)$,

- There exists an integer i where $a = b(i)$ holds.
- There exists an integer i where $a(i) = b(i)$ and $a_{i+1} < b_{i+1}$ hold.

For example, if $s = 2$ and $h = 3$, then we have;

$$\epsilon \prec_R 0 \prec_R 01 \prec_R 010 \prec_R 1 \prec_R 10 \prec_R 101.$$

For a node v, define the path P_v which goes from the root to the node v. For a string $v = a_1 a_2 \cdots a_k$, let $w(v)$ be the number of queue-type edges included in the path P_v as follows.

$$w(v) = \sharp\{i \mid 1 \leq i \leq k, a_i \geq s\}$$

Note that $u \in R$ if and only if $w(u) = 0$.

Then if $w(v) > 0$, we can decompose a node (string) v uniquely as follows:

$$v = p(v)q(v)r(v)$$

Where,

$$p(v) \in V(T)$$

$$s \leq q(v) < s + q$$

$$r(v) \in R.$$

Note that $q(v)$ is the rightmost queue-type edge in the string of v. For two nodes $u, v \in V(T)(u \neq v)$, define the node ordering $u \prec v$ when one of the following three conditions $(R_1 - R_3)$ holds. Then, the node ordering $u \prec v$ gives a total order on $V(T)$.

- $w(u) < w(v)$ (R_1)
- $u, v \in R$ and $u \prec_R v$ (R_2)
- $w(u) = w(v) > 0$ and one of the following conditions holds: (R_3)
  - $p(u) \prec p(v)$
  - $p(u) = p(v)$ and $q(u) < q(v)$
  - $p(u) = p(v)$, $q(u) = q(v)$ and $r(u) \prec_R r(v)$

For example, if $s = 2, q = 1$ and $h = 3$, then the node set is as follows; also see Figure 1.

$$V(T) = \{\epsilon, 0, 1, 2, 01, 02, 10, 12, 20, 21, 22, 010, 012, 020, 021, 022,$$

$$101, 102, 120, 121, 122, 201, 202, 210, 212, 220, 221, 222\}.$$

Using this ordering, we have a 1-stack 1-queue mixed layout of T.

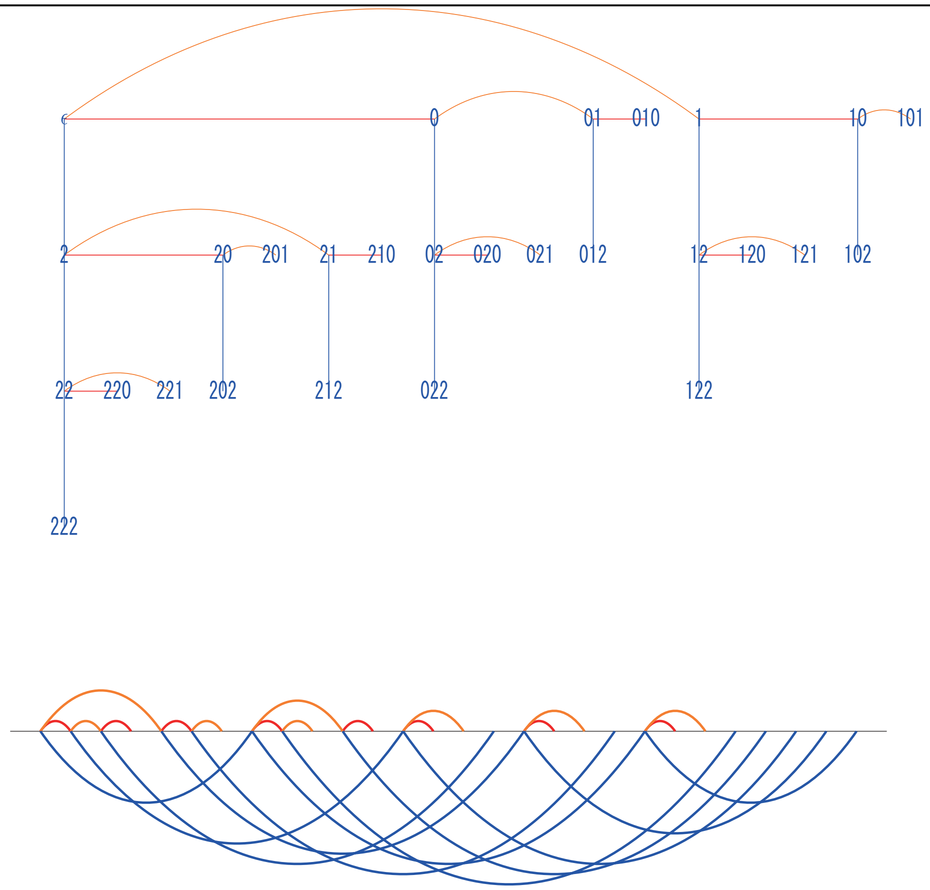

Figure 1: (2,1)-ary tree of height 3 and its 1-stack 1-queue mixed layout

Proposition 3 *Queue-type edges form a 1-queue and stack-type edges form a 1-stack.*

Proof. For two nodes $a, b \in T$, let $a = a_1 a_2 \cdots a_k$, $b = b_1 b_2 \cdots b_\ell$, $a' = a(k-1) = a_1 a_2 \cdots a_{k-1}$, and $b' = b(\ell - 1) = b_1 b_2 \cdots b_{\ell-1}$.

First, we will show that queue-type edges form a 1-queue, i.e., no two edges in the queue nest each other.

Assume that two edges (a, a') and (b, b') nest in the queue. We may assume that endpoints of the two edges are laid out from left to right in the order

$$a' \prec b' \prec b \prec a. \quad (*)$$

Because both two edges are in the queue page, we have $w(a) = w(a')+1$ and $w(b) = w(b')+1$. Then from the assumption

$$w(a') \leq w(b') \leq w(b) = w(b') + 1 \leq w(a) = w(a') + 1,$$

we have $w(a') = w(b')$ and $w(a) = w(b)$. Thus, if this assumption (*) is true, then both the first and the last inequlities have to hold the condition R_3. However, if $a' \prec b'$ then we have $a \prec b$ beacause $p(a) = a'$ and $p(b) = b'$ which contradicts the last inequality of the assumption (*).

Thus all queue-type edges form a 1-queue.

Next, we will show that stack-type edges form a 1-stack, i.e., no two edges in the stack cross each other.

Assume that two edges (a', a) and (b', b) cross in the stack. We may assume that endpoints of the two edges are laid out from left to right in the order

$$a' \prec b' \prec a \prec b.$$

Because both two edges are in the stack page, we have $w(a) = w(a')$ and $w(b) = w(b')$. Then from the assumption

$$w(a') \leq w(b') \leq w(a)(= w(a')) \leq w(b)(= w(b')),$$

we have $w(a') = w(a) = w(b') = w(b)$.

Consider the case of $w(a') = w(a) = w(b') = w(b) = 0$, i.e., $a, b \in R$ (condition R_2). Then we can rewrite the assumption as

$$a' \prec_R b' \prec_R a \prec_R b. \quad (**)$$

Suppose $a = b(i)$ for $i \leq \ell$. Then we have $i = \ell$ because the second inequality of the assumption (**) is

$$b' = b_1 b_2 \cdots b_{\ell-1} \prec_R a = b(i).$$

Then $a = b(i) = b(\ell) = b$ which contradicts the assumption.

Suppose $a(i) = b(i)$ and $a_{i+1} < b_{i+1}$ for $i < \ell$.

1. If $i < \ell - 1$, the second inequality of the assumption (**)

$$b' = b_1 b_2 \cdots b_{\ell-1} \prec_R a = a_1 a_2 \cdots a_k$$

   contradicts the assumption $a_{i+1} < b_{i+1}$.

2. If $i = \ell - 1$, then we have $a(\ell - 1) = b(\ell - 1)$ and $a_\ell < b_\ell$.

   (a) If $k \leq \ell$ then from the assumption (**)

$$b' = b(\ell - 1)(= a(\ell - 1)) \prec_R a_1 a_2 \cdots a_k,$$

   we have $\ell - 1 < k(\leq \ell)$. Thus we have $\ell = k$. Then we have $a' = a(k-1) = a(\ell - 1) = b(\ell - 1) = b'$ which contradicts the assumption $a' \neq b'$.

   (b) If $\ell < k$, then it contradicts the first inequality of the assumption (**) because,

$$a_1 a_2 \cdots a_{k-1} (= a(\ell - 1) a_\ell \cdots a_{k-1}) \prec_R b(\ell - 1) = a(\ell - 1).$$

   Consider the case of $w(a') = w(a) = w(b') = w(b) > 0$ (condition R_3). If, $w(a') = w(b') > 0$, then we have

$$p(a') = p(b'),\ q(a') = q(b'),\ r(a') \prec r(b'),\ r(a) = r(a')a_k \prec r(b) = r(b')b_\ell.$$

   Thus in this case, for the same reason as the case of $w(a') = w(a) = w(b') = w(b) = 0$ (i.e. $a, b \in R$), we can derive the contradiction.

Thus we have proved that stack-type edges form a 1-stack, i.e., no two edges in the stack cross each other.

■

3 Stack-queue mixed layouts of graph subdivisions

In this section, we prove Theorem 2.

The flow of the proof with respect to constructing a stack-queue mixed layout of graph subdivisions by using the stack number is as follows.

First we constract a special tree T called (s, q)-ary tree. Consider the stack layout S of graph G, which has $\mathrm{sn}(G)$ pages. Let call the maximum value among levels in the tree T "height." We use the (s, q)-ary tree T in which the number of leaves of the tree, which is one level lower than the height of T, is greater than or equal to $\mathrm{sn}(G)$.

Next, we constract a stack-queue layout of a given graph G by using the tree T and the stack layout S as follows.

Allocate all the vertices of G to the root of T. For example see Figure 4. Vertices x_i^* are allocated in the leftmost part of this layout which correspond to the root of T.

Allocate each page of the stack layout S of G to each leaf that is incident to a queue-type edge. For example, a stack which is drawn in Figure 2 is deformed as shown in Figure 3 and allocated in the rightmost part of this layout which correspond to a leaf of T as shown in Figure 4.

Connect vertices x and y in G along the (s, q)-ary tree T through the leaf corresponding to the page of S on which x and y are incident: That is, let $e = (x, y) \in S$. Then connect vertices $x, y \in G$ from x^* to y^* (in the root of T) along the (s, q)-ary tree T through $v_{x,e}$ and $v_{y,e}$ in a leaf of T. Where both $v_{x,e}$ and $v_{y,e}$ are the endvertices of the edge e in the page of the stack-queue mixed layout which is corresponding to the leaf of T where we construct a page contained e of S as shown in Figure 4.

In detail, if the m-th page of the stack layout S is as shown in Figure 2, we separate each edge independently as shown in Figure 3. Then attach these edges to the corresponding leaef of T that is incident to a queue-type edge.

We construct a layout of a subdivided graph G^* where each edge goes to the leaf along T that corresponds to the page of the stack layout S of G that contains the edge. That is, we arrange the vertices of G as in the stack layout S of G, and then put all subdivided vertices on the right of them with appropriate order as follows. See also Figure 4.

Split the root of the tree T into $\mid V(G) \mid$ vertices $\{x_i^*\}$. Also, split the leaf corresponding m-th stack page into $2 \mid E_m \mid$ vertices $\{x_{i,k}\}$, where E_m is the set of edges on the m-th stack page. Connect vertices $x_{i,k}$ and $x_{j,k}$ in G^* when there is an edge k of G that connects vertices x_i and x_j in S. And connect x_i^* and $x_{i,k}$ along the 1-stack 1-queue mixed layout of the tree T from root to the leaf m (see Figure 4). Each node in the level ℓ in T between the path from x_i^* and $x_{i,k}$ in G^* (which corresponds to the half of the edge k in G) become division vertices of G^*.

Here, the height of the tree is $h(\mathrm{sn}(G)) + 1$ (as to the proof, see Page 7). Then the number of subdivisions is "the height of the tree $\times 2$." Note that if the number of stack-type edges contained in the path from the root to a leaf is odd then we have to reverse the vertex ordering of $\{x_{i,k}\}$.

Remember that in section 2, we symbolize the node set $V(T)$ and an edge set $E(G)$ of this (s, q)-ary tree T with depth h as follows:

$$V(T) = \{a_1 a_2 \cdots a_k \mid 0 \le k \le h, 0 \le a_i < s + q, a_{i+1} \ne a_i \text{ if } a_i < s\}.$$

$$E(T) = \{(v, va) | v, va \in V(T), 0 \le a < s + q\}.$$

The pages used for edges obtained by the subdivision are defined as follows; for a division-edge e which corresponds to the stack-type edge (v, va) in the (s, q)-ary tree T, color the edge

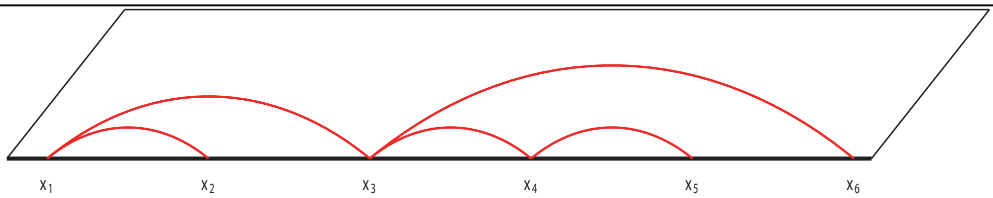

Figure 2: The mth page of the stack layout S of G

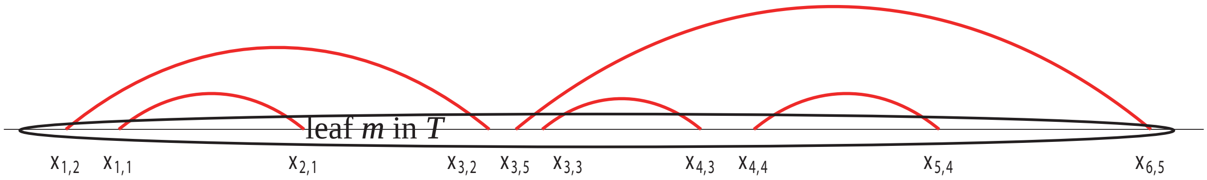

Figure 3: Edges attached to the leaf m of the layout of G^*

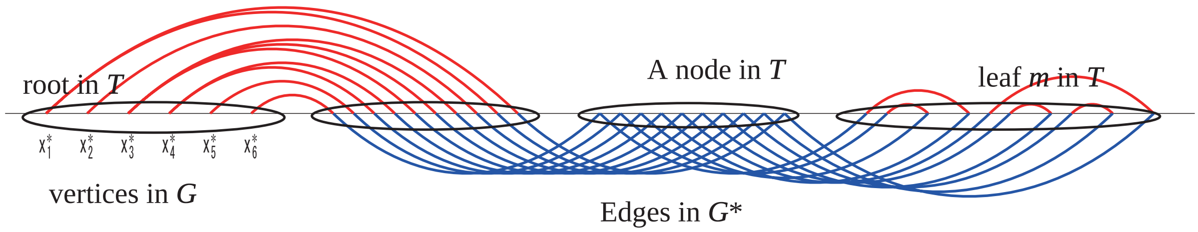

Figure 4: Layout of G^*

e as a.

The proof with respect to constructing a stack-queue mixed layout of graph subdivisions by using the queue number is also accomplished in the same way. Consider the queue layout Q of graph G, which has $\mathrm{qn}(G)$ pages. We use the (s, q)-ary tree T in which the number of leaves of the tree one level lower than T is greater than or equal to $\mathrm{qn}(G)$. Allocate all vertices of G to the root of T. Allocate each page of the queue layout Q of G to each leaf that is incident to a queue-type edge. Connect vertices x and y in G along T through the leaf corresponding to the page of Q on which x and y are incident. However, in the case of queue number, it is necessary to devise the layout so that the queue pages of Q attached on the leaves of T should not nest the queue edges laid out along the tree T in the layout of the graph subdivision.

Thus we devise the parts of the leaves of the 1-stack 1-queue mixed layout of T as follows. We focus on the fact that, on each page of 1-stack 1-queue mixed layout of T, there are no stack-type edges in the above area of pages which correspond to the leaves of T incident to queue-type edges.

(The reason is as follows. Suppose that there is a stack-type edge above the leaf v that is incident to a queue-type edge. Let the stack-type edge be e and $L(e) = u$. Then $R(e) = ua$ $(0 \leq a < s)$ and $L(e) \prec v \prec R(e)$.

Decompose u and v as follows. Note that $w(u) > 0$ because $w(v) > 0$ and $v \prec R(e)$.

$$u = p(u)q(u)r(u)$$
$$v = p(v)q(v)r(v).$$

As the leaf v is incident to a queue-type edge, we have

$$s \leq q(v) < s + q, r(v) = \epsilon.$$

From conditions $u \prec v$ and $v \prec ua$, we can derive the contradiction. That is, because of $ua = p(ua)q(ua)r(ua)$, we have $p(ua) = p(u)$, $q(ua) = q(u)$, $r(ua) = r(u)a$. Because of $w(u) = w(ua)$, and the definition of $u \prec v$ and $v \prec ua$, we have $w(u) = w(v) = w(ua)$. If $p(u) \prec p(v)$, then we have $p(ua) \prec p(v)$, and this contradicts $v \prec ua$. Thus we have $p(u) = p(v) = p(ua)$. If $q(u) < q(v)$, we have $q(ua) < q(v)$ and this contradicts $v \prec ua$. Therefore, we have $q(u) = q(v) = q(ua)$. In this case, because the condition $u \prec v$, we have $r(u) \prec_R r(v)$. Here, this contradicts $r(v) = \epsilon$. Thus we have proved the nonexistence of such an edge.)

Therefore, we take all leaves incident with queue-type edges to the right end of the stack-queue mixed layout by using stack-type edges on any stack-type pages. Then, there are no queue edges above leaves, and so we can construct a stack-queue mixed layout of graph subdivisions by attaching each page of Q to each leaf locally. In this layout, the height of the tree increases by 1, and so we need two extra subdivision points for each page.

In this section, we add $2h(\mathrm{sn}(G)) + 2$ or $2h(\mathrm{qn}(G)) + 4$ subdivisions to each edge of G to construct a stack-queue mixed layout of G^*. Where $h(n) + 1$ is the height of T, which is used to construct a stack-queue mixed layout.

The number of $h(n)$ is as follows. Let s_k be the number of stack-type edges and q_k be the number of queue-type edges between levels $k - 1$ and k. Then we have,

$$s_1 = s$$

$$q_1 = q$$

$$s_{k+1} = (s-1)s_k + sq_k$$

$$q_{k+1} = q(s_k + q_k).$$

Note that $s_h + q_h$ is the number of leaves in the height h of the tree T.

By solving the recursive relations, we get

$$s_k = \frac{(\alpha^k - \beta^k)s}{\alpha - \beta}, \quad q_k = \frac{(\alpha^{k+1} - \beta^{k+1}) - (s-1)(\alpha^k - \beta^k)}{\alpha - \beta}.$$

Here,

$$h(n) = \min\{k \mid s_k + q_k \geq n\}$$

is “the height of the (s, q)-ary tree -1”, where

$$s_k + q_k = \frac{\alpha^{k+1} - \beta^{k+1} + \alpha^k - \beta^k}{\alpha - \beta}.$$

Therefore we have proved Theorem 2.

Let $k = \lceil \log_\alpha n \rceil$. Then

$$s_k + q_k - \alpha^k = \frac{(\alpha^k - \beta^k)(\beta + 1)}{\alpha - \beta}.$$

Here, $s = (\alpha+1)(\beta+1) > 0$ and $\alpha\beta = -q \leq -1$, thus $-1 < \beta < 0$ and $\alpha > 1$. Then, $|\alpha| > |\beta|$. Therefore $s_k + q_k - \alpha^k > 0$. Thus $s_k + q_k > \alpha^k \geq n$, and so we have $h(n) \leq \lceil \log_\alpha n \rceil$.

Also, because of $\alpha > s + q - 1$, if $s \geq 2$ then

$$\alpha > s + q - 1 > \sqrt{(s+q)q}.$$

If $s = 1$, by using $q \geq 1$, we have $\alpha > \sqrt{(q+1)q}$.
Thus Theorem 2 improves Theorem 1.

The following below table is the table comparing the values of $\sqrt{(s+q)q}$ and α when $2 \leq s \leq 5$ and $q = 1$. Each value is rounded in the third decimal place.

	$\sqrt{(s+q)q}$	α
$s = 2$	1.73	2.41
$s = 3$	2	3.30
$s = 4$	2.24	4.24
$s = 5$	2.45	5.19

Also, the following below table is the table comparing the numbers of leaves in two different types of $(s, 1)$-ary trees. One was the number of leaves in the tree used by V. Dujmović and D. R. Wood [3], and the other is the number of leaves which are incident to queue-type edges in the tree we have newly defined in this paper.

	Depth of $(s, 1)$-ary tree	2	3	4	5	6	7
$s = 2$	Number of leaves in previous result	3		9		27	
	Number of leaves in our result	3	7	17	41	99	239
$s = 3$	Number of leaves in previous result	4		16		64	
	Number of leaves in our result	4	13	43	142	469	1549
$s = 4$	Number of leaves in previous result	5		25		125	
	Number of leaves in our result	5	21	89	377	1597	6765
$s = 5$	Number of leaves in previous result	6		36	6	216	
	Number of leaves in our result	6	31	161	836	4341	22541

4 Conclusion

In this paper, by constructing a 1-stack 1-queue mixed layout of a newly defined (s, q)-ary tree, we have proved that every graph G has s-stack q-queue mixed subdivisions with at most $2h(\mathrm{sn}(G)) + 2$ and $2h(\mathrm{qn}(G)) + 4$ division vertices per edge, respectively, where, $h(n) = \min\{k \mid s_k + q_k \geq n\}$. The base of the number of subdivision's logarithm of our result is greater than that of the mixed layout constructed by Dujmović and Wood [3]. Thus we have improved the Dujmović and Wood's result. This result improves the Dujmović and Wood's result [3]. We don't know whether this result is best possible or not.

References

[1] G. A. Atneosen, "On the embeddability of compacta in n-books," Ph.D. Dissertation, Michigan State University, 1968.

[2] F. Bernhart and P. C. Kainen, "The book thickness of a graph," J. Combi. Theory B, vol. 27, pp. 320-331, 1979.

[3] V. Dujmović and D. R. Wood, "Stacks, queues and tracks: layouts of graph subdivisions", Discrete Math. Theor. Comp. Sci., vol. 7, pp. 155-202, 2005.

[4] H. Enomoto and M. S. Miyauchi, "Embedding a graph into a three-page book with $O(m \log n)$ edge-crossings over the spine," SIAM J. Discrete Math., vol. 12, no. 3, pp. 337–341, 1999.

[5] H. Enomoto, M. S. Miyauchi, and K. Ota, “Lower bounds for the number of edge-crossings over the spine in a topological book embedding of a graph,” Discrete Appl. Math., vol. 92, pp. 149-155, 1999.

[6] H. Enomoto, T. Nakamigawa, and K. Ota, “On the pagenumber of complete bipartite graphs,” J. Combi. Theory B, vol. 71, pp. 111-120, 1997.

[7] L. S. Heath and A. L. Rosenberg, “Laying out graphs using queues.” SIAM J. Comput., vol. 21, no. 5, pp. 927-958, 1992.

[8] M. Miyauchi, “Topological book embedding of bipartite graphs,” IEICE Trans. Fundamentals, vol. E89-A, no. 5, pp. 1223-1226, 2006.

[9] M. Miyauchi, “Queue layout of bipartite graph subdivisions,” IEICE Trans. Fundamentals, vol. E90-A, no. 5, PP. 896-899, 2007.

[10] D. J. Muder, M. Weaver and D. B. West, “Pagenumber of complete bipartite graphs,” J. Graph Theory, vol. 12, no. 4, pp. 469-489, 1988.