# Transforming Stacks into Queues: Mixed and Separated Layouts of Graphs

**Julia Katheder** ✉ 🆔
Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany

**Michael Kaufmann** ✉ 🆔
Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany

**Sergey Pupyrev** ✉ 🆔
Menlo Park, CA, USA

**Torsten Ueckerdt** ✉ 🆔
Institute of Theoretical Informatics, Karlsruhe Institute of Technology, Germany

───── **Abstract** ─────

Some of the most important open problems for linear layouts of graphs ask for the relation between a graph's queue number and its stack number or mixed number. In such, we seek a vertex order and edge partition of $G$ into parts with pairwise non-crossing edges (a stack) or with pairwise non-nesting edges (a queue). Allowing only stacks, only queues, or both, the minimum number of required parts is the graph's stack number $\mathrm{sn}(G)$, queue number $\mathrm{qn}(G)$, and mixed number $\mathrm{mn}(G)$, respectively.

Already in 1992, Heath and Rosenberg asked whether $\mathrm{qn}(G)$ is bounded in terms of $\mathrm{sn}(G)$, that is, whether stacks "can be transformed into" queues. This is equivalent to bipartite 3-stack graphs having bounded queue number (Dujmović and Wood, 2005). Recently, Alam et al. asked whether $\mathrm{qn}(G)$ is bounded in terms of $\mathrm{mn}(G)$, which we show to also be equivalent to the previous questions.

We approach the problem by considering *separated* linear layouts of bipartite graphs. In this natural setting all vertices of one part must precede all vertices of the other part. Separated stack and queue numbers coincide, and for fixed vertex orders, graphs with bounded separated stack/queue number can be characterized and efficiently recognized, whereas the separated mixed layouts are more challenging. In this work, we thoroughly investigate the relationship between separated and non-separated, mixed and pure linear layouts.

## 1 Introduction

In this paper we study linear layouts of graphs[1]. That is, for a graph $G = (V, E)$, we consider a total order $\sigma$ of its vertex set $V$, while $\sigma$ defines the relative position of its edges. In particular, we investigate for a graph $G$ its well-known parameters stack number, $\mathrm{sn}(G)$, and its queue number, $\mathrm{qn}(G)$, as well as their common generalization called its mixed number, $\mathrm{mn}(G)$. Their precise definitions go as follows.

---

[1] Throughout, all graphs in this paper are simple, undirected, finite, and have at least one edge.

**Stack, Queue, and Mixed Numbers.** Given a vertex order $\sigma$, two edges $(u,v)$ and $(x,y)$ in $E$ are said to *cross* in $\sigma$ if any order of their endvertices symmetric to $u <_\sigma x <_\sigma v <_\sigma y$ is prescribed by $\sigma$. Roughly speaking, for the stack number of $G$, we want a vertex order $\sigma$ in which not many edges pairwise cross. Formally, for an integer $s \geq 1$, an *s-stack layout* of $G = (V, E)$ is a total order $\sigma$ of $V$ together with a partition of $E$ into subsets $E_1, \ldots, E_s$, called *stacks*, such that no two edges in the same stack cross. The minimum number $s$ of stacks needed for a stack layout of graph $G$ is called its *stack number* and denoted by $\mathrm{sn}(G)$. Stack numbers were first investigated by Bernhart and Kainen [6] in 1979, building on Kainen and Ollman [20, 24][2].

As a concept "dual" to $s$-stack layouts, for an integer $q \geq 1$, a *q-queue layout* of $G = (V, E)$ is a total order $\sigma$ of $V$ together with a partition of $E$ into subsets $E_1, \ldots, E_q$, called *queues*, such that no two edges in the same queue *nest*; that is, there are no edges $(u,v)$ and $(x,y)$ in a queue with $u <_\sigma x <_\sigma y <_\sigma v$ (or any symmetric order). The minimum number $q$ of queues needed for a queue layout of graph $G$ is called its *queue number* and denoted by $\mathrm{qn}(G)$. Queue numbers were introduced by Heath and Rosenberg [18] in 1992.

Stack and queue layouts are generalized through the notion of a *mixed* layout. For integers $s, q \geq 1$, an *s-stack q-queue layout* of $G = (V, E)$ is a total order $\sigma$ of $V$ together with a partition of $E$ into $s$ stacks and $q$ queues. The minimum value of $s + q$ needed for an $s$-stack $q$-queue layout of graph $G$ is called its *mixed number* and denoted by $\mathrm{mn}(G)$. Mixed layouts were already considered by Heath and Rosenberg [18] in 1992, while a thorough study started only recently [16, 22, 26]. In contrast to mixed layouts, we call a layout *pure* if only stacks or only queues are being used.

**Comparing Stack, Queue, and Mixed Numbers.** Besides their similar definitions, there are more similarities between *k-stack graphs*, *k-queue graphs*, and *k-mixed graphs*, where a $k$-(stack, queue, mixed) graph is defined as a graph admitting a $k$-(stack, queue, mixed) layout. For example, in each case we have sparse graphs with only $\mathcal{O}(kn)$ edges for $n$ vertices [6, 18, 25]. Moreover, stack, queue, and mixed numbers are all bounded for planar graphs [5, 8, 14, 30] and for bounded treewidth graphs [1, 9, 15, 28]. On the other hand, neither stack nor queue number is bounded for 3-regular graphs [3, 29]. Already in 1992, Heath, Leighton, and Rosenberg [17] investigated the relationship between stack and queue layouts; in particular whether stack layouts and queue layouts can be transformed into each other. They introduced the following fundamental question, which remains unanswered despite a wealth of studies on linear graph layouts.

▶ **Open Problem 1** (Heath, Leighton, and Rosenberg [17], see also [7, 11, 13]).
*Do graphs with bounded stack number have bounded queue number?*

We emphasize that a companion question (*"do graphs with bounded queue number have bounded stack number?"*) has been recently resolved in the negative by Dujmović et al. [7]. One attempt to resolve Open Problem 1 was made by Dujmović and Wood [11] who showed that the problem is equivalent to the question of whether bipartite 3-stack graphs have bounded queue number. Note that 1-stack graphs and 2-stack graphs are planar, and hence, they have a bounded queue number [8], but the question remains open for 3-stack graphs.

Turning to mixed layouts, it follows from the definition of the mixed number that for every graph $G$ we have $\mathrm{mn}(G) \leq \mathrm{sn}(G)$ and $\mathrm{mn}(G) \leq \mathrm{qn}(G)$. However for some graphs $G$, their mixed number $\mathrm{mn}(G)$ is strictly less than $\mathrm{sn}(G)$ and $\mathrm{qn}(G)$; for example, for the

---

[2] We remark that $s$-stack layouts are also known as $s$-page book embeddings, where stacks are then called pages. Similarly, the stack number $\mathrm{sn}(G)$ is sometimes called the book thickness or page number of $G$.
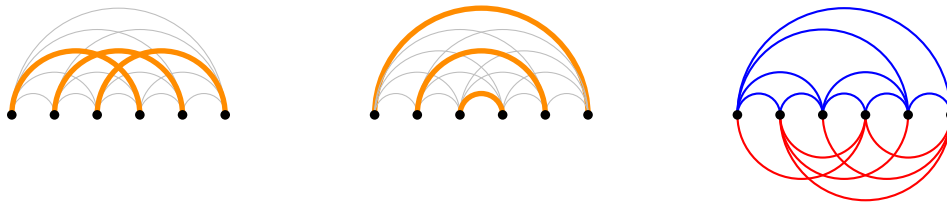
**Figure 1** Linear layouts of $K_6$ verifying that $\text{sn}(K_6) \geq 3$ due to a 3-twist (left), $\text{qn}(K_6) \geq 3$ due to a 3-rainbow (middle), and $\text{mn}(K_6) \leq 2$ due to a 1-stack 1-queue layout (right).

complete graph $K_6$ it holds that $\text{sn}(K_6) = \text{qn}(K_6) = 3$ but $\text{mn}(K_6) = 2$, as illustrated in Figure 1. In particular, graphs with bounded mixed number potentially form a strictly larger class than those of bounded stack number. Thus, the following problem of Alam et al. [22] asks for something (potentially) stronger than Open Problem 1.

▶ **Open Problem 2** (Alam et al. [22]).
*Do graphs with bounded mixed number have bounded queue number?*

As one of our results (cf. Theorem 4), we shall show that Open Problems 1 and 2 are in fact equivalent. That is, either both questions have a YES-answer or both have a NO-answer. In that sense, it is easier to prove a NO-answer by finding graphs of unbounded queue number but bounded mixed number (instead of bounded stack number). As a second result, also in Theorem 4, we shall prove that it indeed suffices to look for graphs of mixed number 2; specifically, graphs that admit 1-stack 1-queue layouts. That is, Open Problems 1 and 2 have a YES-answer if and only if all 1-stack 1-queue graphs have bounded queue number.

**Separated Linear Layouts.** Since Open Problems 1 and 2 are likely very challenging in their full generality, we investigate a special case for *separated mixed* layouts of bipartite graphs. A vertex order $\sigma$ of a bipartite graph $G$ with bipartition[3] $(A, B)$ is *separated* if all vertices in $A$ precede all vertices in $B$ (or vice versa). This naturally gives rise to separated $k$-stack, separated $k$-queue, and separated $s$-stack $q$-queue layouts of bipartite graphs $G$, as well as the corresponding separated stack number $\overline{\text{sn}}(G)$, separated queue number $\overline{\text{qn}}(G)$, and separated mixed number $\overline{\text{mn}}(G)$.

Observe that a separated $s$-stack $q$-queue layout can be transformed into a separated $q$-stack $s$-queue layout by reversing the vertex order of one of the parts. It follows that $\overline{\text{sn}}(G) = \overline{\text{qn}}(G)$ for every bipartite graph $G$. Furthermore, every separated vertex order $\sigma$ of $G = (V, E)$ with bipartition $(A, B)$ induces an injective mapping of the edges of $G$ to points of the $|A| \times |B|$ integer grid; see Figure 2. This grid is sometimes referred to as a *reduced adjacency matrix* of $G$. One can easily verify that a subset $S \subseteq E$ of edges forms a queue (resp. a stack) if and only if the corresponding points form a monotonically increasing (resp. decreasing) subset in the $|A| \times |B|$ grid. The stack-queue transformation mentioned above then corresponds to mirroring the grid along the $x$-axis or along the $y$-axis.

This separated setting has been widely studied (sometimes, implicitly) under different names, such as *2-track layouts* [10, 27] or *2-layer drawings* [2, 12, 23]. For the present paper, let us introduce the following special variant of Open Problem 2.

▶ **Open Problem 3.**
*Do bipartite graphs with bounded **separated** mixed number have bounded queue number?*

---

[3] $(A, B)$ is a bipartition if $A \cap B = \emptyset$, $A \cup B = V$, and both induced subgraph $G[A], G[B]$ are edgeless.

**(a)** Separated layout.          **(b)** 2-layer drawing.          **(c)** Grid representation.
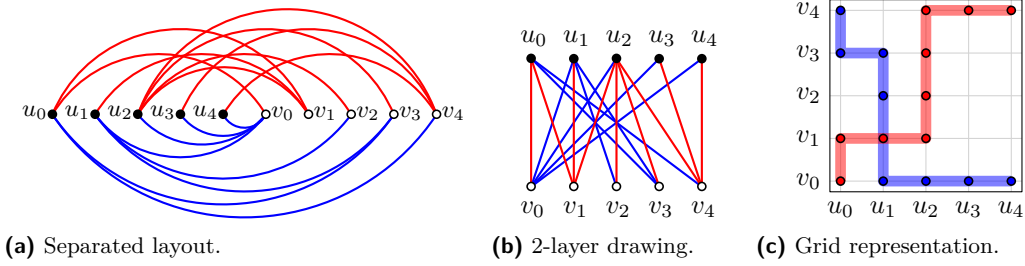
**Figure 2** Various representations of a separated 1-stack 1-queue bipartite graph.

**Our Contributions.** We make progress towards the resolution of Open Problems 1 and 2, as well as our new Open Problem 3. Each of these problems asks whether graphs with low stack number (Open Problem 1), low mixed number (Open Problem 2), or low separated mixed number (Open Problem 3) have also low queue number. Formally, we say that the *queue number is bounded by stack number* (similarly, *bounded by mixed number*, and *bounded by separated mixed number*) if for every graph $G$ with stack number $\mathrm{sn}(G)$ and queue number $\mathrm{qn}(G)$, it holds that $\mathrm{qn}(G) \leq f(\mathrm{sn}(G))$ for some function $f$ (independent of $G$). The question of whether such functions, $f$, exist are Open Problems 1–3.

Clearly, a function $f$ that bounds the queue number in terms of the stack number exists if and only if for all $k \in \mathbb{N}$ there is a constant $C = C_k$ such that every $k$-stack graph has queue number at most $C$. Surprisingly, it is enough to consider just one particular value for $k$. In fact, Dujmović and Wood [11] show that Open Problem 1 is equivalent to the existence of a constant $C$ such that every 3-stack graph has queue number at most $C$. (It is known that 1-stack and 2-stack graphs have queue number at most 2 [17] and at most 42 [4], respectively.)

Here, we extend the list to four equivalent statements, showing in particular that Open Problems 1 and 2 are equivalent, and that it is enough to consider 1-stack 1-queue graphs.

▶ **Theorem 4.** *The following are equivalent:*
**(1)** *every $s$-stack graph has queue number at most $f(s)$ for some function $f$*
    *("queue number is bounded by stack number");*
**(2)** *every 3-stack graph has queue number at most $C$ for some constant $C$;*
**(3)** *every 1-stack 1-queue graph has queue number at most $C$ for some constant $C$;*
**(4)** *every $s$-stack $q$-queue graph has queue number at most $f(s, q)$ for some function $f$*
    *("queue number is bounded by mixed number").*

Turning to separated layouts of (necessarily bipartite) graphs, our main contribution is also a list of four equivalent statements, one of which, namely (1), is Open Problem 3. Statements (3) and (4) show that it is enough to consider 1-stack $q$-queue graphs, respectively even only 1-stack 6-queue graphs, for solving Open Problem 3. Additionally, we discuss a natural approach for Open Problem 3 where we start with the grid representation of a separated $s$-stack $q$-queue layout, and then permute the rows and columns so as to obtain a separated $f(s, q)$-queue layout. Another surprising contribution is that it is always enough to apply the same permutation to the rows and the columns, which is statement (2).

▶ **Theorem 5.** *The following are equivalent:*
**(1)** *every separated $s$-stack $q$-queue graph $G$ has queue number at most $f(s, q)$ for some function $f$     ("queue number is bounded by mixed number");*
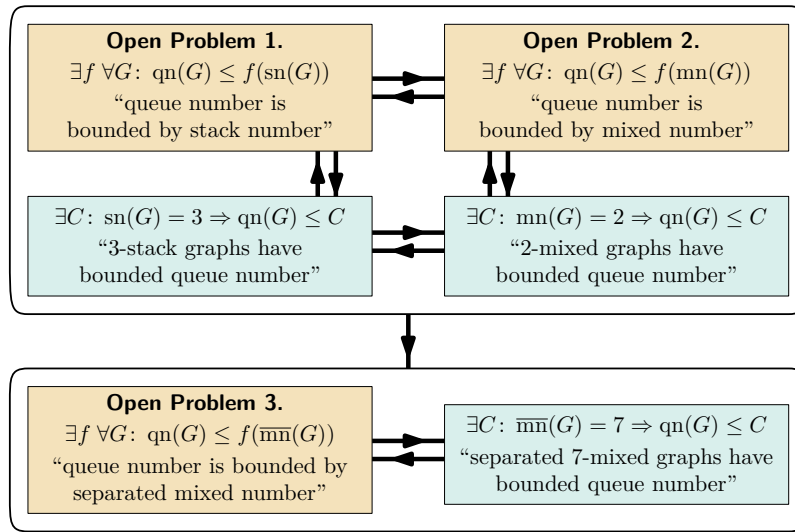
**Figure 3** An overview of the new and known relationships between different linear layouts.

(2) *every separated s-stack q-queue layout of a graph $G = (A \cup B, E)$ with $|A| = |B|$ can be transformed into a separated $f(s, q)$-queue layout for some function $f$ by applying the same permutation to $A$ and $B$;*

(3) *every separated 1-stack q-queue graph has queue number at most $f(q)$ for some function $f$;*

(4) *every separated 1-stack 6-queue graph has queue number at most $C$ for some constant $C$.*

Finally, let us discuss the connections between non-separated and separated layouts. Standard techniques for queue layouts (which we present in Section 2) easily give that Open Problem 2 implies Open Problem 3. In fact, Corollary 7 states that for every bipartite graph $G$ we have $\overline{\mathrm{qn}}(G) \leq 2\,\mathrm{qn}(G)$. However, it remains open whether Open Problem 3 implies Open Problem 2 (or equivalently Open Problem 1). The situation is summarized in Figure 3.

## 2 Preliminaries

In this section we collect several facts about manipulating vertex orders in linear layouts, which keep the stack and/or queue numbers within a constant factor from each other.

▶ **Lemma 6** (Riffle-Lemma). *Let $G = (V, E)$ be a graph with a given q-queue layout using $\sigma$ as the vertex order and $V_1, \ldots, V_k$ be a partition of $V$. Then, for any vertex order $\sigma'$ where for vertices $u, v \in V_i$ it holds that $u <_{\sigma'} v$ whenever $u <_\sigma v$:*

(1) *$G$ admits a $k^2 \cdot q$-queue layout.*

(2) *$G$ admits a $2 \cdot \ell \cdot (k - \ell) \cdot q$-queue layout if $G$ is bipartite, $\bigcup_{i=1}^{\ell} V_i = A$ and $\bigcup_{j=\ell+1}^{k} V_j = B$.*

**Proof.** We show that every queue $Q \subseteq E$ for $\sigma$ can be split into $k^2$ queues for $\sigma'$; see Figure 4a. Overall, this results in the desired $k^2 \cdot q$-queue layout. For $i, j \in [k]$ let $E_{i,j} \subseteq Q$ contain all edges $(u, v) \in Q$ such that $u <_\sigma v$ and $u \in V_i$ and $v \in V_j$. This partitions $Q$ into $k^2$ many edge sets $E_{i,j}$, and we claim that each $E_{i,j}$ is a queue for $\sigma'$. For $i = j$, the relative order of any $u, v \in V_i$ is unchanged in $\sigma'$, and hence $E_{i,i}$ is a queue for $\sigma'$. For $i \neq j$, let $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$ be two edges in $E_{i,j}$ with $u_1, u_2 \in V_i$ and $v_1, v_2 \in V_j$. Without loss of generality assume that $u_1 <_{\sigma'} u_2$ (hence $u_1 <_\sigma u_2$) and assume for a contradiction that $e_1$ nests $e_2$ in $\sigma'$, that is, $u_1 <_{\sigma'} u_2 <_{\sigma'} v_2 <_{\sigma'} v_1$ or $v_2 <_{\sigma'} v_1 <_{\sigma'} u_1 <_{\sigma'} u_2$. In either case,
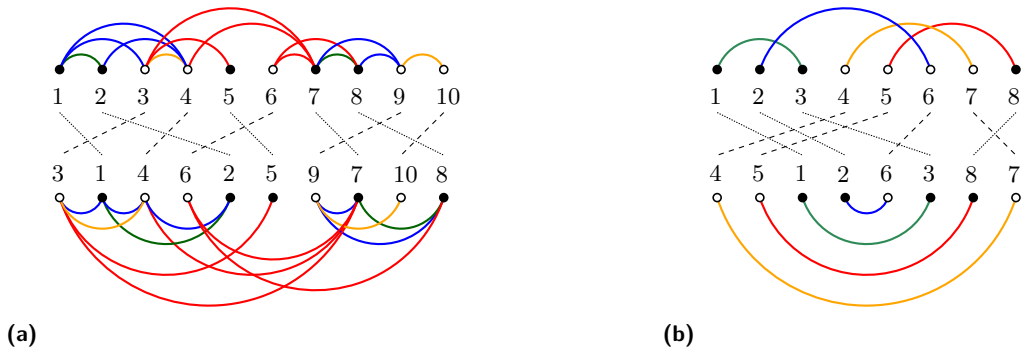
**Figure 4** **(a)** Illustrating Lemma 6 (1) and its tightness for $k = 2$ **(b)**. Edge colors illustrate the partition of the edges with respect to $V_1$ (black) and $V_2$ (white). The black lines signify the orders within $V_1$ (dotted) and $V_2$ (dashed).

it follows that $v_2 <_{\sigma'} v_1$ and hence $v_2 <_\sigma v_1$. Together with $u_2 <_\sigma v_2$ (as $(u_2, v_2) \in E_{i,j}$) this yields $u_1 <_\sigma u_2 <_\sigma v_2 <_\sigma v_1$ and $e_1$ nests $e_2$ in $\sigma$ – a contradiction to $Q$ being a queue. It follows that each $E_{i,j}$ requires at most one queue in the layout with vertex order $\sigma'$, which concludes the first case of the proof.

In the bipartite case, for each queue $Q$ in the original layout, let $e = (u, v)$ be an edge in $Q$ with $u \in V_i$ with $i \leq \ell$ and $v \in V_j$ with $\ell < j \leq k$. As there are $\ell \cdot (k - \ell)$ combinations of $i$ and $j$, and we require two queues, namely $E_{i,j}$ and $E_{j,i}$, for each combination, the resulting layout with vertex order $\sigma'$ requires $2 \cdot \ell \cdot (k - \ell) \cdot q$-queues in total. ◄

Applying the lemma to bipartite graphs with $\ell = 1$ and $k = 2$ such that $V_1 = A$ and $V_2 = B$ yields the following (well-known) fact; see [25] for an alternative proof.

▶ **Corollary 7.** *Every $q$-queue graph with bipartition $(A, B)$ has a separated $2q$-queue layout.*

The next two results concern graph subdivisions. A *subdivision* of a graph $G$ is a graph obtained from $G$ by replacing each edge $(u, v)$ in $G$ by a path with one or more edges. Internal vertices on such a path are called *division vertices*.

▶ **Lemma 8** (Lemma 2 in [11]). *Let $G = (V, E)$ be a graph and $G' = (V', E')$ be a subdivision of $G$ with at most one division vertex per edge. If $G$ admits a $q$-queue layout with vertex order $\sigma$, then $G'$ with bipartition $V$ and $V' \setminus V$ admits a separated $(q + 1)$-queue layout in which $V$ is ordered as in $\sigma$.*

A converse operation to a subdivision is a (path-)contraction. We use a more general transformation here. An *$r$-shallow minor* is a restricted form of a graph minor in which each (connected) subgraph that is contracted to form the minor has radius at most $r$, where the radius is defined as the minimum of the eccentricities of its vertices. For an *$s$-stack $q$-queue graph $G$, we will combine the following result with finding an $s'$-stack $q'$-queue subdivision $H$ where $s' \leq s$ and $q' \leq q$, which has $G$ as an $r$-shallow minor in order to prove equivalence statements in Theorem 4 and Theorem 5.

▶ **Lemma 9** (Lemma 12 in [19]). *For every graph $G$ and $H$, where $G$ is a $r$-shallow minor of $H$, it holds that $\mathrm{qn}(G) \leq (2r + 1)\big(2\,\mathrm{qn}(H)\big)^{2r+1}$.*

## 3 Non-Separated Layouts

In this section we explore Open Problem 2. Dujmović and Wood [11] studied graph subdivisions of stack and queue layouts. They proved that every $s$-stack graph has

(i) a 3-stack subdivision with $2\lceil \log_2 s \rceil - 2$ division vertices per edge, and

(ii) a 1-stack 1-queue subdivision with $4\lceil \log_2 s \rceil$ division vertices per edge.

Similarly, every $q$-queue graph has

(i) a 3-stack subdivision with $2\lceil \log_2 q \rceil + 1$ division vertices per edge, and

(ii) a 2-queue subdivision with $2\lceil \log_2 q \rceil + 1$ division vertices per edge, and

(iii) a 1-stack 1-queue subdivision with $4\lceil \log_2 q \rceil + 2$ division vertices per edge.

Notice some asymmetry in the statements of the two results: It is not always possible to subdivide an $s$-stack graph with $f(s)$ division vertices per edge (for an arbitrary function $f$) such that the result admits an $\mathcal{O}(1)$-queue layout. Otherwise, such a subdivision combined with Lemma 9 would imply that every $s$-stack graph admits an $\mathcal{O}(1)$-queue layout, contradicting the main result of Dujmović et al. [7].

The crux of the contributions of Dujmović and Wood [11] is a technical construction, which we formalize next. Let $T$ be a rooted tree with nodes $V(T)$ and edges $E(T)$. Given a graph $G = (V, E)$, a *$T$-partition* of $G$ is a pair $\big( T, \{T_x : x \in V(T)\} \big)$ consisting of a tree $T$ and a partition of $V$ into sets $\{T_x : x \in V(T)\}$, such that for every edge $(u, v) \in E$ one of the following holds: (i) $u, v \in T_x$ for some $x \in V(T)$, or (ii) there is an edge $(x, y)$ of $T$ with $u \in T_x$ and $v \in T_y$. The sets $T_x, x \in V(T)$ are called the *bags* of the $T$-partition. A *$T$-layout* of a graph $G$ is a $T$-partition of $G$ in which the bags are ordered, that is, $<_x$ is a total order of vertices in $T_x$; see Figure 5a for an example. The vertex orders are described in terms of two functions, $\mathcal{S} : V(T) \to \mathbb{N}_0$ and $\mathcal{Q} : V(T) \to \mathbb{N}_0$, defined for the nodes of $T$. We prescribe each bag to contain either only stacks or only queues formed by intra-bag edges. Here $\mathcal{S}(x)$ (respectively, $\mathcal{Q}(x)$) denotes the stack (queue) number of the intra-bag edges on $T_x$ under vertex order $<_x$ such that if $T_x$ is prescribed to contain stacks, i.e., if $\mathcal{S}(x) > 0$ then $\mathcal{Q}(x) = 0$, and if $T_x$ is a bag containing queues, then $\mathcal{Q}(x) > 0$ and $\mathcal{S}(x) = 0$. Similarly, for the edges of $T$, $\mathcal{K}(x, y) : E(T) \to \mathbb{N}_0$ is the minimum number of edge sets $E_1, \ldots, E_{\mathcal{K}(x,y)}$ needed to partition the inter-bag edges of $G$ between $T_x$ and $T_y$ such that they are pairwise non-crossing. Under the concatenation of the orders $<_x$ and $<_y$, each $E_i$, $1 \le i \le \mathcal{K}(x, y)$ will form a queue, while each $E_i$ forms a stack when concatenating $<_x$ and $>_y$. The leaf nodes of a tree $T$ are denoted $\widetilde{V}(T)$.

In this paper, we work with *simple $T$-layouts*, where

- for every non-leaf node $x \in V(T) \setminus \widetilde{V}(T)$, bag $T_x$ forms an independent set in $G$, that is, $\mathcal{S}(x) = \mathcal{Q}(x) = 0$;
- for every leaf node $x \in \widetilde{V}(T)$, bag $T_x$ admits a 1-stack or a 1-queue layout under $<_x$, that is, $\mathcal{S}(x) = 1, \mathcal{Q}(x) = 0$ or $\mathcal{S}(x) = 0, \mathcal{Q}(x) = 1$;
- for every edge $(x, y) \in E(T)$, it holds that $\mathcal{K}(x, y) = 1$.

In all our constructions of tree-partitions, we utilize a *binary tree*, that is, a rooted tree in which every node has at most two children. The following result provides a subdivision and a tree-layout for a given graph.

▶ **Lemma 10** (a special case of Lemma 21 in [11]). *Let $G$ be a graph that admits a $k$-stack (respectively, $k$-queue) layout with vertex order $\sigma$, and $T$ be a binary tree of height $h$ with $k$ or more leaves. Then $G$ has a subdivision, $D$, with an even number of division vertices per edge such that $D$ has a simple $T$-layout in which $\mathcal{S}(x) = 1$ (respectively, $\mathcal{Q}(x) = 1$) for every leaf node $x \in \widetilde{V}(T)$. The number of division vertices per edge is at most $2h$, or exactly $2h$*

*if all the leaves of $T$ are at depth $h$. Moreover, the vertices of $G$ correspond to vertices in
the root bag of $T$ and their order in the $T$-layout is $\sigma$, whereas all other bags contain only
division vertices.*

In what follows we consider a (not necessarily proper) 2-coloring of edges of a tree
using colors blue and red, i.e., we allow edges with the same color at a common endvertex.
Throughout, we associate blue colors with stacks, and red colors with queues. The set of
blue edges is $E^s(T) \subseteq E(T)$ and the set of red edges is $E^q(T) \subseteq E(T)$.

▶ **Lemma 11** (a special case of Lemma 22 in [11]). *Let $G$ be a graph with a $T$-layout for
some tree $T$. Suppose that edges of $T$ are 2-colored in red and blue, and its nodes, $V(T)$, are
ordered by $\sigma$ such that the blue edges, $E^s$, form a stack and the red edges, $E^q$, form a queue.
Let*

$$\lambda_s = \max_{x \in V(T)} \left( \mathcal{S}(x) + \sum_{(y,x) \in E^s :\, y <_\sigma x} \mathcal{K}(y,x) + \sum_{(x,y) \in E^s :\, x <_\sigma y} \mathcal{K}(x,y) \right), \text{ and}$$

$$\lambda_q = \max_{x \in V(T)} \left( \mathcal{Q}(x) + \max_{y \in V(T) :\, y \leq_\sigma x} \sum_{(y,z) \in E^q :\, x \leq_\sigma z} \mathcal{K}(y,z) \right).$$

*Then $G$ admits a mixed $\lambda_s$-stack $\lambda_q$-queue layout. Furthermore, the order of vertices in
the root bag of the $T$-layout is preserved in the mixed layout.*

Now we can state the new results of the section.

▶ **Lemma 12.** *Let $G$ be an $s$-stack $q$-queue graph. Then $G$ has a 3-stack subdivision with at
most $2\lceil \log_2(\max(s,q)) \rceil + 3$ division vertices per edge.*

**Proof.** First we show how to obtain a tree-layout for an $s$-stack $q$-queue graph $G = (V, E)$
with the vertex order $\sigma$. Denote $h = \lceil \log_2(\max(s,q)) \rceil$ so that $\max(s,q) \leq 2^h$. Assume that
$E = S_1 \cup \cdots \cup S_s \cup Q_1 \cup \cdots \cup Q_q$, where $S_i, 1 \leq i \leq s$ are stacks and $Q_i, 1 \leq i \leq q$ are queues.

Consider the subgraph of $G$ induced by all stack edges, $G^s = (V, S_1 \cup \cdots \cup S_s)$. Let $T_s$ be
a binary tree of height $h + 1$ in which the root node has a single child, each internal non-root
node has exactly two children, and all leaves are at the same depth. By Lemma 10, there
exists a subdivision of $G^s$, denoted $D^s$, with exactly $2(h + 1)$ division vertices per edge and
a simple $T_s$-layout of $D^s$. It holds that $\mathcal{S}(x) = 1, \mathcal{Q}(x) = 0$ for $x \in \widetilde{V}(T_s)$, $\mathcal{S}(x) = \mathcal{Q}(x) = 0$
for $x \in V(T_s) \setminus \widetilde{V}(T_s)$, and $\mathcal{K}(x,y) = 1$ for $(x,y) \in E(T_s)$; see Figure 5.

Next consider the subgraph of $G$ induced by the queue edges, $G^q = (V, Q_1 \cup \cdots \cup Q_q)$.
Start with a binary tree of height $h + 1$, denoted $T'_q$, which is a copy of $T_s$ (that is, the
root node has one child and each internal non-root node has two children). There exists
a subdivision of $G^q$ with exactly $2h + 2$ subdivision vertices and its simple $T'_q$-layout by
Lemma 10. We modify the subdivision and the tree-layout as follows. Consider a leaf
node $x' \in \widetilde{V}(T'_q)$ and let $G_{x'}$ be the graph induced by vertices in $T_{x'}$ and the corresponding
intra-bag edges. Observe that $\mathcal{Q}(x') = 1$ in the $T'_q$-layout, and hence, $\mathrm{qn}(G_{x'}) = 1$ with $<_{x'}$
as the vertex order. Now, we apply Lemma 8 by subdividing every edge of $G_{x'}$ once and let
$G_x$ be the resulting graph. This results in a 2-queue layout of the subdivision $G_x$ in which
the vertices of $G_{x'}$ are separated from the new division vertices and are ordered as in the
$T'_q$-layout. Thus, we can build a new tree, denoted $T_q$, by appending a child, $x$, to every
node $x' \in \widetilde{V}(T'_q)$. This results in a (non-simple) $T_q$-layout for $G_q$, where the leaf nodes of
$T_q$ contain the division vertices in the order given by Lemma 8. Since the queue number of
$G_x$ is at most 2, the inter-bag edges between $G_{x'}$ and $G_x$ can be partitioned into two sets
of pairwise non-crossing edges, i.e., we have that $\mathcal{K}(x',x) = 2$ for all $(x',x) \in E(T_q)$, where
$x \in \widetilde{V}(T_q)$; see Figure 5a for an illustration.
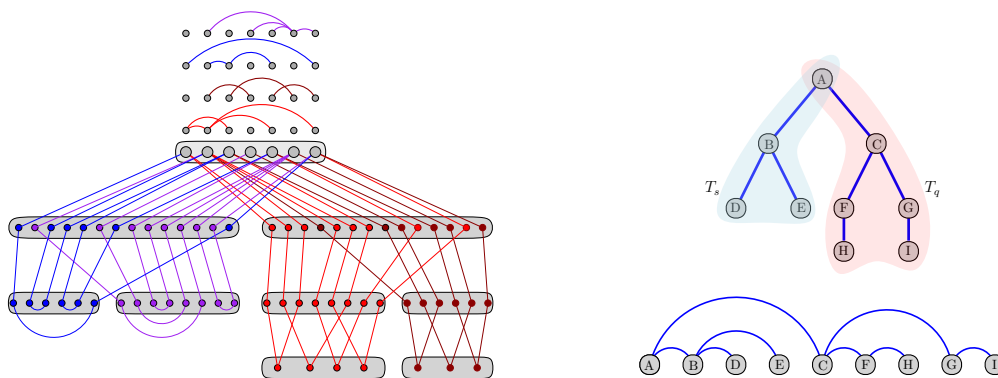
**(a)** A tree-layout of the subdivision of a 2-stack 2-queue graph. Edge colors in the tree-layout correspond to the original queues and stacks of $G$ shown above the root node.

**(b)** A tree-partition and a 1-stack layout of the corresponding binary tree, $T_{sq}$, where the blue edge color corresponds to Lemma 11.

**Figure 5** An illustration for Lemma 12: Subdividing a 2-stack 2-queue graph $G$ with at most $2\lceil \log_2(\max(s,q))\rceil + 3 = 5$ division vertices per edge to obtain a 3-stack graph.

By Lemma 10, the root bags of the $T_s$-layout and the $T_q$-layout contain the same vertices, $V$, both ordered by $\sigma$. Thus, we can merge the two tree-layouts into a joint one, called $T_{sq}$-layout; see Figure 5b. The corresponding subdivision of $G$, denoted $D^{sq}$, has stack edges subdivided $2h + 2$ times and queue edges subdivided $2h + 3$ times. To apply Lemma 11 for the $T_{sq}$-layout, we color all the edges of $T_{sq}$ blue and find a 1-stack layout of the tree via a depth-first search traversal such that every node precedes its children in the order. Let us argue that the result of applying Lemma 11 is a 3-stack layout, that is, $\lambda_s = 3$ and $\lambda_q = 0$.

Note that $\mathcal{Q}(x) = 0$ for all nodes of $T_{sq}$ and the tree contains no red edges; thus, $\lambda_q = 0$. For the stack number, consider four disjoint groups of nodes of $T_{sq}$:

- for $x \in \widetilde{V}(T_s)$, it holds that $\mathcal{S}(x) = 1$ and $\mathcal{K}(y, x) = 1$ for $(y, x) \in E^s$;
- for $x \in V(T_s) \setminus \widetilde{V}(T_s)$ and $x \in V(T_q') \setminus \widetilde{V}(T_q')$, it holds that $\mathcal{S}(x) = 0$, there exists one edge $(y, x) \in E^s$ with $\mathcal{K}(y, x) = 1$ and two edges $(x, y) \in E^s$ with $\mathcal{K}(x, y) = 1$;
- for $x \in \widetilde{V}(T_q')$, it holds that $\mathcal{S}(x) = 0$, $\mathcal{K}(y, x) = 1$ for a single edge $(y, x) \in E^s$, and $\mathcal{K}(x, y) = 2$ for a single edge $(x, y) \in E^s$;
- for $x \in \widetilde{V}(T_q)$, it holds that $\mathcal{S}(x) = 0$, and $\mathcal{K}(y, x) = 2$ for a single edge $(y, x) \in E^s$.

Combining everything together, we get

$$\lambda_s = \max_{x \in V(T_{sq})} \left( \mathcal{S}(x) + \sum_{(y,x) \in E^s} \mathcal{K}(y, x) + \sum_{(x,y) \in E^s} \mathcal{K}(x, y) \right) \leq 3.$$

Therefore, subdivision $D^{sq}$ of $G$ admits a 3-stack layout by Lemma 11.                  ◀

With a similar technique we can find a 1-stack 1-queue subdivision (Lemma 13) instead of a 3-stack subdivision (Lemma 12). The proof of the next result is given in [21].

▶ **Lemma 13.** *Let $G$ be an $s$-stack $q$-queue graph. Then $G$ has a 1-stack 1-queue subdivision with* at most $4\lceil \log_2(\max(s,q))\rceil + 6$ *division vertices per edge.*

We now prove Theorem 4, in particular that Open Problems 1 and 2 are equivalent.

▶ **Theorem 4.** *The following are equivalent:*
**(1)** *every s-stack graph has queue number at most $f(s)$ for some function $f$*
     *("queue number is bounded by stack number");*
**(2)** *every 3-stack graph has queue number at most $C$ for some constant $C$;*
**(3)** *every 1-stack 1-queue graph has queue number at most $C$ for some constant $C$;*
**(4)** *every s-stack q-queue graph has queue number at most $f(s, q)$ for some function $f$*
     *("queue number is bounded by mixed number").*

**Proof.** It is immediate that (4) implies (1), (2), and (3). That (1) $\iff$ (2) is proven by
Dujmović and Wood [11].

Now we show that $(2) \Rightarrow (4)$. Assume that every 3-stack graph has a bounded queue
number, $C \in \mathcal{O}(1)$. Let $G$ be an $s$-stack $q$-queue graph. By Lemma 12, $G$ admits a 3-stack
subdivision, $D$, with at most $k = 2\lceil \log_2(\max(s, q)) \rceil + 3$ division vertices per edge. By the
assumption, $\mathrm{qn}(D) \leq C$. Note that $G$ is an $r$-shallow minor of $D$ for $r = (k + 1)/2 =$
$\lceil \log_2(\max(s, q)) \rceil + 2$. By Lemma 9, we get

$$\mathrm{qn}(G) \leq (2r + 1)\big(2\,\mathrm{qn}(D)\big)^{2r+1} = (2\lceil \log_2(\max(s, q)) \rceil + 5)\big(2C\big)^{2\lceil \log_2(\max(s,q)) \rceil + 5},$$

which proves that $G$ has a bounded queue number, as long as $s, q$ and $C$ are constants.

Similarly we show that $(3) \Rightarrow (4)$. Suppose that every 1-stack 1-queue graph has queue
number at most $C \in \mathcal{O}(1)$. Let $G$ be an $s$-stack $q$-queue graph. By Lemma 13, $G$ admits a
1-stack 1-queue subdivision, $D$, with $k = 4\lceil \log_2(\max(s, q)) \rceil + 6$ division vertices per edge.
By the assumption, $\mathrm{qn}(D) \leq C$, and $G$ is an $r$-shallow minor of $D$ for $r = \lceil (k+1)/2 \rceil$. Again
by Lemma 9, we get

$$\mathrm{qn}(G) \leq (2r + 1)\big(2\,\mathrm{qn}(D)\big)^{2r+1} = (4\lceil \log_2(\max(s, q)) \rceil + 8)\big(2C\big)^{4\lceil \log_2(\max(s,q)) \rceil + 8},$$

which shows that $G$ has a bounded queue number, as long as $s, q$ and $C$ are constants.     ◀
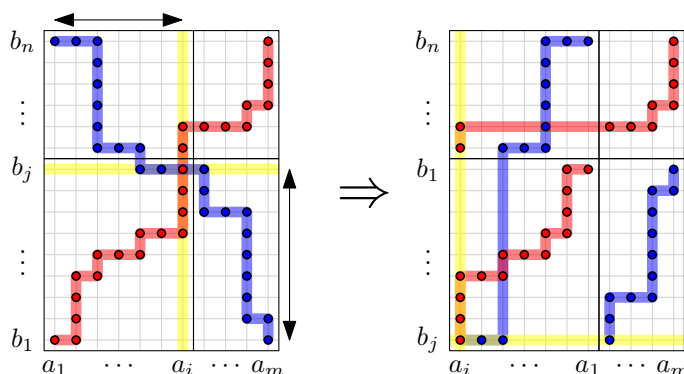
## 4     Separated Layouts

In this section we investigate separated mixed layouts of bipartite graphs and Open Problem 3.
Recall that a vertex order $\sigma$ of a bipartite graph $G = (V, E)$ with bipartition $(A, B)$ is *separated*
if all vertices of $A$ precede all vertices of $B$ in $\sigma$ (or vice versa).

▶ **Observation 14.** *Every separated s-stack q-queue layout of a bipartite graph $G = (V, E)$*
*with bipartition $(A, B)$ can be transformed to a separated q-stack s-queue layout by reversing*
*the order of all vertices in $A$ (or alternatively all vertices in $B$).*

Reversing the order of some consecutive vertices (as in Observation 14) is the simplest
modification we could do to a given layout. It turns out that this is already enough to show
that graphs with separated 1-stack 1-queue layouts have a constant queue number.

▶ **Theorem 15.** *Every separated 1-stack 1-queue graph admits a separated 4-queue layout.*

**Proof.** Let $G = (V, E)$ be a bipartite graph with bipartition $(A, B)$ admitting a separated
1-stack 1-queue layout with vertex order $\sigma$. Consider the reduced adjacency matrix $M$ with
columns $A = (a_1, \ldots, a_m)$ and rows $B = (b_1, \ldots, b_n)$ ordered according to $\sigma$. The edges
$S \subset E$ in the stack form a monotonically weakly decreasing subset of $|A| \times |B|$. The edges
$Q \subset E$ in the queue form a monotonically weakly increasing subset of $|A| \times |B|$. Without
loss of generality we can assume (by adding edges to the graph) that $S$ and $Q$ correspond to
inclusion-maximal such subsets; see Figure 6.

**Figure 6** Obtaining a separated 4-queue layout from a separated 1-stack 1-queue layout by reversing column and row orders. The blue edge color of the transformed stack edges is preserved for clarity.

Let $(a_i, b_j) \in A \times B$ be a point/edge that is contained in both, $S$ and $Q$. Now consider the vertex order $a_i, \ldots, a_1, a_{i+1}, \ldots, a_m$ of the columns obtained by reversing the order of $a_1, \ldots, a_i$. Similarly, consider the vertex order $b_j, \ldots, b_1, b_{j+1}, \ldots, b_n$ of the rows obtained by reversing the order of $b_1, \ldots, b_j$. Let $\sigma'$ denote the resulting separated vertex order of $G$. Now the edges in $Q$ incident to $a_1, \ldots, a_i$ form a queue $Q_1$ in $\sigma'$. Also the remaining edges in $Q$, incident to $a_{i+1}, \ldots, a_m$ form a queue $Q_2$ in $\sigma'$. Similarly, the edges in $S$ incident to $b_1, \ldots, b_j$ form a queue $Q_3$ in $\sigma'$. Also the remaining edges in $S$, incident to $b_{j+1}, \ldots, b_m$ form a queue $Q_4$ in $\sigma'$; see again Figure 6. This is a separated 4-queue layout of $G$.    ◀

Note that by applying Observation 14, separated 1-stack 1-queue graphs also admit a separated 4-stack layout. However, we do not know whether the bound of 4 in Theorem 15 is best-possible and remark that $K_{3,3}$ admits a separated 1-stack 1-queue layout but its separated stack/queue number is 3.

The simple operation of reversing the order of some consecutive vertices can be employed in a "checkerboard" fashion. Consider a given separated mixed layout (e.g., with $s$ stacks and $q$ queues). Assume we have partitioned the reduced adjacency matrix by means of a superimposed grid with a checkerboard odd-even pattern, in such a way that odd grid cells contain no stack edges and even grid cells contain no queue edges. Then, the vertex order of every second row and column can be reversed to derive a separated pure queue layout. Finding such a grid refinement is always possible, that is, down to individual rows and columns, however the derived $\overline{\mathrm{sn}}$ and $\overline{\mathrm{qn}}$ are dependent on the grid granularity and thus may not be bounded by $\overline{\mathrm{mn}}$. An example for specific separated $s$-stack $q$-queue layouts with bounded $\overline{\mathrm{qn}}$ number are grids where each cell contains either an increasing or decreasing diagonal. In this case, grid columns and rows can be halved in order to apply the checkerboard approach; see Figure 7.

However there exist graphs, where such operations do not suffice and more involved row and column permutations are necessary. In [21] we provide as a challenging example a subcubic bipartite graph that admits a separated 1-stack 2-queue layout and a fairly simple 4-queue pure layout. However, the grid granularity has to be super-constant in order to transform the mixed into the pure layout. We conclude that even for low-degree graphs with $\overline{\mathrm{mn}}(G) = 3$, it is not obvious how to transform a separated mixed layout into a separated pure layout, say with few queues.

**(a)**         **(b)**         **(c)**         **(d)**
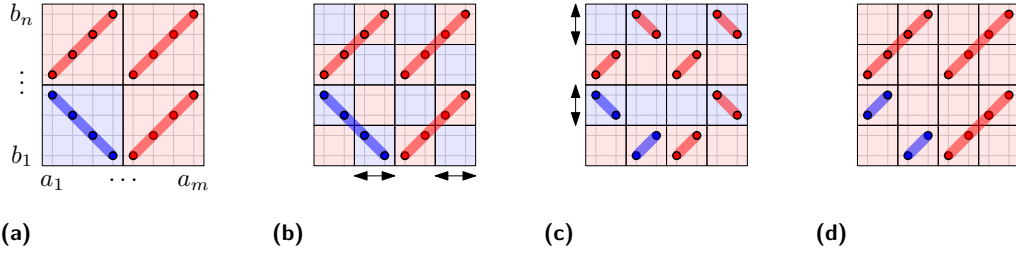
■ **Figure 7** Transforming a separated mixed layout of a grid with diagonals **(a)** into a separated pure layout **(d)** by halving columns and rows **(b)** and reversing every second row and column **(c)**. The blue edge color of the transformed stack edges is preserved for clarity.

While we do not know how to generalize Theorem 15 even just to separated 1-stack 2-queue layouts, we can narrow down the difficult case of Open Problem 3. If Open Problem 3 is answered positively, we can find an $f(s, q)$-queue layout for any bipartite graph admitting a separated $s$-stack $q$-queue layout. In Theorem 5 we show three equivalent formulations of this problem. In the challenging example presented in our preprint [21], the transformation from the separated mixed to the separated pure layout applies the same column and row permutation. Now, Theorem 5 states that we may always assume that rows and columns are identically permuted when transforming a separated mixed layout into a separated pure layout.

Further, we show in Lemma 17 that every separated $s$-stack $q$-queue layout can be transformed into a separated 1-stack $f(s, q)$-queue layout, while this transformation does not change the separated queue number by too much. This implies that one can restrict the attention to separated 1-stack $q$-queue layouts, for solving Open Problem 3.

▶ **Theorem 5.** *The following are equivalent:*
**(1)** *every separated $s$-stack $q$-queue graph $G$ has queue number at most $f(s, q)$ for some function $f$     ("queue number is bounded by mixed number");*
**(2)** *every separated $s$-stack $q$-queue layout of a graph $G = (A \cup B, E)$ with $|A| = |B|$ can be transformed into a separated $f(s, q)$-queue layout for some function $f$ by applying the same permutation to $A$ and $B$;*
**(3)** *every separated 1-stack $q$-queue graph has queue number at most $f(q)$ for some function $f$;*
**(4)** *every separated 1-stack 6-queue graph has queue number at most $C$ for some constant $C$.*

**Proof.** Observe that (1) immediately implies (3), which implies (4). That (2) implies (1) is also immediate, when adding isolated vertices to guarantee $|A| = |B|$. The proofs of the other directions are deferred, in particular see Lemma 16 for (1) $\Rightarrow$ (2) and Lemma 17 for (3) $\Rightarrow$ (1). For (4), we show equivalence by proving that (4) implies (3) in Lemma 18. ◀

▶ **Lemma 16.** *If there is a function $f$ such that every separated $s$-stack $q$-queue graph admits a separated $f(s, q)$-queue layout, then every separated $s$-stack $q$-queue layout of a bipartite graph $G = (A \cup B, E)$ with $|A| = |B|$ can be transformed into a separated $2f(s, q + 1)^2$-queue layout by applying the same permutation to $A$ and $B$.*

In other words, Lemma 16 proves that (1) implies (2) in Theorem 5.

**Proof.** Let $u_1, \ldots, u_n$ and $v_1, \ldots, v_n$ be the order of $A$ and $B$ in the given separated $s$-stack $q$-queue layout of $G$. We add the "identity" queue $Q$, where for each $i$, $1 \leq i \leq n$, there is the edge $(u_i, v_i)$ (if not already present in $G$). Let $G' = (A \cup B, E \cup Q)$ be the resulting bipartite

**(a)**                          **(b)**                          **(c)**
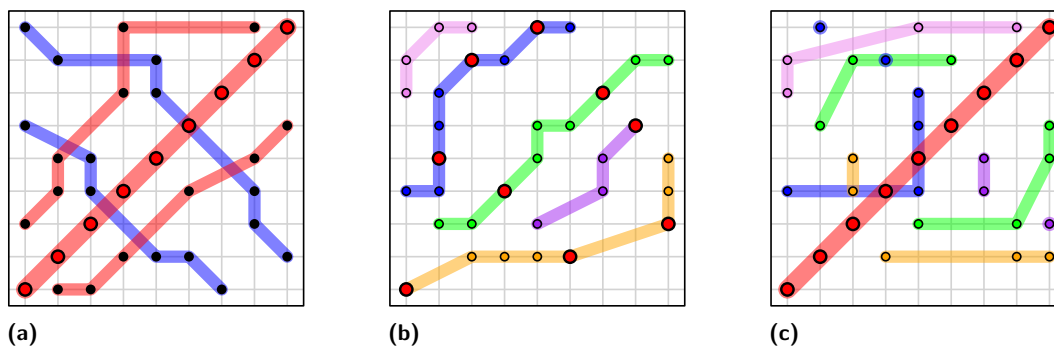
**Figure 8** Illustration of the proof of Lemma 16: Adding the identity queue **(a)**, obtaining a separated queue layout **(b)**, and restoring the identity queue **(c)**.

graph. By assumption, there exists a separated $f(s, q + 1)$-queue layout of $G'$ with vertex order $\sigma_0$. We are looking for another vertex order $\sigma$ such that, compared to the initial order, $A$ and $B$ are permuted the same, that is, the columns and rows in the reduced adjacency matrix are identically permuted. Since we added the identity queue $Q$ to obtain $G'$ in the beginning, this is equivalent to restoring $Q$ to the diagonal in the matrix; see Figure 8.

To that end, we shall apply Lemma 6 to change only the order of vertices in $A$, which corresponds to permuting the columns in the matrix. Let $k = f(s, q + 1)$ and $E_1, \ldots, E_k$ be the queues in the $f(s, q+1)$-queue layout of $G'$ and $A_i = \{u \in A \mid (u, v) \in Q \cap E_i\}$, $1 \leq i \leq k$. We want to apply the Riffle-Lemma (Lemma 6-(2)) with the partition of $V$ into $A_1, \ldots, A_k$ and $B$. To this end, we keep the order of $B$ (the rows), and restore $Q$ as a diagonal by simply sorting the columns according to their row in $Q$. Since each $E_i$ is a queue, the relative order within each $A_i$ is kept intact, as required by Lemma 6. Hence, by Lemma 6-(2), the resulting layout has at most $2 \cdot f(s, q + 1) \cdot f(s, q + 1)$ queues in total.                                              ◄

▶ **Lemma 17.** *Let $G$ be a bipartite graph with a separated $s$-stack $q$-queue layout. Then there exists another bipartite graph $H$ admitting a separated 1-stack $(s + q - 1)$-queue layout, where $G$ is a 1-shallow minor of $H$, such that $\mathrm{qn}(G) \in \mathcal{O}(\mathrm{qn}(H)^3)$.*

If (3) of Theorem 5 holds, then the separated 1-stack $(s + q - 1)$-queue layout of $H$ has bounded queue number, i.e, $\mathrm{qn}(G) \in \mathcal{O}(\mathrm{qn}(H)^3) \leq f(s + q - 1)$ for some function $f$. Hence $\mathrm{qn}(G) \leq f'(s, q) = f(s + q - 1)$, implying (1) of Theorem 5.

**Proof.** Assume we have a bipartite graph $G$, that admits a separated $s$-stack $q$-queue layout. We will build a new bipartite graph $H$, such that $H$ has a separated 1-stack $(s + q - 1)$-queue layout. We will then show that for the separated queue numbers of $G$ and $H$, it holds that $\mathrm{qn}(G) \in \mathcal{O}(\mathrm{qn}(H)^3)$. That is, a small queue number of $H$ implies a small queue number of $G$. Next we make the argument formal.

Let $G = (A \cup B, E), |A| = n, |B| = m$, $S_1, \ldots, S_s$ be the stacks and $Q_1, \ldots, Q_q$ be the queues in the separated $s$-stack $q$-queue layout. We build $H$ as follows. The vertices of $H$ are $A \cup (\bigcup_k U^k) \cup B \cup (\bigcup_k V^k)$, where $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$ are copied from $G$, while $U^k \subseteq \{u_1^k, \ldots, u_n^k\}$ and $V^k \subseteq \{v_1^k, \ldots, v_m^k\}$ for any $1 \leq k \leq s - 1$ are new. $U^k$ contains a vertex $u_i^k$ for each vertex $a_i$ incident to an edge $(a_i, b_j) \in S_k$. Likewise, $V^k$ contains a vertex $v_j^k$ for each such $b_j$. Vertices in $A$ and $B$ that are not incident to an edge in $S_k$ are omitted in $U^k$ and $V^k$. (Note that for each stack $S_k$, the new vertices in $U^k$ and $V^k$ are added to construct parts of the graph $H$ starting from $G$, while the last stack $S_s$ is left as is.) In the reduced adjacency matrix of $H$, the new vertices are represented by additional
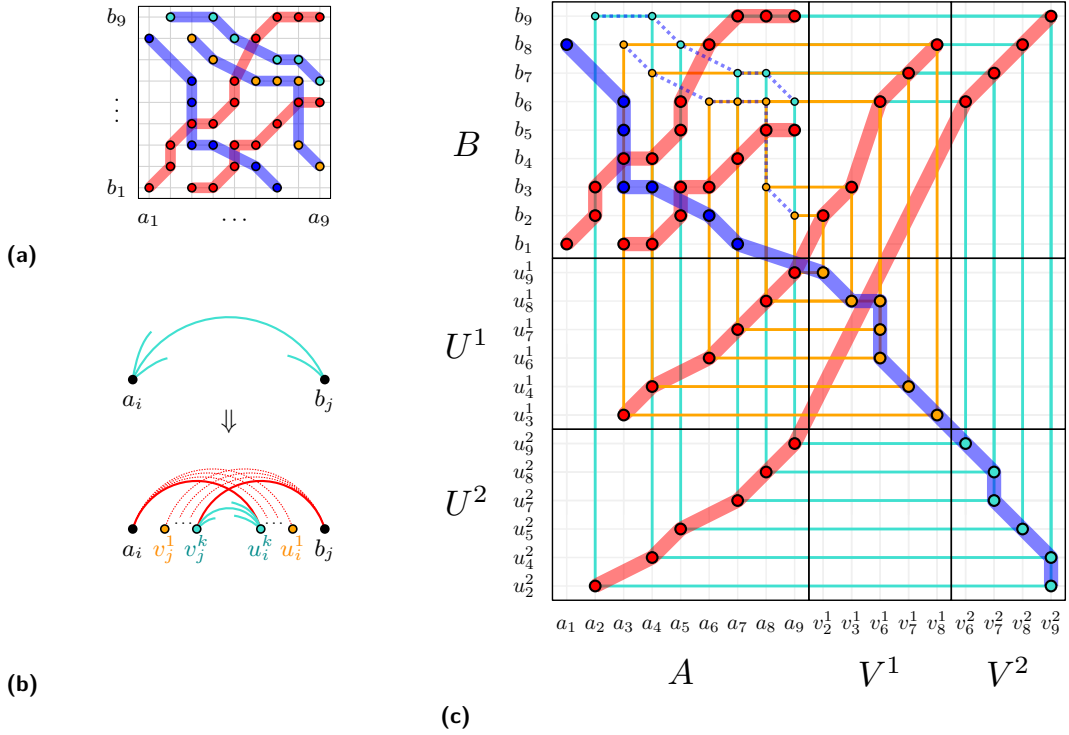
**Figure 9** Transforming a separated 3-stack 2-queue layout into a separated 1-stack 4-queue layout: The initial graph $G$ **(a)**, modifying a stack edge $(a_i, b_j)$ **(b)**, the constructed graph $H$ **(c)**.

blocks of $n_k$ consecutive rows for each $U^k$ and $m_k$ consecutive columns for each $V^k$, where $n_k$ $(m_k)$ is the number of vertices of $A$ $(B)$ having an edge in $S_k$. Row blocks of $U^k$ and column blocks of $V_k$ are subsequent for $k = 1, \ldots, s - 1$. Intuitively, we move the stack $S_k$ from $A \times B$ to $U^k \times V^k$, $k = 1, \ldots, s - 1$. We leave the last stack $S_s$ with all $q$ initial queues in $A \times B$. Finally, we put one new queue each in $A \times U^k$ and $V^k \times B$, $k = 1, \ldots, s - 1$. Figure 9 shows the construction and in particular the order of the rows $U^1, \ldots, U^{s-1}$ and columns $V^1, \ldots, V^{s-1}$.

More formally, for every edge $(a_i, b_j) \in Q_1 \cup \ldots \cup Q_q$, there is an edge $(a_i, b_j)$ in $H$. For every stack edge, $(a_i, b_j)$ of stack $S_k$, $1 \leq k \leq s - 1$, there are three edges $(a_i, u_i^k), (u_i^k, v_j^k), (v_j^k, b_j)$ in $H$. (In case of multiple edges between a pair of vertices, we keep only one instance.) It is easy to verify that $H$ admits a separated 1-stack $(s + q - 1)$-queue layout, as for every original stack $S_k$, $1 \leq k \leq s - 1$ the edges $(a_i, u_i^k)$ and $(v_j^k, b_j)$ for every $(a_i, b_j) \in S_k$ form a queue respectively, due to the increasing column and row indices of each block $U^k$ and $V^K$ starting from the bottom left of the matrix. In fact, edges between vertices in $A$ and $U^k$ and edges between vertices in $B$ and $V^k$ fit in a single queue, since the former lie below and to the left of the latter in the matrix. The original queues remain, yielding $q + s - 1$ queues in total. For a stack $S_k$, the edges $(a_i, b_j) \in S_k$ are a decreasing subset in $A \times B$ and due to the increasing ordering of indices in every column and row block of the matrix, the edges $(u_i^k, v_j^k)$ in $U^k \times V^k$ have the same property. The ordering of the row blocks $U^k$ and column blocks $V^k$, where $A \times B$ and $U^k \times V^k$, $1 \leq k \leq s - 1$ are positioned diagonally from the top left to the bottom right of the matrix, then allows to combine the $s$ stacks of $G$ into one single stack in $H$.

Finally, we show that $\mathrm{qn}(G) \in \mathcal{O}(\mathrm{qn}(H)^3)$. To this end, consider a $\mathrm{qn}(H)$-queue layout of $H$. Now, contract for each $i = 1, \ldots, n$ the vertex $a_i$ with its neighbors of the form $u_i^k$, $1 \leq k \leq s - 1$. Similarly, contract for each $j = 1, \ldots, m$ the vertex $b_j$ with its neighbors of the form $v_j^k$, $1 \leq k \leq s - 1$. The result is a 1-shallow minor of $H$, and most crucially, the result is exactly the initial graph $G$. Hence, Lemma 9 with radius $r = 1$ gives that $\mathrm{qn}(G) \leq (2r + 1)(2\,\mathrm{qn}(H))^{2r+1} = 24 \cdot \mathrm{qn}(H)^3$. ◀

Using the techniques from Section 3, we can provide the final missing piece of Theorem 5.

▶ **Lemma 18.** *Let $G$ be a bipartite graph with a separated 1-stack $q$-queue layout. Then $G$ has a subdivision $D$ with at most $2\lceil \log_2 q \rceil$ division vertices per edge that admits a separated 1-stack 6-queue layout.*

In other words, Lemma 18 proves that (4) implies (3) in Theorem 5, as applying Lemma 9 for the $r$-shallow minor $G$ of $D$ with $k = 2\lceil \log_2 q \rceil$ and $r = \lceil (k + 1)/2 \rceil$ implies bounded queue number of $G$ for a function $f(q)$.

$$\mathrm{qn}(G) \leq (2r + 1)\big(2\,\mathrm{qn}(D)\big)^{2r+1} = (2\lceil \log_2 q \rceil + 2)\big(12\big)^{2\lceil \log_2 q \rceil + 2}$$

**Proof.** Let $G = (V, E_s \cup E_1 \cup \cdots \cup E_q)$ be a bipartite graph that admits a separated 1-stack $q$-queue layout such that $E_s$ is a stack and $E_i, 1 \leq i \leq q$ are queues. Denote $h = \lceil \log_2 q \rceil$ so that $q \leq 2^h$. We consider the subgraph of $G$ induced by the queue edges, $G^q = (V, E_1 \cup \cdots \cup E_q)$. Let $T$ be a complete binary tree of height $h$. By Lemma 10, there exists a subdivision of $G^q$, denoted $D^q$, and a simple $T$-layout of $D^q$ in which $\mathcal{S}(x) = 0, \mathcal{Q}(x) = 1$ for all $x \in \widetilde{V}(T)$, $\mathcal{S}(x) = \mathcal{Q}(x) = 0$ for all non-leaf nodes $x \in V(T) \setminus \widetilde{V}(T)$, and $\mathcal{K}(x, y) = 1$ for all $(x, y) \in E(T)$. Note that $D^q$ contains exactly $2h$ division vertices per edge and it is bipartite; see Figure 10.

Now, color all edges of $T$ red and find a 1-queue layout of $T$ via a breadth-first search traversal such that every node precedes its children in the order. By Lemma 11 applied to graph $G^q$ and its simple $T$-layout, we obtain a linear layout of $D^q$. Let us argue that this result is in fact a 3-queue layout of $D^q$, i.e., that $\lambda_s = 0$ and $\lambda_q \leq 3$ as defined in Lemma 11. On the one hand, $\mathcal{S}(x) = 0$ for all the nodes of $T$ and there are no blue edges in $T$. Thus, we have $\lambda_s = 0$. On the other hand, every node $x \in V(T)$ has at most two outgoing edges in the layout of $T$, i.e., the set $\{(y, z) \in E^q \mid y \leq_\sigma x \leq_\sigma z\}$ contains at most two edges. Hence,

$$\lambda_q = \max_{x \in V(T)} \Big( \mathcal{Q}(x) + \max_{y \in V(T):\, y \leq_\sigma x} \sum_{(y,z) \in E^q:\, x \leq_\sigma z} \mathcal{K}(y, z) \Big)$$

$$\leq \max_{x \in V(T)} \Big( 1 + \max_{y \in V(T):\, y \leq_\sigma x} 2 \Big) \leq 3.$$

The final step is to convert the 3-queue layout of $D^q$ into a separated layout. This is accomplished by Corollary 7, which in worst case doubles the number of queues. The transformations of Lemma 11 and Corollary 7 keep the relative order of the original vertices $V$ unchanged. Thus, the resulting layout of $D^q$ can be joined with the stack edges $E_s$ to finally yield a subdivision $D$ of $G$ (in which the stack edges are not subdivided) with a separated 1-stack 6-queue layout. ◀

## 5 Conclusions and Open Questions

We have explored the relation between mixed and pure stack or queue layouts, in the separated as well as in the non-separated setting. An interesting (and somewhat surprising) result is the equivalence of Open Problems 1 and 2, which we believe sheds some light on the famous open problem, whether bounded stack number always implies bounded queue number. Let us highlight a few intriguing questions and possible directions in the area.
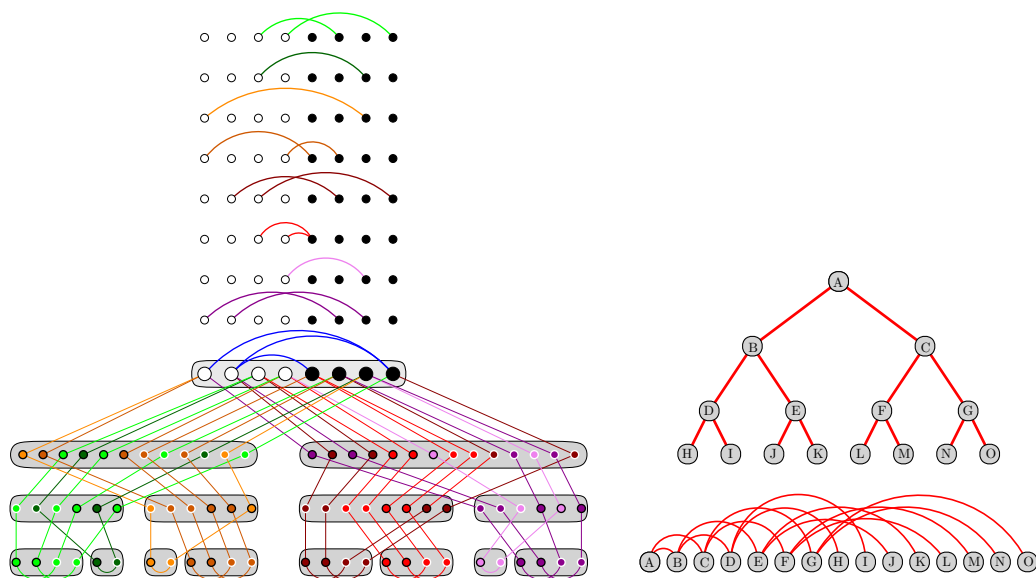
**(a)** A tree-layout of the subdivision of a separated 1-stack 8-queue graph $G$. Colors of edges and division vertices in the tree-layout correspond to the original queues and stack of $G$ shown above the root node, while vertex outline colors in non-root nodes correspond to the vertex partitions in G.

**(b)** A tree-partition and a layout of the corresponding binary tree $T$, where the red edge color corresponds to Lemma 11.

■ **Figure 10** An illustration for Lemma 18: Subdividing a separated 1-stack 8-queue graph $G$ with at most $2\lceil \log_2 q \rceil = 6$ division vertices per edge to obtain a mixed 1-stack 6-queue graph.

1. Do graphs with (non-separated) 1-stack 1-queue layouts have a constant queue number? This might be hard in general, but one could for example start with subcubic graphs.

2. Do graphs with separated 1-stack 2-queue layouts have a constant queue number? Lemmas 16 and 18 combined show that separated 1-stack 6-queue graphs are as hard as the general case, and we feel that the same holds for separated 1-stack 2-queue graphs. On the other hand, Theorem 15 solves the separated 1-stack 1-queue case.

3. Is there a constant $C$ such that every bipartite 1-stack 1-queue graph admits a separated $C$-stack $C$-queue layout? A positive answer would imply that Open Problems 1–3 are all equivalent.

## References

1   Jawaherul Md Alam, Michael A Bekos, Martin Gronemann, Michael Kaufmann, and Sergey Pupyrev. Queue layouts of planar 3-trees. *Algorithmica*, pages 1–22, 2020. `doi:10.1007/s00453-020-00697-4`.

2   Patrizio Angelini, Giordano Da Lozzo, Henry Förster, and Thomas Schneck. 2-layer k-planar graphs density, crossing lemma, relationships and pathwidth. *The Computer Journal*, 67(3):1005–1016, April 2023. `doi:10.1093/comjnl/bxad038`.

3   János Barát, Jirí Matousek, and David R. Wood. Bounded-degree graphs have arbitrarily large geometric thickness. *The Electronic Journal of Combinatoric*, 13(1), 2006. `doi:10.37236/1029`.

4   Michael A. Bekos, Martin Gronemann, and Chrysanthi N. Raftopoulou. On the queue number of planar graphs. In *International Symposium on Graph Drawing and Network Visualization*, pages 271–284, Berlin, Heidelberg, 2021. Springer-Verlag. `doi:10.1007/978-3-030-92931-2_20`.

**5**   Michael A. Bekos, Michael Kaufmann, Fabian Klute, Sergey Pupyrev, Chrysanthi N. Raftopoulou, and Torsten Ueckerdt. Four pages are indeed necessary for planar graphs. *Journal of Computational Geometry*, 11(1):332–353, 2020. `doi:10.20382/jocg.v11i1a12`.

**6**   Frank Bernhart and Paul C. Kainen. The book thickness of a graph. *Journal of Combinatorial Theory, Series B*, 27(3):320–331, 1979. `doi:10.1016/0095-8956(79)90021-2`.

**7**   Vida Dujmović, David Eppstein, Robert Hickingbotham, Pat Morin, and David R Wood. Stack-number is not bounded by queue-number. *Combinatorica*, pages 1–14, 2021. `doi:10.1007/s00493-021-4585-7`.

**8**   Vida Dujmović, Gwenaël Joret, Piotr Micek, Pat Morin, Torsten Ueckerdt, and David R Wood. Planar graphs have bounded queue-number. *Journal of the ACM*, 67(4):1–38, 2020. `doi:10.1145/3385731`.

**9**   Vida Dujmović, Pat Morin, and David R Wood. Layout of graphs with bounded tree-width. *SIAM Journal on Computing*, 34(3):553–579, 2005. `doi:10.1137/S0097539702416141`.

**10**  Vida Dujmović, Attila Pór, and David R Wood. Track layouts of graphs. *Discrete Mathematics & Theoretical Computer Science*, 6(2):497–522, 2004. `doi:10.46298/DMTCS.315`.

**11**  Vida Dujmović and David R Wood. Stacks, queues and tracks: Layouts of graph subdivisions. *Discrete Mathematics and Theoretical Computer Science*, 7:155–202, 2005. `doi:10.46298/dmtcs.346`.

**12**  Peter Eades and Nicholas C Wormald. Edge crossings in drawings of bipartite graphs. *Algorithmica*, 11(4):379–403, 1994. `doi:10.1007/BF01187020`.

**13**  David Eppstein, Robert Hickingbotham, Laura Merker, Sergey Norin, Michał T Seweryn, and David R Wood. Three-dimensional graph products with unbounded stack-number. *Discrete & Computational Geometry*, pages 1–28, 2023. `doi:10.1007/s00454-022-00478-6`.

**14**  Henry Förster, Michael Kaufmann, Laura Merker, Sergey Pupyrev, and Chrysanthi N. Raftopoulou. Linear layouts of bipartite planar graphs. In Pat Morin and Subhash Suri, editors, *Algorithms and Data Structures - 18th International Symposium, WADS 2023, Montreal, QC, Canada, July 31 - August 2, 2023, Proceedings*, volume 14079 of *Lecture Notes in Computer Science*, pages 444–459. Springer, 2023. `doi:10.1007/978-3-031-38906-1_29`.

**15**  Joseph L. Ganley and Lenwood S. Heath. The pagenumber of k-trees is O(k). *Discrete Applied Mathematics*, 109(3):215–221, 2001. `doi:10.1016/S0166-218X(00)00178-5`.

**16**  Deborah Haun, Laura Merker, and Sergey Pupyrev. Forbidden patterns in mixed linear layouts. In *42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025)*, 2024. `doi:10.48550/arXiv.2412.12786`.

**17**  Lenwood S Heath, Frank Thomson Leighton, and Arnold L Rosenberg. Comparing queues and stacks as machines for laying out graphs. *SIAM Journal on Discrete Mathematics*, 5(3):398–412, 1992. `doi:10.1137/0405031`.

**18**  Lenwood S Heath and Arnold L Rosenberg. Laying out graphs using queues. *SIAM Journal on Computing*, 21(5):927–958, 1992. `doi:10.1137/0221055`.

**19**  Robert Hickingbotham and David R. Wood. Shallow minors, graph products, and beyond-planar graphs. *SIAM Journal on Discrete Mathematics*, 38(1):1057–1089, 2024. `doi:10.1137/22M1540296`.

**20**  Paul C. Kainen. Some recent results in topological graph theory. In Ruth A. Bari and Frank Harary, editors, *Graphs and Combinatorics*, pages 76–108, Berlin, Heidelberg, 1974. Springer Berlin Heidelberg. `doi:10.1007/BFb0066436`.

**21**  Julia Katheder, Michael Kaufmann, Sergey Pupyrev, and Torsten Ueckerdt. Transforming stacks into queues: Mixed and separated layouts of graphs. *CoRR*, abs/2409.17776, 2024. `doi:10.48550/arXiv.2409.17776`.

**22**  Jawaherul Md. Alam, Michael A. Bekos, Martin Gronemann, Michael Kaufmann, and Sergey Pupyrev. The mixed page number of graphs. *Theoretical Computer Science*, 931:131–141, 2022. `doi:10.1016/j.tcs.2022.07.036`.

**23**   Hiroshi Nagamochi. An improved bound on the one-sided minimum crossing number in two-layered drawings. *Discrete & Computational Geometry*, 33(4):569–591, 2005. `doi:10.1007/S00454-005-1168-0`.

**24**   L Taylor Ollmann. On the book thicknesses of various graphs. In *Proc. 4th Southeastern Conference on Combinatorics, Graph Theory and Computing*, volume 8, page 459. Utilitas Math., 1973.

**25**   Sriram Venkata Pemmaraju. *Exploring the powers of stacks and queues via graph layouts.* PhD thesis, Virginia Polytechnic Institute and State University, 1992.

**26**   Sergey Pupyrev. Mixed linear layouts of planar graphs. In *International Symposium on Graph Drawing and Network Visualization*, pages 197–209. Springer, 2017. `doi:10.1007/978-3-319-73915-1_17`.

**27**   Sergey Pupyrev. Improved bounds for track numbers of planar graphs. *Journal of Graph Algorithms and Applications*, 24(3):323–341, 2020. `doi:10.7155/JGAA.00536`.

**28**   Veit Wiechert. On the queue-number of graphs with bounded tree-width. *The Electronic Journal of Combinatorics*, 24(1):P1.65, 2017. `doi:10.37236/6429`.

**29**   David R. Wood. Bounded-degree graphs have arbitrarily large queue-number. *Discrete Mathematics and Theoretical Computer Science*, 10(1), 2008. `doi:10.46298/DMTCS.434`.

**30**   Mihalis Yannakakis. Embedding planar graphs in four pages. *Journal of Computer and System Sciences*, 38(1):36–67, 1989. `doi:10.1016/0022-0000(89)90032-9`.