A Note on Mixed Linear Layouts of Planar Graphs

Michael Kaufmann[1] **and Maria Eleni Pavlidi**[2]

1 **Department of Computer Science, University of Tübingen, Germany**
michael.kaufmann@uni-tuebingen.de
2 **Department of Mathematics, University of Ioannina, Greece**
m.e.pavlidi@uoi.gr

Abstract

In this work, we study mixed linear layouts of graphs. Our motivation stems from a result by Pupyrev [15], who disproved a conjecture by Heath and Rosenberg [14] by showing the existence of planar graphs not admitting layouts with one stack and one queue. Since stacks and queues form special cases of the recently-introduced riques, we strengthen this result by showing that there exist planar graphs that do not admit a layout with one rique and either one stack or one queue.

1 Introduction

Linear layouts of graphs [12] have a long tradition of research, e.g., in Algorithm Design, Combinatorics, Graph Theory and Graph Drawing. The ones that we consider in this paper are defined using an associated data structure [3, 4, 9, 14]. The task is to find a so-called *linear order* of the vertices of the input graph and a partition of its edges into as few parts (called *pages*) as possible, such that the edges of each part can be processed by the given data structure. Namely, assuming that the vertices are left-to-right ordered according to their linear order, each edge is added to the data structure when its left endpoint is encountered in the order and is removed from the data structure when its right endpoint is encountered.

In this context, the most prominent types of linear layouts are the *stack* [9, 17] and the *queue layouts* [11, 14] that are defined using the stack and the queue data structures, respectively. A page of the former is called *stack* and does not allow two crossing edges, while a page of the latter is called *queue* and does not allow two nesting edges; see Fig. 2. Both these layouts form special cases of the so-called *deque layouts* [3], which are defined using the *double-ended queue* (or *deque*, for short) data structure. It is well-known that a page of a deque layout, called *deque*, has the following properties: the union of (i) two stacks, or (ii) two queues or (iii) a stack and a queue forms a deque [3]. In particular, (i) and (ii) imply that the *deque-number* (i.e., the minimum required number of deques over all deque layouts) of a graph cannot be more than half its stack- or its queue-number (i.e., the corresponding required numbers of stacks and queues, respectively).

In this work, we focus on planar input graphs and mixed linear layouts consisting of two pages; one that is either a stack or a queue, and one that is *rique* [4]; such a page is defined by the *restricted-input double-ended queue* (or *rique*, for short) data structure; refer, e.g., to Fig. 1 for a sample linear layout consisting of a single rique and to Section 2 for formal definitions. The rique data structure forms a special case of the deque data structure as follows. While in a deque insertions and removals occur both at the head and at the tail of it, in a rique insertions occur only at the head (removals occur both at the head and at the tail). Our work is motivated by a result by Pupyrev [15], who disproved a conjecture by Heath and Rosenberg [14] by showing the existence of planar graphs not admitting mixed layouts with one stack and one queue. Here, we show that there exist planar graphs that do not admit mixed layouts with one rique and either one stack or one queue. In other words,

40th European Workshop on Computational Geometry, Ioannina, Greece, March 13–15, 2024.
This is an extended abstract of a presentation given at EuroCG'24. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

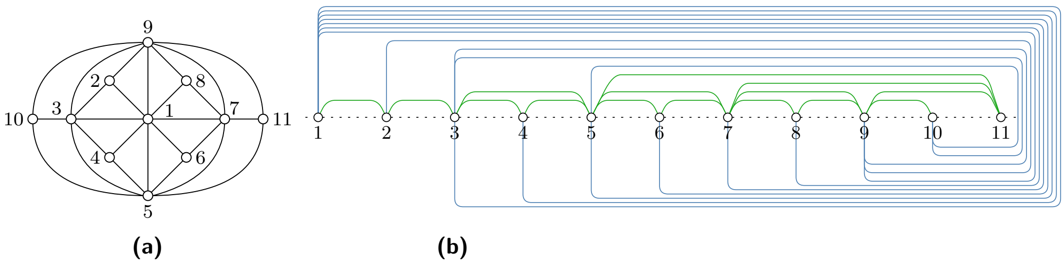


Figure 1 Illustration of: (a) the Goldner-Harary graph without the edge connecting its topmost vertex with its bottommost one, and (b) a rique layout of it with a single rique, in which the green edges are head-head, while the blue ones are head-tail.

with respect to the previously mentioned result by Pupyrev [15], our result implies that substituting one of the pages by a rique is still not enough for a positive answer to Heath and Rosenberg's conjecture.

Our work is also related to the rique-number (i.e., the minimum required number of riques over all rique layouts) of planar graphs. More precisely, since the stack-number of planar graphs is 4 [7, 17], it follows that the deque-number of planar graphs is 2, as also observed by Auer et al. [3]. So, it is natural to ask whether the rique-number of planar graphs is also 2; the obvious upper bound is 4, since a stack page is trivially a rique [4]. Unfortunately, we have not managed to completely settle this question, as our result does not close the gap on the rique-number of planar graphs (this ranges between 2 and 4, as noted). It forms, however, an indication that it might be not 2 (as observed above, both stacks and queues form special cases of riques).

2 Preliminaries

A *vertex order* $\prec$ of a graph G is a total order of its vertices, such that for any two vertices u and v of G, we write $u \prec v$ if and only if u precedes v in the order. Let F be a set of $k \geq 2$ pairwise independent edges (u_i, v_i) of G, that is, $F = \{(u_i, v_i);\ i = 1, \ldots, k\}$. If $u_1 \prec \cdots \prec u_k \prec v_k \prec \cdots \prec v_1$, then the edges of F form a k-*rainbow*, while if $u_1 \prec \cdots \prec u_k \prec v_1 \prec \cdots \prec v_k$, then the edges of F form a k-*twist*; see Fig. 2. Two edges that form a 2-twist (2-rainbow) are commonly referred to as *crossing* (*nested*). A *stack* is a set of pairwise non-crossing edges in $\prec$, while a *queue* is a set of pairwise non-nested edges in $\prec$.

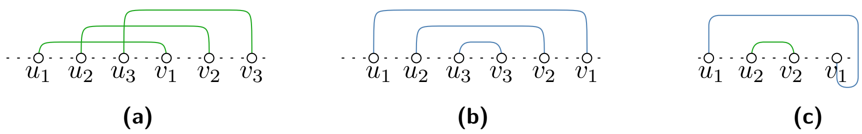


Figure 2 Illustration of: (a) a 3-twist (i.e., three pairwise crossing edges), (b) a 3-rainbow (i.e., three pairwise nesting edges), and (c) a rique page with two edges; a head-head and a head-tail.

A *rique* is a set of edges that does not contain three edges (a, a'), (b, b') and (c, c') such that $a \prec b \prec c \prec b' \prec \{a', c'\}$ in $\prec$ [4]. A more intuitive definition of a rique is the following. Assume that the vertices of the input graph are arranged on a horizontal line ℓ from left to right according to $\prec$ (say, w.l.o.g., equidistantly). Then, each edge (u, v) with $u \prec v$ can be represented either (i) as a semi-circle that is completely above ℓ connecting u and v, or (ii) as two semi-circles on opposite sides of ℓ, one that starts at u, lies above ℓ and ends at a point p of ℓ to the right of the last vertex of $\prec$ and one that starts at p, lies below ℓ and ends at v. Then, a rique is a set of edges each of which can be represented with one of the two types (i) or (ii) that avoids crossings (such a representation is called *cylindric* in [3, 4]); see Fig. 2c. A type-(i) edge is called *head-head*, while a type-(ii) edge is called *head-tail*[1]; refer to the green and blue edges of Fig. 2c, respectively. It is not difficult to see that the subset of the head-head edges of a rique induces a stack in $\prec$, while the corresponding set of the head-tail edges of a rique induces a queue in $\prec$ [4]. Thus, in a sense, a rique is a special case of a stack and a queue; not every pair of a stack and a queue however forms a rique.

3 Our result

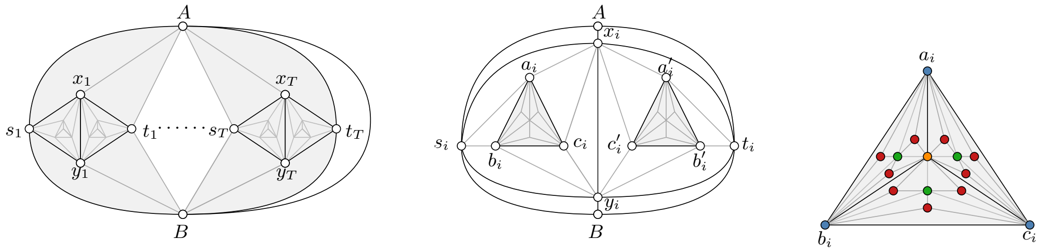
In this section, we prove that there exist planar graphs that do not admit mixed layouts with one rique and either one stack or one queue. To achieve this, we establish a recursive definition of a planar graph (Section 3.1) and we prove that every layout of it with one rique and either one stack or one queue contains at least two edges that either cross in the cylindric representation of the rique or that cross (nest) in the stack (queue). Our proof contains several combinatorial arguments (Section 3.2) but the case analysis that needs to be performed in order to obtain the desired result is deferred to the computer (Section 3.3). The reason for this is that there exist several cases that one needs to consider arising from the two different types that each edge may have; in addition to this, each edge assigned to the rique may be head-head or head-tail (Section 3.4). For the last step in the proof, we exploit a known formulation of the problem of testing whether a given (not necessarily planar) graph admits a layout with a certain number of pages (stacks, queues or riques) as a SAT instance [8]. In our approach, we use properties from our combinatorial analysis to reduce the size of the search space and to introduce several symmetry-breaking constraints in the SAT instance, which made the latter verifiable in reasonable amount of time (less than 10 minutes) using a standard SAT solver [10]. Note that, the actual implementation has become part of [5] and the corresponding code is available to the community as part of the following `GitHub` repository:

`https://github.com/linear-layouts/SAT`

3.1 The graph supporting the proof

We start with the description of the graph, which contains a set of $2T$ independent vertices s_i and t_i, with $1 \leq i \leq T$, called *terminals*. For each $i = 1, \dots, T$, we connect each of s_i and t_i to two adjacent vertices A and B, called *poles*. Each pair of such terminals delimits a so-called *component* C_i in G_T (colored gray in Fig. 3a) as follows: For $i = 1, \dots, T-1$, we add two vertices x_i and y_i that are connected by an edge; each of these two vertices is connected with s_i and t_i; additionally, x_i is connected with A, and y_i with B. In a second step, we construct a 3-cycle $\langle a_i, b_i, c_i \rangle$ and we connect vertex a_i with x_i and s_i, vertex b_i with s_i and y_i, and vertex c_i with x_i and y_i. Symmetrically, we construct a 3-cycle $\langle a'_i, b'_i, c'_i \rangle$ and we connect vertex a'_i with x_i and t_i, vertex b'_i with t_i and y_i, and vertex c'_i with x_i and y_i; see Fig. 3b. Aiming to introduce in G_T several subgraphs, which are neither 2-stack nor 2-queue embeddable [1, 13], the construction continues by *stellating* several already formed

[1] Note that a deque additionally supports *tail-tail edges* (semi-circles below ℓ) and *tail-head edges* (two semi-circles, one that starts at the left endpoint of the edge, lies below ℓ and ends at a point p of ℓ to the right of the last vertex of $\prec$ and one that starts at p, lies above ℓ and ends at the other endpoint).



(a) Graph G_T **(b)** Component C_i **(c)** Stellating $\langle a_i, b_i, c_i \rangle$

Figure 3 Illustrations for the construction of graph G_T: Each gray subgraph in (a) corresponds to a copy of the graph in (b); each gray subgraph in (b) corresponds to a copy of the graph in (c).

faces, where the operation of stellating a face f bounded by a cycle C introduces a vertex u in f and connects u to each of the vertices of C. In particular, we proceed by stellating the resulting faces $\langle a_i, b_i, c_i \rangle$ and $\langle a'_i, b'_i, c'_i \rangle$, introducing two new vertices d_i and d'_i, respectively (refer to the yellow vertex in Fig. 3c). Afterwards, a second round of stellations occurs involving the faces $\langle a_i, d_i, b_i \rangle$, $\langle a_i, d_i, c_i \rangle$, $\langle b_i, d_i, c_i \rangle$, $\langle a'_i, d'_i, b'_i \rangle$, $\langle a'_i, d'_i, c'_i \rangle$ and $\langle b'_i, d'_i, c'_i \rangle$ (refer to the green vertices in Fig. 3c). The final graph G_T is obtained by stellating each of the newly formed faces once more (refer to the red vertices in Fig. 3c). We refer to two vertices (edges) of two different components C_i and C_j that correspond to the same vertex (edge) in the construction above as *twin* vertices (edges), e.g., the vertices $x_1, \ldots, x_T$ are twin vertices, while the edges $(A, x_1), (A, x_2), \ldots, (A, x_T)$ are twin edges.

3.2 The combinatorial part of the proof

Assume that G_T has a mixed linear layout $\mathcal{L}$ with one rique and either one stack or one queue. By symmetry, we may assume w.l.o.g. that $A \prec B$ and $s_i \prec t_i$ holds in $\mathcal{L}$, for each $i = 1, \ldots, T$. Since each component in G_T is of fixed size, if we set T to be large enough, then we can assume by pigeonhole principle that there is a certain number, say k, of copies of components, w.l.o.g. $C_1, \ldots, C_k$, of G_T that have exactly the same layout in $\mathcal{L}$. Namely, for any two components C_i and C_j, with $1 \le i, j \le k$, (i) the order in which any two vertices u and v of C_i appear in $\mathcal{L}$ is the same as their twin vertices u' and v' of C_j, while (ii) any two twin edges of C_i and C_j are assigned to the same page and additionally are of the same type (e.g., both head-head or both head-tail) if assigned to the rique of $\mathcal{L}$. Using Ramsey's theory (and assuming that T is even larger), we can further guarantee that (iii) each group of twin edges form a rainbow or a twist or a necklace in the underlying linear order.

In the following, we assume that T is large enough such that we can identify $k = 4$ components C_1, C_2, C_3 and C_4 with the aforementioned properties. In this case, by symmetry, we can further assume that $t_1 \prec t_2 \prec t_3 \prec t_4$. Let w_1 be any vertex connected to t_1 that is not one of the poles A or B of G_T. Let w_2, w_3 and w_4 be the twins of w_1 in C_2, C_3 and C_4, respectively. Since the edges (t_1, w_1), (t_2, w_2), (t_3, w_3) and (t_4, w_4) are twin edges (thus, forming a rainbow or a twist or a necklace), it follows that either $w_1 \prec w_2 \prec w_3 \prec w_4$ or $w_4 \prec w_3 \prec w_2 \prec w_1$ holds in $\mathcal{L}$. Extending this argument to the neighbors of w_1, w_2, w_3 and w_4 and further, one may conclude that for every quadruple of twin vertices z_1, z_2, z_3 and z_4 in C_1, C_2, C_3 and C_4, respectively, it holds that either $z_1 \prec z_2 \prec z_3 \prec z_4$ or $z_4 \prec z_3 \prec z_2 \prec z_1$. Twin vertices satisfying this property are said to be *monotonically ordered*.

3.3 The computer-aided part of the proof

With the observations that we made in Section 3.2, we were able to prove that, for large enough values of T, graph G_T does not admit a mixed linear layout with one rique and either a stack or a queue using the SAT formulation described in [8]. More precisely, assuming to the contrary that G_T admits such a layout, the subgraph of G_T formed by the poles A and B and by the four components C_1, C_2, C_3 and C_4 that we described in Section 3.2 must also admit a corresponding layout under the following constraints:

1. Pole A precedes pole B.
2. Terminal s_i precedes terminal t_i for each $i = 1, 2, 3, 4$.
3. Every quadruple of twin edges is assigned to the same page.
4. For every quadruple of twin vertices, we require them to be (i) monotonically ordered, (ii) either all before or all after pole A and (iii) either all before or all after pole B.

Note that, by our discussion in Section 3.2, Constraints 1–4 preserve the satisfiability of the SAT instance. However, with the online implementation [6] of [8], which already provides support for encoding Constraints 1–4 in SAT, we verified that the subgraph of G_T formed by the poles A and B and by the four components C_1, C_2, C_3 and C_4 admits a mixed linear layout neither with one rique and one stack nor with one rique and one queue when Constraints 1–4 are imposed, contradicting the fact that G_T also admits such a layout. The total time needed to verify the unsatisfiability was less than 10 minutes on a single-node 4-core 3.3 GHz Intel Core i5-4590 machine with 16GM RAM. We summarize this finding in the next theorem.

▶ **Theorem 3.1.** *There exist planar graphs that do not admit mixed linear layouts with one rique and either one stack or one queue.*

3.4 Some remarks towards a purely combinatorial proof

We conclude this section by mentioning that a purely combinatorial proof is possible to be derived by further extending the arguments that we introduced in Section 3.2. As a matter of fact, the next step in the proof is to consider the six possible permutations that may arise for the poles A and B with respect to the terminals s_1, t_1, s_2, t_2, s_3, t_3, s_4 and t_4 of the components C_1, C_2, C_3 and C_4, namely: **(P.1)** $s_i \prec A \prec B \prec t_i$, **(P.2)** $A \prec s_i \prec B \prec t_i$, **(P.3)** $s_i \prec A \prec t_i \prec B$, **(P.4)** $A \prec B \prec s_i \prec t_i$, **(P.5)** $s_i \prec t_i \prec A \prec B$ and **(P.6)** $A \prec s_i \prec t_i \prec B$. Then, one has to argue on the feasible positions of the remaining (twin) vertices contained in C_1, C_2, C_3 and C_4 within each of P.1–P.6. However, these positions depend on the page that each edge is assigned (rique, stack or queue) and of its type (head-head or head-tail, if the edge is in the rique). This make the number of starting cases for the edges connecting A, B and the terminals s_1, t_1, s_2, t_2, s_3, t_3, s_4 and t_4 already very large and the resulting purely combinatorial proof very tedious.

4 Conclusions

In this work, we demonstrated planar graphs that do not admit mixed linear layouts with one rique and either one stack or one queue strengthening a corresponding result by Pupyrev [15] limited to layouts with one stack and one queue. We also made a step towards answering a question in [8] related to the rique number of planar graphs that ranges between 2 and 4; we feel that to show a lower bound of 3 is a realistic goal. Nevertheless, we consider closing this gap as an interesting open problem for future consideration.

Related to our work is also a result by Angelini et al. [2], who also provided a strengthened version of the result by Pupyrev [15] by demonstrating 2-trees that do not admit mixed linear layouts with one stack and one queue. Their result implies that 2-trees do not admit rique layouts with one rique. On the other hand, 2-trees admit stack layouts with two stacks [16], which trivially implies that they also admit mixed linear layouts with one rique and one stack. In this regard, it would be interesting to study whether this result transfers to planar 3-trees, namely, whether planar 3-trees admit mixed linear layouts with one rique and either one stack or one queue; note that planar 3-trees admit stack layouts with three stacks [13], which implies that they also admit mixed linear layouts with one deque and one stack. So, in a sense, our question is whether substituting the deque page with a rique one still suffices for such a positive result.

Acknowledgements. The authors thank M. A. Bekos for numerous fruitful discussions.

References

1 Jawaherul Md. Alam, Michael A. Bekos, Martin Gronemann, Michael Kaufmann, and Sergey Pupyrev. Queue layouts of planar 3-trees. *Algorithmica*, 82(9):2564–2585, 2020. doi:10.1007/s00453-020-00697-4.

2 Patrizio Angelini, Michael A. Bekos, Philipp Kindermann, and Tamara Mchedlidze. On mixed linear layouts of series-parallel graphs. *Theor. Comput. Sci.*, 936:129–138, 2022. URL: https://doi.org/10.1016/j.tcs.2022.09.019, doi:10.1016/J.TCS.2022.09.019.

3 Christopher Auer, Christian Bachmaier, Franz-Josef Brandenburg, Wolfgang Brunner, and Andreas Gleißner. Plane drawings of queue and deque graphs. In Ulrik Brandes and Sabine Cornelsen, editors, *Graph Drawing*, volume 6502 of *LNCS*, pages 68–79. Springer, 2010. doi:10.1007/978-3-642-18469-7_7.

4 Michael A. Bekos, Stefan Felsner, Philipp Kindermann, Stephen G. Kobourov, Jan Kratochvíl, and Ignaz Rutter. The rique-number of graphs. In Patrizio Angelini and Reinhard von Hanxleden, editors, *Graph Drawing and Network Visualization*, volume 13764 of *LNCS*, pages 371–386. Springer, 2022. doi:10.1007/978-3-031-22203-0_27.

5 Michael A. Bekos, Mirco Haug, Michael Kaufmann, and Julia Männecke. An online framework to interact and efficiently compute linear layouts of graphs. *CoRR*, abs/2003.09642, 2020. URL: https://arxiv.org/abs/2003.09642, arXiv:2003.09642.

6 Michael A. Bekos, Mirco Haug, Michael Kaufmann, and Julia Männecke. An online framework to interact and efficiently compute linear layouts of graphs. *CoRR*, abs/2003.09642, 2020. online version http://alice.math.uoi.gr/; source code available at https://github.com/linear-layouts/SAT. URL: http://arxiv.org/abs/2003.09642, arXiv:2003.09642.

7 Michael A. Bekos, Michael Kaufmann, Fabian Klute, Sergey Pupyrev, Chrysanthi N. Raftopoulou, and Torsten Ueckerdt. Four pages are indeed necessary for planar graphs. *J. Comput. Geom.*, 11(1):332–353, 2020. URL: https://journals.carleton.ca/jocg/index.php/jocg/article/view/504.

8 Michael A. Bekos, Michael Kaufmann, Maria Eleni Pavlidi, and Xenia Rieger. On the deque and rique numbers of complete and complete bipartite graphs. In Denis Pankratov, editor, *Canadian Conference on Computational Geometry*, pages 89–95, 2023. URL: https://wadscccg2023.encs.concordia.ca/assets/pdf/CCCG_2023_proc.pdf.

9 Frank Bernhart and Paul C. Kainen. The book thickness of a graph. *J. Comb. Theory, Ser. B*, 27(3):320–331, 1979. doi:10.1016/0095-8956(79)90021-2.

10 Armin Biere. Lingeling, Plingeling and Treengeling entering the SAT Competition 2013. In A. Balint, A. Belov, M. Heule, and M. Järvisalo, editors, *SAT Competition 2013*, volume B-

2013-1 of *Department of Computer Science Series of Publications B*, pages 51–52. University of Helsinki, 2013. Source code available at `https://github.com/arminbiere/lingeling`.

11 Vida Dujmovic, Louis Esperet, Cyril Gavoille, Gwenaël Joret, Piotr Micek, and Pat Morin. Adjacency labelling for planar graphs (and beyond). In *FOCS*. IEEE, 2020. `doi:10.1109/FOCS46700.2020.00060`.

12 Vida Dujmović and David R. Wood. On linear layouts of graphs. *Discrete Mathematics & Theoretical Computer Science*, 6(2):339–358, 2004. URL: `http://dmtcs.episciences.org/317`.

13 Lenwood S. Heath. Embedding planar graphs in seven pages. In *FOCS*, pages 74–83. IEEE Computer Society, 1984. `doi:10.1109/SFCS.1984.715903`.

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

15 Sergey Pupyrev. Mixed linear layouts of planar graphs. In Fabrizio Frati and Kwan-Liu Ma, editors, *Graph Drawing and Network Visualization*, volume 10692 of *LNCS*, pages 197–209. Springer, 2017. `doi:10.1007/978-3-319-73915-1\_17`.

16 S. Rengarajan and C. E. Veni Madhavan. Stack and queue number of 2-trees. In Ding-Zhu Du and Ming Li, editors, *COCOON*, volume 959 of *LNCS*, pages 203–212. Springer, 1995. `doi:10.1007/BFb0030834`.

17 Mihalis Yannakakis. Embedding planar graphs in four pages. *J. Comput. Syst. Sci.*, 38(1):36–67, 1989. `doi:10.1016/0022-0000(89)90032-9`.